edX101: How to Create an edX Course
<table>
<thead>
<tr>
<th>Version</th>
<th>Author</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1.0</td>
<td>DT</td>
<td>Original draft</td>
</tr>
</tbody>
</table>
| v2.0    | SP     | Added:  
Studio Tutorial  
Studio Advanced Tutorial  
Images for videos  
Updated:  
Your About Page  
Open Response Assessments |
| v2.1    | SP     | Added:  
Exercise Gallery  
Add a Link to a Course Unit  
Specify Video Settings  
Preface  
Visibility  
Links to live course for every unit title (except in XML Documentation)  
Updated:  
Create a Video Component  
Deleted:  
Gallery of Response Types |
| v3      | SP     | Added:  
Homework 1 lesson  
Problem Settings in Studio unit  
Updated:  
Add Transcripts to Videos unit  
Open Response Assessments lesson  
Annotation unit  
Word Cloud unit |
Open Response Assessments

Open Response Assessments Roadmap
Introduction to Open Response Assessments
Components of an Open Response Assessment
Open Response Assessment Types
A Few Notes about Open Response Assessments
Past Use of Open Response Assessments
Effective Questions
Rubrics
Self Assessment
AI Assessment
Peer Assessment
Accessing Scores and Feedback
Action: Try a Combined Assessment
Action: Create an ORA Problem in Your Practice Course
Action: Grade Student Responses
Best Practices
Discussion: Open Response Assessments Lesson

Building, Running, and Ending

Building

Running

Overview
Welcome Email
Welcome Announcement
The Forum
The Wiki
Weekly E-Mail and Announcements

Ending

Certificates of Mastery

Appendix

Course Readiness Checklists
Sample HTML Templates .............................................................................................................. 320
  Simple HTML Template ............................................................................................................. 320
  Dynamic HTML Schedule ......................................................................................................... 322
  HTML Syllabus ......................................................................................................................... 326
  Long Course Description ......................................................................................................... 334
  Course Overview ..................................................................................................................... 336

XML Documentation .................................................................................................................. 339
  Option Response (Drop-Down List) ......................................................................................... 339
  Multiple Choice Response (Drop-Down List) ......................................................................... 342
  Checkbox Response ................................................................................................................. 346
  String Response ...................................................................................................................... 349
  Numerical Response ............................................................................................................... 352
  Formula Response .................................................................................................................. 357
  Image Response ..................................................................................................................... 362
  Custom Response: Script Evaluation ....................................................................................... 365
  Custom Response: Chemical Equations .................................................................................. 370
  Schematic Response: Circuit Schematic ................................................................................... 372

edX Glossary ................................................................................................................................ 376
  edX Glossary ........................................................................................................................... 377
Preface

Welcome to edX101-beta!

This course is designed to walk you through the process of planning, building, and running your own online course. In the process, you will experience an edX course from a student's perspective.

The course is organized into five sections.

**Introduction: A Tour of EdX** In this section, you will hear faculty members share their own perspectives on teaching an online course. You'll also walk through an edX course just as a student would, and you'll learn about Studio, edX's tool for creating edX courses.

**Structuring Your Course** This section takes you through the preliminary stages of preparing your course: creating a syllabus, choosing a textbook, writing a course description for the course catalog page, etc.

**Producing Videos** Most people creating online courses have little or no experience producing professional quality videos. This section teaches you the basics of video production, including how to adapt your presentations to online courses, selecting a video style, and how to achieve good audio quality.

**Writing Exercises** One of the most exciting aspects of teaching an online course is the opportunity to embed interactive activities within a lesson. In this section, you'll learn about the types of activities that are possible on the edX platform and the best practices surrounding them.

**Building, Running, and Ending** This section will take you through the mechanics of building your course, the challenges of running your course, and the procedures for ending your course.

Online courses provide the opportunity to embed interactivity into every step of the learning process. We will call the online equivalent of a lecture a "lesson" or a "learning sequence." Most lessons are a sequence of videos, text, exercises, and discussion spaces. EdX101 contains exercises so that you can have a more student-like experience as well as "Actions," which are tasks that are designed to facilitate your course creation process. At the end of each Action is a check box. If you complete all of the actions, then by the end of the course, you will have built, run, and ended your own course.

**Note** Studio, edX's tool for course creation, is currently in closed beta and only accessible to our partner institutions.

--The edX101 team
Introduction: A Tour of edX
What Does an edX Course Look Like?

Welcome to edX 101

Welcome to edX 101!

(Embedded video, transcript follows)

ANANT AGARWAL:

I am Anant Agarwal. I am the president of edX.

So, edX is looking to reinvent education. The idea is to offer education to people worldwide. Dramatically increase access to education, but at the same time, dramatically improving on-campus education from blended models of learning.

Learning is not going to be the same anymore, whether on campus or worldwide. We have brought online technologies, computing technologies, internet to bear on something that we’re all passionate about—learning and education. By making a concerted attempt at improving education, I think that this is having a real impact. Online learning is like a rising tide that is
going to lift all boats. So as a professor, to be able to spread the word and increase access, and have an impact on students all around the world, it's very exciting.

Here, you record your videos once. Yeah, of course, you can improve them over time. But the idea is that in subsequent semesters after you record the videos and assessments, you can now think of ways in which you can be much more creative, and think of really cool ways in which students can learn things. And the courses involve content like video snippets interleaved with exercises. It involves simulations, online simulations. It involves discussion forums. So there's a way of teaching and there's a way of creating assessments that is quite particular to online learning. And so we can create very high-quality education.

If you have passion for teaching, if you care about students, and you're willing to try new techniques, I think you will do a great job of teaching on the edX platform.
Action: Browse through the Course Tabs

Get to know edX101 by clicking each of the tabs at the top of the page.

- **Course Info**: This tab contains course updates and announcements. It also includes the **Course Handouts** sidebar, which contains links to supplemental information for the course. In all edX courses, this tab opens by default every time a student accesses a course through the dashboard, so students can see any important updates the instructor has posted.

- **Discussion**: The **Discussion** tab shows the discussion forum for the course. Take a minute to explore the current discussions in edX101. You'll introduce yourself on the forum just a bit later in this lesson.

- **Wiki**: The edX101 wiki provides a space for learners to share ideas. It has links to resources for course creators and even includes a "playground" where you can experiment before editing the actual wiki page.

- **Progress**: On this tab you can see your progress through the course, including how many exercises you've completed, your score on each exercise, and how many exercises remain.

- **New Features**: EdX101 is under constant revision. Here you can see a list of what's changed recently.

- **Open Ended Panel**: This is a tab used with open response assessments, which you'll learn about in a later lesson.

- **Table of Contents**: This tab contains a linked list of every lesson in the course, along with each individual topic (unit) in that lesson.

When you're done browsing through each tab, click the **Courseware** tab to return to the lesson.

**Note** The **Courseware** tab is always visible in the top left corner of the page. You can click this tab at any time from any other tab to return to the lesson you were learning.
A Learner’s Journey

(Embedded video, transcript follows)

ANANT AGARWAL:

Let me briefly introduce you to the edX platform through the eyes of our students.

So a typical learner would come to edx.org. And they would like to find out what course they would like to take. So they browse down. And let us say they decide to check out PH 207x, Health in Numbers. This is a public health course from HarvardX. So they check out the About page. And then they decide to register for the course. When the student logs into the website, they will now see this course on their dashboard since they have registered for the course. They can now directly dive into the course by clicking on View Courseware. There is Courseware, Course Info, Textbook, a discussion forum, a wiki for collaboration, and a Progress tab for the students to view their own progress.

Next, let’s go to the Courseware tab within the student’s dashboard. The courseware for this course is organized by week. In the Overview section, there is a welcome tutorial and some basic tutorials for tools that are found within the course. So in week three, we notice that the learner gets to do two learning sequences.

What is a learning sequence? A learning sequence is a series of video snippets. Each of these is a video snippet. And within those video snippets, are interleaved some learning exercises. You can see there are three learning exercises here and a number of video snippets. So as you can see, during a given week, students may go through a couple of learning sequences, which are interleaved videos and exercises. They do a homework. And they might also do a laboratory. Certain weeks of the course may also have exams. Here is an example of a midterm exam that a learner might have to go through.

Now, let’s take a look at some of the other tabs. Notice that there is a Textbook tab. Here, students can access textbooks that supplement the rest of the course materials.

So let’s go into the discussion forum. On the left-hand pane, you notice the subject lines of all the student postings. The right-hand pane is where a student can post a response or ask a question. A student can sort posts by date, votes, or comments. Each of the posts can also have a number of comments. And you can also sort by the number of comments on the various posts.

The wiki is an area where students collaborate. Here, they can view or edit the page to share resources and co-create understanding.
Finally, let's go and take a look at the Progress tab. The Progress tab is where learners keep track of their own progress. Notice that on the x-axis of the Progress page, you see all the homeworks. You see the labs. You see the midterm exam, final, and a total score. And when a student finishes all of her assignments and tests, she gets a certificate of completion. That is a brief overview of some of the most familiar aspects of the edX platform.
**Action: Introduce Yourself**

Please help us create a lively discussion forum and strong sense of community by introducing yourself on the discussion forum. To do this, click the **Discussion Forum** tab, find the post titled "Official Introductions," scroll to the bottom, and post a response with your own introduction. When you're done, click the **Courseware** tab to return to the lesson.

Please take a moment to explore and contribute to the edX101 wiki. To do this, click the **Wiki** tab. When you’re done exploring the wiki, click the **Courseware** tab to return to the lesson.

(Discussion space provided)
**The Interactive Classroom**

(Embedded video, transcript follows)

GRACE LYO:

Teaching an online course is a wonderful opportunity to inject interactivity into your classroom. In this new medium, you're not going to be teaching 100 students. You're not going to be talking to 100 students, or even 30 students. You're going to be talking to one student at a time, even though you'll be reaching 30,000 students, or even more, possibly. In this video, we're going to show you examples of effective uses of this online medium. And we're also going to show you some of the tools that you'll have at your disposal in creating your course.

Now, there's no way we're going to be able to show you all of the tools, because as I'm talking right now, there are engineers out there creating new tools for you. So this video is going to be immediately obsolete. But we'll show you a few of the tools that you can use. Let's start with an example of interactivity embedded into lessons.

So this is a lesson from Public Health 207. This is a Harvard course. And as you can see, we're in week one. This is the second lesson. And this ribbon at the top shows you the different pieces that make up this lesson. So there are a bunch of videos. These little icons indicate videos. And then this is an exercise. These are more videos.

So now we're on the sixth piece of this lesson. And this is a video in which Professor Pagano explains what the mean is to the students. Now, after going through some examples, he then talks about a concrete situation in which you're looking at hospitals. And you guess that maybe mortality is a good way of determining whether a hospital is a good hospital. So after students finish watching the video, in the next segment, they have this little exercise in which they investigate mortality rates and hospital rankings.

So here you see that in general, if you have young people having a mortality rate of 2 deaths per 1,000 visits, middle age of 5, and old of 8, then if you take a hospital and you adjust the composition of middle, old, and young people in the hospital-- so let's do that here, let's make it so that all of the patients are elderly-- so now you can see we have 100% old and 0% middle and young. And it automatically computes the average mortality, which in this case would be 8 deaths per 1,000 visits.

Now, on the other hand, you can take these sliders and adjust it so that the population is as young as possible. And now, the average mortality rate calculation goes down to 3.5 deaths per 1,000 visits. And the students get to discover for themselves that mortality rate is not
necessarily a good indicator of whether a hospital is a good hospital. Because if you have a children's hospital, it's going to look much better than if you have a hospital that focuses on the elderly.

So in this course, you see Michael Sandel lecturing. This was captured live in front of a student audience in Sanders Theater at Harvard. And Michael Sandel discusses some ethics questions and asks the students to answer the questions for themselves. Now, the online student, after watching the video, is given the opportunity to answer the exact same question themselves.

Now, if we answer this-- let's see. Do we think it's permissible? Let's say yes. Now, as soon as we answer this question we get to see statistics about what the other students answered. So as you can see, I'm in the majority here. And now immediately below, we have this discussion which has appeared, where students can discuss this extra question that Professor Sandel has posed, which is right here. After participating in this discussion, students can continue through the lesson, and they're asked further questions. And finally at the end, they're asked whether they've changed their mind. And again, they're given statistics about what other students said-- whether other students changed their minds as well.

So the point here is that this is all this interactivity embedded into the lesson that might be more difficult to do in an on campus setting. In an online course, you don't have just multiple choice or fill in the blank. You actually have all these rich types of exercises and questions you can ask your students. You can embed discussion spaces easily into your lessons using Studio.

Let's look at another example of interactivity embedded into a course using novel tools. This is an example from MIT's course on chemistry, with Professor Michael Cima, 309.1. So after students watch this video about this, they can now go to this exercise. And in this exercise, they're shown a graph. And they're asked to click on the area in this graph where the electrons have enough energy to jump across the band gap. So students can actually pick a place-- and you see, where I click is where this green dot shows up-- and then they can click Check to see whether they got it right. And sadly, I got it wrong. One of the great things about this is as you can see, the student can keep trying and trying over again. And they can find out immediately whether they got the answer right or wrong. After getting my wrong answer, maybe I stop and think about it, do some computations, and I say, maybe it's here. And I click Check, and yes, we got it right.

So the examples we've seen so far are all lessons. They're video sequences with exercises built into them. There are also plenty of graded assignments that students do that are not in the context of a video sequence.

Now let's look at CS188, Berkeley's course on artificial intelligence with Dan Klein and Pieter Abbeel. This is a coding project about Pac-Man. So as you can see here, Pac-Man is this yellow guy here. And there's this little pellet here that he wants to get. And he has to find his way through this gigantic maze to find the pellet. Now at this point in the course, students have learned quite a bit about search algorithms and different types of search algorithms. So this whole sequence explains to them all of the different things they're asked to do in this project.

So the first page, they're shown all of the different files they're going to need to use together, how to use the auto grader, and then they're given specific tasks for their code to do. And at the very end, once they've written their code, what they do is they come to this submission page. They click on Choose Files, and a window pops up. And they choose the file that they want to submit. And they click Submit. And they find out whether they got it right or wrong. And they also get some feedback from the auto grader. So this, again, gives the students the opportunity to check their answer over and over, as many times as they want. And they really can't guess
the right answer. They really have to understand what they're doing to be able to get the right answer. And because they get immediate feedback, it's very rewarding and game-like for them.

So now we're going to look at an example from 600.2-- MIT's course on circuits and electronics. And you might wonder how you can possibly have a course on circuits and electronics that's online. Because a fundamental part of circuits and electronics is building things. So what the creators of this course did was they designed a virtual circuit kit, where students can create their own circuits. Now, students can add pieces to this. They can move things around. And they can extend edges like this to connect things. And they can extend wires to make the ends connect. And then when they're done testing it out, they can scroll down here and click Check and find out whether they got it correct.

So again, there are many, many different possible answers that the student can submit. And they can try out lots of different things. There are a couple more examples I'd like to show you, and there are many, many different types that are possible, of course.

So the circuits and electronics course used PowerPoint slides that were annotated. So when we start this video here, you can see that this is a prepared slide. And now, Professor Agarwal is writing on the slide and explaining how to build the digital gate.

If we look at one of the tutorials from 600.2, we have a video that is an overhead camera capture with a piece of paper. And the presenter here, Piotr Mitros, is writing on the piece of paper and explaining something about the circuit. He also has these little pieces of paper prepared that he uses to cover up pieces of the circuit so that he can talk about different situations.

So as you can see, there are lots of different ways of bringing interactivity into your course. Just get creative and enjoy yourself. In the next segment, you're going to see the mechanics of how to do this.
New Technologies: Computer-Graded Essays

Read the following article from the April 4, 2013 New York Times:

Essay-Grading Software Offers Professors a Break

Click Show Discussion below, and then read the posts from your classmates. Respond to one or more of them by clicking Respond below the student's original post.

Post your own response to the article by clicking New Post. Some questions you can consider for your post are:

- How do you think computer-graded essays (also called “open response assessments”) will change education in your field?
- What types of questions do you think would work best and worst with open response assessments?

Computer-graded essays are just one facet of open response assessments. For more information about open response assessments, see the Open Response Assessments lesson.

(Discussion space provided)
GRACE LYO:

EdX Studio is a software tool that edX created to make writing your new course easy and intuitive. EdX 101 is the first course that was created entirely in Studio. And now, most of our courses are created in Studio. In this video, we’re going to look at Studio and see what it’s capable of doing. Let’s take a look now.

So here we are in edX Studio. Let’s sign in. And as you can see, I have exactly one course, called "How to Create an edX Course." This is edX 101. And you are going to get to see the secret inner life of edX 101. So you see what edX 101 looks like from the outside. Now we’re going to see what it looks like from the inside.

So now we’re going to click the View Live button. And we’re going to view Studio and edX 101 side by side. So on the left here, we have edX Studio. It's the insides of edX 101. And here, we have the outsides of edX 101, the part that you’re used to seeing. And I want you to notice here that Studio publishes to edX. Any course that you create on edX Studio is going to show up for students on edX. There’s actually another Studio, studio.edge.edx.org. And as you can see, it looks exactly the same as studio.edx.org. And I want you to notice here that studio.edge publishes directly to edge.edx.org. Now, any course that you create on studio.edx.org is going to show up on edx.org.

So now there’s a question-- what is the difference between Edge and edX? Why do we have these two different platforms? Well, it’s kind of like the difference between the minor leagues and the major leagues. So Edge is like the minor leagues. EdX is like the major leagues. And on Edge, you have complete freedom to do whatever you want. You can put test courses up there, and you can release them to whoever you feel like releasing it to. Once you want to make your course go into the big leagues, then you talk to your dean of online learning, you talk to edX, and the course can then be moved and polished and put onto the edX platform. So that's the general process.

All right, let's take a little look around edX 101. So if we click on Courseware here, we see the beginning of the course, Introduction-- A Tour of edX. And as you can see here, you have the exact same section on the left, Introduction-- A Tour of edX. Similarly, the second section, Structuring Your Course, shows up on the right here. And going down, you can see this is really the same course.
Now, I can do all kinds of funny things here. For example, I can open up this unit here, and I can open up this over here. And now, remember, this is live. So every change I make here is going to be seen by students who are looking at the course right now.

As you can see here, we have Welcome to edX 101, Tabs, edX Highlights, and Introduce Yourself, all in that order. Now, I can take this Introduce Yourself piece, and I can actually just drag it down into this other section called Your About Page. So let's stick it here.

Now, remember, this is live. So if I go over here and I hit Refresh and I go into Structuring Your Course, which is where we moved it, Your About Page-- and where did we put it? It's over here. It's the fourth one down-- one, two, three, four. And here, we have Introduce Yourself. So we've just moved it. This is visible to the whole world. Now let's fix it so that nobody is too confused. So now that I've moved it back into place, I can come here, I can refresh, and I can see it disappeared. It's gone now.

Another thing I can do is I can come up here into Course Settings and click on Course Teams. Now, I can add collaborators here. So as you can see, I have a large course team here. There are lots of different people who can also edit the edX 101 course. So this enables me to not have to do all of the work myself. I can also click here to set the course dates-- the start date, start time, end date, end time, enrollment date and time, et cetera. You can also work on your course after it's been released.

So edX 101 is under constant revision. And there's actually a private page here, edX's Most Guarded Secrets. And if you look here, on the student side, you will not find this page. This only has four. And over here, there are five. As you can see here, it's grayed out and it says Private. If I click into it, there's this visibility setting that I can change to Public if I want.

Now, all of these pages, if I want to edit them, it's very easy. I just click here on Edit, and I start typing. So we can make this bold. So this, as you can see, it's all visual. This is now bold and italics. You can also add static pages. So as you can see over here, there are these tabs across the nav bar-- Courseware, Course Info, Discussion, Wiki, and Progress. These are all the standard tabs. I've added a couple for edX 101. There's Table of Contents and New Features. This Instructor one appears for all of the authors on the course. And you can see here in Studio, there's Table of Contents and New Features. If I click Edit, I'll see the table of contents. And I can edit it.

So that's a really brief overview of edX Studio. And if you want, you can actually go and create your own online course right now. Just go to Studio, and click the New Course button.

**Studio Question**

What is the difference between edx.org and edge.edx.org?

a. edX.org is the “major leagues,” while edge.edx.org is the “minor leagues.”

b. edx.org is visible to the whole world, whereas edge.edx.org is visible only to people who have permission.

c. Courses that you create on Edge appear on edx.org.

d. Anyone with access to studio.edge.edx.org can release a course to the entire world without additional authorization.

e. Both A and D
f. Both B and C

Answer: Both A and D

Discussion: Improve the Question

If you think the question above has several significant flaws, you're right! We deliberately designed it that way to help you get a sense of what students may experience. In the discussion space below, please share your thoughts on ways we could improve the question.

(Discussion space provided)
Structure of a Course

Introduction

In the next lesson, you will create your own practice course using Studio, the tool that you just learned about. You do not need to have a plan for your course--we've provided materials for you. All you need for right now is a name for your course, the name of your institution or organization, and a course number. For your practice course, these can be whatever you want.

Before you get started on your practice course, we suggest that you familiarize yourself with the way that Studio is structured.

Structure of a Course

Just as in an offline course, content in an online course is broken down into smaller pieces. In Studio, these pieces are categories called sections, subsections, and units. Units, in turn, are made up of components, which contain the actual content of your course.

Sections are composed of

- Subsections, which are composed of
- Units, which are composed of
  - Components, each of which is a
    - Discussion,
    - HTML,
    - Problem, or
    - Video.

You have seen all of these already in edX101:
Sections, subsections, units, and components also appear in Studio.

You’ll learn more about the difference between Studio and the LMS, which is where your students see your course, a little later on.

**Sections, Subsections, Units, and Components**

**Section**

A section is the topmost category that you use to organize your course. Sections, for example, may correspond to weeks in your course—for example, section 1 may be named Week 1,
section 2 may be named Week 2, and so on. Section names appear in the course accordion in the left pane of the LMS.

**Subsection**

A subsection is a subcategory of a section. It corresponds to a lesson, which is an interwoven selection of units of different types, such as videos, text, images, discussions, learning exercises, and assessments (quizzes or exams). It is an interactive representation of the material that you would cover in a typical class period. Subsection names also appear in the course accordion in the left pane of the LMS.

Subsections should have at least one unit.

**Unit**

A unit is a further category that helps you organize your course materials. Units do not appear in the left pane with section and subsection headings. Instead, each unit appears as a part of the course ribbon at the top of the page. For example, the "What Does an edX Course Look Like?" subsection contains seven units.

Units should have at least one component.

**Component**

A component is the part of a unit that contains your actual course content. The names of all components in a unit appear when you hover over the unit in the course ribbon at the top of the page.

**The Four Components**

There are four types of components: Discussion components, HTML components, Problem components, and Video components. You'll use each of these in your practice course. Generally, a unit can contain as many types of components as you want, in any order. The only exception is the Discussion component: a unit can only contain one of these.

**Discussion Component**

When you add a Discussion component to a unit, you create a discussion space where you and your students can add posts, comments, and responses to a discussion of specific material right in the body of your course. Discussions that appear in individual units also appear with all the other discussions for the course in the discussion forum, which users can access at any time by clicking the Discussion tab.

**HTML Component**

The HTML component is the most basic component type. It is used to add information to the unit, such as text, lists, links, and images.
**Problem Component**

To add interactive, automatically graded exercises to your course content, use the Problem component. There are many different types of problems you can create on the edX platform, and a number of different ways you can include these problems in the flow of your course.

**Video Component**

You use the Video component to add video segments to your course.
Studio Tutorial

Roadmap for the Studio Tutorial

This Studio tutorial walks you through the mechanical process of creating a practice course that contains basic elements--text, videos, discussion spaces, and simple problems such as multiple choice and checkbox problems. You'll start by creating a user account in Studio and creating the course, and then you'll create individual pieces of content using materials that we've provided.

This tutorial doesn't cover pedagogical information, such as how to adapt your course to an online audience or create engaging videos. It also doesn't go into detail about how to perform complex tasks in Studio. Rather, it's designed for faculty members who have course teams and who want a general idea of the tools that are available to the team. The Advanced Studio Tutorial lesson, which is optional, provides more detailed information about advanced tasks.

- Introduction: This unit gives you an overview of what your practice course may look like when you're done.
- Create an Account in Studio: In this unit, you create your user account in Studio.
- Create and Register for the Course: This unit walks you through the process of creating your course. To see your course as students will see it, you then need to register for your course.
• Create a Subsection and Unit: This unit teaches you how to set up the framework for your content.

• Create an HTML Component: In this unit, you create a component that can contain text, images, and other content.

• Create a Video Component: This unit covers adding a video to your unit.

• Create a Discussion Component: In this unit, you add a discussion component to a unit so that students can discuss the content in that unit.

• Create Problem Components: This unit describes how to create several basic problem types—multiple choice, check box, and numerical input.

• Reorder the Unit: In this unit, you use Studio's drag-and-drop feature to easily reorder the components in your unit.

• Visibility: This unit describes how to make your content visible to your students.

• Add a Welcome Announcement: In this unit, you create and upload a welcome announcement that your students see when they first log on to your course.
Introduction

In this tutorial, you're going to create the lesson content for a very basic practice course. Your course will include one section, one subsection, and one unit. The unit will contain each of the four main components that are available in Studio.

Your practice course won't be a complete course—for example, you won't set up grading for your course, add a syllabus, or create the About page for your course yet. You'll add the other information and materials in the Structuring Your Course section. Rather, this tutorial covers the basics of creating lessons in Studio so that you can then develop your own course with your own content. It will also help you to estimate the time that it will take for you to create your own course later.

While you're creating your practice course, you'll be able to view it in Studio and on the LMS. Studio is where you create your course, and the LMS is where your students view your course. The LMS is where you're viewing edX101 right now. If you've watched the video in the Introduction to edX Studio unit, you've seen a side-by-side comparison of Studio and the LMS. In this tutorial, Studio pictures have magenta frames and LMS pictures have blue frames.

The Studio view will resemble the following.

---

edX101: How to Create an edX Course
To view the course on the LMS, click **Preview** in the **Unit Settings** box in the upper-right corner of any unit page.

The LMS view will resemble the following.

---

**The Fruits of Learning**

'Tis one of our goals here at edX

To circumspect end-of-course headX

We hope that our students

1. Will demonstrate **prudence**
   - And faithfully do extra credX.

**WELCOME TO EDX101!**

---

edX101: How to Create an edX Course
Create an Account in Studio

Note Studio, edX’s tool for course creation, is currently in closed beta and only accessible to our partner institutions. If you are at one of our partner institutions and should be granted access, please speak with your local edX administrative representatives for further instructions.

1. To create a user ID in Studio, go to http://studio.edge.edX.org. The following page opens.

2. In the upper-right corner, click Sign up. The following screen appears.

3. In the Email box, type your e-mail address.
   
   Note We strongly encourage you to use your institutional e-mail address if you are at an edX consortium university.

4. In the Password box, type a unique password.
5. In the **Public Username** field, type the name you want students to see when you post on the user forum.

6. In the **Full Name** box, type your full name. This name will only appear on your user dashboard. Your students will not see this name.

7. Click **Create My Account**.

   After you click **Create My Account**, you will receive an activation e-mail message. To finish creating your account, click the link that you receive in the activation e-mail.

---

**Access and Create Courses**

After you create your account and click the link in the activation e-mail, the My Courses page opens.

If another user has listed you as staff on a course, you can immediately see and edit that course.

If you should be staff on a course but that course doesn't appear in your list, you'll have to contact the main course author separately and ask the author to add you as staff. The course will then appear in your list, and you'll be able to see and edit it.

If you want to create new courses of your own, you'll request access from edX. To do this, click **Request the Ability to Create Courses**. (The permission to create new courses is different from the permission to edit existing courses, so you'll do this whether or not you're a staff member of another course.)
After you request the ability to create courses, you receive a "Pending" notification. The edX team will respond to your request within one business day.

Request Granted

If the request is granted, you receive the following e-mail message.

On the My Courses page, the New Course button appears in the upper-right corner, and the Create Your First Course button appears in the body of the page. You can now create a new course at any time.
Request Denied

If the request is denied, you receive an e-mail message, and a "Denied" message appears on the My Courses page.
Create Your Practice Course

After you create an account in Studio and have been granted the ability to create courses, you’re ready to create your practice course.

1. Sign in to your account. The My Courses page opens.
2. Click New Course.
3. The Create a New Course screen opens.
4. For Course Name, enter the title of your course. Use title capitalization for the course title.
5. For Organization, enter the name of your university. Do not include whitespace or special characters.
6. For Course Number, enter both a subject abbreviation and a number. Do not include whitespace or special characters in the course number.
   
   **Note** If your course will be open to the world, be sure to include the "x". If it is exclusively an on-campus offering, do not include the "x".
7. For Course Run, enter the year and term in which your course will run. This field cannot include white space or special characters, but it can include underscores (_).
   
   **Note** Because seasons are different in the northern and southern hemispheres, we recommend that you specify a term (e.g., 2013_Term3) rather than a season (e.g., 2013_Fall).
8. Click Save.
Your new course opens to the **Course Outline** page.

In your browser's address bar, notice that the URL of your course includes the course organization, number, and course run. In this example, the organization name is “edX,” the course number is “PRACTICE101x,” and the course run is “2013_Term3.” Thus, the URL is https://studio.edge.edx.org/edX/PRACTICE101x/course/2013_Term3.

The course name appears in the upper-left corner, next to the edX Studio logo. In this example, the course name is “Practice Course.”

**Register for Your Practice Course**

If you’re creating your course on **Edge**, you can go on to the next unit.

If you’re creating your course on **edX**, your next step is to register for your course.

1. On the **Course Outline** page, click the blue **View Live** button in the upper-right corner of your screen.

Your course registration page opens in a new tab on the LMS.
2. Click the blue **Register** button to register for your course.

3. In your browser, switch back to the tab that shows Studio. You will still be on the **Course Outline** page.
Create a Section, Subsection, and Unit

To build a lesson in Studio, you'll create a new section, subsection, and unit. Units are made up of components, which you’ll learn about shortly.

1. Sign in to Studio, and then click the course that you want.

2. On the Course Content menu, click Outline. You see the following screen.

3. On the right, click New Section.

4. In the New Section Name box, type Section 1, and then click Save. The name you enter appears under Course Outline, on the left.

5. To create a new subsection in your section, click New Subsection.

6. In the New Subsection box, enter the name for this subsection, and then click Save. For example, if you enter Section 1 as the section title and Subsection 1a as the subsection title, you see the following.
7. Click the new subsection that you just created. In this example, you would click **Subsection 1a**. You see the following screen.

![Display Name: Subsection 1a](image)

8. Click **New Unit**. The following screen appears.

![Display Name: New Unit](image)

9. In the **Display Name** box, type **Subsection 1a Unit 1**. This name appears in Studio as follows.

![Display Name: Subsection 1a Unit 1](image)

This name also appears in the course ribbon at the top of the page in the LMS, as follows.

![Course ribbon](image)

To see what the unit looks like in the LMS, you can click **Preview** under **Unit Settings** in the right pane at any time.
Next, you’ll create an HTML component. HTML components are the basic building blocks of your course.
Create an HTML Component

Introduction

HTML components are most often used to add and format text for your course. You can also add links, images, and other content. The component that you’re looking at right now in edX101 is an HTML component.

Create an HTML Component

1. In the unit that you just created, under Add New Component, click html.

2. In the list that appears, click Text.

A blank component appears.

3. In the blank component that appears, click Edit.

The HTML component editor opens.

4. Copy and paste the following text in the HTML component editor.

The Fruits of Learning

’Tis one of our goals here at edX
To circumvent end-of-course headX.
We hope that our students
Will demonstrate prudence
And faithfully do extra credX.

5. Use the buttons at the top of the component editor to change the formatting any way you want. For example, you can enclose the title in heading tags, create bulleted or numbered lists, or apply bold, italic, or underline formatting. Feel free to enter your own text and play around with it.

6. Create a link to the dictionary.com definition of the word "prudence."
a. Select the word **prudence**, and then click the link icon in the toolbar.

![Image of a HTML component editor with text: The Fruits of Learning.](image)

b. In the **Insert/Edit Link** dialog box, enter the following in the **Link URL** field, and then click **Insert**:

   http://dictionary.reference.com/browse/prudence?s=t

7. In the upper-right corner of the component editor, click **Settings**.

8. In the **Display Name** box, replace **Text** with **Limerick**.

![Image of the component editor with the display name set to Limerick.](image)

This title appears when you hover your mouse over the course ribbon on the LMS.

**Note** The name of the first component in a unit replaces the display name of that unit in the course ribbon. Therefore, after you add this component, you see **Limerick** when you hover over the course ribbon instead of seeing Subsection 1a Unit 1.

9. Click **Save**.

You can also include images in an HTML component. You'll learn more about this in the Advanced Practice Course lesson.
Create a Video Component

Introduction

When you add a video to your course, you first post the video online, and then create a link to that video in the body of your course. The video then plays from the edX video player. The edX video player supports videos in .mp4, .ogg, and .mpeg format.

All course videos should be posted to YouTube. By default, the edX video player accesses your YouTube videos. However, because YouTube is not available in all locations, we recommend that you also post copies of your videos on a third-party site such as Amazon S3. When a student views a video in your course, if YouTube is not available in that student's location or if the YouTube video doesn't play, the video on the backup site starts playing automatically. The student can also click a link to download the video from the backup site.

When you post your videos on a backup site, you can use any site that you want. Keep in mind, however, that the site where you post the videos may have to handle a lot of traffic.

Create a Video Component

To add a video to the unit, you must obtain the YouTube ID for the video, obtain the URL for the backup video, and then create a video component.

To determine the YouTube ID for your own video, locate the video on YouTube and make a note of the code that appears between watch?v = and &feature in the URL. This code appears circled below.

**Note** If &feature does not appear in the URL, just use the code that follows watch?v= in the URL.

![YouTube Video](https://www.youtube.com/watch?v=edX101Feature)

For this practice course, you'll use the Welcome to edX101! video. Its information is below.

<table>
<thead>
<tr>
<th>Video</th>
<th>Backup URL</th>
<th>YouTube Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome</td>
<td><a href="https://s3.amazonaws.com/edx-course-videos/edx-">https://s3.amazonaws.com/edx-course-videos/edx-</a></td>
<td>26T82YVSCT0</td>
</tr>
</tbody>
</table>

edX101: How to Create an edX Course
Instructions

1. Under the HTML component that you just created, click Add New Component, and then click Video.

![](media.png)

2. When the default video component appears, click Edit.

![](media2.png)

3. When the video component editor opens, locate the following fields:
   - Display Name
   - Download Video
   - Video Sources
   - YouTube ID
4. In the **Display Name** field, replace the default text with *Welcome to edX101! (1:43)*.

   The **Display Name** field contains the name of your video as well as, optionally, your video's duration. We encourage you to include your video's duration so that students can see it when they hover the mouse over the course ribbon and decide whether they have time to complete the unit.

   The **Display Name** text appears as a header for the video in Studio.

   ![Video display name](image)

   It also appears when the student hovers the mouse over the unit in the course ribbon, and as a heading above the video.
5. In the Download Video field, enter the URL for the "Welcome to edX101!" video from the table above. A link to this URL appears below the video player. The student can click the link to download a copy of the video.

6. In the Video Sources field, click Add. In the field that appears, enter the URL for the "Welcome to edX101!" video. If the YouTube video does not play, this backup video will start playing in the edX video player.

   Note This URL may be the same URL that you enter in the Download Video field.

   If you have more copies of the video hosted in different places, click Add again, and then enter the URL of the video in the field that appears. Repeat this action for each additional video.

7. In the YouTube ID field, replace the existing numbers and letters with the YouTube ID for the "Welcome to edX101!" video from the table above. This ID is case sensitive, so be careful to watch for uppercase and lowercase letters in the YouTube code.

8. Click Save.

The video component editor includes several other fields as well. You’ll learn about these in the Advanced Studio Tutorial lesson.
Create a Discussion Component

Introduction

In this unit, you create a Discussion component under the video that you just added.

Before you add a Discussion component, it is generally a good idea to add an HTML component that introduces the topic to be discussed. The Discussion component itself does not contain any text and may be easy for students to overlook. In this unit, you’ll create an HTML component followed by a Discussion component.

Create a Discussion Component

1. Directly under the Video component that you just created, create a new HTML component. Name the HTML component "Discussion Question: Welcome to edX 101!" and add the following text. (Make sure to format "Discussion Question: Welcome to edX101!" as a heading 2.)

   Discussion Question: Welcome to edX101!

   Anant Agarwal mentions several ways to present content in an online medium. In the discussion space below, mention one of these and share your thoughts about how to effectively incorporate it in an online course.

2. Directly under the HTML component that you just created, create a Discussion component by clicking Discussion under Add New Component.

3. When the following box appears, click Edit.

   The Discussion component editor opens.
4. Fill in the fields for your discussion. For this practice discussion component, you’ll name the categories and discussion according to the Sal Khan video.
   a. In the **Category** box, type **Videos**.
   b. In the **Display Name** box, type **Discussion: Welcome to edX101!**
   c. In the **Subcategory** box, type **Welcome to edX101!**

   The category and subcategory names appear in the list of discussions on the left side of the **Discussion** tab in the LMS. They do not appear in Studio or in the discussion space inside the unit.

   The display name appears when a student hovers the mouse over the course ribbon.

   ![Editing Discussion: Welcome to edX101!](image)

5. Click **Save**.

6. On the unit page, click **Preview**. The unit opens in the LMS.

In the LMS, the HTML and Discussion component names appear in the course ribbon at the top of the page.

   ![Preview](image)

The discussion space appears under the Video and HTML components that you just created. It doesn’t have a label in the body of the unit. Instead, you’ll see “Show discussion” or “Hide discussion” on the left, and a blue **New Post** button on the right.
If you click the **Discussion** tab at the top of the page, you can find the category and subcategory of the discussion in the left pane.
Create Problem Components

In this unit, you’ll create three relatively simple problems: multiple choice, dropdown, and numerical input.

Studio has many different problem templates that you can use to create problems in your course. All but one of these templates includes sample text. (The exception is the blank common problem template, which allows you to create a custom problem.) The sample text includes an example problem as well as information about the way that the problem works. To create a problem, you’ll replace the sample text with your own text.

There are also more advanced templates for complex problems. We’ll cover an example advanced problem in the Advanced Studio Tutorial.

Create a Multiple Choice Problem

1. Under Add New Component, click Problem.
2. In the Select Problem Component Type screen, click Multiple Choice.

3. When the new Problem component appears, click Edit.

The component editor for the multiple choice problem opens.

By default, the multiple choice problem type includes the following text.
4. Replace the sample problem text with the following text.

Which of the following is a real edX course?

CS101x: Introduction to Social Media
POET507x: Writing Engaging Limericks and Haiku
6.002x: Circuits and Electronics
MOOD304x: Keeping Students Happy

[explanation] Anant Agarwal's "6.002: Circuits and Electronics" is an online version of the course he taught at MIT. [explanation]

5. Select the following text, and then click the multiple choice icon in the menu bar at the top of the component editor.

CS101x: Introduction to Social Media
POET507x: Writing Engaging Limericks and Haiku
6.002x: Circuits and Electronics
MOOD304x: Keeping Students Happy

The component editor adds the formatting for multiple choice options. Each answer option is preceded by a pair of parentheses. Add an "x" between the parentheses next to 6.002x: Circuits and Electronics to designate the correct answer.

6. In the component editor, click Settings in the upper-right corner.
7. In the **Display Name** field, change the text to **edX Courses**. Leave the other attributes as-is for now. You'll learn about them in the next step.

8. Click **Save**.

The problem appears as follows in the LMS.

![Image of problem settings]

**Edit Problem Settings**

As you saw in the multiple choice problem that you just created, every problem has several settings, or attributes, that appear when you click **Settings** in the component editor.
These settings are **Display Name, Maximum Attempts, Problem Weight, Randomization,** and **Show Answer.**

- **Display Name** is the name of your problem. The name appears as a heading over the problem in the LMS. It also appears in the course ribbon at the top of the page.

- **Maximum Attempts** indicates the number of times a student can try to answer the problem. By default, a student has an unlimited number of attempts.

- **Problem Weight** determines the maximum number of points the problem is worth. You can change the maximum number of points possible by changing the **Problem Weight** value.

  It is also possible to create a problem that has more than one answer space—for example, you may create a problem that has two questions, one with a multiple choice answer and one with a numerical value answer. By default, each answer space is worth one point. Thus this problem would be worth two points. For now, we suggest that you leave the default value.

- **Randomization** defines how often inputs are randomized when a student loads the problem. This setting only applies to problems that can have randomly generated numeric values.

- **Show Answer** defines when the problem shows the answer to the student. There are seven options.
  
  o **Always**: Always show the answer when the student clicks the **Show Answer** button.
  
  o **Answered**: Show the answer after the student has submitted her final answer.
  
  o **Attempted**: Show the answer after the student has tried to answer the problem one time, whether or not the student answered the problem correctly.
  
  o **Closed**: Show the answer after the student has used up all his attempts to answer the problem or the due date has passed.
  
  o **Finished**: Show the answer after the student has answered the problem correctly, the student has no attempts left, or the problem due date has passed.
  
  o **Past Due**: Show the answer after the due date for the problem has passed.
  
  o **Never**: Never show the answer. In this case, the **Show Answer** button does not appear next to the problem in Studio or in the LMS.

Now that you've learned about the problem settings, open the multiple choice problem that you just created. Change the settings any way you want. When you're done, click **Save** to save the settings, or click **Cancel** to discard your changes.
Additionally, each problem has a Check button. By default, each Check button corresponds to one problem. If you want students to be able to check their answers to multiple problems at a time, you can include more than one problem in each Problem component.

Create a Numerical Input Problem

1. Under Add New Component, click Problem.
2. In the Select Problem Component Type screen, click Numerical Input.
3. When the new Problem component with sample template text appears, click Edit.

The component editor for the numerical input problem opens.

4. Replace the sample problem text with the following, and then click Save.

How many different countries do edX students live in as of May 2013?

= 190 +− 5%

[explanation]As of edX's first birthday, in May 2013, edX students live in 193 different countries. [explanation]

Note that the answer appears after the equal sign, and that we have included a margin of error of plus or minus 5%. If you do not want to include a margin of error, delete = +− 5% from your problem text.

5. Click Settings, and then change the text in the Display Name field to edX Students.
Leave the other attributes as-is.
6. Click Save.

Create a Checkbox Problem

1. Under Add New Component, click Problem.
2. In the Select Problem Component Type screen, click Blank Common Problem.
3. In the Problem component that appears, click Edit.
4. In the component editor, type the following.
What types of poetry currently appear in edX101? Select all that apply.

cinquain
elfchen
haiku
limerick
Petrarchan sonnet
Shakespearean sonnet

5. In the component editor, select all the answer options, and then click the checkbox icon.

6. Add an x between the brackets for "haiku" and "limerick." The options should now appear as follows.

[ ] cinquain
[ ] elfchen
[x] haiku
[x] limerick
[ ] Petrarchan sonnet
[ ] Shakespearean sonnet

7. Click Save.

View the Problems in the LMS

In the LMS, the three problems that you've just created appear in the course ribbon at the top of the page.

They also appear after the Discussion component in the body of the unit.
EDX COURSES
Which of the following is a real edX course?
- CS101x: Introduction to Social Media
- POET102x: Writing Engaging Limericks and Haiku
- 6.002x: Circuits and Electronics
- MODD104x: Keeping Students Happy

EDX STUDENTS
How many different countries do edX students live in as of May 2013?

EDX101 POETRY
What types of poetry currently appear in edX101? Select all that apply.
- cinquain
- eftren
- haiku
- limerick
- Petrarchan sonnet
- Shakespearean sonnet
Reorder the Unit

Now that you've created example components for each component type, return to the unit page in Studio. (You return to the unit page every time you save a component.)

Studio includes a drag-and-drop feature that you can use to change the order of the components in the unit. To use this feature, hover the mouse over the right edge of a
Drag and drop a component until the gray bar turns blue and the multi-directional arrow cursor appears. A tooltip that says Drag to reorder may also appear.

Hold down the mouse button, drag the component to the location where you want it, and then release the mouse button.

Go ahead and experiment with the drag-and-drop feature, reordering the components in this unit any way you want. When you’re done, click Subsection 1a in the Unit Location box in the right pane to return to the Subsection 1a page.

Note Drag and drop also works for units, subsections, and sections.
Visibility

When you create a course on Studio, students cannot see any of your course content until the course start date has passed. After the course start date has passed, whether a student can see your course materials depends on two settings that you can control: release dates and visibility.

- **Release dates** specify the dates when content appears on the internet. They are set at the section and subsection levels. Neither a section nor its contents are visible to students until the release date passes. However, course staff can see content regardless of its release date.

  By default, subsections have the same release date as the section they are in. However, you can change the release date of each subsection so that you release individual subsections at different times after you release the section.

- **Visibility** specifies whether individual units are set to Private or Public. This setting is specified at the unit level. By default, all new units are set to Private. Neither students nor staff can see content if its visibility is set to Private.

You can view your content on the LMS to make sure that it appears the way you want, even if the release date hasn't passed or the content is set to Private. There are three ways to view your content on the LMS.

- **In Student view in Live mode.** In Student view, you see your content as a student would. You can only see content if its release date has passed and the unit's visibility is set to Public.

- **In Staff view in Live mode.** In Staff view, you view content as a staff member would see it. You can see content even if its release date hasn't passed, but you cannot see content unless the unit's visibility is set to Public.

- **In Preview mode.** In Preview mode you can see all the content for your course, regardless of release date or visibility setting. Preview mode is the only way to see content whose visibility is set to Private.

**LMS STUDENT VIEW**

When you use Student view, you can only see material that meets all three publishing conditions:

- The course start date has passed.
- The release dates for the section and subsection have passed.
- The unit that contains the material is set to Public.

To view your course as a student, click **View Live** in Studio. **View Live** appears on the **Course Outline** page, on any subsection page, and on the unit page if you have changed the Visibility setting of the unit from Private to Public.
When you click **View Live**, the LMS opens in a new browser tab. In the LMS, look at the upper-right corner. If **Student view** appears in the upper-right corner, you are in Student view. (If **Staff view** appears instead, click **Staff view** to switch to Student view.)
Again, whether you can see your content in Student view depends on its release date and visibility. If the release date for the section has passed, but the release date for the subsection has not passed, the student can see the section heading in the left pane. However, the student cannot see the subsection heading or any of the subsection’s content.

In the following example, the release date for Section 1 is January 1, 2000. The release date for Subsection 1a is January 1, 2010. The release date for Subsection 1b is January 1, 2099.

In the LMS, students can see Subsection 1a. However, Subsection 1b does not appear in the left pane.
Remember that in addition to release dates, content visibility depends on the individual unit’s Visibility setting. For students to see your content, its visibility must be set to Public.

In the following example, the release date for the subsection is January 1, 2010. Thus, the students can see all content in the subsection that is set to Public. The visibility of unit 1 is set to Public. However, the visibility of unit 2 is set to Private.

In the LMS, students can only see content for unit 1. Unit 2 does not appear in the course accordion.
If you change the visibility of unit 2 to Public, students can see content for both units, and both units appear in the course accordion.

**LMS STAFF VIEW**

When you use Staff view, you can see any units whose visibility is set to Public, whether or not the release dates have passed. You cannot see any content whose visibility is still set to Private.

To view your course as a staff member, click **View Live** in Studio. **View Live** appears on the **Course Outline** page, on any subsection page, and on the unit page if you have changed the Visibility setting of the unit from Private to Public.
When you click **View Live**, the LMS opens in a new browser tab. In the LMS, look at the upper-right corner. If **Staff view** appears in the upper-right corner, you are in Staff view. (If **Student view** appears instead, click **Student view** to switch to Staff view.)

In the following example, the release date for Section 1 is January 1, 2000.
The release date for Subsection 1a is January 1, 2010. In Subsection 1a, Unit 1 is public, and Unit 2 is private.

The release date for Subsection 1b is January 1, 2099. In Subsection 1b, Unit 1 is private, and Unit 2 is public.

In the LMS, both subsections are listed in the left pane. However, in Subsection 1a, you cannot see Unit 2.

In Subsection 1b, you cannot see Unit 1.
If you change the visibility of all the units to Public, you can see both units in Subsection 1a and in Subsection 1b.

Subsection 1a

Subsection 1b

edX101: How to Create an edX Course
PREVIEW MODE

When you use Preview mode, you see all the units of your course, regardless of whether they are set to Public or Private and regardless of whether the release dates have passed. Only instructors can use Preview mode.

Using Preview mode is the only way to see content that is set to Private.

You can enter Preview mode from a subsection page or from a unit page.

On any subsection page, click Preview Drafts.

On a unit page for a unit whose visibility is set to Private, click Preview.

If you have changed the visibility of a unit to Public, the View Live button appears on that unit’s page, but the Preview button does not appear.
If you click **View Live**, you see the content as it appears in the current course on the LMS.

**Example**

In the following example, the release date for Section 1 is January 1, 2000.

The release date for Subsection 1a is January 1, 2010. In Subsection 1a, Unit 1 is public, and Unit 2 is private.

The release date for Subsection 1b is January 1, 2099. In Subsection 1b, Unit 1 is private, and Unit 2 is public.

In the LMS, you can see both units in Subsection 1a and in Subsection 1b.

**Subsection 1a**
REVISING CONTENT

In Studio, the visibility of every unit is set to Private by default. When the visibility is set to Private, the Preview button appears on the unit's page. The View Live button does not appear.
When you change the visibility setting of a unit from Private to Public, you publish the unit and its contents to the web. After you publish the content, the Preview button no longer appears for that unit. Instead, the View Live button appears.

If you want to make revisions to content that has been published, you create and edit a draft of that unit. To create a draft, go to the unit's page, and then click edit a draft in the right pane.

When you edit a draft of a unit, you can view the unit's contents in two ways.

- To view the already-published content as it appears in the live course, click View the Live Version in the upper-right corner of the page.
- To view the unpublished content as you're working on it, click Preview.
For this example, the already-published content appears as follows when you click View the Live Version. (Remember, unit 2 of Subsection 1a is set to Private. It does not appear in the live course.)

The unpublished draft appears as follows when you click Preview.
When you’re ready to publish the draft of your revised content, click **replace it with this draft** in the right pane. If you decide you don’t want to keep the revised content, click **Delete Draft**.
Add a Welcome Announcement

Now that you have your instructional content ready, you can create a welcome announcement.

You can add announcements or updates to welcome students to the class, remind students of exams, notify students of changes in the course schedule, and call out important steps students need to keep in mind. These updates appear on the Course Info tab in the LMS. Because your course automatically opens to the Course Info page, students will see the update as soon as they sign in.

1. On the Course Content menu, click Updates.

2. On the Course Updates page, click New Update.

The Course Updates editor opens.

Note that the content for the editor must be formatted in HTML. We've provided some HTML for you in the next step.

In the Course Updates editor, paste the following content, making sure to include all tags.
<h1>Great work!</h1>

<p>You're almost done with the Studio tutorial. We hope you've enjoyed creating a practice course!</p>

<p>This sample template for your welcome announcement uses basic HTML. If you don't want to do anything fancy with your text, you can just surround it in paragraph tags. Since this is a basic tutorial, that's what we're going to do for now.</p>

<p>If you want to learn about how to add images to your welcome announcement or to any unit in your course, continue on to the advanced Studio tutorial in the next section. The advanced tutorial will also teach you how to add transcripts to your video, create internal links, and more. We hope to see you there!</p>

If you want to add your own text, replace the text between the heading (h1) or paragraph (p) tags with your text.

3. **Click Save.** Your welcome announcement appears in Studio.

To view your announcement in your course, follow these steps.

1. **On the Course Content menu, click Outline.**

2. **In the upper-right corner of the Course Outline page, click View Live.**

The LMS opens in a new browser tab.
3. In the course navigation bar, click the **Course Info** tab. Your course announcements appear on the **Course Info** page, in reverse chronological order.

![Course Updates & News](image)

**Great work!**

You’re almost done with the Studio tutorial. We hope you’ve enjoyed creating a practice course!

This sample template for your welcome announcement uses basic HTML. If you don’t want to do anything fancy with your text, you can just surround it in paragraph tags. Since this is a basic tutorial, that’s what we’re going to do for now.

If you want to learn about how to add images to your welcome announcement or to any unit in your course, continue on to the advanced Studio tutorial in the next section. The advanced tutorial will also teach you how to add transcripts to your video, create internal links, and more. We hope to see you there!
Advanced Studio Tutorial (optional)

Roadmap for the Advanced Studio Tutorial

Now that you've worked with each of the components in Studio, you can go back and add advanced functions to those units--including images, multi-speed video, transcripts, advanced problems, and more. We'll do this in the next few units. You can use the links below to skip to a specific unit at any time.

- Add a Link to a Course Unit: In this unit, you'll add a link from one unit of your course to another unit.
- Add an Image to a Unit: In this unit, you'll add a picture to one of the Problem components that you created in the Studio tutorial. You'll then add a picture to an HTML component.
- Add an Advanced Problem: This unit covers adding an image response problem, one of the advanced problem types, to your course.
- Add Transcripts to Videos: This unit describes how to add transcripts to your course videos.

- Specify Video Settings: In this unit, you'll specify additional settings for your video, such as a start or end time and whether you want transcripts to appear by default.

- Seed a Discussion Space: In this unit, you'll get a discussion started by adding your own anonymous post to a discussion space.

- Create a Mini-Course: In this unit, you'll use the “Introduction to edX Studio” video to create your own course, complete with problems and discussions.
Add a Link to a Course Unit

Adding a link from one unit to another in your course is very similar to adding a link to an external website, which you did in the Create an HTML Component unit in the Studio Tutorial lesson. The main difference is that instead of an external URL, you need the the **unit identifier** of the unit that you want to link to.

1. Determine the unit identifier of the unit you’re linking to. To do this, open the unit page in Studio, and locate the **Unit Identifier** field under **Unit Location** in the right pane.

2. Double-click the text in the **Unit Identifier** field, and then press CTRL+C or COMMAND+C to copy the text.

3. Open the unit that you want to link from, and then open the HTML component where you want to add the link.

4. Select the text that you want to make into the link. (In the Create an HTML Component unit, you selected the word "prudence.")

5. Click the link icon in the toolbar.

6. In the **Insert/Edit Link** dialog box, enter the following in the **Link URL** field. Make sure to replace `<unit identifier>` (including the brackets) with the unit identifier that you copied in step 2, and make sure to include both forward slashes (/).

   `/jump_to_id/<unit identifier>`

   The text in the **Insert/Edit Link** resembles the following.
7. If you want the link to open in a new window, click the drop-down arrow next to the Target field, and then select **Open Link in a New Window**. If not, you can leave the default value.

8. Click **Insert**.
Add an Image to a Unit

Introduction

Adding an image to a unit in your course has two steps:

1. Upload the image file to the Files & Uploads page.
2. Create a link to that image file in the unit.

Upload a File to the Files & Uploads Page

To upload a file to the Files & Uploads page:

1. Download the 6002x_CircuitsAndElectronics.gif file below, and then save it on your desktop or in a place you'll remember. (To do this, right-click the image, click Save picture as or Save image as, and then select the location where you want to save the file.)

2. On the navigation bar, click Course Content, and then click Files & Uploads.


4. In the Upload New File dialog box, click Choose File.
5. In the **Open** dialog box, locate the 6002x_CircuitsAndElectronics.jpg file that you just downloaded, and then click **Open**.

   **Note** You can upload more than one file at a time. Use the **SHIFT** or **CTRL** keys (**SHIFT** or **COMMAND** on a Mac) to select the files that you want, and then click **Open**.

6. When you see the **Upload completed** message, make a note of the URL that appears in the URL field. The URL uses the format `/static/<name of file>`. You will use this URL when you add the image to a component in your course. (You can also see this URL at any time to the right of the file on the **Files & Uploads** page.)

   ![Upload New File](image)

7. To add more files, click **Load Another File**. To close the dialog box, click the x in the top right corner.

   When you close the dialog box, the file appears on the **Files & Uploads** page. You can see the file's URL to the right of the file name and date added.

   ![Files & Uploads](image)

   If you ever want to replace this file, you can upload a new file that has the same name as the file that you want to replace. The new file replaces the current file.

   If you want to delete a file, click the trash can icon to the right of the file's URL.

### Create a Link to the Image in a Problem Component

To create a link to the image in your course, you add the code for the image in the component where you want the image. The code is the URL for your image, plus a tag. You access the code by opening the Problem component that you want, and then clicking **Advanced Editor** to open the advanced editor.

In this example, you'll add an image to the "edX Courses" problem that you created in the Create Problem Components unit in the Studio Tutorial lesson.
1. In your practice course, open the **Subsection 1a** unit. This unit contains the **edX Courses** component.

2. Open the **edX Courses** component by clicking **Edit** in the upper-left corner of the component.

3. In the component, click **Advanced Editor**. Click **OK** in the dialog box that appears.

4. In the advanced editor, add the following to the text of the problem.

   (Hint: The course's About page image appears below the problem.)

5. Below the `<multiplechoicerresponse>` tag and above the `<solution>` tag, add the code for the `6002x_CircuitsAndElectronics.gif` image that you uploaded to the **Files & Uploads** page. The code is an image tag that includes the URL of your image file. It uses the following format.

   `<img src="/static/NAME_OF_FILE" />`

   The URL for the `6002x_CircuitsAndElectronics.gif` image is the following. (You can copy and paste this URL from the **Files & Uploads** page.)

   `/static/6002x_CircuitsAndElectronics.gif`

   Thus, the code for this image is the following.
The code appears as follows between the `<multiplechoiceresponse>` and `<solution>` tags in the component editor.

6. Click Save to save and close the component editor.

On the unit page, the component appears as follows.

In the LMS, the problem appears as follows.
Create a Link to an Image in an HTML Component

Creating a link to an image in an HTML component is very similar to creating the link in a Problem component. In both cases, you add code that includes an image tag and the image URL to the component.

In this example, you'll create an HTML component, then add an image to that component.

1. Download the Haiku_Picture.gif image below. Save it to your desktop or another location that you'll remember.

2. Upload the Haiku_Picture.jpg image to the Files & Uploads page. Make a note of the URL for the file.

3. Create an HTML component. Name it Haiku, and then add the following text:

   An edX Haiku
   Start an online course
   And teach in your pajamas
   Nobody will know

4. Format the text any way you want. In this example, the title "An edX Haiku" uses Heading 2 tags.
5. In the HTML component editor, click the HTML tab.

![HTML tab](image)

AN EDX HAIKU
Start an online course
And teach in your pajamas
Nobody will know

6. After the last line, add an image tag that contains the URL for the image. This tag is the same as the tag that you added in the last unit. Only the file name is different. In this example, the image tag is the following.

```html
<img src="/static/Haiku_Picture.gif" />
```

7. To see the component in Visual view, switch back to the Visual tab.

8. Click Save.

In the LMS, the component appears as follows.
AN EDX HAiku

Start an online course
And teach in your pajamas
Nobody will know
Add an Advanced Problem

In this unit, you'll add an image mapped input problem. This problem type, like all advanced problem types, opens directly in the advanced editor that you saw in Add an Image to a Unit.

In an image response problem, students click inside a defined area in an image. You define this area by including coordinates in the body of the problem.

1. Add the image that you'll use in your problem. To do this, download the Shapes.gif image below to your computer, and then upload it to the Files & Uploads page.

2. In the unit where you want to add the problem, click Problem under Add Component, and then click Image Mapped Input.

3. In the upper-left corner of the component that appears, click Edit.
   The component editor opens, and you can see the sample problem.
4. Replace all of the text in the problem with the following code. The placeholders are explained right after the code.

```html
<p>In the image below, click anywhere on the triangle.</p>
</problem>

<p>A triangle has three sides. Thus, the yellow object is the triangle.</p>

>Note All coordinates start with (0,0) at the upper left corner and increase in value toward the lower right corner, very similar to the progression of reading English. Thus coordinates of (1,1) would represent a point in the top left corner of the image, (1,75) would represent a point on the far left side and vertical center of the image, (220,75) would represent a point on the far right side and vertical center of the image, and (220,150) would represent a point in the lower right corner.

/static/Shapes.gif is the path to the image file that you uploaded.

220 is the width of the image, in pixels.

150 is the height of the image, in pixels.

The values inside each set of parentheses represent the starting coordinates (80,40) and ending coordinates (130,90) of the area where the student should click. The coordinates form two opposite corners of a rectangle that designates the correct area in the image. The first number in each pair represents the value on the x axis, and the second represents the value on the y axis.

Note All coordinates start with (0,0) at the upper left corner and increase in value toward the lower right corner, very similar to the progression of reading English. Thus coordinates of (1,1) would represent a point in the top left corner of the image, (1,75) would represent a point on the far left side and vertical center of the image, (220,75) would represent a point on the far right side and vertical center of the image, and (220,150) would represent a point in the lower right corner.
5. If you want to include an explanation for the problem, you can add the following after the `<imageresponse>` tags and before the closing `<problem>` tag. The text of the explanation appears between the `<p>` tags.

```html
<solution>
<div class="detailed-solution">
<p>A triangle has three sides. Thus, the yellow object is the triangle.</p>
</div>
</solution>
```

The complete code for this problem appears as follows in the component editor.

6. Click **Save**.

7. Click **Preview** on the unit page to view the problem in the LMS.

In the LMS, the problem appears as follows. Feel free to click inside the problem, and then click **Check** to check your answer.
Specify Video Settings

In addition to Display Name, Download Video, Video Sources, and YouTube ID, the video component editor has the following settings:

- **Download Track**: This feature will be implemented in a future version of Studio.

- **End Time**: This setting specifies the number of seconds after 0:00 when the video will automatically stop playing. Note that you must enter the value as the number of seconds rather than using the minute:second format. For example, if you have a 10-minute video and you want the video to stop playing at 4:23, you would enter 263 in the **End Time** field.

- **HTML5 Subtitles**: This feature will be available in a future version of Studio.

- **Show Captions**: This setting specifies whether you want the transcript to show by default. Students can always turn transcripts on or off while they watch the video.

- **Start Time**: This setting specifies the number of seconds after the technical start of the video when the video will start playing. Note that you must enter this value as a number of seconds rather than using the minute:second format. For example, if you have a 10-minute video and you want the video to start at 3:47 instead of at 0:00, you would enter 227 in the **Start Time** field.

- **YouTube ID for .75x speed**: If you have a separate YouTube video for the .75x-speed version of your video, you can enter that YouTube ID here. If you don't use this setting, students with the most updated versions of the major browsers can still adjust the speed of the video while they view it.

- **YouTube ID for 1.25x speed**: If you have a separate YouTube video for the 1.25x-speed version of your video, you can enter that YouTube ID here. If you don't use this setting, students with the most updated versions of the major browsers can still adjust the speed of the video while they view it.

- **YouTube ID for 1.5x speed**: If you have a separate YouTube video for the 1.5x-speed version of your video, you can enter that YouTube ID here. If you don't use this setting, students with the most updated versions of the major browsers can still adjust the speed of the video while they view it.
Add Transcripts to Videos

Many students appreciate seeing a transcript of your videos--particularly if they are not native English speakers. Students can view the entire transcript even if the video isn't playing.

The Studio video player uses a custom edX transcript format, **.srt.sjson**. You have two options for obtaining your .srt.sjson files.

- Obtain the files from [3Play Media](https://www.3playmedia.com) *(strongly recommended)*. 3Play Media can create files that use the .srt.sjson format, and you can then download these files. You don’t have to make any changes to the content of the file.

- Obtain an .srt file, and then convert the .srt file to an .srt.sjson file. For information about how to do this, see [Converting .srt to .srt.sjson](#). This process is fairly complex, however, and requires you to run a Python script. Again, we recommend that you obtain your .srt.sjson files from 3Play Media.

When you have your .srt.sjson file, all you have to do to add a transcript is change the name of the .srt.sjson file, and then upload that file to the **Files & Uploads** page. You don’t have to make any changes in the Video component.

Instructions

1. Obtain your .srt.sjson file, either from 3Play or by converting an .srt file. Save the file to your desktop or to another location that's easy to remember.

2. Obtain the YouTube ID for your video.

3. Change the name of the .srt.sjson file to the following, where **YOUTUBEID** is the YouTube ID for the video.

   ```
   subs_YOUTUBEID.srt.sjson
   ```

   For example, if the YouTube ID for your video is **AbCdEf12345**, the new name of the .srt.sjson file is **subs_AbCdEf12345.srt.sjson**.

4. Upload the .srt.sjson file to the **Files & Uploads** page. *(For more information about how to do this, see Upload a File to the Files & Uploads Page.)*

When you open the unit that contains your video, you can see the transcript to the right of the video image. The transcript plays automatically when you start the video.
Seed a Discussion Space

In the Studio tutorial, you created a Discussion component. In this advanced tutorial, you'll get the discussion started by posting your own answer, or "seeding" the discussion space.

Many students may feel hesitant to be the first to post in a discussion. When you seed the discussion space, you can post anonymously or as a student, so that students will be more comfortable replying if they disagree with your post.

To post as a student, follow the steps below. If you later want to reply as yourself, log back into your usual account and omit steps 1 and 2.

1. Set up a test account on edX or Edge with an e-mail address that is not associated with your Course Team.
2. Go to your course URL and register for your course.
3. On edX or Edge, locate the unit that contains the Discussion component.
4. In the unit, locate the discussion space.
5. Click New post.
6. Type a title for your post in the Title box, and then enter text for your post.
7. If you want to, select the post anonymously check box or the follow this post check box.
8. When you are satisfied with your post, click Add Post.

Your new post appears at the top of list in the unit. Posts are listed in reverse chronological order.
Create a Mini-Course

Now that you have created a practice unit that includes each of the components--Discussion, HTML, Problem, and Video--and you’ve added images, transcripts, and an advanced problem, you can create a new course of your own. You can base your course on the "Introduction to edX Studio" video, or you can use your own materials. The YouTube ID for the "Introduction to edX Studio" video is 26T82YVScT0.

Feel free to create as many sections, subsections, and units as you want, and add the video, HTML components, problems, and discussion spaces in any order you want. You can add an unlimited number of components in a unit. In general, however, we recommend that you include just two or three components in a unit.

When you've created some content, use drag and drop to rearrange it in any way you like. If you want to delete any units, subsections, or sections, you can click the trash can icon to the right of that item on the Course Outline page. Be careful when you delete items, however--this action can't be undone.

After you've created your mini-course, you can share it with other students in edX101 by adding your course's URL to the discussion space below. As your fellow edX101 students share their courses, you can look at and comment on each other’s work.

(Discussion space provided)
Homework 1

Introduction

In this homework assignment, you'll create your own mini-course and then share the course with your peers. We'll provide you with videos and a textbook, and you can use these materials to create exercises, discussion spaces, quizzes, and anything else you want your mini-course to include.
Mini-Course: The Stable Marriage Problem

In 2012, Dr. Lloyd Shapley received a share of the Nobel Prize in Economics for his work on the "stable marriage problem"--a problem in game theory that mathematically creates stable marriages in any given population of equal numbers of men and women. In the next several units, Harvard post-doc Emily Riehl will explain this problem in four videos. We'll also provide a textbook that further describes the problem. You'll then create a mini-course, using these videos and the textbook along with simple exercises that you design. When you're done, you and your peers will share and discuss your mini-courses in a provided discussion space.

Note that you don't have to create an About page, a syllabus, a schedule, or other materials for this exercise.

This project is a collaboration with Girls’ Angle, a non-profit dedicated to nurturing girls' interest in mathematics.
EMILY RIEHL: Hi, I'm Emily Riehl. I'm a post-doc at Harvard University. And today I'm going to tell you about the stable marriage problem.

So the stable marriage problem was introduced by David Gale and Lloyd Shapley in a 1962 paper on college admissions and the stability of marriage. The authors thought of this problem as a mere mathematical diversion. But it turns out there are actually a surprising number of real world applications, in recognition for which Shapley received a share of the 2012 Nobel Prize in Economics.

So what is the problem? The stable marriage problem is something of a matchmaker's dilemma. So imagine we have a fixed dating pool of single men and single women, each of whom know each other. More precisely I guess we'll assume that each individual has a ranked list of preferences of the members of the opposite sex, and that these preferences are unchanging over time. So the objective then is to find what we will call a stable marriage configuration. And what I mean by this is we want to arrange marriages for everyone in the dating pool-- these are the one man, one woman sort of marriages-- in such a way that the following does not occur. So I'll draw a little cartoon to illustrate.

If we have one couple whose marriage we've arranged, this man and this woman, and we have a second couple here, we'll say that our marriage arrangement is unstable, so not stable, if the following occurs: If this woman prefers this man to her assigned husband, and simultaneously, if this man prefers this woman to his assigned wife. In this situation then this unmatched couple, this man and this woman, will be tempted to elope, to run away from the arrangement and break our assigned marriages. So this is what we want to avoid. This is the situation we're hoping does not occur.

So the mathematical problem then: Is there always a stable marriage configuration? That is, for any dating pool, for any collection of single men and single women, for any list of individual preferences, is it always possible to arrange marriages for everyone in the dating pool so that this sort of instability does not arise? It's not at all obvious that the answer is yes. But if it
happens to be, a further question is, how do you do it? How do you arrange the marriages? We’ll see that this problem can in fact be solved. And the solution is due to a 1962 algorithm of Gale and Shapley, which is the subject we will turn to next.
EMILY RIEHL: So now I'd like to tell you about the Gale-Shapley algorithm. So we're supposing before we start that we're given a dating pool consisting of equal numbers of single men and single women, and that for each individual in the dating pool, we know their preferences over each member of the opposite sex. And again, we're assuming these preferences won't change over time. Now it turns out the algorithm works in essentially the same way if we have unequal numbers of men and women, and if some individuals don't rank every one of the members of the opposite sex. But let me make these simplifying assumptions.

OK, so how does the algorithm work? To begin, on the very first day, each man in the dating pool proposes to his top choice woman. Some women will receive multiple proposals. Others will receive none. But then each woman rejects all but her top suitor. So each woman who receives multiple proposals rejects all but her favorite. At the end of the day, the men who have not been rejected become tentatively engaged to the woman who they proposed to.

Then on the second day, on day two, and in fact on all subsequent days, day two, three, four, and so on, each unengaged man, so each man who was rejected the night before, again proposes to his next best choice. So the women, a woman, may be tentatively engaged to somebody who has proposed to her on a previous day, and she may receive new proposals as well. The women now have an opportunity to trade up. So they again, exactly as in the first day, will reject all but their preferred suitor. In particular, tentative engagements can be broken. And men who had been engaged previously will become single again, in which case they'll make a new proposal on the next day.

Now eventually either everyone is engaged, all the men are engaged, there are no single men to make new proposals, or everyone has run out of new people to propose to. And at that moment, the algorithm stops. And these tentative engagements become real engagements. And we have found a marriage configuration. The claim is that this is a stable marriage configuration, that this solves the problem we posed at the beginning. But before we prove this, let's consider an example.
An Example (4:19)

EMILY RIEHL: All right, let's see how the Gale-Shapley algorithm works in an example. So we'll consider a dating pool that has four women, Amy, Bev, Cat, and Dot, and four men, Jim, Ken, Leo, and Max, whose preferences are listed here in decreasing order. So Bev likes Jim the best, then Max, then Leo, and then Ken.

So the Gale-Shapley algorithm says that on the first day, on day one, each of the men propose to their top choice, which I'll indicate by circling. Jim proposes to Amy. Amy receives a proposal from Jim. Ken also proposes to Amy. Amy receives a proposal from Ken. Leo proposes to Bev. Bev receives a proposal from Leo. And Max proposes to Cat. Amy is the only one who has received multiple proposals. So she has a choice, and as she prefers Jim to Ken, she'll reject Ken. Ken is rejected by Amy. And at the end of the first day, the conclusion is that Amy has rejected Ken.

On the second day, Ken is the only one who is unengaged, having been rejected the day before. So he proposes to his next choice, Bev. Bev receives a second proposal from Ken. But she prefers Leo, to whom she is tentatively engaged from day one. So Bev then rejects Ken. Ken is rejected by Bev. And at the conclusion of the second day, Bev rejects Ken.

On the third day, Ken is the only one who is single. So he proposes to his next choice, who is Cat. Cat receives this proposal from Ken. And she prefers Max, to whom she had been tentatively engaged. So she breaks her engagement with Max, rejecting him. And Max is going to go back on the singles market. The conclusion then is that Cat rejects Max.

On the fourth day, Max is now single. So he'll propose to his next choice, which is Amy. Amy receives this new proposal from Max, but she prefers Jim, to whom she's been tentatively engaged since the first day. So Amy will reject Max. And Max is going to have to try again tomorrow. Conclusion, Amy rejects Max.

Now on the fifth day, Max proposes to his next choice, his next choice being Bev. And Bev prefers Max to Leo, to whom she had been engaged previously. So she now rejects Leo. And Leo, even though he had been tentatively engaged since the very first day, is now single once more. Bev rejects Leo.
Now on the sixth day, Leo, who is newly single, proposes to Dot, his next choice. And you can see that Dot is not too happy with Leo. He's her last choice, in fact. But she doesn't have any better offers. So she has no choice but to accept. And it is a match.

So now we have four tentative engagements that become permanent engagements, Amy to Jim, Bev to Max, Cat to Ken, and Dot to Leo. And we can check that this is, in fact, a stable marriage configuration. So Bev would rather be with Jim than with Max. But Jim is happier with Amy then he would have been with Bev. So he's not in danger of eloping with her. Dot would rather be with anybody but Leo. But each of these men is happier with his assigned wife than he would be with Dot. So that couple won't run off together either.

Now of course the fact that we've found a stable marriage configuration in this one example doesn't prove that we can do it always, that this algorithm works always. But that's the subject we'll turn to next.
EMILY RIEHL: So as a mathematician what excites me about the Gale-Shapley algorithm and the stable marriage problem more generally are the theorems that we can prove about it. So we won't be able to quite give full proofs to all of the results I'm going to talk about today right at this moment. But maybe if you go away and think about it on your own, you'll be able to discover them yourself.

So the first theorem is that the Gale-Shapley algorithm solves the stable marriage problem. So the formal way to say this-- we'll call this theorem one-- is that the Gale-Shapley algorithm finds a stable marriage configuration. That is, for any dating pool, for any preference list, this algorithm produces stable marriages, according to our definition before. And actually, the proof of this is fairly clear from our description of the algorithm. So to show that the resulting marriage configuration is stable it suffices to prove that none of the men are in danger of eloping, none of the men will ever leave their wives.

But let's think now from the perspective of one of the men, say Max in our example. Now, there are a number of women that he preferred to the wife he was assigned by the Gale-Shapley algorithm. But each of these women is then someone he proposed to along the way and who then rejected him in favor of somebody she liked better. So even though Max in this case would rather be with Cat or Amy, neither Cat nor Amy would consent to being with Max because they're happier with their assigned match. So therefore, we do in fact solve the stable marriage problem using this Gale-Shapley algorithm.

I'd like to pause now to comment on how remarkable it is that this problem does always have a solution for any dating pool. One way to illustrate this is you can imagine related problems that in fact don't have a solution. So for instance, it being 2013, you could imagine a stable same sex marriage problem. Mathematically what I mean by this of course is that we have just a single dating pool, and each individual lists preferences over all the other members in that single dating pool. Now unfortunately, this problem cannot always be solved. And indeed, you can see that our discussion of the Gale-Shapley algorithm is heteronormative, and the heteronormativity is an essential feature of our description of the algorithm. There's no way to adapt the algorithm to solve this other problem.
But returning to the stable marriage problem, now that we know that it can be solved, that there always exists a stable marriage configuration for any dating pool, we might imagine that there are multiple solutions. That is, for a given fixed dating pool, there might be multiple ways to arrange the marriages that satisfy the stability condition we talked about before.

What's really remarkable about the stable marriage configuration found by the Gale-Shapley algorithm is the following, and this is our second theorem. The theorem is that the Gale-Shapley algorithm simultaneously gives each man his best possible wife. What's meant by this is, among all the women that a given man is matched to in one of these hypothetical stable marriage configurations, his top choice among all those women is the one he's matched to by this Gale-Shapley algorithm. And this is true for every single man. The results for all the men are simultaneously optimized by the Gale-Shapley algorithm. It's really remarkable that there is a stable marriage configuration with this property, much less that this algorithm finds that particular one. Now both of these theorems are due to Gale and Shapley, and their proofs can be found in the original 1962 paper.

But there's one more result that I want to discuss that reveals, unfortunately, the hidden sexism of the Gale-Shapley algorithm. The proof of this last theorem-- we'll call it theorem 3-- is in fact identical to the proof of theorem 2. But the original authors, Gale and Shapley, happen to have missed it.

So we've just seen that the stable marriage configuration produced by the Gale-Shapley algorithm has a really remarkable property in that it simultaneously optimizes the results for all the men. Unfortunately at the same time, the Gale-Shapley algorithm gives each woman, every single woman, her worst possible husband. Again, what this means is that among all the men that a given woman might be married to in one of these stable marriage configurations, her least favorite, her worst choice, is the one that she's matched to by the Gale-Shapley algorithm. And this is true simultaneously for every single woman in the dating pool. So the men all get their best results, and the women all get their worst results. And we can illustrate this by returning to our example.

Now the presentation of the Gale-Shapley algorithm I gave was very patriarchal, you might say. But the mathematics is unchanged if I instead had the women proposing to the men, as opposed to the men proposing to the women. The same proof of theorem 1 will show that the matching produced if the women propose is again a stable marriage configuration. And let's see what happens in the example.

So if the women act as the proposers, then on the first day of the algorithm, Amy proposes to Leo. Leo is proposed to by Amy. Bev proposes to Jim. Jim is proposed to by Bev. Cat proposes to Ken. Ken receives a proposal from Cat. And Dot proposes to Max. Max is proposed to by Dot.

Now each of the four men has just a single proposal. So he has no choice but to accept. And the algorithm stops on the very first day. But we can see now that the marriages that are arranged by the women proposing, Amy to Leo, Bev to Jim, Cat to Ken, Dot to Max, are preferred by each single one of the women to the marriages arranged by the male proposing algorithm. And simultaneously, the men all prefer their partners when they proposed to their partners than when the women proposed. This is illustrating the results of theorem 2 and theorem 3 in this particular example.

That's all we have time for today. Thanks very much.
The Stable Marriage Problem
Textbook

In addition to the videos describing the stable marriage problem, we've provided a textbook that you can use in your mini-course. Click each of the five links below to open a PDF of the front matter and four chapters of the textbook. You can then save each PDF to your computer. When you create your course, you can create a textbook out of these PDFs by uploading each of them to the Textbooks page.

(The Stable Marriage Problem textbook is available in edX101.)
Create Your Mini-Course

Now that you've watched all four stable marriage videos and downloaded the textbook, create a new course based on the videos. You can set up your course any way you want, and you can include as many exercises as you want. (For more information about the exercises that you can create in Studio, see the Exercise Gallery.)

After you create your mini-course, you'll evaluate your course according to our rubric.
Evaluate Your Mini-Course

When you're done creating your mini-course, evaluate your course using the following rubric. (You'll use this same rubric to evaluate your peers' courses in the next unit.)

Does the first unit provide instructions on how to get to the next unit?
   Yes
   No

Do the videos have transcripts?
   Yes
   No

Are the videos interleaved with exercises?
   Yes
   No

Are the exercises clear and grammatically correct?
   Yes
   No

Do the exercises provide sufficiently clear instructions to the students on what to do?
   Yes
   No

Are the exercises effective in enhancing the learning experience?
   Yes
   No
Post and Discuss Your Mini-Course

Now that you've created a mini-course, you'll share your course with your peers and provide feedback for others' mini-courses.

Post Your Mini-Course

1. Find the URL for your mini-course:
   a. Click the Settings menu in Studio, and then click Schedule & Details.
   b. In the Basic Information section, locate the Course Summary Page field. The URL that appears in that field is the URL for your course.

2. In the discussion space below, post a link to your mini-course. Make sure the post title is "Request for feedback on mini-course: <URL>" so that the post is easy to identify when you're looking at the Discussion tab.

Provide Feedback for Other Courses

For this part of the assignment, you'll take one of your peers' courses, and then provide feedback for that course.

When you provide feedback you'll use the same rubric you used for your own course, but you'll use it slightly differently. This time, instead of answering "yes" or "no" for each category, you'll give the course a score of 0-5. Write the score for each category in the comments for the post for that course. If you give the course less than a 5 in any category, include a comment explaining the score along with your general comments.

1. Does the first unit provide instructions on how to get to the next unit?
2. Do the videos have transcripts?
3. Are the videos interleaved with exercises?
4. Are the exercises clear and grammatically correct?
5. Do the exercises provide sufficiently clear instructions to the students on what to do?
6. Are the exercises effective in enhancing the learning experience?
Structuring Your Course
Your Course Info

Overview

In this lesson, you create and post the preliminary materials for your course. These generally include materials such as a schedule, textbook, and syllabus. In this early version of Studio you will use HTML to create some of these materials; we have provided templates when that’s the case.

The first part of this lesson covers your options for presenting your course materials—custom pages and the Course Handouts sidebar. You will then create online versions of the following course materials.

- Schedule
- Textbook
- Grading policies
- Syllabus

In addition, you may include materials around specific aspects of online learning, such as the following.

- Collaboration guidelines
- Forum guidelines

Note that there is more than one way to create any of the course materials that this lesson covers. The purpose of your practice course is to give you leeway to find methods and items that work best for you and your course. Feel free to adapt any of our suggestions to your own situation, and be sure to post best practices in the edX101 wiki or discussion forum!
Intellectual Property

A note about intellectual property:

It is your university's responsibility to ensure that your online course complies with applicable copyright laws. Copyright laws in the MOOC context can be very different from copyright laws for individual classes at a university. For example, some instructors use pictures or movie clips to illustrate a point in a traditional classroom. However, use of the same pictures or clips may not be permitted in online courses without explicit permission from the copyright holder.

Please check with your university counsel and research librarians for guidance about proper licensing and permissions in the MOOC context, and to ensure that your online course is compliant.
Presenting Course Materials

There are two natural places to present your preliminary course materials:

- Custom pages
- The Handouts sidebar on the Course Info page

Custom Pages

Custom pages (also called “static pages”) appear on the navigation bar in your course along with the standard pages (including Courseware and Course Info). For example, the following navigation bar shows Schedule, Poor Economics, and Additional Readings custom pages.

Custom pages are accessible from any page in the class. They are ideal for material that students will need to access easily and often.

Action: Create a Custom Page in the Practice Course

Create a custom page for your practice course. We’ve provided an HTML template that includes placeholders for you to use.

1. Follow the step-by-step directions for creating a custom page in Getting Started with Studio.

   Note When the editor opens in Studio, you can use the default Visual view to add content without working directly in HTML. However, if you use the provided HTML template, you need to click the HTML tab on the right side of the editor to open HTML view, and then replace all of the existing HTML in the editor with the HTML from the template.

2. After you create your custom page using the template, view the course either in Live mode or Preview mode. To do this, click the name of your course in the upper left corner of the screen, and then click View Live in the upper right corner of the Course Outline page. You can then click the name of your new custom page at the top of the screen from any page in the course.

3. Re-open your custom page and make changes, either in Visual view or in HTML view. Save your changes.

4. View your course in Live or Preview mode again.
Examples of Custom Pages

The CS 169.1 course has a custom Syllabus page that includes the course’s grading rubric, expected student workload, and schedule. The following image shows the Syllabus page for CS 169.1.

The following are more examples of the ways various instructors have used custom pages.
CS 169.1: Tutorials & Resources

Tutorials & Resources

There are quite a few free online resources available for learning the material covered in this course. Below is a list of some that you may find useful. You can also check the wiki resources page.

- Ruby:
  - tryruby.org (Beginner)
  - rubymon.com (beginner/intermediate)
  - rubyvoice.com (intermediate/advanced)
- HTML & CSS:
  - W3 Schools (Beginner)
- Git & GitHub:
  - try.github.com (beginner)
  - Understanding Git Conceptually (Intermediate)
  - git weekly (Beginning to advanced articles, e.g. installing Git, pushing and pulling)
  - An Illustrated Guide to Git on Windows
  - Git for Computer Scientists (advanced)
  - Git Magic (advanced)

Harvard ER22x Weekly Forum Digest

WEEKLY FORUM DIGEST

In the “Weekly Forum Digest” the teaching staff will highlight and comment on a few important themes and questions that have emerged in your lively discussions. The goal of the digest is to provide clarification and guidance for future discussions.

The forum digest for a given week will typically be posted over the weekend (a couple of days before the release of the new lectures).

Lectures 1 and 2 (the release date of these lectures was March 12th)
The weekly forum digest for lectures 1 and 2 can be found here.

Lectures 3 and 4 (the release date of these lectures was March 19th)
The weekly forum digest for lectures 3 and 4 can be found here.

Lectures 5 and 6 (the release date of these lectures was March 26th)
The weekly forum digest for lectures 5 and 6 can be found here.

Lectures 7 and 8 (the release date of these lectures was April 2nd)
The weekly forum digest for lectures 7 and 8 can be found here.

Lectures 9 and 10 (the release date of these lectures was April 9th)
The weekly forum digest for lectures 9 and 10 can be found here.

Lectures 11 and 12 (the release date of these lectures was April 16th)
The weekly forum digest for lectures 11 and 12 can be found here.

Lectures 13 and 14 (the release date of these lectures was April 23rd)
The weekly forum digest for lectures 13 and 14 can be found here.

edX101: How to Create an edX Course
Handouts Sidebar

When you include materials in the Course Handouts sidebar on the Course Info page, the materials appear in a list on the right side of the page. For example, 6.002x keeps all of its course information in its Course Handouts sidebar.

Using the Course Handouts sidebar provides a convenient way for the course team to provide documents, since documents do not have to be formatted as HTML. However, documents on the Course Handouts sidebar are harder for students to access. Additionally, if your students have downloaded one of the course handouts and you then make changes to that handout, you have to upload the new version of the handout and then tell your students to download a new copy.

Action: Post Materials on the Course Handouts Sidebar in the Practice Course

For this action, you can use any PDF or other document that you want.

Follow the steps in Getting Started with Studio to upload a document of your choice into your practice course and create a link to that document on the Course Handouts sidebar.
View the course in Preview mode or Live mode, and click your link to make sure you can download the document.

**Note** After you finish creating and running your course, you may want to run it again. To make sure you can easily find and update your course materials the next time that you run the course, you can upload all your original course materials (e.g. .docx files) into Studio. The next time you want to use your materials, you can download them from Studio, make changes, and then re-upload them.

Now that you have created some sample custom pages and uploaded a test document to the **Course Handouts** sidebar, you will create the materials for your specific course.
Course Schedule

In this unit, you will adapt a traditional course schedule to your online course.

Just like a schedule for a traditional course, the schedule for your online course should cover important dates for your students. However, there are also some adjustments you should make for your new audience. Your online students are different from your on-campus students: most have full-time jobs and/or families, and they live in many different countries (and, therefore, different time zones).

The scheduling and release guidelines below will help you adjust your course to this very different audience.

**Action: Create a Schedule for Your Course**

In the word processing program of your choice, open or create the kind of schedule that you would use in a traditional class.

Read through the guidelines below, and then make adjustments in your word processing program based on the needs of your online students.

In the next unit, you'll create and then post this schedule in Studio. You can create a Google calendar, a dynamic HTML calendar, or both.

**Scheduling Guidelines**

**General**

- Most course schedules include the following information.
  - Reading assignments
  - Lesson and video release dates
  - Homework release and due dates
  - Quiz release and due dates
  - Exam release and due dates

- We have found that students are more likely to complete a shorter course. Consider splitting an existing course into two or more shorter courses.

- Major multi-day holidays across the world, such as Chinese New Year or the last week of the year in many countries, may affect a number of your students. You may want to consider these dates when you plan your course. For an online calendar of domestic and international public holidays, see http://www.officeholidays.com.

- If your course staff includes undergraduate students, you should keep in mind that most undergraduate students won’t be on campus during school breaks.
Material, Assignments, and Exams

- Most instructors release materials once or twice a week, on the same weekdays, to avoid confusing students. Some courses, for example, release lessons on Tuesday and release homework on Thursday.

  Monday is a common choice for a release day. However, releases on Monday obligate staff to work over the weekend and are also more likely to coincide with US holidays—which may cause some instructors to delay releasing materials. If a release day falls on a US holiday, we recommend that you not delay releasing materials to be considerate of non-US students.

- To prevent students from worrying about a missed release, we recommend that you specify a time when you will release material, or release material early during the day (such as midnight of the first time zone). Late releases generate a large volume of e-mails and forum posts.

- Many instructors allow 14-21 days between the release date and the due date for homework. You can still release homework weekly.

- Some instructors do not assign homework covering materials released during the final two weeks of a course, but then include the material in the final exam. Other instructors have a study period in which no new materials are released before the final.

- Most courses schedule assignments so that they are due on the same day each week or every two weeks. Our user feedback indicates that students prefer Monday due dates so that they can work on the homework during the weekend.

- When you schedule an exam, you may want to provide a 4-7-day window in which a student has a fixed amount of time after opening the exam to work on it (24-48 hours is a good amount, since many students have full-time jobs and families and cannot free a three-hour block). During this time, the discussion forum will need to be either closed or vigilantly staffed 24-7 to prevent blatant cheating.

Time Zones

Daylight saving time and time zones can cause confusion about deadlines. Even if you cover time zones in your supplementary course materials, we recommend mentioning this issue in your first lesson or in your welcome video as well to help prevent a high volume of e-mails and forum posts.

By default, Studio uses Coordinated Universal Time (UTC). When you set due dates and times for assignments, you should make sure that you convert from UTC to your local time. A simple time zone converter is available at http://www.thetimezoneconverter.com.

To prevent confusion around deadlines and time zones, you may want to try one of the following methods.

- Tell students to submit the problems by midnight of their local time, and then set the actual deadline at 1:00 a.m. of the last inhabited time zone, UTC-11:00.

- Specify a specific time and time zone for the assignment to be due, and include a link to the online time converter at http://www.thetimezoneconverter.com.
Type of Schedule

Now that you have modified your schedule based on guidelines for online students, you’re ready to create the calendar for your course. You can create several different kinds of schedule or calendar. Most instructors create a Google calendar, a dynamic HTML calendar, or both. Course staff can change either type of schedule on the fly.

Google Calendar

When you create a Google calendar, students can easily sync your course calendar with their personal Google calendars. You can also link to the calendar from anywhere in your course.

Additionally, Google calendars automatically convert the times you enter to a student’s local time zone—that is, if you set an assignment to be due Friday at 5:00 PM US Eastern time, the calendar for a student in Thailand will show the due date and time as Saturday at 4:00 AM.

Your calendar may look like the following calendar for BerkeleyX’s CS 169.2x course.

Action: Create and Post a Google Calendar for Your Course

1. Create a Google calendar (even if you don’t currently have a Google account). Your students won’t need a Google account to access your calendar.
2. Make the calendar public.
3. Share the calendar with people who don’t have a Google account.
4. Embed the calendar in a custom page.
   a. Obtain the iframe code for your calendar. It should look something like this:
      <iframe
         src="https://www.google.com/calendar/embed?src=gracelyo%40gmail.com&ctz=America/New_York" style="border: 0"
         width="800" height="600" frameborder="0"
         scrolling="no"></iframe>
   b. Create a new custom page. Name the page Calendar.
   c. In the custom page editor, switch to the HTML tab.
   d. Replace the existing content with the iframe code for your calendar, and then click Save.

5. (Optional) Print the calendar.

6. (Optional) Save the calendar as a PDF and upload the PDF calendar to the Course Handouts sidebar (see "Add Course Handouts" in Getting Started with Studio).

**Dynamic HTML Schedule**

You can also use HTML to create a dynamic schedule that has links to your course materials or lessons. This schedule can be a custom page that students can access from anywhere in the course.

The following is a dynamic HTML schedule.

![Dynamic HTML Schedule](image)

Creating a dynamic HTML schedule is fairly advanced, however, so we recommend that you be familiar with HTML if you want to do this. For more information, see the Advanced unit.

In the next lesson, you’ll select an online textbook for your course.
Textbook

In this unit, you will choose the textbook for your course.

The primary difference between choosing a textbook for an on-campus course and for an edX course is that any textbook required for an edX course needs to be available for free. There are many ways to locate and provide free quality textbook materials for your students. The textbook can be available on the web, or you may be able to provide the textbook inside your course as a custom page. For example, the Poor Economics textbook for MITx’s “The Challenges of Global Poverty” course is available as a custom page.

Action: Explore Options for Selecting an Online Textbook

Three primary options are discussed below. You might also use a combination of options within your course.
**Option 1: Make an existing textbook available**

Whether the text has been authored by you, a colleague, or someone else, in order to offer a traditional text for free we must work directly with the publisher to provide a free digital copy of the text in the course.

To determine if we can make a published text available to students for free, we need to begin work as soon as possible to receive permission. Contact your program manager for more information.

**Option 2: Use free and open textbooks**

The open education resource movement has created several great non-profit and academic ventures where scholars can self-publish textbooks and articles through a peer review process. Below are learning object repositories that provide thousands of books, articles, and media that might be appropriate for your course.

The Open Educational Resource Commons (OER) website lists over 700 open textbooks and over 40,000 open source learning objects and articles. In addition to subject, you can search for content by a number of criteria including grade level, media format, and material type.

Connexions, affiliated with Rice University, offers nearly 22,000 modules, or small “knowledge chunks,” and over 1000 collections, or groups of modules organized into books, reports, and other content. Much of this content can be downloaded to mobile devices. Connexions also offers low-cost hard copies of its content.

Merlot, affiliated with the California State University system, offers peer-reviewed online teaching materials including about 2500 open textbooks as well as articles, simulations, tutorials, and other learning objects. Readers can browse through personal collections assembled by others. Merlot also has a list of over 110,000 members that include students, faculty, librarians, and other educators so that users can find colleagues in their own and other disciplines.

Wikibooks includes over 2,500 open-content, freely editable textbooks covering a variety of nonfiction subjects. Texts on Wikibooks are required to be instructional and cannot include “verbatim copies of pre-existing works,” although the site does permit annotated texts.

Project Gutenberg has put over 42,000 out-of-copyright ebooks online. The site offers several types of downloads, including HTML and Kindle. Because the copyright has expired on most books on the site, Project Gutenberg can be a very good source for classics and literature, but may not be a good resource for science or technology.

Open Culture offers approximately 160 free textbooks on subjects from the history of painting and basic music theory to organic chemistry and creating iPhone apps. It also includes over 400 free ebooks formatted for Kindle, iPad or iPhone, and Nook (mostly literature whose copyright has expired) and over 500 classic movies.

LibriVox is a collection of free audiobooks, read by volunteers. LibriVox’s goal is “to make all public domain books available as free audio books.” Again, this may be a good resource for classics and literature, but not for science or technology.
Option 3: Work with your institutional librarians

Talk about your options for journal articles and other digital resources that might be available for free to you and your students. Librarians are experts at copyright law and can help you navigate what might be possible for distribution in your online course.

Action: Select a Textbook for Your Course

Select the textbook that you want to use and make a note of the URL where the book is available. You will include this information when you create your syllabus later in this lesson.
Grading Overview

Setting up grading has two steps.

1. Establish an overall grading policy—including whether your course is pass-fail or graded by letter, and what the thresholds are for each grade. You can also decide whether to include a grace period for late assignments and whether to drop one or more assignments from a student’s final grade.

2. Determine what kinds of assignments you want to include and the percentage of the student’s total grade that each assignment type will make up. For example, your assignment types might include homework assignments, labs, projects, and quizzes. This information appears on each student’s Progress page.

You will learn more about grading and creating assignments later in edX101.

Action: Set Up Grading in the Practice Course

Set up the following grading system in your sample course. For step-by-step instructions, see Getting Started with Studio.

- Your course is graded by letter, with the following thresholds.
  - A: 90% - 100%
  - B: 80% - 89%
  - C: 70% - 79%
  - D: 60% - 69%
  - F: 0% - 59%
- Your course includes a grace period of 12 hours.
- Your course assignments are the following.
  - 10 homework assignments, worth a total of 25% of the grade.
  - 10 labs, worth a total of 25% of the grade.
  - Three midterm exams, worth a total of 30% of the grade.
- One final exam, worth 20% of the grade.

**Action: Set Up Grading In Your Course**

- Establish your grading policy in your course.
- Determine grade thresholds.
- Decide whether to use a grace period.
- Plan assignment types.
  - Decide what types of assignments to include.
  - Determine the weight of each assignment type.
  - Determine the number of assignments of each type.
Collaboration and Discussion
Forum Guidelines

In addition to the materials that every course provides, such as the schedule, textbook, and grading information, you may want to provide your students with information about acceptable uses of the course discussion forum and about collaboration with other students.

Collaboration

Because your online students are located in different countries throughout the world, they cannot physically get together in study groups the way your traditional students might. However, they may still want to work together on difficult assignments. You can encourage students to collaborate—lively discussions can help students understand material better. However, you will need to make sure they know that acceptable collaboration does not include sharing answers.

The following are the collaboration guidelines for one edX course.

Action: Create Collaboration Guidelines For Your Course

Based on the example above, write out collaboration guidelines for your students in your word processing program and save the document as a PDF.

Post your collaboration guidelines on your Course Handouts sidebar (for step-by-step instructions, see Getting Started with Studio).
**Discussion Forum**

Much of student collaboration will take place through the discussion forum. In your discussion forum guidelines, you can provide information that covers etiquette, best practices, and formatting hints. You can also let students know when your forum moderators will be online and available to answer questions.

The following images show discussion forum guidelines for several edX courses.

---

**Discussion Forum Best Practices:**

Here are a couple of helpful hints to using the discussion forum.

- Enclose your equations with the dollar sign ($) to activate "math mode". If you want to practice, go to the discussion forum and try typing the following text into your text box: $E \cdot \text{mass} = m \cdot \text{c \cdot c}^2$
- Actively upvote other posts, and other people will upvote yours! The most upvotes your post has, the more likely it is to be seen. Don’t forget to upvote your own posts!
- On each video page of the lecture sequences, there is a link at the bottom that says "show discussion". Clicking on this link will show all discussion on the forum associated with this particular video.
- If you are reporting a bug, tag your post with "bug". If you need the teaching staff to see your post, please tag "staff".

---

**Discussion Forum Guidelines:**

The discussion forum is the main way for you to communicate with the course team and other students. We hope it contributes to a sense of community and serves as a useful resource for your learning.

- **Observe the honor code.** We encourage collaboration and help, but please avoid asking for and posting blatant answers. Those caught cheating may have their accounts disabled and their progress erased.
- **Be polite.** We have learners from all around the world and with different backgrounds. Something that is easy for you may be challenging for someone else. Let’s build an encouraging community.
- **Upvote good posts.** This applies to questions and answers. Click on the green plus button so that good posts can be found more easily.
- **Search before asking.** The forum can become hard to use if there are too many threads, and good discussions happen when people participate in the same thread. Before asking a question, use the search feature by clicking on the magnifying glass on the left-hand side.
Most discussion forums employ community TAs, who are students from the course who have been designated to help moderate the forum and answer questions. Good candidates for community TAs are students who have been actively participating in and contributing to your class discussions. In the discussion forum section of your syllabus, you can outline criteria that course staff will consider when they select community TAs.

One course used the following criteria to select community TAs.

Discussion Forum Schedules

6.002x course TAs will be online and available to answer questions on the discussion forum at the following times each week:

- **Monday-Friday:**
  - 9:00 am – 9:30 am
  - 5:30 am – 6:00 pm
  - 9:00 am – 9:30 pm
- **Saturday:**
  - 10:30 am – 11:00 am
  - 1:30 am – 2:00 pm
  - 4:30 pm – 5:00 pm
- **Sunday:**
  - 12:00 pm – 12:30 pm
  - 3:00 pm – 3:30 pm
  - 4:30 pm – 5:00 pm
  - 6:00 pm – 8:00 pm
You will learn more about the course discussion forum in the Discussion Component unit.

**Action: Create Discussion Forum Guidelines for Your Course**

Based on the example above, write out discussion forum guidelines for your students.

Post your discussion forum guidelines on your Course Handouts sidebar (for step-by-step instructions, see Getting Started with Studio).
Syllabus

Just like a syllabus for a traditional course, your syllabus for your online course should cover all the information that your students will need to successfully complete the course. For your online course, your syllabus could also include a description of some elements that are unique to online learning so that your students know what to expect.

Action: Create the Syllabus for Your Course

1. Open the syllabus for your traditional course.
2. Read through the following suggestions.
3. Based on these suggestions, make adjustments to the syllabus.

Some courses have included the following information.

Welcome or Overview

Students coming into an online course may not know what to expect. In addition to the information a traditional welcome section would cover, your welcome or overview section can include descriptions of the way homework and labs in your online course are different from homework and labs in a traditional course. You can also tell your students about the format of your lessons. For example, if you have embedded interactive questions between your videos in a lesson, you can let students know they should expect these questions.

Your welcome section can also include general information. One course syllabus included a “basic statement about the ideals” of the course.

A Basic Statement About the Ideals of this Course

What does it mean to be human? This course takes a close look at the human condition, as viewed through the lens of classical Greek civilization. the basic organizing principle is an objective study of a model of humanity, the hero.

By the time the course comes to an end, students will have learned that there are different definitions of the “hero” in different historical times and places. In the end, though, the one true “hero” of this course will be the logos or “word” of logical reasoning, as activated by Socratic dialogue. The logos of dialogue in this course will require careful thinking, realized in close reading and reflective writing. The “last word” about this logos comes from Plato’s memories of words spoken in dialogue by Socrates during the last days of his life, which will be read towards the very end of the course. Such a “last word” shaped by a deep understanding of the concept of the hero in all its varieties throughout the history of Greek civilization, will become the “last word” for students who earnestly engage in dialogue, by way of writing as well as reading, with heroic expressions of the human condition. This course is driven by a sequence of dialogues that lead to such an engagement, guiding the attentive reader through many of the major works of the ancient Greek Classics.

Another course syllabus also included information about the course’s interactive video sequences and optional tutorials.

edX101: How to Create an edX Course
Prerequisites

To make sure your students have the skills they need to succeed in your course, describe your course’s prerequisites as specifically as you can. If your course has prerequisite courses, make sure to include a detailed description of the courses from your university’s course catalog. If your course requires knowledge of certain topics, make sure to include specific information about the requirements—for example, if a student needs to be familiar with calculus, make sure to specify which subtopics of calculus the student should know (for example, single-variable calculus, multi-variable calculus, or integrals and derivatives).
Workload

In the workload section, you can include the amount of time you expect students to spend on the course each week, based on what you would expect from a student at your school.

How many hours am I expected to work for the course?

Here are the expectations for a week:

- **50% of the students spent 6 hours or less per week on the course**
- **40% spent 6 to 12 hours a week**
- **10% spent more than 12 hours a week**

Textbook and Resources

Information in the textbook and resources section should include where to find the electronic textbooks and other online resources your class will use. If the textbook is also available for sale as an electronic or printed book, you can include links to the websites where students can purchase the text.

Required Reading: All required texts can be accessed for free on the course website.

- **Sourcebook of Original Greek Texts Translated into English**
  
- **The Ancient Greek Drama in 24 Hours** by Gregory Nagy. This e-book, referred to within the course and in this syllabus as **A24H**, contains 24 chapters. Each chapter is called an “hour,” because each one of the “hours” is keyed to each one of the 24 dialogues that make up the course. The book is designed to provide you with close readings and analysis of some of the most important parts of the relevant primary sources found in the Sourcebook. These “hours” take the place of formal lectures in this course, and the close readings in these “hours” are meant to be part of the “homework” that prepares you for each class session.

Recommended Reading:

For further background on and interpretation of the required reading, two books by Professor Nagy are available for free online: The Best of the Achaeans: Concepts of the Hero in Lyric, Epic, and Drama; and Homer’s Homer: The Lyric Association of an Epic Hero: These books supply background for some, but hardly all, of the topics to be presented in the course.

The following works are in the Sourcebook but are not required except for those sections addressed in A24H: the Homer’s Hymn to Demeter and Hymn to Aphrodite; Aeschylean Seven Against Thebes; Euripidean Helen; selections from Aristotle; and Sophocles’ Antigone.

Note that if the textbook is required for the course, it has to be available to students free of charge. Textbooks that are recommended but not required don’t have to be free.

Grading and Certification

The grading and certification section can include a brief overview of your grading policies and list the requirements to earn a certificate for your course. For your grading policies, you can specify whether your course is pass/fail or graded by letter and list the threshold for each grade. If you use a course-wide grace period, or if any lowest-scoring assignments are dropped, you can mention that here. This section can also cover information about homework, including the
type, format, and number of homework assignments, as well as release and due dates. You can also describe the format, number, and weight of midterms and the final exam.

**Earning a certificate of mastery**

To receive a certificate of mastery, students must take five short quizzes each consisting of 5 multiple-choice questions and a final exam consisting of 25 multiple-choice questions. Each quiz counts for 5% of the overall grade with the final exam accounting for the remaining 75%. A total score of 60% or higher overall qualifies as a passing grade for the course. While you are encouraged, throughout the semester, to discuss the topics of the course with your friends and fellow students, you must do the quizzes and the final exam on your own, without consulting others. We strongly recommend that students take the five short quizzes in sequence, to assess their progress. For those who want to earn a certificate of mastery, all graded work (the five short quizzes and the final exam) must be completed by **July 31st at 7 pm EDT** (Eastern Daylight Time) (which is the same as 23:00 UTC (Coordinated Universal Time)).

9. **Grading**

Letter grades will be based on the following weighting: homework 15%, labs 15%, midterm 30%, and final exam 40%. Each of the homework and labs carries equal weight. You will need to get a total mark of 60% for a C, 70% for a B, and 87% for an A.

Homework and labs will be graded based on the best ten out of twelve individual grades. Therefore, two homework assignments and two labs may be missed in total without a grade penalty.

In this section, you can also tell your students about the Progress page, where they can see a graphical representation of their progress through the course.

![Course Progress](image)

**Discussion Forum and Collaboration Guidelines**

In your syllabus, you can provide an overview of your discussion forum and collaboration guidelines. For your discussion forums, you can include suggestions around topics and etiquette, indicate when course staff will be available online, and outline criteria you will consider when you select community TAs.
If you want to provide more detailed information about discussion forum guidelines as well as what are and are not acceptable forms of collaboration, you can create separate pages for these topics and create links to those pages in your syllabus.

**Action: Post Your Syllabus**

After you have modified your syllabus according to the guidelines, post your syllabus on the Course Handouts sidebar.

You can also create an HTML syllabus that you can post on a custom tab together with the schedule. If you want to, you can include your schedule in this HTML syllabus. Creating a dynamic HTML schedule is fairly advanced, however, so we recommend that you be familiar with HTML if you want to do this. For more information, see the Advanced unit.
Advanced

You can create a syllabus and schedule that appear as custom HTML pages instead of posting them as separate items on the Course Handouts sidebar or embedding your Google calendar into a custom page. We have provided an HTML Syllabus template, adapted from Berkeley's CS169.1x and CS169.2x, that you can use to create a syllabus that includes your schedule.

This template has some advanced functionality, such as collapsible sections, that will make it easier for your students to find the information they need.

**Action: Create a Syllabus**

1. Open the HTML Syllabus template.
2. Copy the HTML template into a document that you can edit.
   - (Recommended) If you have a Google account, click File, and then click **Make a copy** to create a document in your Google drive.
   - If you don’t have a Google account, select and copy all the text in the template, and then paste it into your word processing program.
3. In your document, scroll down to the colored **Welcome to the Course!, Prerequisites, Workload, and Grading** sections. (You can ignore the **Schedule** section for now.)
4. Replace the placeholder text in these sections with the information for your course, and then save your document in an easily findable place. (You will use it again below.)
   If you want to add a section such as a Discussion Forum section, copy all the code for a section (starting with the `<div onClick...>` tag and ending with the `<hr width="100%" align="center"/>` tag), and paste the code in the location where you want the new section. Be sure to change the colored numbers and letters inside the `<div>` tags so that they're different for each section.
5. Select and then copy all the text in your template document.
6. In Studio, create a new custom page.
7. In the custom page editor, click **Settings**, type **Syllabus** in the **Display Name** field, and then click **Editor**.
8. Switch to the **HTML** tab.
9. Paste the HTML from the template into the custom page, and then save the page.

View your course in Live mode to see what the template looks like with the placeholder text.

**Note** If you have technical problems, make sure that your word processor has not replaced straight quotation marks (‘ and ”) with curly quotation marks (‘ and “).
Action: Add a Dynamic HTML Schedule to Your Syllabus

You can use HTML to create a dynamic schedule that has links to your course materials or lessons. When you use an HTML schedule, students can both see release dates for all your course materials and access the released materials in one place. Students can also access the schedule from anywhere in the course.

The following is a dynamic HTML schedule.

<table>
<thead>
<tr>
<th>Courseware</th>
<th>Course Info</th>
<th>Discussion</th>
<th>Wiki</th>
<th>Progress</th>
<th>Open Ended Panel</th>
<th>Syllabus</th>
<th>Schedule</th>
<th>Instructor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Course Schedule</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week of</td>
<td>Topic</td>
<td>Reading</td>
<td>Slides</td>
<td>Lecture Sequence</td>
<td>Assignment</td>
<td>Due</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mo 9/24</td>
<td>Software Engineering and SaaS Architecture</td>
<td>Ch. 1</td>
<td>Lecture 1</td>
<td>Lecture 1: Course Introduction</td>
<td>Quiz 0</td>
<td>10/7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ch. 2</td>
<td>Lecture 2</td>
<td>Lecture 2: SOA and Cloud Computing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mo 10/1</td>
<td>Ruby</td>
<td>Ch. 3</td>
<td>Lecture 3</td>
<td>Lecture 3: Introduction to Ruby</td>
<td>HW 1</td>
<td>10/16</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lecture 4</td>
<td>Lecture 4: Ruby Continued</td>
<td>Quiz 1</td>
<td>10/14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mo 10/8</td>
<td>Rails</td>
<td>Ch. 4</td>
<td>Lecture 5</td>
<td>Lecture 5: SaaS Architecture &amp; Intro to Rails</td>
<td>HW 2</td>
<td>10/21</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When you use an HTML schedule, the dates for all of your planned lessons are always visible, but the schedule does not contain any live links when the course begins. Course staff members manually create live links to your lessons and materials as materials are released.

(Alternatively, you can include "live" links to all of your course materials—including those that haven't been released. However, note that until the release date for your course materials has passed, students will receive an error message when they click the links for those materials.)

The following short sample schedule was adapted from Berkeley's CS169.2x course. The sample shows the way the schedule would look during week 1 of the course. There are no links for material after week 1, since that information has not yet been released.

Course Schedule

<table>
<thead>
<tr>
<th>Week of</th>
<th>Topic</th>
<th>Reading</th>
<th>Slides</th>
<th>Lecture Sequence</th>
<th>Assignment</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mo 9/24</td>
<td>Software Engineering and SaaS Architecture</td>
<td>Ch. 1</td>
<td>Lecture 1</td>
<td>Lecture 1: Course Introduction</td>
<td>Quiz 0</td>
<td>10/7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ch. 2</td>
<td>Lecture 2</td>
<td>Lecture 2: SOA and</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

edX101: How to Create an edX Course
### Create a Dynamic HTML Schedule

1. Open the document that contains the HTML template that you created above.
2. Scroll down to the **Course Schedule** section.
3. In your document, replace the placeholder text in the **Course Schedule** section with the schedule information for your course. Note that the template is colored by week to make each week easier to find.
4. In Studio, open the custom page that you created.
5. In the custom page editor, replace all of the existing content with all of the content from your HTML template document, and then save the page.
6. View your course in Live mode, and click the **Syllabus** tab at the top of any page to see your syllabus and schedule.

Your About Page
Introduction

When you visit edx.org and click Courses at the top of any page, you can see the edX course catalog, or the list of the courses that edX offers.

edX Course Catalog

When you click one of the courses in the list, the About page for that course opens. The About page corresponds to a traditional course catalog entry, but it's more detailed and colorful. The About page is where you tell potential students about your course: what the course covers, when it starts, how long it lasts, and how many hours you expect students to devote to the course each week. Additionally, along with a text description, an About page has a course picture and even an About video--a "movie trailer" for the course.

Below is a sample course About page.
Software as a Service

CS169.1x teaches the fundamentals for engineering long-lived software using Agile techniques to develop Software as a Service (SaaS) using Ruby on Rails. Students will understand the new challenges and opportunities of SaaS versus shrink-wrapped software. They will understand and apply fundamental programming techniques to the design, development, testing, and public cloud deployment of a simple SaaS application. Students will use best-off-the-shelf tools that support modern development technologies including behavior-driven design, user stories, test-driven development, velocity, and pair programming. Students will learn how modern programming language features like metaprogramming and reflection can improve productivity and code maintainability.

Students will work individually on weekly coding projects. Those who successfully complete each assignment and earn a passing grade will get an honor code certificate from Berkeley.

Course Detail

School: UC Berkeley
Course code: CS-169.1x
Classes start: 2 Jul 2013
Course length: 5 weeks

Prerequisites: Programming proficiency in an object-oriented programming language such as Java, C#, Python, or Ruby is required. We will teach the basics of Ruby at a very accelerated pace that assumes thorough familiarity with OOP inheritance.

Register for CS-169.1x

Armando Fox
Armando Fox is an Adjunct Associate Professor at UC Berkeley and was previously an Assistant Professor at Stanford, where he received teaching and mentoring awards from the Associated Students of Stanford University, the Society of Women Engineers, and Tau Beta Pi Engineering Honor Society. In previous lives, he helped design the Intel Pentium Pro microprocessor and founded a successful startup to commercialize his UC Berkeley dissertation research on mobile computing. His degrees in electrical engineering and computer science are from MIT, the University of Illinois, and UC Berkeley. He is an ACM Distinguished Member.

David Patterson
David Patterson is the Ramanujan Professor of Computer Science at UC Berkeley and the recipient of the Distinguished Teaching Award (UC Berkeley), the Kesternich Outstanding Educator Award (ACM), the Mulliken Education Medal (IEEE), and the Undergraduate Teaching Award (IEEE). He has served as Chair of Berkeley's CS

edX.org-hosted About Page

Note that the About page above is for a course that is hosted on edx.org. About pages for courses that are hosted on edge.edx.org, or Edge courses, aren't accessible from the course catalog on edx.org. To share the About page for an Edge course, you have to send students the URL for the page.

About pages for Edge-hosted courses also look a little different from edx.org About pages. The following About page is from a course that's basically identical to the one above, but this course is on Edge instead of edx.org.
Edge-hosted About Page

**Entering the About Page Information in Studio**

Similar to the Your Course Info lesson, this lesson teaches you to prepare all of the material for your course's About page so that you or your staff can enter the information into Studio later. For step-by-step instructions for entering your course's About page material in Studio, see the Action: Add the About Page Information unit.
Long Course Description

The long course description is essentially the text that would appear in a traditional course catalog. It’s one of the first things a prospective student sees on your About page.

Your long course description should have between 150 and 300 words, though it can be longer or shorter. The description should be as specific as possible--it should describe not only the topics and skills you’ll cover, but how those topics and skills relate to the external world. Additionally, because online students are more likely to have jobs and families than on-campus students, it is important to address the following issues:

- Will there be homework assignments?
- When do due dates occur (for example, every Friday or every Monday)?
- Will there be exams? How many? Timed or untimed?

You’ll cover these topics when you write your syllabus as well. However, a student who is considering taking your course won’t have access to the syllabus yet, and may want to know what to expect before he decides to register. Your long course description may be an abbreviated version of your syllabus.
Course E-mail Address

Many course administrators have found it helpful to create an e-mail address specifically for an individual course, or even an individual run of a course. If your course doesn't have an e-mail address, students may use the e-mail address that they find in your school's directory or in other online sources to contact you. Your online course may have tens of thousands of students, and if even a small percentage of these students want to contact you, the amount of e-mail you receive may be difficult to manage. Your course e-mail address should be clearly affiliated with your organization—for example, class_name@georgetown.edu is a more trusted e-mail address than class_name@gmail.com or even class_name@comcast.net.

When you have created an e-mail address for the course, make sure that you add this information to the long course description so that students can contact the course team.

Long Course Description Examples

The following are a few examples of long course descriptions.

CS169.1x: Software as a Service

CS169.1x teaches the fundamentals for engineering long-lasting software using highly-productive Agile techniques to develop Software as a Service (SaaS) using Ruby on Rails. Students will understand the new challenges and opportunities of SaaS versus shrink-wrapped software. They will understand and apply fundamental programming techniques to the design, development, testing, and public cloud deployment of a simple SaaS application. Students will use best-of-breed tools that support modern development techniques including behavior-driven design, user stories, test-driven development, velocity, and pair programming. Students will learn how modern programming language features like metaprogramming and reflection can improve productivity and code maintainability.

Students will work individually on weekly coding projects. Those who successfully complete each assignment and earn a passing grade will get an honor code certificate from BerkeleyX.

BE101x: Behavioural Economics in Action

How can we get people to save more money, eat healthy foods and engage in healthy behaviours, and more generally make better choices? There has been a lot written about the fact that human beings do not process information and make decisions in an optimal fashion. This course builds on much of the fascinating work in the area of behavioural economics and allows the student to develop a hands-on approach by learning its methods and more importantly, how it can be harnessed by suitably designing contexts to “nudge” choice. In three modules, students will be able to a) explain and interpret the principles underlying decision-making and compare the nudging approach to other methods of behaviour change, b) learn how to critique, design and interpret experiments; and c) design nudges and decision-tools to help people make better decisions. Students will also witness and participate in weekly topical debates on topics like “does irrationality impact welfare?” or “Is nudging manipulative?” If you’ve been fascinated with the buzz surrounding behavioural economics but are not sure how to actually use it, this course is for you.

Guest Appearances: Several leading scholars, policy makers, businesspeople and commentators will make short appearances in our debate and discussion sections. These
guests include Professor Sendhil Mullainathan (Harvard University), Professor John Lynch (University of Colorado), Rory Sutherland (Ogilvy Group), Owain Service (Behavioural Insights Team, UK Cabinet Office), Shankar Vedantam (NPR Columnist and Author – The Hidden Brain), Professors Andrew Ching, Avi Goldfarb, Nina Mazar, and Claire Tsai (University of Toronto) and many others!

8.MReVx: Mechanics ReView

Mechanics ReView is a second look at introductory Newtonian Mechanics. It will give you a unified overview of mechanics that will dramatically increase your problem-solving ability. It is open to any students who meet the prerequisites (see right), but is especially designed for teachers and those who want to improve their existing understanding of mechanics.

Newtonian mechanics is the study of how forces change the motion of objects. This course begins with force, and moves on to straight-line motion, momentum, mechanical energy, rotational motion, and angular momentum. Optional units include oscillations, planetary orbits, and a review of multi-concept problems.

Why Take Our Course?

Our approach to mechanics is a unique one. Through worked examples and online texts we introduce a strategic overview of core concepts in mechanics. This overview couples with a new approach to problem-solving that will help you break down and solve multi-concept problems. By choosing a system of objects, defining their interactions, and deciding on a model to use to describe them, you will come to solve physics problems more easily and more systematically.

The road to "easy" goes through "hard." This course is more challenging than a standard high school or introductory college physics offering. We will help you become a more expert problem solver, and that process involves solving a variety of problems, in many different ways, with our new approach. The reward is stronger problem-solving ability that carries over to other areas in physics.

Take this course to better organize your mechanics knowledge, to prepare for AP or advanced standing exams, to teach more effectively, or if you enjoy attacking challenging problems!

For Teachers

If you are a teacher looking to improve your knowledge of mechanics, or to learn new approaches to teach your students, we encourage you to sign up. Our approach has been researched carefully and has proven effectiveness when it comes to preparing students for later courses.

Teachers in the United States, and especially in Massachusetts, can receive extra benefit from this course. We offer Professional Development Points (PDPs) at no charge to teachers in Massachusetts who complete our course. If you are in a different state, we instead offer Continuing Education Units through the American Association of Physics Teachers. There is a fee for this certificate.

Action: Create Your Long Course Description

For this action, you’ll write your long course description and enter the text into the open response assessment that follows. You’ll then perform a self-assessment of the description
based on the rubric. Open response assessments are a new feature that edX offers, and you'll learn more about them in the Open Response Assessments lesson.

If your course will be hosted on Edge and you want to add the long course description to Studio, you can skip to the Action: Add the About Page Information unit.

**Open Response Assessment: Write Your Long Course Description**

Based on what you've read, write the long course description for your course. When you're done, you'll give yourself a score based on how well your long course description meets the criteria in our rubric. Read through each topic, and then select the option that best fits your response.

**Rubric**

The description lists the specific topics your course covers. For example, instead of saying "students will learn about calculus," your description should say "the course covers derivatives, L'Hôpital's Rule, integrals, Riemann sums, sequences, and series."

- The description only gives a general idea of what topics are covered or doesn't list topics.
- The description lists very specific topics.

The description explains why students will enjoy learning the material in your course and how they can apply the information to the external world.

- The description does not include this information.
- The description includes this information.

The description outlines the specific skills you expect your students to learn through the course. For example, they may learn to identify European haplogroups or tune up a bicycle.

- The description covers only general skills the students should learn or does not mention skills.
- The description lists specific skills that the students should learn.

The description provides scheduling information. For example, students may have two weeks to complete homework assignments and five days to complete exams. You can also provide the dates when you will release your exams.

- The description does not provide scheduling information, or provides only vague information.
- The description provides specific scheduling information.

The description covers the form of homework assignments and exams. For example, your homework assignments may require students to write computer code or essays. Your exams may be timed or untimed.

- The description does not provide information about what homework and exams may be like or only provides a general idea.
- The description provides detailed information about homework and exams.
The description mentions the requirements for a certificate, if one will be available.

- The description does not specify the requirements for a certificate.
- The description lists specific requirements for a certificate.
Short Course Description

Now that you have written your long course description, you can use it to help write your short course description. Your short course description appears in the course catalog on edx.org.

Note Because the short course description only appears in the edX course catalog, courses that are hosted on Edge don't need a short description.

Since your short course description is the first information about your course that a student sees, it should be both informative and catchy. You can model your short course description on any of the following examples.

Note Your short course description must contain fewer than 146 characters (including spaces).

Behavioural Economics in Action - Learn to use principles and methods of behavioural economics to change behaviours, improve welfare and make better products and policy.

Introduction to Computer Science - An introduction to the intellectual enterprises of computer science and the art of programming.

Ideas of the Twentieth Century - Learn how philosophy, art, literature, and history shaped the last century and the world today.

Software as a Service II - CS169.2x teaches sophisticated SaaS+Agile skills, such as working with legacy code, building on basics from CS169.1x.

Introduction to Human Evolution - An overview of human evolutionary history viewed through the human fossil, archaeological and genetic records.

Introduction to Water Treatment - Learn about urban water services, focusing on basic drinking water and wastewater treatment technologies.

Globalization’s Winners and Losers: Challenges for Developed and Developing Countries - Who are the winners and losers of globalization? What should be done to improve outcomes for all?

Introduction to Philosophy: God, Knowledge and Consciousness - This course will focus on big questions. You will learn how to ask them and how to answer them.
Action: Write Your Short Course Description

This action has two steps:

If your course will be hosted on edX, write your short course description in the discussion space below.

After you add your short description, read and comment on a peer’s short course description.

(Discussion space provided)
Staff Biographies

The About page has a section for biographical information about your course staff.

Armando Fox
Armando Fox is an Adjunct Associate Professor at UC Berkeley and was previously an Assistant Professor at Stanford, where he received teaching and mentoring awards from the Associated Students of Stanford University, the Society of Women Engineers, and Tau Beta Pi Engineering Honor Society. In previous lives he helped design the Intel Pentium Pro microprocessor and founded a successful startup to commoditize his UC Berkeley dissertation research on mobile computing. He has degrees in electrical engineering and computer science from MIT, the University of Illinois, and UC Berkeley, and he is an ACM Distinguished Member.

David Patterson
David Patterson is the Pardee Professor of Computer Science at UC Berkeley and the recipient of the Distinguished Teaching Award (UC Berkeley), the Karlstrom Outstanding Educator Award (ACM), the Mulligan Education Medal (IEEE), and the Undergraduate Teaching Award (IEEE). He has served as Chair of Berkeley's CS Division, Chair of the CRA, and President of the ACM. His best-known research projects are Reduced Instruction Set Computers (RISC), redundant Arrays of Inexpensive Disks (RAID), and Network of Workstations (NOW). Among other distinctions, he’s been elected to the National Academy of Engineering, the National Academy of Sciences, and the Silicon Valley Engineering Hall of Fame, and he is a Fellow of the Computer History Museum. He received all his degrees from UCLA.

Sam Joseph
Sam Joseph is an Associate Professor at Hawaii Pacific University and was previously an Associate Researcher at University of Hawaii at Manoa (UHM). He is a recipient of the Baymond-Hole prize for Astrophysics and a Travelski Fellowship. He teaches courses on game & mobile programming & design, software engineering and scientific research methods exclusively online from London, UK. He runs the funniest computer ever competition as part of his research on creating humorous chatbots, which dovetails nicely with his other research interests in software to support online collaborative learning, in particular remote pair programming. His degrees in Astrophysics, Cognitive Science, and Computer Science are from the University of Leicester, the University of Edinburgh, and UHM, and he is an ACM Member.
Your staff bios can not only list your qualifications, but also provide glimpses of your personality. Your students will be curious about you, so feel free to express yourself!

Along with your written biography, your students will enjoy seeing photographs of each staff member. These should be a minimum of 110 x 110 pixels in size, with higher resolution preferred.

The following are the bios for edX101 staff.

**Edx101 Staff Biographies**

![Anant Agarwal](image)

**Anant Agarwal**

President of edX and Professor of Electrical Engineering and Computer Science at MIT. His research focus is in parallel computer architectures and cloud software systems, and he is a founder of several successful startups, including Tilera, a company that produces scalable multicore processors. Prof. Agarwal won MIT’s Smullin and Jamieson prizes for teaching and co-authored the course textbook "Foundations of Analog and Digital Electronic Circuits."

![Grace Lyo](image)

**Grace Lyo**

Grace Lyo is the primary author for edX101 and a program manager for all rerun courses, Seoul National University, and Kyoto University. Her background is in theoretical mathematics: after receiving her bachelor's degree at Harvard and her PhD at UC Berkeley, she held a faculty-level position for several years in the MIT mathematics department. In 2007 she also won the Clay Liftoff Fellowship, an honor awarded to only 16 new mathematics PhDs across the nation.

Grace began teaching as a calculus TA during her sophomore year of college, and has always received stellar evaluations and accolades: one course administrator remarked that he frequently received emails from would-be students complaining that her course was full. She is excited to be at the front lines of this new movement in education and to work with universities as they transform their campuses.

![Piotr Mitros](image)

**Piotr Mitros**

As Chief Scientist at edX and one of its co-founders, Piotr Mitros is charged with developing and applying technology to optimize the learning process. His teaching experience around the world has given him a unique perspective on education. As a graduate student, Mitros took breaks from his thesis to spend time teaching in China, working in India, and facilitating educational technology projects in Nigeria. His observations of university systems in the developing world inspired Mitros to find innovative ways to dramatically increase access to education. Following a stint in industry as an analog designer, most recently at start-up Rhythmia Medical where he designed the analog front end for a novel medical imaging modality, Mitros returned to MIT to help lead the creation of the original MITx platform. He brings a broad interdisciplinary background that combines teaching, engineering, computer science, and math, and has been interested in teaching and education
since he was a child. Mitros enjoys making things -- curtains, bicycle parts, electronics, furniture, and speakers. Mitros holds a B.S. in Math and Electrical Engineering, a Masters of Engineering in EECS, and a Ph.D. in EECS, all from MIT.

Rob Rubin

Vice President of Engineering. Rob Rubin has been building infrastructures of Internet companies since 1995. (He’s been collecting books on mathematical puzzles and baseball for even longer.) As Vice President of Engineering at Carbonite (IPO 2011), a leading provider of online backup and cloud-based applications, he was responsible for a massively scalable infrastructure, and hyper-productive, flourishing scrum teams. He is the founder of Vopinion, an innovative website where people can discover and debate issues important in their local town. Rubin has taught in the United States and Israel, and has co-developed patents for computer applications tools. At edX, Rubin leads a team of engineers and developers who are responsible for building and providing the open platform technology behind edX, and for launching edX courseware. Rubin holds a B.A. in Computing Science from Columbia University, a Sc.M. in Computer Science from Brown University, and a Ph.D. in Computer Science from Brown University. He is an avid gardener and has a particular affection for Raleigh Twenty folding bikes, often seen riding his single-speed in and around town.

Sylvia Pearce

Sylvia Pearce, the writer for edX101, is an Arkansas transplant who got her book learnin’ at Brigham Young University and now calls Cambridge home. A writer and editor since ages 6 and 7, respectively, she’s spent the last 15 or so years professionally translating between technical jargonese and comprehensible English. She has been delighted to compose instructional content and occasional poetry for edX101.

Julia Zimmerman

Julia Zimmerman created the illustration for An edX Haiku. She went to River Oaks Elementary, Lanier Middle School, HSPVA, and MIT, and now she works at Veson Nautical. She likes to draw things. If you want her to draw something for you, contact her at jwzimmer1990@gmail.com.

Action: Write Your Staff Biographies

Write a 75- to 100-word biography for each member of your course staff.

If your course will be hosted on Edge and you want to add your staff biographies to Studio, you can skip to the Action: Add the About Page Information unit.
Frequently Asked Questions

Many edX course About pages also have a Frequently Asked Questions (FAQ) section.

In this section, you can provide answers to common questions about your course. Students may want to know how long materials will be available after the week in which they're released, whether they have to watch videos at specific times, or even if they can take a vacation during the course run.

Note that edX.org has a list of FAQs, and there is a link to this FAQ list at the bottom of every course's About page. It's a good idea to make sure that your questions aren't already covered on the edX FAQ.
The following are some example FAQ sections from edX courses.

8.MReVx Mechanics ReView

FAQS

Q. Is there a required textbook?
A. You do not need to buy a textbook. All material is included in the edX course and is viewable online. If you would like to use a textbook with the course (for example, as a reference), most calculus-level books are suitable. Introductory physics books by Halliday and Resnick, Young and Freedman, or Knight are all appropriate.

Q. What if I take a vacation?
A. The course schedule is designed with this in mind! You can work three weeks ahead in the course, and the exams are one week after the material. Even if you have a four-week vacation, you do not need to miss any deadlines and can still complete all of the material.

Q. Will I earn a certificate?
A. Yes! This course awards certificates to all who satisfactorily complete the required portion of the course.

Q. How are grades assigned?
A. There are three parts of the course that are worth points: Checkpoint problems that are folded in with the reading, homework problems that come at the end of each unit, and quizzes that are at the end of every 1-2 units. Each is worth a varying number of points, and you will not have to do every problem.

The course consists of 11 required units and three optional units. You do not need to complete the optional units in order to receive a certificate.

There is no final exam.

Q. How does this course use video?
A. We use video only when appropriate, to convey specific concepts or tackle specific questions. There are no required videos to watch each week. The majority of our course is delivered through text, worked examples, and problems.
ANTH207x Introduction to Human Evolution

FAQs

Q. What do I need to know about the topic prior to enrolling in this course?
A. Nothing. This course is intended as an introduction to human evolution and the only experience you need in advance is the experience of being human.

Q. If I have prior experience with this subject, will the course be useful?
A. Yes! While the course is introductory in nature, many of the questions presented for discussion represent active areas of research and debate within the field. This course will not only provide you with Prof. Van Arsdale's perspective on these issues, but also the views and ideas of the thousands of other peers who will be taking the course.

Q. What do I have to buy in order to participate in the course?
A. Nothing. All edX courses are free. The lectures, virtual labs, explorations and course readings are all available online.

Q. Will certificates be awarded?
A. Yes. Online learners who achieve a passing grade in a course can earn a certificate of mastery. These certificates will indicate you have successfully completed the course, but will not include a specific grade. Certificates will be issued by edX under the name of WellesleyM, designating the institution from which the course originated.

Q. When can I watch the lectures?
A. After the release of the learning sequences for the week, you can watch the lectures at any time.

Q. Can I register to the class after the start date has passed?
A. Yes, you can register at any time between the start date and the end date. To obtain the certificate, however, you will have to get at least 60% of the total available points for the class.

Q. Other questions
A. If you have any questions about edX generally, please see the edX FAQ.

Action: Create Your FAQ Section

This action has two steps:

In the discussion space below, write out a list of questions that you've been asked frequently or that you expect many students to have.

Read and comment on a peer’s FAQ section.

If your course will be hosted on Edge and you want to add your FAQ section to Studio, you can skip to the Action: Add the About Page Information unit.
**Prerequisites**

You can also list prerequisites for your course on your About page. The prerequisites appear in two places: in the sidebar on the right side of the page, and in the body of the page. The prerequisites look a little different on edx.org and on Edge.

If your course is hosted on edx.org, the first 200 characters of your prerequisites appear in the sidebar on the right side of the About page. The full text appears in the **Prerequisites** section at the bottom of the page.

If your course is hosted on Edge, the sidebar specifies whether the course has prerequisites, and the text describing your course's specific prerequisites appears in the **Prerequisites** section.
It's best to be as specific as possible when you write your prerequisites. For example, instead of saying "students should be familiar with calculus," your description should say "students should be familiar with derivatives, L'Hôpital's Rule, integrals, Riemann sums, sequences, and series." If your course doesn't have prerequisites, you can mention that too.

The following are examples of prerequisites sections for edX courses.

**8.MReVx Mechanics ReView**

An introductory physics course in high school or college is strongly recommended. Very good algebra and trigonometry skills are essential, introductory calculus and vector calculus are helpful.

**UT4.01x Take Your Medicine - The Impact of Drug Development**

There are no prerequisites for this course. This course is designed to allow participation by everyone, including the general population and scholars, without prior knowledge.

**CS169.1x Software as a Service**

Programming proficiency in an object-oriented programming language such as Java, C#, C++, Python, or Ruby is required. We will teach the basics of Ruby at a very accelerated pace that assumes thorough familiarity with OOP inheritance, static/class vs. instance methods and attributes, recursion, hash tables/hash maps, list comprehensions, higher-order functions, lambda expressions. This course is NOT a good first course in programming. Basic Unix command-line skills are helpful; we will provide a cheat sheet.
**Action: List Your Prerequisites**

This action has two steps:

1. In the discussion space below, write a detailed list of prerequisites for your course.
2. Read and comment on a peer’s list of prerequisites.

If your course will be hosted on Edge and you want to add your prerequisites information to Studio, you can skip to the Action: Add the About Page Information unit.
Descriptive Picture

Every course has a descriptive picture associated with it. This picture appears in many places:

- In the course catalog, if your course is hosted on edx.org.

- On the student dashboard after a student has registered for the course.

- On the About page.
When you choose an image for your course, select an eye-catching, colorful image that captures not only the content of your course but the atmosphere that you want to create. Additionally, we recommend the following best practices for your descriptive picture.

- Avoid using a picture that includes text or headlines--those elements may be partially covered by the Register button for the course.
- Make sure that the picture is a high-resolution image that is a minimum of 660 pixels in width by 240 pixels in height.
- Consider working with a professional photographer or designer to create a picture that will communicate effectively with prospective students.

Some descriptive pictures show a specific moment during the course.

Classical Mechanics

Others portray the course’s content and atmosphere in a more abstract way.

Introduction to Human Evolution
Action: Create a Descriptive Picture

Create a descriptive picture for your course.

If your course will be hosted on Edge and you want to add your descriptive picture to Studio, you can skip to the Action: Add the About Page Information unit.
The Course About Video

(Embedded video, transcript follows)

ERIC: The about video is like a movie trailer for your class. Here's an example from Harvard's Introductory Computer Science course, CS50x.

NARRATOR: Computer Science 50, otherwise known as CS50, is Harvard College's introduction to the intellectual enterprises of computer science, and the art of programming, for majors and non-majors alike.

ERIC: The best about videos are short-- no longer than three minutes-- and describe the course content, introduce the course staff, and state the learning objectives of the class in a general and succinct way.

Here’s another example of an about video, from MIT's introductory computer science class, 600x.

NARRATOR: For seven years, MIT's CS department has taught a course, 600, that helped people who have never programmed learn to use a computer to help them become more productive, at whatever it is they want to do.

ERIC: It describes the goals of the class from a more global perspective, using historical clips to emphasize the importance of computer science. Our video team can help you make a great video. It should reflect the character and personality of you and your course. The next two units are more examples of about videos.

Planning an About Video

When you plan your video, think of effective ways to answer the following questions.

- Why should students take this course?
- Why is this subject important?
- What will a student be able to do after taking the course?
- Who is the instructor? (Including something interesting and personal about the instructor makes for a more compelling video.)
To help you generate ideas, watch the example About videos in the next two units. You can also watch additional About videos on the main edX site. After you watch the example videos, you'll create a plan for your course's video.

**Action: Create a Course About Video**

Create a plan for your course About video. For the purposes of this exercise, imagine that you have unlimited resources. After you have an idea of what you want your video to look like, you can contact your university's administration about working with you to create a professional quality video.

If your course will be hosted on Edge and you want to add your About video to Studio, you can skip to the Action: Add the About Page Information unit.
About Video for 8.01X: Classical Mechanics

(MUSIC PLAYING)

[CHEERING]

WALTER LEWIN: It's very difficult to measure that. Very hard.

[MUSIC PLAYING]

WALTER LEWIN: Hello!

I'm Walter Lewin, and I am back in my beloved lecture hall, 26-100. This is where I gave the course that I think you should take. You're going to look at the world in a way that you never have before. Whether you like it or not, I'm going to make you love physics.

[MUSIC PLAYING]
(Embedded video, transcript follows)

The 20th century was a century of unprecedented technological progress. More people lived longer, healthier, more prosperous, and freer lives than ever before.

But it was also a century of unprecedented destruction. Friedrich Nietzsche, a German philosopher, predicted that the 20th century would be a century of great wars. He was right. 100 million people died in two world wars. Another 100 million died at the hands of their own governments.

Artists, philosophers, and writers lost their faith-- their faith in God, in traditional morality, in any kind of value at all. They struggled to find new ways of thinking and creating.

From the ashes of destruction and disillusionment, a new generation of leaders redefined values. They created intellectual and political systems to justify and defend them.

This is our heritage. It's your heritage. You can't understand yourself or the world around you without understanding the intellectual and political revolutions of the past 100 years.

The 20th century-- it's how we got here.
Other Course Information

In addition to the information that appears in the body of the About page, some at-a-glance information about your course appears in the sidebar on the right.

On edX.org, the sidebar appears as follows.

On Edge, the sidebar looks similar.

The school, course code, and prerequisites are filled in automatically. You can also specify some other basic information.

edX101: How to Create an edX Course
• **Classes start**: If the exact start date has not yet been determined, you can temporarily indicate a general timeframe such as "2013 Term 1."
  **Note** Because students from all over the world will be enrolled in your course, and seasons are different in the northern and southern hemispheres, use timeframes such as "2013 Term 1" instead of "Spring 2013."

• **Course length**: The number of weeks your course will run.

• **Estimated effort** (optional): The number of hours per week you would expect an average student at your university to spend on your course.

The sidebar also has links to social media including Twitter, Facebook, and Meetup, as well as e-mail, so that your students can easily share their excitement about your course with their social networks.

**Action: Determine Other Course Information**

Determine and collect the additional information that should appear in the right sidebar on your About page.

If your course will be hosted on Edge and you want to add your other course information to Studio, you can skip to the Action: Add the About Page Information unit.
**Action: Add the About Page Information**

After you've collected all the information for your About page, the process for adding it to your course differs depending on whether your course is hosted on edx.org or on Edge.

**edx.org**

If your course is hosted on edx.org, you'll work with your project manager (PM) to add the About page content.

**Edge**

If your course is hosted on Edge, you'll upload your descriptive picture in Studio, and then you'll add the About page content on the Schedule & Details page.

To add your content in Studio:

1. Add your descriptive picture. To do this, change the file name of the picture that you want to use to `images_course_image.jpg`. The picture that has this file name automatically appears on the course About page. Then, upload that file to the Files & Uploads page.

2. On the Settings menu at the top of any page, click Schedule & Details.

3. Under Course Schedule, enter your course start and end dates, along with the enrollment start and end dates. You can also specify times.

4. Under Introducing Your Course, enter the following in the Course Overview field.
- Long course description
- Prerequisites
- Staff bios and pictures
- FAQs

Note that this information must be formatted in HTML. By default, the Course Overview field contains a basic template for this information. For a more detailed template that includes step-by-step instructions and placeholder text, see the Appendix.

5. Under Course Introduction Video, enter the YouTube ID for your course’s About video in the Your YouTube video’s ID box.

After you add the YouTube ID for your About video, an image from your video appears in the large gray field above the your YouTube video’s ID box.

6. Under Requirements, enter the time you expect your students to spend on your course each week in the Hours of Effort per Week box.

**Action: Add the About Page Information to Studio**

Add your About page information to the Schedule & Details page in Studio.
Producing Videos
Creating Video Segments

Introduction

(Embedded video, transcript follows)

CLAYTON HAINSWORTH:

Hi. I'm Clayton Hainsworth. I'm the head of the Media Department here at edX, and I've been in video production for about the last 15 years of my career. I wanted to share with you some of our best practices about creating content for the edX platform, and also create a space for you to be able to share those back with us. As we go through this, we'll learn different things and we want to definitely partner with all of our institutions so that we create this community of learning. We want to share best practices so that you can be successful in delivering your content within the edX platform.

It's really important to understand that this is complicated. Video production requires a lot of support and is very different from your traditional classroom. This is a modality shift from theatrical in the classroom, where you're reaching one to many, so you may be having 20, 50, 100 students, to a far more cinematic mode, where you're reaching one to one. You may be talking to thousands of students through the edX platform, but you're reaching them one at a time, from the camera to the screen on the other side.

This is a really exciting time for you to think about how you are going to redesign your course for the online platform. Before you start recording, we've provided a number of questions that we use to help you start thinking through this. They boil it down to a number of simple topics. How
are you most comfortable presenting? What's the environment that makes the most sense? And finally, do you have any pre-existing materials that you need to use in order to teach your content?

In the next segment, we're going to talk about practical tips for the recording process for the edX platform.
Creating Engaging Video Content

CLAYTON HAINSWORTH:

I want to talk to you about how to create the best engaging video content that you can. Here at edX, we're always learning and sharing. And we're focused on learning new techniques and using new technologies to create new capture methods. Right now what I want to do is share with you some of our best practices and then talk about some common modes of capture.

One of those best practices is delivering the content from a one-to-one relationship. What that means is that you need to create good eye contact, as well as try and speak in a conversational tone. What I'm really trying to do is have a conversation with you, just on the other side of that screen.

One of the other things that we often encourage is creating short segments for your content. Ideally, three to seven minutes long. This gives you the opportunity to think about your content in a way that is extremely modular and extremely succinct. What you can do is take these segments and put them side by side, and then put exercises after them that underscore the ideas that you're trying to set forth for the learner.

The final best practice that I want to focus on is capturing good quality audio. Your video is only as good as your audio, and it's important that when you start recording that you do a quick test of that recording, and check it back. That way if there is an issue, you can correct it before you spend a lot of time recording, and find out that you have either marginalized, or unusable audio.

I now want to talk about one of the common modes of capture, which is a tablet capture. This is where you're using some sort of interface that allows you to present slides and either write out formulas or draw directly on them. What we encourage you to do is try and create as much visual interest as you can. And what we've actually found is that drawing on a slide is quite effective. You may think that drawing on your slides could be problematic in that you would be delivering your content, pausing. But through the editorial process it allows us to actually condense that and make you write and talk at the same time.

One of the other common modes of capture is a two-stream capture where you're using something like a tablet and a camera. That allows us to do two things. It allows us to show the tablet as you're working on it. Obviously your head would be down and you would be looking at the tablet. And you wouldn't want to show that. And what we really want to do is give you the direction of, when you are creating the content and you've gone through and you've developed...
this tablet capture, and then you want to deliver that teachable moment, you look up. And you follow that first best practice, which is to deliver to camera conversationally eye to eye. So here’s the moment where I would want to look directly in the camera and let you know that this is an important piece for you to understand.

We look forward to seeing your content on edX.
Tips on Turning a Lecture into Video Segments

Videos for the online format come in small pieces called "segments". A segment should be less than 15 minutes long (probably 5 to 10 minutes) and differs from a lecture in the following ways.

- It focuses on one specific topic.
- You can leave computations as lecture-exercises.
- You can talk whatever pace is comfortable to you, even if your classroom students complain that you talk too quickly or slowly; our video player has a multispeed function that allows students to speed up or slow down the video.
- To the extent possible, is best for video segments to be self-contained. This makes it easier to change the ordering later if you want to.

**Note** Using copyrighted material without permission is more problematic in this context than in the classroom because there is much more public visibility. Please do not use copyrighted material without proper licensing! This includes stock footage, images, documents, and other graphic representations (such as video games) not available under Creative Commons.
Summary of Video Styles

Keep in mind that not every style works for everyone. Even the most engaging classroom lecturer can come off as wooden and staged when faced with only a camera. Play your strengths!

- Tablet Capture
- PowerPoint with annotations
- Filmed blackboard lecture
  - Live
  - Not live but classroom
  - Studio
  - Home video

Advantages, Disadvantages, and Logistics

Tablet Capture

- (+) It is easy, fast, and excellent for conveying math heavy concepts
- (+) Can allow for 'one-on-one' feel between the instructor and online learner
- (+) Because the speaker is not visible, these videos can be heavily and easily edited, so you can start and stop multiple times
- (-) Engagement is only conveyed via voice, annotations and drawing, which is not everyone's strong suit

PowerPoint with Annotations

- (+) Easy and fast
- (+) If done well, can allow for 'one-on-one' feel between the instructor and the online learner
- (-) Usually comes across as a presentation not a conversation
- (-) Can be difficult to spark and sustain learner engagement because of static visuals

Filmed Blackboard Lecture

Live Audience

- (+) Learning atmosphere for the online student can feel more dynamic than with tablet capture
- (+) Allows for more human connection because student can see the instructor
- (+) Environment is comfortable and familiar to the lecturer
- (+) Lecturer can guage subtle audience response to material and modulate accordingly. This can be surprisingly important.
- (-) Content, video and audio errors are difficult to correct
• (-) Focus is on classroom students not the online learner, in terms of material as well as the film production (it is like a film of a play, doing disservice to the best aspects of both)
• (-) Not all of the relevant written information is always visible
• (-) In-class students may ask questions the online student cannot hear; even if the question is repeated by the professor, this can feel disruptive and alienating
• (-) Requires camera person who understands when to zoom in or pan out, based on material covered

Not Live but Classroom (similar to above, with the following additions)
• (+) Allows for more production options in terms of camera angles (e.g. camera can be closer to front of classroom, creating a more intimate feeling)
• (+) Can be interrupted and re-shot
• (-) Need a really good camera operator and lighting
• (-) Doesn't use the medium to its maximum effect (without an audience, why still be in a classroom?)
• (-) Requires a lot of performance on the part of the lecturer
• (-) Requires time outside of on-campus teaching responsibilities

Studio
• (+) Controlled environment with professional, skilled camera operators
• (-) Potentially sterile, limited and expensive
• (-) May be difficult to schedule studio time

Home Video
• (+) Easy to schedule
• (+) Can be really personable and convey one-on-one feel
• (-) Audio and lighting quality can be weak and convey lack of planning and forethought
• (-) Requires a lot of filming skills
Sal Khan on How to Do Khan-style Videos

Sal Khan became famous for his instructional YouTube videos, which he created when he was tutoring his cousin. He now runs the Khan Academy. He graciously created this video for edX on how he does his magic.

(Embedded video, transcript follows)

SAL KHAN:

Anant Agarwal at MIT, or I should say edX, has come up with an acronym, which, I guess, is based to some degree on what I'm doing now, which he calls KSV, which very flatteringly refers to Khan Style Videos. And he graciously asked me if I could do a KSV--a Khan-style video--on making KSVs. So I will make my best attempt at this very meta-level task. And this is just my guess of why KSVs may resonate with people.

And the first one is that they're just conversational. So if you were to make a video, or if you want to make a video like this, I encourage you to talk in the videos the way that you would talk to another human being, the way you would actually have a conversation with another human being, maybe in a tone not too different than I'm speaking right now. If you were in the room with me, I would be talking to you just like this, as strange as that might seem.

Now, what is not conversational? Well, there's a couple of non-conversational styles. The least conversational style, and, unfortunately, this is the style that we see the most in education materials, is what I would call the highly scripted, highly polished, but emotionless style. So let me write this down this way. So not emotionless. And if you want an example of emotionless, go on YouTube and do a search for some concept, I don't know, in chemistry, or biology, you know, photosynthesis or something. And you'll find these videos that are very polished. And they'll have some computer graphics with mitochondria in them, and all things like that. And then they'll have some type of a voice over during those videos that sound not too different than your GPS device. It'll sound--the next step in mitochondria is when the ATP gets a new--and even though it sounds very professional, it actually is not what human beings, or at least in my opinion, what human beings want to hear. They don't relate to it. That clearly does not sound like another human being. You'll see it in a lot of education videos, where it's clear that
someone wrote a script, and that they're paying someone, or they hired someone else to read
the script. Or maybe they are reading their own script. But it just doesn't come off like it that
person is thinking. Human beings, as listeners, are highly, highly, highly sensitive to what is
going on in the speaker's brain. If the speaker's brain is not thinking in real time, if they're just
reading something, then your cue is, well, why should I be thinking it through? Or why should I
care? If the speaker is emotionless, doesn't really care about what they're talking about, there's
no intonation in their voice, then it's a big cue. Especially if it's dense material, it's a big cue for
the user not to care too much about it.

Now the other, I guess I would categorize as not to do or, at least, that I try not to do, is talking
above the student. Be respectful of the student, but don't talk above them. And I think most of
us recognize what talking above is. This is, OK, the next step is going to be really easy. OK,
well, we can skip those steps, and then we can do that. And then, obviously, this is intuitively
obvious. So we can move onto the next. That's usually talking above. And I think most people
recognize that that's not always the most helpful thing. Now what's not obvious is, oftentimes,
when someone is talking above, and you say, hey, look, just, slow down a little bit. Not
everyone's an expert. They go into the other mode, which might be just as bad, or worse, which
is talking below-- talking below the student. And this is where it gets really, really, it almost gets
condescending, where it's, the next-- so let's do an algebra problem. The next step-- and you
talk to the user like they are a six-year-old. And one thing that I've found is, not even six-year-
olds like to be talk like they are six-year-olds. They want to be engaged with where you
respect them. You respect their ability to understand things. But you also respect that they are
new to the topic, and you're willing to, essentially, work through things.

Now, the next big point, the next big point, and I'll use an appropriate color for this, which is very
important on a meta level, is to use colors and visuals. So I'll write visuals first. Visuals and
colors with the one caveat that it doesn't have to be too fancy-- not too fancy. And not too fancy,
there's two implications for not too fancy. My personal opinion is, hand-drawn-- I don't know if
there's a hyphen there. I'll write it with a hyphen. I personally believe that, when someone sees
something hand drawn, it actually resonates with their brain better. They can process it better.
They can see it written out. If they just see it in a PowerPoint, or they just see something pop
up, it doesn't quite connect with their brain in exactly the same way. The other thing is, you get
higher output, more output, more output. If I took the time to make computer graphics for this
little thing that I'm making here-- and it's not ideal. I'm obviously overloading the use of this
yellow and all the rest. But it would have taken a lot more time. And if I'd written the script, and
all these computer graphics, it would have taken a lot more time. And it's not clear to me that it
would have been that much more impactful. So conversational, visuals, and colors. And I'm a
big fan of diagramming. I don't know exactly whether I have something to diagram just at this
moment but, if I did, I would enjoy diagramming it.

The next idea is the idea around preparation. And here the preparation I'd advocate, or at least
what works for me is very different than what sometimes is associated with preparation. For me,
it's very important to prepare your mind, but once you prepare your mind-- and this is nothing to
sneeze at. I mean, preparing your mind, you might spend a lifetime preparing your mind in order
to be able to teach something. But prepare your mind, make sure it's very distilled in your mind
that you have clear visuals in your brain of how this concept can be depicted. I often go on hour-
long walks to think about what's the best way to visualize one concept or another. But once you
prepare your mind, speak directly from it. Speak from my mind, and, if possible, your heart as
well, not to get a little bit too touchy-feely.

The thing not to do, and this goes back to the "nots" right over here is the emotionless-- I know it
does work for some folks, but I'm personally not a fan of scripts. So let me write this this way. So
no scripts. Scripts are OK. Actually, oftentimes, the process of writing a script, or writing an outline, could prepare your mind, at which point I would encourage you to speak from your mind or your heart. And you often over think things. You often, oh, I said um once, or I mumbled a little bit. But people actually don't mind that as long as they can tell it's coming from you, and as long as you're able to articulate it, eventually, in a fairly clean way. Scripts, also, I've seen people who can read the script in a very believable way. I think they're good at letting their brain re-engage with the concept. So there's a famous quote. I actually forget who it's who it's by. I have a feeling it was Mark Twain, or Benjamin Franklin, or one of those characters, who say, famously, a lecture is a transferring from the professors notes to the students notes without going through the brains of either of them. And that's essentially what a script is doing. A good script reader, if you've got to read scripts, will read the script, but actually process it in their brain. So they'll process it in their brain right there. And when they process it in their brain, what comes out of their mouth is something that a student's brain will actually want to process and actually want to listen to.

Now, the last point I'll make, and this was something that was initially forced on to me by YouTube-- I don't want to make the YouTube logo-- but it's the idea of length. It's the idea of, try to be as short as possible, but not too short. I've found that most concepts can be articulated reasonably well in about 10 minutes. If 10 minutes is not enough time, so if you feel like you need 60 minutes to articulate a concept, you should be able to chunk that down into 10, or 6 minute, or 8 minute chunks. And what the value of that is, well there's several values there. One is, as a content producer, especially if you're speaking from your mind, you're much less likely to make a careless mistake in 10 minutes then you are in 60 minutes. But it also becomes much easier for the student to navigate. Maybe the student is only having trouble, right, over here with Fourier transforms and doesn't have to listen to this part over here. And so they can just immediately move in to that part of it. It's like a good action novel, you know one of those Dan Brown Da Vinci Code type novels, where you feel like, oh I can read the next chapter. The next chapter is three minutes, or three pages, but by doing that, you reach a milestone, you say, oh, I'll read another one. And so, to a large degree, the person's mind, because they are able to rest, would be more likely to stay engaged over a longer period of time. And there's actually a whole bunch of research to this effect that students actually have trouble, in one stretch, especially, with really deep dense content, they have trouble paying attention for more than 10 minutes. So not to be a hypocrite, I realize that I'm at 9 minutes and 44, so I will leave you there.
How to Be an Engaging Presenter in this Medium

Unless you are videotaping live lectures with a student audience, what you need to do with your voice and body language is not the same online as in your classroom.

**General Tips**

- Speak comfortably fast; students can use our multispeed function to slow down or speed up your speech.
- Look directly into the camera - imagine a person where the camera is and imagine a conversation.
- For tablet capture: imagine you are sitting next to your student, explaining the material.
- It may be helpful to record with an audience present.
- For tablet capture: If you make a mistake, stop for a second and then restart the last sentence. In the Tutorial "How to Create a Tablet Capture," you will see the raw footage for the "edX Highlights" video. You can see in that video the drastic effect that good editing can have.
Audio Guidelines

Capturing quality audio is the most important technical aspect of creating a good video. Here are some guidelines.

- It is important to use an external microphone! Most external microphones are fine.
- Proximity of the microphone to you is more important than the microphone quality.
- Don’t forget to turn it on! Test it out before beginning.
- Before recording, check to make sure the volume is correctly adjusted.
- Make sample recordings and have someone else give you feedback about the sound quality.
Writing Exercises
Writing Online Exercises

Overview

Below are some links to video demonstrations of some specialized edX interactive tools.

- Interactive Periodic Table
- Crystallography exercises
- VSEPR Diagram Exercises
- Point Charge Simulation

In this lesson, we will discuss how to adapt your exercises to the online format. We will cover the following topics:

Anatomy of an exercise. Every edX exercise is composed of many customizable pieces.

Context. Exercises generally appear in lessons, homework assignments, quizzes, and exams; we will discuss the differences between these types of exercises.

Considerations for the online format. Some things to keep in mind when writing exercises for a global audience.

Please note that this section is a high-level overview of pedagogy and available features. For information on the technicalities of creating problems in Studio (e.g., for a walk-through on how make an exercise appear in your courseware), refer to Building.
Anatomy of an Exercise

In this lesson, you create and post the preliminary materials for your course. These generally include materials such as a schedule, textbook, and syllabus. In this early version of Studio you will use HTML to create some of these materials; we have provided templates when that’s the case.

3.091x Problem Set 1

1. Problem text. The problem text can contain any standard HTML formatting.

2. Student input. There are many different types of student input. These will be referred to as Response Types. All of the Response Types are described in the Exercise Gallery.

3. Rendered answer.

4. Check button. All problems have a Check button. The student can press the Check button to get feedback on whether he or she did a problem correctly. When the student presses Check, the grade and current state of the problem is automatically saved.

5. Save button. This is a button the student can press to save the current state of the problem without having to submit it for a grade. This allows the student to begin working on a problem and then finish it later.

6. Show Answer button. When the student clicks Show Answer, both the correct answer (see 2 above) and the explanation (see 7 below) are shown to the student. The instructor can set whether the Show Answer button is visible.

7. Attempts. For each problem, the instructor may set either a specific number of attempts or “unlimited” attempts.

After you click Check followed by Show Answer, the problem looks like this:
3.091x Problem Set 1

8. Feedback. After clicking Check, all problems return a green check or a red X.

9. Correct answer. Most problems require that the instructor specify a single correct answer in advance.

10. Revealed explanation. For all response types, the instructor may include an explanation that gets revealed when the Show Answer button is pressed.

11. Reset button. Clears the student input, so that the problem looks the way it did originally.

12. Hide Answer button.

There are also some attributes of problems that are not immediately visible.

13. Grading. The instructor may specify whether a group of problems is graded.

14. Due date. A problem with an elapsed due date will not have a Check button. It will neither accept answers nor provide feedback.

15. Randomization. The instructor may specify whether a problem will use numbers that are randomly generated and very from student to student.

16. Grace period. The instructor may set a single grace period that extends the due date of all problems in the course by a fixed amount.

17. Weight. Different problems in a particular problem set may be given different weights.
Context

In this lesson, you create and post the preliminary materials for your course. These generally include materials such as a schedule, textbook, and syllabus. In this early version of Studio you will use HTML to create some of these materials; we have provided templates when that’s the case.

Context

In general, exercises can be integrated into an edX course as

- In-lesson exercises,
- Homework questions,
- Quiz problems, or
- Exam problems.

In-Lesson Exercises

In-lesson exercises usually focus on one or two small ideas, and should not be so difficult as to take the student away from the lesson. It is important to remember that in-lesson exercises can be used for instruction and not just for feedback and assessment!

Some considerations for placing exercises in a sequence:

- Flow. In-lesson exercises are part of the flow of the lesson.
- Lead-in transitions. While modular exercises are useful, you can still provide context for an exercise with preceding text. Also, the default assumption is that an exercise will follow up on a concept just covered by video, and students will review previous material to answer the question. If this is not the case, some text describing the transition may help orient the reader.
- Length. Keep in mind whether you intend students to answer quickly and return to lecture.

Homework Questions

These tend to be longer questions that connect several different concepts and require deeper thinking.

Quiz Questions

These tend to be shorter graded exercises to give students a chance to test their understanding of a section of the material and prepare for exams.
Exam Questions

These tend to be more serious problems that test the students' understanding of a section they have finished. Usually exams happen at a fixed time and have a shorter time limit. Discussion forums can be turned off during exams.
Considerations for the Online Format

Some considerations for phrasing exercises:

- The student body is global. Culture-specific references or phrases like "at this stage" and "all set" can translate badly into other languages.

- Be specific and concrete. Vague phrases or open ended phrases like "what do you expect" and "predict the result" can be difficult to interpret. Ambiguous wording can be distracting and can create the perception of subjectivity or unfairness.

- The internet is always available. Students have search engines, libraries, and other external data readily available. If they need to look up particular resources they can be directed to the relevant sources.

- Modularity will pay off both now and next semester. An exercise that does not depend on its surrounding videos can be easily reused. (Example: An exercise that says "What is the solution to the problem in the previous video?" is a question tied to a particular point in a particular lesson, while "What is the solution to $y = mx + b$?" allows more flexibility. Also, self-contained questions are easier to understand and harder to misinterpret.)
Introduction

You can use Studio to create many different types of exercises, from simple multiple choice exercises to more advanced exercises that require the student to "build" a virtual circuit. Studio even provides the ability to grade open-ended short answer and essay questions. This lesson provides a list of many of the exercise types that are currently available, organized according to the way you create them in Studio.

- **Common problems** appear on the **Common Problem Types** tab when you create a new Problem component in Studio. Studio provides templates for these problems, and you can create them without using XML.

- **Advanced problems** appear on the **Advanced** tab when you create a new Problem component. Studio also provides templates for these problems, but they have to be created in XML.

- **Specialized problems** are advanced problems that aren't available by default. To add these problems, you first have to modify the advanced settings in your course. The Advanced component then appears under **Add New Component** in each unit.

For each exercise type in this lesson, you'll be able to try out a working example of the exercise, read a description of the exercise, and learn how to create the exercise in Studio.
This lesson concludes with a list of the settings that are available for each problem type, as well as a description of each setting.

**List of Exercise Types**

**Common Problem Types**

- **Checkbox**: In checkbox exercises, students select one or more options from a list of possible answers.
- **Dropdown**: In dropdown exercises, students select one answer from a dropdown list.
- **Multiple choice**: Multiple choice exercises require students to select one answer from a list of choices that appear directly below the question.
- **Numerical input**: Numerical input exercises require answers that include only integers, fractions, and a few common constants and operators.
- **Text input**: In text input exercises, students enter a short text answer to a question.

**Advanced Problems**

- **Circuit schematic builder**: In circuit schematic exercises, students create and modify circuits on an interactive grid and submit computer-generated analyses of the circuits for grading.
- **Write-your-own-grader ("Custom Python-evaluated input")**: Write-your-own-grader exercises evaluate students’ responses using an embedded Python script that you create. These exercises can be any type.
- **Drag and drop**: Drag and drop exercises require students to drag text or objects to a specific location on an image.
- **Image mapped input**: Image mapped input exercises require students to click a specific location on an image.
- **Math expression input**: Math expression input exercises require students to enter a mathematical expression as text, such as \( e=m*c^2 \).
- **Problem written in LaTeX**: This problem type allows you to convert problems that you’ve already written in LaTeX into the edX format. Note that this problem type is still a prototype, however, and may not be supported in the future.
- **Problem with adaptive hint**: These exercises can give students feedback or hints based on their responses. Problems with adaptive hints can be text input or multiple choice exercises.

**Specialized Problems**

- **Annotations**: Annotation exercises ask students to respond to questions about a specific block of text. The question appears above the text when the student hovers the mouse over the highlighted text so that students can think about the question as they read.
- **Open Response Assessments**: Open response assessments allow students to enter short answer or essay responses that students or a computer algorithm can then grade.
• **Word cloud**: Word cloud exercises show a colorful graphic of the words that students enter as responses to a prompt.

**Problem Settings**

• **Problem Settings in Studio**: Lists the settings available for the different problem types in Studio. Note that all problem types except open response assessments and word clouds have the same collection of settings.
Common Problems

Common problems are typical problems such as multiple choice problems and other problems whose answers are simple for students to select or enter. You can create all of these problems using the Simple Editor in Studio. You don't have to use XML or switch to the Advanced Editor.

These problems are easy to access in Studio. To create them, click Problem under Add New Component, click the Common Problem Types tab, and then click the name of the problem. (Note that Checkbox doesn't appear in the list of common problem types. To create a checkbox problem, click Blank Common Problem.)

More information about how to create each problem is provided in the unit for that problem type.
Checkbox

In checkbox exercises, the student selects one or more options from a list of possible answers. The student must select all the options that apply to answer the exercise correctly.

Example Checkbox Exercise

This exercise first appeared in MITx's 14.73x The Challenges of Global Poverty course, spring 2013.

Learning about the benefits of preventative healthcare can be particularly difficult. Check all of the reasons below why this may be the case.

- A large amount of time passes between undertaking a preventative measure and seeing the result.
- Non-immunized people will always fall sick.
- If others are immunized, fewer people will fall sick regardless of a particular individual’s choice to get immunized or not.
- Trust in healthcare professionals and government officials is fragile.

EXPLANATION

People who are not immunized against a disease may still not fall sick from the disease. If someone is trying to learn whether or not preventative measures against the disease have any impact, he or she may see these people and conclude, since they have remained healthy despite not being immunized, that immunizations have no effect. Consequently, he or she would tend to believe that immunization (or other preventative measures) have fewer benefits than they actually do.

How to Create This Exercise

For step-by-step instructions that describe how to create a checkbox problem in Studio, see the Create Problem Components unit in the Studio tutorial, and scroll down to Create a Checkbox Problem.

For the example problem above, the text in the Problem component is the following.

Learning about the benefits of preventative healthcare can be particularly difficult. Check all of the reasons below why this may be the case.

[x] A large amount of time passes between undertaking a preventative measure and seeing the result.
[ ] Non-immunized people will always fall sick.
[x] If others are immunized, fewer people will fall sick regardless of a particular individual’s choice to get immunized or not.
[x] Trust in healthcare professionals and government officials is fragile.
[Explanation]
People who are not immunized against a disease may still not fall sick from the disease. If someone is trying to learn whether or not preventative measures against the disease have any impact, he or she may see these people and conclude, since they have remained healthy despite not being immunized, that immunizations have no effect. Consequently, he or she would tend to believe that immunization (or other preventative measures) have fewer benefits than they actually do.

[Explanation]
For more information, see the Template and XML Documentation for this exercise type.
Dropdown

Dropdown exercises allow the student to choose from a collection of answer options, presented as a dropdown list. Unlike multiple choice exercises, whose answers are always visible directly below the question, dropdown exercises don’t show answer choices until the student clicks the dropdown arrow.

Example Dropdown Exercise

This exercise first appeared in HarvardX’s PH207x Health in Numbers: Quantitative Methods in Clinical & Public Health Research course, fall 2012.

What type of data are the following?

Age:

(dropdown box—choices nominal, discrete, continuous)

Answer: Continuous

Age, rounded to the nearest year:

(dropdown box—choices nominal, discrete, continuous)

Answer: Discrete

Life stage - infant, child, and adult:

(dropdown box—choices nominal, discrete, continuous)

Answer: Nominal

How to Create This Exercise

To create a dropdown problem, follow these steps.

1. Under Add New Component, click Problem.
2. In the Select Problem Component Type screen, click Dropdown on the Common Problem Types tab.
3. In the new Problem component that appears, click Edit.
4. Replace the default text with the text for your exercise.
   - Enter each of the possible answers, separated by commas, between the two sets of square brackets ([ and ]).
   - Surround the correct answer with parentheses.

For the example problem above, the text in the Problem component is the following.

What type of data are the following?
Age:
[[Nominal, Discrete, (Continuous)]]
Age, rounded to the nearest year:
[[Nominal, (Discrete), Continuous]]
Life stage – infant, child, and adult:
[[Nominal), Discrete, Continuous]]

For more information, see the Template and XML Documentation for this exercise type.
Multiple Choice

In multiple choice exercises, students select one option from a list of answer options. Unlike with dropdown exercises, whose answer choices don't appear until the student clicks the drop-down arrow, answer choices for multiple choice exercises are always visible directly below the question.

Example Multiple Choice Exercise

This exercise first appeared in HarvardX's PH278x: Human Health and Global Environmental Change course, spring 2013.

Lateral inhibition, as was first discovered in the horsehoe crab...

- is a property of touch sensation, referring to the ability of crabs to detect nearby predators.
- is a property of hearing, referring to the ability of crabs to detect low frequency noises.
- is a property of vision, referring to the ability of crabs eyes to enhance contrasts.
- has to do with the ability of crabs to use sonar to detect fellow horseshoe crabs nearby.
- has to do with a weighting system in the crabs skeleton that allows it to balance in turbulent water.

EXPLANATION

Horseshoe crabs were essential to the discovery of lateral inhibition, a property of vision present in horseshoe crabs as well as humans, that enables enhancement of contrast at edges of objects as was demonstrated in class. In 1967, Haldan Hartline received the Nobel prize for his research on vision and in particular his research investigating lateral inhibition using horseshoe crabs.

How to Create This Exercise

For step-by-step instructions that describe how to create a multiple choice problem in Studio, see the Create Problem Components unit in the Studio tutorial, and scroll down to Create a Multiple Choice Problem.

For the example problem above, the text in the Problem component is the following.

Lateral inhibition, as was first discovered in the horsehoe crab:

( ) is a property of touch sensation, referring to the ability of crabs to detect nearby predators.
( ) is a property of hearing, referring to the ability of crabs to detect low frequency noises.
(x) is a property of vision, referring to the ability of crabs eyes to enhance contrasts.
( ) has to do with the ability of crabs to use sonar to detect fellow horseshoe crabs nearby.
( ) has to do with a weighting system in the crabs skeleton that allows it to balance in turbulent water.

[Explanation]
Horseshoe crabs were essential to the discovery of lateral inhibition, a property of vision present in horseshoe crabs as well as humans, that enables enhancement of contrast at edges of objects as was demonstrated in class. In 1967, Haldan Hartline received the Nobel prize for his research on vision and in particular his research investigating lateral inhibition using horseshoe crabs.

[Explanation]

For more information, see the Template and XML Documentation for this problem type.
Numerical Input

In numerical input exercises, students enter numbers or specific and relatively simple mathematical expressions to answer a question. You can specify a margin of error for students' responses.

Responses for numerical input exercises can include integers, fractions, and constants such as $\pi$ and $g$. Responses can also include text representing common functions, such as square root (sqrt) and log base 2 (log2), as well as trigonometric functions and their inverses, such as sine (sin) and arcsine (arcsin). For these functions, Studio changes the text that the student enters into mathematical symbols. The following example shows the way Studio renders students' text responses in numerical input problems. To see more examples, scroll down to More Information.

Example Numerical Input Exercise

This exercise first appeared in BerkeleyX's CS188.1x Artificial Intelligence course, fall 2012.

How many nodes are in the complete search tree for the given state space graph? The start state is S. You may find it helpful to draw out the search tree on a piece of paper.

Answer: 3.14159
How to Create This Exercise

For step-by-step instructions that describe how to create a numerical input problem in Studio, see the Create Problem Components unit in the Studio tutorial, and scroll down to Create a Numerical Input Problem. You can also view the Template and XML Documentation for this problem type.

More Information

The following are a few more examples of the way that Studio renders the text that students enter.

\[
\sin(\pi/5)
\]

\[
\sin\left(\frac{\pi}{5}\right)
\]

\[
\sqrt{(\pi - 0)^2 + (0 - 3)^2}
\]

\[
\sqrt{\pi^2 + e^2}
\]

\[
1 + e^{\sin(\pi/2) + 2 + i)}
\]

\[
1 + e^{\sin\left(\frac{\pi}{2}\right) + 2 + i}
\]
Text Input

In text input exercises, students enter text into a response field. The response can include numbers, letters, and special characters such as punctuation marks. Because the text that the student enters must match the instructor’s specified answer exactly, including spelling and punctuation, we recommend that you specify more than one attempt for text input exercises to allow for typographical errors.

Example Text Input Exercise

*This exercise first appeared in MITx's 14.73x The Challenges of Global Poverty course, spring 2013.*

What is the technical term that refers to the fact that, when enough people sleep under a bednet, the disease may altogether disappear?

Answer: herd immunity

**EXPLANATION**

The correct answer is **herd immunity**. As more and more people use bednets, the risk of malaria begins to fall for everyone – users and non-users alike. This can fall to such a low probability that malaria is effectively eradicated from the group (even when the group does not have 100% bednet coverage).

How to Create This Exercise

To create a text input problem, follow these steps.

1. Under Add New Component, click Problem.
2. In the Select Problem Component Type screen, click Text Input on the Common Problem Types tab.
3. In the new Problem component that appears, click Edit.
4. Replace the default text with the text for your exercise.
   - Insert an equal sign (=) before the answer.
   - Optionally, add additional information in [explanation] tags. This information appears after the student clicks Show Answer.
5. In the upper right corner of the component editor, click Settings. Enter the name of your exercise in the Display Name field, and then click Save.

For the example problem above, the text in the Problem component is the following.

What is the technical term that refers to the fact that, when enough people sleep under a bednet, the disease may altogether disappear?
= herd immunity

[explanation]
The correct answer is herd immunity. As more and more people use bednets, the risk of malaria begins to fall for everyone – users and non-users alike. This can fall to such a low probability that malaria is effectively eradicated from the group (even when the group does not have 100% bednet coverage).
[explanation]

For more information, see the Template and XML Documentation for this problem type.
Advanced Problems

Advanced problems are problems such as drag and drop, circuit schematic builder, and math expression problems. These problems appear on the Advanced tab when you create a new Problem component. Studio provides templates for these problems, but the problems open directly in the Advanced Editor and have to be created in XML.

These problems are easy to access in Studio. To create them, click Problem under Add New Component, click the Advanced tab, and then click the name of the problem that you want to create.

More information about how to create each problem is provided in the unit for that problem type.
Circuit Schematic Builder

In circuit schematic builder exercises, students can arrange circuit elements such as voltage sources, capacitors, resistors, and MOSFETs on an interactive grid. They then submit a DC, AC, or transient analysis of their circuit to the system for grading.

Example Circuit Schematic Builder Exercise

This exercise first appeared in MITx's 6.002x: Circuits and Electronics course, spring 2012.

A circuit that combines two or more signals is called a mixer. In this lab, your goal is to build a mixer that combines the signals generated by two voltage sources, V1 and V2, where:

- V1 is a 1 kHz square wave that varies between 0V and +1V, and
- V2 is a 5 kHz sine wave that varies between -1V and +1V.

Please design a circuit that mixes V1 and V2 to produce V_{out} such that

\[ V_{out} \approx 12V_1 + 16V_2. \]

The maximum value of the resulting output should be approximately 667mV and the minimum value should be approximately −167mV.

Enter your circuit below, using the appropriate configuration of resistors. Please do not modify the wiring or parameters of the voltage sources -- your goal is to take the signals they generate and combine them, not to change what is generated. Run a 5ms transient analysis to verify the correct operation of your circuit. We will be checking for the transient waveform at the "output" node.

EXPLANATION:

The goal is to design a mixer circuit with characteristics of \[ V_{out} = \frac{1}{2} \cdot V_1 + \frac{1}{6} \cdot V_2 \] You might have started to design your mixer with two resistors only as the example suggests. But working
through the math, soon you'll realize that the equations return no non-zero value for the resistor components. Thus you have to change the design. The next simplest design will be to add a resistor $R_3$ that connects the node $V_{out}$ to ground. See the schematic below:

Since we are going to use only linear elements in this circuit (resistors are linear), superposition will hold and thus one can look at the effect of each source $V_1$ and $V_2$ one at the time:

$$V_{out1} = V_1 \cdot \frac{(R_2 \parallel R_3)}{(R_2 \parallel R_3 + R_1)}$$

$$V_{out2} = V_2 \cdot \frac{(R_1 \parallel R_3)}{(R_1 \parallel R_3 + R_2)}$$

$$V_{out} = V_{out1} + V_{out2} = V_1 \cdot \frac{(R_2 \parallel R_3)}{(R_2 \parallel R_3 + R_1)} + V_2 \cdot \frac{(R_1 \parallel R_3)}{(R_1 \parallel R_3 + R_2)} = \frac{1}{2} \cdot V_1 + \frac{1}{6} \cdot V_2$$

Therefore:

$$\frac{(R_2 \parallel R_3)}{(R_2 \parallel R_3 + R_1)} = \frac{1}{2}$$

$$\frac{(R_1 \parallel R_3)}{(R_1 \parallel R_3 + R_2)} = \frac{1}{6}$$

So we have to solve for the resistors given these two equations. You might notice that we have 2 equations and 3 unknowns, and that there is therefore not a unique solution. That is okay, though. We only have to worry if there is no solution, not if there are too many solutions. We will simply find one of the many possible correct answers by arbitrarily choosing a value for one of the variables later.

The first equation simplifies to $R_1 = R_2 \parallel R_3$ and the second simplifies to $R_2 = 5 \cdot R_1 \parallel R_3$ Expanding the notation gives:
\[
\frac{1}{R_1} = \frac{1}{R_2} + \frac{1}{R_3}
\]
\[
\frac{1}{R_1} + \frac{1}{R_3} = \frac{5}{R_2}
\]

Subtracting these two equations will yield \( R_2 = 2 \cdot R_3 \) and putting this back to the starred equation, will result in \( R_1 = \frac{2}{3} \cdot R_3 \) So now we have \( R_2 \) and \( R_1 \) in terms of \( R_3 \) with the following ratios:

\[
R_2 = 2 \cdot R_3
\]
\[
R_1 = \frac{2}{3} \cdot R_3
\]

Since the design hadn't mentioned anything about the resistances, one can use a simple value of \( R_3 = 3 \Omega \) and find the rest accordingly:

\[
R_1 = 2 \Omega
\]
\[
R_2 = 6 \Omega
\]
\[
R_3 = 3 \Omega
\]

With these resistor values, doing a transient analysis shows a result which meets the required specs of \( V_{out} \).

**How to Create This Exercise**

1. In the unit where you want to create the exercise, click **Problem** under **Add New Component**, and then click the **Advanced** tab.
2. Click **Circuit Schematic Builder**.
3. In the component that appears, click **Edit**.
4. In the component editor, replace the example code with your own code.
5. Click **Save**.

For more information, see the [Schematic Response Reference](#) and the [Template and XML Documentation](#) for this exercise type.
Write-Your-Own-Grader ("Custom Python-Evaluated Input") Exercise

In write-your-own-grader exercises (also called "custom Python-evaluated input" exercises), the grader evaluates a student's response using a Python script that you create and embed in the problem. These exercises can be any type. Numerical input and text input exercises are the most popular write-your-own-grader exercises.

Example Write-Your-Own-Grader ("Custom Python-Evaluated Input") Exercise

Enter two integers which sum to 10:
3 and 7
Enter two integers which sum to 20:
11 and 9

EXPLANATION
Any set of integers on the line $y=10-x$ and $y=20-x$ satisfy these constraints.

How to Create This Exercise

To create a write-your-own-grader exercise:

1. In the unit where you want to create the exercise, click Problem under Add New Component, and then click the Advanced tab.
2. Click **Custom Python-Evaluated Input**.
3. In the component that appears, click **Edit**.
4. In the component editor, replace the example code with your own code.
5. Click **Save**.
Drag and Drop

In drag and drop exercises, students respond to a question by dragging text or objects to a specific location on an image.

Example Drag and Drop Exercise

The following drag and drop exercise first appeared in MIT's 7.00x: Introduction to Biology - The Secret of Life course in March 2013.

Allopurinol is a drug used to treat and prevent gout, a very painful form of arthritis. Once only a “rich man’s disease”, gout has become more and more common in recent decades – affecting about 3 million people in the United States alone. Deposits of needle-like crystals of uric acid in connective tissue or joint spaces cause the symptoms of swelling, stiffness and intense pain. Individuals with gout overproduce uric acid because they cannot eliminate it efficiently. Allopurinol treats and prevents gout by stopping the overproduction of uric acid through inhibition of an enzyme required for the synthesis of uric acid.

You are shown one of many possible molecules. On the structure of allopurinol below, identify the functional groups that are present by dragging the functional group name listed onto the appropriate target boxes on the structure. If you want to change an answer, you have to drag off the name as well. You may need to scroll through the names of functional groups to see all options.

Answer:
How to Create This Exercise

To create a drag and drop exercise:

1. In the unit where you want to create the exercise, click Problem under Add New Component, and then click the Advanced tab.
2. Click Drag and Drop.
3. In the component that appears, click Edit.
4. In the component editor, replace the example code with your own code.
5. Click Save.

For more information, see the Template and XML Documentation for this exercise type.
Image Mapped Input

In an image mapped input problem, students click inside a defined area in an image. You define this area by including coordinates in the body of the problem.

Example Image Mapped Input Exercise

In the image below, click inside any of the green rectangles.

How to Create This Exercise

To create a image mapped input exercise:

1. In the unit where you want to create the exercise, click Problem under Add New Component, and then click the Advanced tab.
2. Click Image Mapped Input.
3. In the component that appears, click Edit.
4. In the component editor, replace the example code with your own code.
5. Click Save.
For step-by-step instructions that describe how to create an image mapped input exercise in Studio, see the Add an Advanced Problem unit in the Advanced Studio Tutorial lesson. You can also view the Template and XML Documentation for this exercise type.

**Image Mapped Input - Irregular Regions**

You can set shapes other than a rectangle as your specified correct region for an image mapped input problem, using 5 coordinate points.

In the image below, click anywhere inside the green pentagon.
Math Expression Input

In math expression input exercises, students enter text that represents a mathematical expression. The edX parser instantly changes the text to a symbolic expression. Unlike numerical input exercises, which only allow integers and a few select constants, math expression exercises can include more complicated symbolic expressions.

Example Math Expression Input Exercise

This exercise first appeared in MITx’s 2.01x Elements of Structures course, spring 2013.

The composite bar $BC$ in the figure is composed of an inner core of cross sectional area $A_2 = 1 \text{ mm}^2$ and a sleeve of cross sectional area $A_1 = 4A_2$. The Young’s modulus of the sleeve is $E_1$ and the modulus of the core is $E_2 = 2E_1$. Under the effects of unknown distributed loading $f_2(x) [\text{N/m}]$, the bar is observed to deform. The measured displacement field in the bar is $u_0(x) = ax^3 - 2aLx$, where $a = 0.05 [\text{1/m}]$ is a dimensional constant and $L = 1 \text{ m}$ is the length of the bar. The origin of the $x$-axis is at the fixed support, $B$. The maximum magnitude (absolute value) of stress in the core is found to be 10 MPa.

Try it (do not include units in your response):

In terms of $L$, $a$, $E_1$, $A_2$, and $x$, write a symbolic expression for $N(x)$:

Answer: $12x^3 + L^2E_1 + A_2(x/L - 1)$

What is the numerical value of $N(x = 0)$ in Newtons?

Answer: -30

Explanation
How to Create This Exercise

To create a math expression input exercise:

1. In the unit where you want to create the exercise, click **Problem** under **Add New Component**, and then click the **Advanced** tab.
2. Click **Math Expression Input**.
3. In the component that appears, click **Edit**.
4. In the component editor, replace the example code with your own code.
5. Click **Save**.
Problem Written in \LaTeX

If you have an exercise that is already written in \LaTeX, you can use this exercise type to easily convert your code into XML. After you paste your code into the \LaTeX editor, you'll only need to make a few minor adjustments. Note that this problem type is still a prototype and may not be supported in the future, so you should use it with caution.

*Note* If you want to *use \LaTeX to typeset mathematical expressions in problems that you haven't yet written, use any of the other problem templates together with MathJax.*

Example Problem Written in \LaTeX

Estimate the energy savings (in J/y) if all the people ($3 \times 10^8$) in the U. S. switched from U. S. code to low flow shower heads.

Energy saved =

Answer: 0.52

How to Create This Exercise

To create a problem written in \LaTeX:

1. In the unit where you want to create the exercise, click **Problem** under **Add New Component**, and then click the **Advanced** tab.
2. Click **Problem Written in \LaTeX**.
3. In the component editor that appears, click **Edit**.
4. In the lower left corner of the component editor, click **Launch \LaTeX Source Compiler**.
5. Replace the example code with your own code.
6. In the lower left corner of the \LaTeX source compiler, click **Save & Compile to edX XML**.
Problem with Adaptive Hint

A problem with an adaptive hint evaluates a student's response, then gives the student feedback or a hint based on that response so that the student is more likely to answer correctly on the next attempt. These exercises can be text input or multiple choice exercises.

Example Problem with Adaptive Hint

If a bat and a ball cost $1.10 together, and the bat costs $1.00 more than the ball, how much does the ball cost? Enter your answer in cents, and include only the number (that is, do not include a $ or a ¢ sign).

(If the student answers 10, a hint appears: Hint: If the ball costs 10 cents, and the bat costs one dollar more than the ball, how much does the bat cost? If that is the cost of the bat, how much do the ball and bat cost together?)

Answer: 5

How to Create This Exercise

To create a problem with an adaptive hint:

1. In the unit where you want to create the exercise, click Problem under Add New Component, and then click the Advanced tab.
2. Click Problem with Adaptive Hint.
3. In the component that appears, click Edit.
4. In the component editor, replace the example code with your own code.
5. Click Save.
Specialized Problems

Specialized problems are advanced problems such as annotations, open response assessments, and word clouds. These problems are available through the Advanced component in Studio. To add the Advanced component to your course, you'll modify your course's advanced settings. The Advanced component then appears under Add New Component in each unit.

Add the Advanced Component to Your Course

By default, when you create a new component in Studio, you see the following options.

To create a specialized problem, you must first add the Advanced component to your course. To do this, follow these steps.

1. On the Settings menu, click Advanced Settings.

2. On the Advanced Settings page, locate the Manual Policy Definition section, and then locate the advanced_modules policy key (this key is at the top of the list).

3. Under Policy Value, place your cursor between the brackets, and then enter the value for the type of problem that you want to create. Make sure to include the quotation marks, but not the period.
   - For annotations, enter "annotatable".
   - For open response assessments, enter "combinedopenended","peergrading". (Include the comma but no spaces between the words.)
   - For word clouds, enter "word_cloud".

edX101: How to Create an edX Course
You can enter more than one problem type at a time. When you do, make sure to surround each problem type with quotation marks and separate each problem type with a comma, but do not include any spaces.

For example, if you wanted to add annotations, open response assessments, and word cloud exercises in your course, you would enter the following between the brackets.

"annotatable","combinedopenended","peergrading","word_cloud"

4. At the bottom of the page, click **Save Changes**.

The page refreshes automatically. At the top of the page, you see the following notification.

The text in the **Policy Value** field now appears as follows.

5. Return to the unit where you want to add the specialized problem. The list of possible components now contains an Advanced component.

When you click the Advanced component, you see the following list.

edX101: How to Create an edX Course
You can now create annotations, open response assessments, and word clouds in your course. More information about how to create each problem is provided in the unit for that problem type.
Annotation

In an annotation exercise, the instructor highlights specific text inside a larger text block and then asks questions about that text. The questions appear when students hover the mouse over the highlighted text. The questions also appear in a section below the text block, along with space for students’ responses.

Example Annotation Exercise

Instructions

Annotation Exercise, Hour 12 (to be done after the Hour 12 Question Set has been completed). Carefully read the following text and answer the accompanying question. To see the question while you read the text, hover your mouse over the highlighted selection in the text below.

Guided Discussion

This exercise first appeared in HarvardX’s CB22x: The Ancient Greek Hero course, spring 2013.

|122 And they [= the Golden Generation of humankind] are superhumans [daimones]. They exist because of the Will of Zeus. |123 They are the good, the earthbound [epi-khthonioi], the guardians of mortal humans. |124 They guard acts of justice [dikē] and they guard against wretched acts of evil. |125 Enveloped in mist, they roam everywhere throughout the earth. |126 They are givers of prosperity. And they had this as a privilege [geras], a kingly one [basilēion].

Annotation Exercise

|124 They guard acts of justice [dikē] and they guard against wretched acts of evil.

Hesiodic poetry declares that heroes are protectors of justice. Is this declaration consistent with what we have read in Homeric poetry?

Type your response below:

(response field)

In your response to this question, which tag below do you choose?

- This declaration in Hesiodic poetry is inconsistent with Homeric poetry, since we see that heroes in the Iliad and Odyssey do in fact engage in unjust as well as just actions.

- This declaration in Hesiodic poetry is consistent with Homeric poetry, since there is no real contradiction here between the two kinds of poetry.

- We cannot be sure whether this declaration in Hesiodic poetry is consistent or inconsistent with Homeric poetry, where the actions of heroes are not evaluated in terms of injustice or justice.

Answer: This declaration in Hesiodic poetry is consistent with Homeric poetry, since there is no real contradiction here between the two kinds of poetry.

Explanation
If you answered “This declaration in Hesiodic poetry is inconsistent with Homeric poetry, since we see that heroes in the Iliad and Odyssey do in fact engage in unjust as well as just actions,” you would be missing the point. The heroes in the Iliad and Odyssey are still alive when they are engaged in their actions. By contrast, the heroes of the Golden Age are not of this world when they are protecting justice. Enveloped in mist and roaming throughout the earth, the heroes of the Golden Age are pictured as otherworldly spirits. This picture corresponds to the idea of cult heroes, not to the idea of epic heroes. If you answered “We cannot be sure whether this declaration in Hesiodic poetry is consistent or inconsistent with Homeric poetry, where the actions of heroes are not evaluated in terms of injustice or justice,” you would be right in saying that Homeric poetry generally avoids evaluating the actions of heroes in terms of injustice or justice (though such evaluations *do* occasionally occur even in Homeric poetry, as I have argued in my overall write-up of this Hour in the book h24h), but you would still be missing the point I already made, that the heroes of the Golden Age are in an otherworldly state of existence when they are protecting justice, by contrast with the heroes of epic as portrayed in the Iliad and Odyssey. A better answer is “This declaration in Hesiodic poetry is consistent with Homeric poetry, since there is no real contradiction here between the two kinds of poetry.” Hesiodic and Homeric poetry picture heroes in different but complementary ways, and they avoid contradicting each other about the basic fact that heroes are different when they are alive and when they are dead.

**How to Create This Exercise**

To create an annotation exercise:

1. Add the Annotation advanced component. To do this, add the "annotatable" key value to the Advanced Settings page. (For more information, see the instructions in Specialized Problems.)

2. Add the Instructions and Guided Discussion segments of the problem.
   a. In the unit where you want to create the exercise, click Advanced under Add New Component.
   b. In the list of problem types, click Annotation.
   c. In the component that appears, click Edit.
   d. In the component editor, replace the example code with your own code.
   e. Click Save.

3. Add the Annotation Exercise segment of the problem.
   a. Under the Annotation component, create a new blank Advanced Problem component.
   b. Paste the following code in the Advanced Problem component, replacing placeholders with your own information.

```
<problem>
<annotationresponse>
<annotationinput>
<text>PLACEHOLDER: Text of annotation</text>
<comment>PLACEHOLDER: Text of question</comment>
```
c. Click **Save**.
Open Response Assessment

EdX has recently been developing new technology that allows instructors to administer "open response assessments": exercises in which students write short open-ended answers or essay responses to questions, and then receive feedback through self assessments, peer assessments, and artificial intelligence (AI) assessments (also called "machine grading"). We're very excited about the implications of this technology in our courses and in the wider world of online learning.

We've included an entire lesson on open response assessments in edX101 that covers what they are, why they're important, and how they're graded. For detailed information about these exercises, including step-by-step instructions for creating and grading them in Studio, see the Open Response Assessments lesson.
Word Cloud

In a word cloud exercise, students enter words into a field in response to a question or prompt. The words all the students have entered then appear instantly as a colorful graphic, with the most popular responses appearing largest. Students can both see the way their peers have answered and contribute their thoughts to the group.

Example Word Cloud Exercise

This exercise first appeared in HarvardX’s PH278x: Human Health and Global Environmental Change course, fall 2012.

In a word or two, state why you think addressing climate change has been so difficult. Please enter each word into a separate box.

(Two boxes provided)

How to Create This Exercise

To create a word cloud exercise:

1. In the unit where you want to create the exercise, click Advanced under Add New Component.

2. In the list of problem types, click Word Cloud.
3. In the component that appears, click **Edit**.

4. In the component editor, specify the settings that you want. You can leave the default value for everything except **Display Name**.

   - **Display Name**: The name that appears in the course ribbon and as a heading above the problem.

   - **Inputs**: The number of text boxes into which students can enter words, phrases, or sentences.

   - **Maximum Words**: The maximum number of words that the word cloud displays. If students enter 300 different words but the maximum is set to 250, only the 250 most commonly entered words appear in the word cloud.

   - **Show Percents**: The number of times that students have entered a given word as a percentage of all words entered appears near that word.

5. Click **Save**.
Problem Settings in Studio

Nearly every problem type that has a template in Studio has the same available settings. The exceptions are open response assessments and word clouds. Click a link below to go to the list of settings for that problem type.

- General Problem Settings
- Open Response Assessment Problem Settings
- Word Cloud Problem Settings

General Problem Settings

Every problem has several settings, or attributes, that appear when you click Settings in the component editor.

These settings are Display Name, Maximum Attempts, Problem Weight, Randomization, and Show Answer.

- Display Name is the name of your problem. The name appears as a heading over the problem in the LMS. It also appears in the course ribbon at the top of the page.
• **Maximum Attempts** indicates the number of times a student can try to answer the problem. By default, a student has an unlimited number of attempts.

• **Problem Weight** determines the maximum number of points the problem is worth. You can change the maximum number of points possible by changing the **Problem Weight** value.

It is also possible to create a problem that has more than one answer space—for example, you may create a problem that has two questions, one with a multiple choice answer and one with a numerical value answer. By default, each answer space is worth one point. Thus this problem would be worth two points. For now, we suggest that you leave the default value.

• **Randomization** defines how often inputs are randomized when a student loads the problem. This setting only applies to problems that can have randomly generated numeric values.

• **Show Answer** defines when the problem shows the answer to the student. There are seven options.
  
  • **Always**: Always show the answer when the student clicks the **Show Answer** button.
  
  • **Answered**: Show the answer after the student has submitted her final answer.
  
  • **Attempted**: Show the answer after the student has tried to answer the problem one time, whether or not the student answered the problem correctly.
  
  • **Closed**: Show the answer after the student has used up all his attempts to answer the problem or the due date has passed.
  
  • **Finished**: Show the answer after the student has answered the problem correctly, the student has no attempts left, or the problem due date has passed.
  
  • **Past Due**: Show the answer after the due date for the problem has passed.
  
  • **Never**: Never show the answer. In this case, the **Show Answer** button does not appear next to the problem in Studio or in the LMS.

Additionally, each problem has a **Check** button. By default, each **Check** button corresponds to one problem. If you want students to be able to check their answers to multiple problems at a time, you can include more than one problem in each Problem component.
Open Response Assessment Problem Settings

Problem settings for open response assessments include options such as the number of responses a student has to peer grade and whether students can upload files as part of their response.

Open response assessments include the following settings.

- **Allow “overgrading” of peer submissions**: This setting applies only to peer grading. If all of the responses for a question have been graded, the instructor can allow additional students to grade responses that were previously graded. This can be helpful if an instructor feels that peer grading has helped students learn, or if some students haven't graded the required number of responses yet, but all available responses have been graded.

- **Allow File Uploads**: This setting specifies whether a student can upload a file, such as an image file or a code file, as a response. Files can be of any type.

- **Disable Quality Filter**: This setting applies to peer grading and AI grading. When the quality filter is disabled (when this value is set to True), Studio allows submissions that are of “poor quality” (such as responses that are very short or that have many spelling or grammatical errors) to be peer graded. For example, you may disable the quality filter if you want students to include URLs to external content—otherwise Studio sees a URL, which may contain a long string of seemingly random characters, as a misspelled word.
When the quality filter is enabled (when this value is set to False), Studio does not allow poor-quality submissions to be peer graded.

- **Display Name**: This name appears in two places in the unit: in the course ribbon at the top of the page and above the exercise. The name of the problem also appears in the list of problems on the **Staff Grading, Problems I Have Submitted**, and **Peer Grading** pages that are accessed through the **Open Ended Console** page.

- **Graded**: This setting specifies whether the problem counts toward a student's grade. By default, if a subsection is set as a graded assignment, each problem in that subsection is graded. However, if a subsection is set as a graded assignment, and you want this problem to be a "test" problem that doesn't count toward a student's grade, you can change this setting to **False**.

- **Maximum Attempts**: This setting specifies the number of times the student can try to answer the problem. Note that each time a student answers a problem, the student's response is graded separately. If a student submits two responses to a peer-assessed problem (for example, by using the **New Submission** button after her first response receives a bad grade or because she wants to change her original response), and the problem requires three peer graders, three separate peer graders will have to grade each of the student's two responses. We thus recommend keeping the maximum number of attempts for each question low.

- **Maximum Peer Grading Calibrations**: This setting applies only to peer grading. You can set the maximum number of responses a student has to "practice grade" before the student can start grading other students' responses. The default value is 6, but you can set this value to any number from 1 to 20. This value must be greater than or equal to the value set for **Minimum Peer Grading Calibrations**.

- **Minimum Peer Grading Calibrations**: This setting applies only to peer grading. You can set the minimum number of responses a student has to "practice grade" before the student can start grading other students' responses. The default value is 3, but you can set this value to any number from 1 to 20. This value must be less than or equal to the value set for **Maximum Peer Grading Calibrations**.

- **Peer Graders per Response**: This setting applies only to peer grading. This setting specifies the number of times a response must be graded before the score and feedback are available to the student who submitted the response.

- **Peer Track Changes**: This setting is new and still under development. This setting applies only to peer grading. When this setting is enabled (set to **True**), peer graders can make inline changes to the responses they're grading. These changes are visible to the student who submitted the response, along with the rubric and comments for the problem.

- **Problem Weight**: This setting specifies the number of points the problem is worth. By default, each problem is worth one point.

- **Required Peer Grading**: This setting specifies the number of responses that each student who submits a response has to grade before the student receives a grade for her response. This value can be the same as the value for the **Peer Graders per Response** setting, but we recommend that you set this value higher than the **Peer Graders per Response** setting to make sure that every student's work is graded. (If no responses remain to be graded, but a student still needs to grade responses, you can
set the **Allow “overgrading” of peer submissions** setting to allow more students to grade previously graded responses.)

**Word Cloud Settings**

<table>
<thead>
<tr>
<th>Display Name</th>
<th>Word cloud</th>
<th>Display name for this module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs</td>
<td>5</td>
<td>Number of text boxes available for students to input words/sentences.</td>
</tr>
<tr>
<td>Maximum Words</td>
<td>250</td>
<td>Maximum number of words to be displayed in generated word cloud.</td>
</tr>
<tr>
<td>Show Percents</td>
<td>True</td>
<td>Statistics are shown for entered words near that word.</td>
</tr>
</tbody>
</table>

Word cloud problems have the following settings.

- **Display Name**: The name that appears in the course ribbon and as a heading above the problem.
- **Inputs**: The number of text boxes into which students can enter words, phrases, or sentences.
- **Maximum Words**: The maximum number of words that the word cloud displays. If students enter 300 different words but the maximum is set to 250, only the 250 most commonly entered words appear in the word cloud.
- **Show Percents**: The number of times that students have entered a given word as a percentage of all words entered appears near that word.
Open Response Assessments

Open Response Assessments Roadmap

In this next lesson, we’ll cover open response assessments (ORAs)—what they are, how they can be used, and how to create, answer, and grade them.

- **The Introduction to Open Response Assessments** unit covers the very basics of ORAs, including why they’re becoming increasingly important in the MOOC world.

- **Components of an Open Response Assessment** provides an introduction to the three components that every ORA has to have. These components are covered in detail in later units.

- **Open Response Assessment Types** introduces learners to the three types of ORAs—self assessment, peer assessment, and artificial intelligence (AI) assessment—and shows an example of each. These assessment types are covered in detail in later units.

- **A Few Notes about Open Response Assessments** contains some guidelines to keep in mind while you’re working with this new technology.
• **Past Use of Open Response Assessments** describes the first ORAs used in edX courses and provides several examples of these assessments.

• **Effective Questions** describes the kind of questions that tend to elicit the best responses from students. It also describes other information that you can include in the body of your question and that can be useful for students.

• **Rubrics** covers the definition of a rubric, how to create rubrics in Studio, and how to design an effective rubric.

• **Self Assessment** describes the self assessment type of ORA. It includes both a discussion with your edX101 classmates about the benefits and drawbacks of self assessments and an opportunity for you to perform your own self assessment.

• **AI Assessment** describes the AI assessment type of ORA and includes a discussion with your edX101 classmates about any new perspectives you have on this technology. You'll also submit a response to a machine-graded question.

• **Peer Assessment** describes the peer assessment type of ORA and includes a discussion with your edX101 classmates about peer assessments. In this unit, you can also perform a peer assessment.

• **Action: Access Scores and Feedback** provides information about receiving scores and feedback for the responses that you've provided.

• In **Action: Try a Combined Assessment**, you'll provide a response to an open response assessment question that includes all three types of assessment--self, AI, and peer.

• **Action: Create an ORA Problem in Your Practice Course** contains step-by-step instructions for adding your own ORAs to your practice course.

• In **Action: Grade Student Responses**, you'll become familiar with the **Staff Grading** page and learn to grade student responses.

• **Best Practices** reviews the information covered in this lesson and lists some hints to make working with ORAs easier.

• **Discussion: Open Response Assessments** contains a discussion space in which you can tell us what you think of this version of the Open Response Assessments lesson.
Introduction to Open Response Assessments

Note Open response assessments are still a very new technology. We’re constantly working to improve them, but you should be aware that these problems are still in beta. We recommend that you test them thoroughly in a practice course and only add them to courses that are not already running. If you plan to use open response assessments, please make sure to work with your edX Program Manager (PM).

Many edX courses include exercises that have relatively straightforward, easy-to-grade answers. For example, there is usually only one correct answer for a fill-in-the-blank chemical equation or multiple choice problem. However, as more of our consortium universities offer more courses, we want to be able to assess student learning through questions that may not have definite answers. A question that asks students to choose between two difficult alternatives and then explain the reasoning behind that choice requires a different kind of assessment—one that is very difficult to implement in a class of tens of thousands of students.

As discussed in the New York Times article in section 1, open response assessments allow tens of thousands of students to receive feedback on written responses of varying lengths as well as files, such as computer code or images, that students upload. The learning sciences team at edX has been working on open response assessment technologies, which include self assessment, peer assessment, and artificial intelligence (AI) assessment (sometimes called "machine assessment" or "machine grading"). These technologies are still under development, but early pilots have shown promising results and we’re continuously making updates to improve our tools. We would love for you to be part of this effort!

In addition to facilitating the grading of many open-ended questions at once, open response assessments allow students to learn critical thinking skills. With self assessments, students learn by comparing their answers to a specific rubric that you create. With peer assessments, students apply their understanding of the rubric to their peers’ answers and provide feedback, receiving feedback in turn. Our student feedback indicates that these methods have led to increased understanding and other benefits.

The next two units provide an overview of what makes up an open response assessment. You’ll read about the components that every open response assessment has, as well as the three open response assessment types.
Components of an Open Response Assessment

An open response assessment has three elements:

- The assessment type or types—self, peer, or artificial intelligence (AI). (AI assessments are sometimes called "computer graded" or "machine graded."). The student sees the type of assessment and the order in which the assessments run in the upper right corner of the ORA problem. In the following example, the student performs a self assessment, then peers perform peer assessments, and then an AI assessment runs.

- The question that you want your students to answer. This appears near the top of the component, followed by a field where the student enters a response. The student can
click **Save** to save a partial response and come back to it later, or **Submit** to submit the response.

- A rubric that you design. After the student enters a response and clicks **Submit**, if the assessment is a self assessment, the student sees the rubric below his answer and compares his answer to the rubric. (If the assessment is an AI or peer assessment, the student sees a "Your response has been submitted" message but doesn't see the rubric.) You'll learn more about rubrics later in this lesson.
By default, when the student submits a response, the question becomes hidden. The student can view the question at any time by clicking the blue **Show Question** text in the upper right corner of the component, below the list of assessment types.
Open Response Assessment Types

You've learned about the three components of an open response assessment: assessment type, question, and rubric. You'll now learn about the different assessment types.

There are three types of assessments for ORAs: self assessment, AI assessment, and peer assessment.

- **Self assessment** allows students to answer a question, and then assess their response according to the rubric you created for the question.

- In **AI assessment**, a computer algorithm learns how to grade according to the rubric from 100 or more instructor-graded responses, and attempts to grade the rest of the student responses in the same way.

- **Peer assessment** allows students to score each other and provide feedback, again using the same rubric.

You can use one or more of these assessments in any problem. For example, you can set a problem to use self assessment only, or you can set it so that the student performs a self assessment, a peer provides a peer assessment, and then a computer performs an AI assessment. Or, you can set a problem to use AI assessment and then peer assessment. You can also set thresholds within the problem for each assessment, so that a response with a low score in one assessment does not move on to the next assessment.

EXAMPLES OF ASSESSMENT TYPES

The following images show an ORA with a student's response. According to assessment type indicator in the upper right corner of the component, this assessment is set to run as a self assessment, then an AI assessment, and finally a peer assessment. You'll see each of these assessments in turn. Click the images for a larger view.

**Question ("Prompt") and Student's Response**
Student's Self Assessment

After the student clicks Submit, the rubric that the instructor created appears below the student's response. In a self assessment, the student reviews the rubric, then selects the options that he feels best describe his response.
AI Assessment

After the student performs his self assessment and clicks **Submit**, his response moves on to AI assessment. The AI algorithm then analyzes the student's response and provides feedback.
Peer Assessment

After the AI assessment is complete, the response moves on to peer assessment. When the required number of students have graded the response (by default, this value is set to 3), and the student has graded enough of his peers’ responses (by default, this value is also set to 3), he can see the score for his response. Peers can provide written feedback as well as point scores.
In this lesson, you'll look at all three assessment types in more detail.
A Few Notes about Open Response Assessments

EdX is excited to be creating and implementing this cutting-edge technology, and our engineers are continually working to make it better. **However, you should be aware that this technology is still in beta.** For a good experience with open response assessments, you'll need to follow a few guidelines.

- Do not create a new open response assessment in a running course. Only create open response assessments in a test course.
- If your course will include open response assessments, add and thoroughly test all the open response assessments *before* the course is live.
- Set open response assessments to be optional, ungraded, or droppable exercises until you’ve used the technology a few times and have become familiar with it.
- Use open response assessments sparingly at first. Only include a few in your course, and make sure that you have contingency plans in case you run into problems.

Finally, if you’re at an edX consortium university and you plan to include open response assessments in a MOOC, make sure to notify your edX project manager (PM).
Past Use of Open Response Assessments

In November 2012, MIT's 3.091x: Introduction to Solid State Chemistry course became the first edX course to use open response assessments. Since then, open response assessments have run in two other university courses, and the EdX Demo course includes four open response assessments. You'll have a chance to respond to all of the ORAs in the EdX Demo course as you work through this lesson.

Both students and course teams have responded positively to open response assessments. Course teams have responded especially well to self assessments, which were simple to implement and required no grading by course staff. For AI assessments, students reported that they felt their scores were accurate even though the AI assessment grader was an early prototype with lower accuracy than our current grader offers. Students have also expressed that seeing their peers' responses in peer assessments has exposed them to viewpoints or information they hadn't before considered.

Students had constructive criticism to offer as well. They indicated that they wanted more feedback from all grader types as well as a way to discuss their scores. Our engineers are working to implement these suggestions.

There are many ways to include open response assessments in your course. Below you'll see examples of open response assessments in three different edX courses.

MIT 3.091x: Introduction to Solid State Chemistry

In the fall of 2012, chemistry course 3.091x piloted self assessment and AI assessment. Among the open response assessments was the following problem.
Harvard PH278x: Human Health and Global Environmental Change

Harvard's PH278x: Human Health and Global Environmental Change course included an AI assessment.
Some have claimed that the carbon dioxide and temperature records presented in class refute the hypothesis that greenhouse gases, and in particular carbon dioxide, drive warming.

Look closely at the time periods in which there is sudden temperature rise, such as those indicated by the arrows. In one sentence or two at most, please describe how the data presented in the graph below runs counter to the hypothesis that present day planetary warming is being driven by increased carbon dioxide concentrations?

![Graph showing CO₂ concentrations in Antarctic ice and Mauna Loa (1958-2013)](image)

Once you have written your response, please click “submit”. An announcement will be made in the Course Info page when all submissions have been graded.
MIT 6.00x: Introduction to Computer Science and Programming

In MIT's 6.00x: Introduction to Computer Science and Programming course, students uploaded an image file for peer assessment. Below you can see a student’s response in the process of being peer graded. You can see an abbreviated version of the question and rubric, as well as the written feedback that the peer grader has provided.
This problem is another chance for you to have a plot reviewed by your classmates. For this problem, please use the file words.txt from Problem Set 3 or 4 produce a plot. What aspect(s) of the data you present, and how you present it, is entirely up to you.

The file words.txt in Problem Set 3 is different than the file words.txt in Problem Set 4. You may wish to choose what data you are using which list you'd prefer to present your work.

For this problem, you'll need to save your graph as a PNG file (the file name does not matter, but a good idea might be "L12_P7_graph.png"). To save your file, first click on the save button:

**Save**

The "save" button

This will pop up a dialog box that will allow you to customize the name of your file as well as the location to which it is saved.

Once you’ve saved your file, upload the image here. You may either enter a URL for the image in the box, or you may simply upload the file directly by clicking the "Choose File" button (recommended).

Full instructions for peer grading are available by clicking here (PDF handout).

**STUDENT RESPONSE**

L12_P7_graph.png

Select the criteria you feel best represents this submission in each category.

**Meaningful Title**

- 0 points: The graph's title does not make sense with respect to the data it presents, or is overly wordy (more than ~10 words).
- 1 points: The graph's title correctly and concisely summarizes the information presented in the graph.

**Presentation of Data**

- 0 points: The graph is confusing or difficult to understand, or does not have anything to do with the data.
- 1 points: The graph chooses the wrong format to display the data with. For example, the axes should be switched or use an altered range, or more, or fewer, pieces of information might be more appropriate.

2 points: The graph provides an interesting take on the given data, and as a whole product stands alone. Looking at this graph with no other information beyond the title and axis labels provides a clear picture of what data is being plotted and why it is important/meaningful.
Effective Questions

You've probably written essay questions for your traditional courses. Open response assessments are comparable to essay questions that ask students to interpret, evaluate, and apply the concepts they've learned. The best questions are focused, asking for specific information in a particular context. For example, rather than "Explain how Per-Erik Moraeus conveys mood in his song 'Koppången,'" an effective question might be "Explain how Per-Erik Moraeus uses suspensions and plagal harmonies in his song 'Koppången' to convey a particular mood."

We recommend that you specify an approximate number of words or sentences that a student's response has to have in the body of your question. Our instructors have required responses of anywhere from 25 to 650 words. The length of the response does not seem to affect grading accuracy.

In addition, your question can provide helpful information about how to use the LMS. For example, if you require students to upload a file as a response, you can provide specific instructions about how to upload and submit their files. You can let students know what to expect after they submit responses. You can also mention the number of times that a student will be able to submit a response for the problem.

The following ORA question used in MIT's 3.091x sets clear expectations about the length of each student's response and specifies that students can only try to respond one time.
The following example question, which comes from a practice course, tells students how to submit an answer as well as what will happen after the student clicks Submit.
Finally, the following question from MIT’s 6.00x includes instructions for students who will upload files in response to the problem.
Action: Write an Open Response Assessment Question

In a word processing program such as Word, write an open response assessment question. Make sure to include information such as the length of the expected response and guidance for submitting the response. You'll learn how to enter this question into Studio in Action: Create an ORA Problem in Your Practice Course.
Rubrics

A rubric for an ORA in an online course is similar to a rubric for traditional course questions. The same rubric is used for all three ORA types—self assessments, peer assessments, and AI assessments—and it can include anything that you want it to include.

In Studio, rubrics are arranged by categories. Each category then has two or more options, and each option has a score. Options are listed in ascending order starting at 0 points. The grader selects one value for each category.

For example, the categories in the following rubric are "Content" and "Organization." The student has given himself one out of three points for content and two out of two points for organization.

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>0 points: Includes little information with few or no details or unrelated details. Unsuccessful in attempts to explore any facets of the topic.</td>
</tr>
<tr>
<td></td>
<td>1 points: Includes little information and few or no details. Explores only one or two facets of the topic.</td>
</tr>
<tr>
<td></td>
<td>2 points: Includes sufficient information and supporting details. Details may not be fully developed. Ideas may be listed. Explores some facets of the topic.</td>
</tr>
<tr>
<td></td>
<td>3 points: Includes in-depth information and exceptional supporting details that are fully developed. Explores all facets of the topic.</td>
</tr>
<tr>
<td>Organization</td>
<td>0 points: Ideas organized logically, transitions weak, and response difficult to follow.</td>
</tr>
<tr>
<td></td>
<td>1 points: Attempts to logically organize ideas. Attempts to progress in an order that enhances meaning and demonstrates use of transitions.</td>
</tr>
<tr>
<td></td>
<td>2 points: Ideas organized logically. Progresses in an order that enhances meaning, includes smooth transitions.</td>
</tr>
</tbody>
</table>

A category can be a topic with various options, as in the above example. It can also contain a longer specific description, and the options can be as simple as "Topic not covered" and "Topic covered." The following example uses this pattern. Click the image for a larger view.

Rubric Design

A good rubric makes assessing a response easier for all graders—self, peer, and AI. The following are several best practices.

- Use specific, concrete language.
- Include detailed examples. For example, instead of specifying that the answer must include references to the Hundred Years’ War, specify that responses must include information about the battles of Crécy, Baugé, and Castillon.
- Use parallel language for all options. For example, begin each option with an action verb.

- Anticipate your students' level. For example, because many of your students are not native English speakers, you may not want to grade them on nuances in American English speech.

**Scoring**

When you create your rubric, keep track of how many points are possible for the problem. If a category has three options, those options are worth 0, 1, and 2 points, respectively. Thus the category has a maximum score of 2 points. Likewise, if a category has four options, those options are worth 0, 1, 2, and 3 points, respectively. Thus this category has a maximum score of 3 points.

If your rubric has two categories, and each category has a maximum score of 2 points, 4 total points are possible for the problem.

If your rubric has four categories, and two categories each has a maximum score of 2 and the other two categories each has a maximum score of 3, 10 total points are possible for that problem.

**Action: Write an Open Response Assessment Rubric**

Write a rubric for the question that you wrote in the last unit. Make sure that you include categories and options, and that you list the options in ascending point order starting at 0 points. You'll learn how to enter this rubric into Studio in Action: Create an ORA Problem in Your Practice Course.
Self Assessment

In a self assessment, the student submits an answer to the ORA question, and then sees the rubric that you created for the question. The student then evaluates her response according to your rubric.

Self assessments provide immediate feedback that helps students to think critically about their performance, as they see the "correct" answer soon after submitting a response. They are also the easiest to implement—you only have to write a question and a rubric—and require no grading effort by course staff. As students may not be the best judge of their own answers, it's important to provide as specific a rubric as you can.

Discussion: Self Assessments

In the discussion space below, share your thoughts on self assessments with your peers. What are some benefits and drawbacks in addition to the ones discussed in this unit?

(Discussion space provided)

Action: Perform a Self Assessment

In this action, as well as in the actions for the AI Assessment and Peer Assessment units, you'll use open response assessments in the EdX Demo course. The EdX Demo course contains a Self-Assessed Essay unit. Go to this unit, submit a response to the open response assessment question, and then perform a self assessment. (You can submit your own response, or you can use one of the sample responses in the Sample Answers unit.) Step-by-step instructions are provided below.

Note If you have technical problems when you use open response assessments, look through the discussions on the edX101 Help forum to see if other users have seen similar issues. If no discussion already exists, you can create a new discussion. (For Category, select edX101.)

Instructions: Perform a Self Assessment

1. In the EdX Demo course, go to the Self-Assessed Essay unit, and then type your response in the Response field under the question. (You can submit your own response, or you can use one of the sample responses in the Sample Answers unit.)
2. Submit your response. To do this, click **Submit**, and then click **OK** in the dialog box to continue.

*Note If you want to save your response and work on it again later, click **Save**. A short message appears directly under the **Save** and **Submit** buttons.*

3. After you submit your response, the rubric for the question appears below your response. Compare your response with the rubric and select the option that you think is appropriate for each category.
4. **Click Submit assessment.**

Your response appears, and you can see the scores that you gave yourself.
AI Assessment

In AI assessment, an instructor grades a sample set of student responses to the ORA. A machine learning algorithm then creates a model based on the instructor’s scores and grades the remaining students’ responses.

AI assessment requires a fairly substantial time commitment from course staff in the beginning, as instructors must grade 100 student responses so that the algorithm can create its model. However, the algorithm can then grade tens of thousands of student responses.

The algorithm attempts to determine instructors’ grading standards based on the supplied rubric as well as criteria including word order, types of phrases, sentence structure, and grammar. After the algorithm creates a model and begins scoring responses, it lists student responses in order of a percentage called a "confidence value"--in other words, the AI algorithm may give the student 10 points out of 10, but list a confidence value of only 50%. After the AI assessment has begun, the instructor sees the responses that AI has graded in order of lowest confidence to highest on the Staff Grading page. You can then regrade the low-confidence responses to refine the grading and improve the AI model.

Many concerns have been raised about the accuracy of AI assessments, and AI has not yet been fully tested on live courses or on a humanities course. However, our learning sciences team has said of AI, "We typically find that in order to fool it, one needs to know enough to be able to write a good answer in the first place." In addition, while AI assessment cannot give human-quality feedback, over 80% of the students in our November 2012 chemistry course pilot reported that their AI assessment scores were accurate. These students also provided us with feedback that we've incorporated into later versions of AI assessments, further improving our tools and processes. Every student is welcome to provide feedback on AI assessments--the more the better! Additionally, future versions of the tools plan to integrate better with the rubric so that students can clearly see the reasoning behind their scores.

Discussion: AI Assessments

Consider your thoughts about AI assessments before and after reading this unit. Does what you've learned change your opinion? In the discussion space below, discuss your thoughts with your peers.

(Discussion space provided)

Action: Submit a Response for an AI Assessment

In the EdX Demo course, submit a response to the open response assessment question in theFeedback and AI-"Graded" Essays unit. Step-by-step instructions are provided below.

Note If you have technical problems when you use open response assessments, look through the discussions on the edX101 Help forum to see if other users have seen similar issues. If no discussion already exists, you can create a new discussion. (For Category, select edX101.)
Instructions: Submit a Response for an AI Assessment

1. In the EdX Demo course, go to the Feedback and AI-"Graded" Essays unit.

2. Submit a response to the question. You can submit your own response, or you can use one of the sample responses in the Sample Answers unit.

After you submit your response for grading, a message appears under your response.

You’ll learn how to access your grade in Action: Access Scores and Feedback.
Peer Assessment

In a peer assessment, the student composes an answer to the ORA question, and then submits the answer to be graded by several of the student's peers. Students provide feedback along with a scored rubric. An algorithm attempts to match students whose activity levels in the course are similar.

A number of students must grade each response before the response returns to the student. In turn, each student must grade a number of peers' responses before he receives feedback. By default, the minimum for both these numbers is 3 and the maximum is 6. You can change these values to any positive number, including 1. After students grade the minimum number of responses, they can grade as many additional responses as they want.

Before students grade each others' responses, they must learn to grade the same way that an instructor would. In this process, called calibration, an instructor grades a minimum of 20 responses. (In future versions of Studio, instructors will be able to change this minimum number.) Students see the instructor-graded essays, without the instructor's feedback, and then grade several of these same essays. If a student's grading is similar to the instructor's, the student begins grading other students' responses to the question. If the student does not score well, the student continues practicing until his grading matches the instructor's. Calibration allows students to think about the problem and why the instructor has given a particular score to a response before they begin grading responses themselves.

Peer assessments offer significant value for both the graders and the students being graded. Calibration essays can serve as learning tools, and students receive feedback from other students rather than a machine. When students are engaged, they can give each other useful feedback.

If students run into problems, they have options. They can skip an essay if they do not know how to grade it. The response is then sent to another student for review. They can also flag responses as inappropriate if necessary, and the course staff will then review the response.

The Peer Grading Interface

The area where students grade each others' responses is the peer grading interface. Each course that has peer assessments has at least one peer grading interface. There may be just one peer grading interface for the whole course, or each individual problem may have its own separate peer grading interface.

Accessing a Peer Grading Interface

There are several ways to access a peer grading interface, depending on the way that the course is set up. (You'll learn how to create peer grading interfaces in your course in Action: Create an ORA Problem in Your Practice Course.)

- Through the Open Ended Console page. This option is always available for every course.
• Through the courseware, in a specific unit. Note that this option can make peer grading interfaces somewhat difficult to find.

• Through the courseware, in a separate section. We encourage course teams to use this option.

**Through the Open Ended Console Page**

You and your students can always access the peer grading interface through the Peer Grading selection on the Open Ended Console page. To access the Open Ended Console page, click the Open Ended Panel tab at the top of any page in the courseware.

When you click Peer Grading on the Open Ended Console page, you can see all the peer assessments in the course, as long as at least one answer has been submitted for that peer assessment.

To access peer grading for a particular problem, you click the name of that problem.

**In a Specific Unit in the Courseware**

Some course creators link a peer grading interface with a specific open response assessment problem and add that peer grading interface inside the courseware. In this case, only the problem that is linked to the peer grading interface appears in that interface. Note, however, that we don’t typically encourage instructors to use this method. Students can always access peer grading through the Open Ended Console page, but they may have difficulty finding the specific problems in the body of the course.

Instructors can create a peer grading interface for each problem in the course and include these interfaces in individual units. The peer grading interface most often appears below the peer assessment, though it can appear anywhere in the course.

For example, the peer grading interface in the Peer-Graded Essay unit of the EdX Demo course appears below the problem in that unit. To access peer grading for the problem, you click the problem name.
Note If you haven’t submitted a response for this problem yet or no responses are available for grading, you see a yellow message in the interface.
Through a Separate Section in the Courseware

Many instructors create a separate section for peer grading interfaces that is always visible in the course accordion. This makes the peer grading interface or interfaces easy to find, no matter where students are in the courseware.

If an instructor creates a separate section for the peer grading interface or interfaces, the instructor can include just one peer grading interface for the entire course, or he can include separate peer grading interfaces for each individual problem. If his course uses multiple peer grading interfaces, he can put all the peer grading interfaces in this section.

For example, in MIT’s 6.00x: Introduction to Computer Science and Programming course, each peer assessment problem has its own peer grading interface. The course has a separate section that holds all these peer grading interfaces, so students can easily access peer grading.
Discussion: Peer Assessments

In the discussion space below, share your thoughts about peer assessments with your peers. Do you think that peer assessments provide an effective way for students to learn? Why or why not?

(Discussion space provided)

Action: Perform a Peer Assessment

In the EdX Demo course, perform a peer assessment in the Peer-Graded Essay unit. Step-by-step instructions are provided below.

Note If you have technical problems when you use open response assessments, look through the discussions on the edX101 Help forum to see if other users have seen similar issues. If no discussion already exists, you can create a new discussion. (For Category, select edX101.)

Instructions: Perform a Peer Assessment

Performing a peer assessment has several steps. Each of these steps is covered in detail below.

1. Submit a response to an open response assessment question.
2. Access responses, either in the body of the course or from the Open Ended Console page.
3. Learn to grade (this process is called calibration).
4. Grade responses from other students.

For this exercise, you'll use the peer assessment problem in the edX Demo course.
1. **Submit a Response**

   1. In the EdX Demo course, go to the Peer-Graded Essay unit.
   
   2. Submit a response to the question. You can submit your own response, or you can use one of the sample responses in the Sample Answers unit.

After you submit your response for grading, a message appears under your response.

```
Open Response | Assessment: Peer

Censorship in the Libraries

"All of us can think of a book that we hope none of our children or any other children have taken off the shelf but I have the right to remove that book from the shelf – that is, the author – then you also have exactly the same right and so does everyone else. And then we have no books left on the shelf for any of us." —Katherine Paterson, Author

Write a persuasive essay to a newspaper reflecting your views on censorship in libraries. Do you believe that certain materials, such as books, music, movies, magazines, etc., should be removed from the shelves if they are found offensive? Support your position with convincing arguments from your own experience, observations, and/or reading.

No, I do not think that books, magazines, movies and music should be removed. However, I do think that the books, magazines, movies, and music should be monitored by age. The reason why it is because younger children should not be reading, listening or watching material that is for adults or teenagers. They have little control over the materials that can be monitored by the child's library card. The card could have their name, age, grade, and date of birth. If there is a certain book, magazine, movie, and music that the child wants a parent to be there with them. The parent needs to show a form of identification so that the librarian knows that the parent is of age. Teenagers should also be monitored. Event though they are almost adults. Some parents not agree with some of the material. So, if their parents do not want them reading certain things their should be stickers on their cards. Even though there are plenty of opinions about what a person's child is reading, watching, or listening to I think there should be certain ways to watch what children are reading, watching or listening to.

Your response has been submitted. Please check back later for your grade.
```

You'll learn how to access your grade in Action: Access Scores and Feedback.

2. **Access Responses from Other Students**

You can access responses to a specific problem inside the courseware, in the peer grading interface for that problem. Or, you can access responses to all problems in the course from the Open Ended Console page.

**Note** When you perform a peer assessment in another course, be aware that you can only grade a response if you've submitted a response to the question, an instructor has already graded at least 20 responses, and there are more essays from other students left to grade. If you click Peer Grading on the Open Ended Console page but any of these conditions aren't met, you see the following message. (Because you're performing a peer assessment in the EdX Demo course, you don't have to worry about these conditions for now--they're all already met.)
Access Responses in the Courseware

- In the Peer-Graded Essay unit in the EdX Demo course, scroll down past the response that you submitted until you see the peer grading interface that appears below the problem. You can start grading other students’ responses from this interface.

Note Grading problems inside the courseware is only possible if the instructor has added a peer grading interface below the specific problem. If no peer grading interface appears below the problem, you can still access peer grading through the Open Ended Console page.

Access Responses from the Open Ended Console Page

1. In the EdX Demo course, click Open Ended Panel at the top of the page.
2. On the Open Ended Console page, click Peer Grading.

The peer grading interface for your course opens, and the problems that have responses available to grade appear in a list. Note that all the problems in the course will appear in this list (as soon as at least one response is available to grade), even if individual problems have specific peer grading interfaces in the body of the course.

3. Click the name of the problem that you want to grade. The problem opens.
3. Learn to Grade

Before you can grade a problem, you have to grade a certain number of problems and compare your scoring with the instructor’s score for those problems. When your score is close to the instructor’s score, you can start grading other students’ responses.

1. Click the name of the problem. When the Learning to grade page opens, click Start learning to grade.

2. When the problem opens, compare the student’s response with the rubric. Select the options that best apply to the response, and then click Submit.

3. Review the How did I do? message that you receive, and then click Continue.
When you click **Continue**, the next student response appears for you to grade, and you see a **Calibration essay saved** message in the top left corner of the page.

4. Continue to grade responses. After you grade the required number of responses correctly, you receive a **Ready to grade!** message. You can then start to grade responses for other students.

4. **Grade a Response**

When you grade a peer assessment response, you can not only select options in the rubric, as you did for your self assessment, but also provide additional feedback for the student who submitted the response.

1. When the response opens, select the options in the rubric that you feel best fit.

2. Under **Written Feedback**, write a comment about the score that you gave the response.
   - If you have concerns about the response, you can select other options. You don't have to fill out the rubric before you select these options.

3. You can flag the response for instructor review if it's offensive or if you suspect that it contains plagiarized material.
4. You can skip the response if you aren't sure how it should be graded. Note that if you skip a response, it doesn’t count toward the number of responses you’ve graded. Thus, if you have to grade three responses before you receive the score for your response, and you skip two responses, you’ll see a total of five responses before you receive your score.

5. Click Submit. You see a Successfully saved your feedback message at the top of the screen, and the next response opens.
6. Continue to grade until you've graded the required number of responses (usually 3). When you've graded enough responses, you receive the following message.

Successfully saved your feedback. Fetching the next essay. You have done the required number of peer assessments but may continue grading if you like.

- When you see this message, you can access the score for your own response. For more information, see Action: Access Scores and Feedback.
- If you want to grade additional responses at any time, you can go back to the Peer Grading page and click the name of the problem.
Accessing Scores and Feedback

After your submit a response for a peer or AI assessment, you'll have to wait to receive a response. The length of the wait depends on how quickly other students submit responses, the instructor grades the minimum number of responses, and (for peer grading) other students grade your response. In most cases, you won't have to wait more than a few days. In some cases, you can receive your scores in just minutes.

Action: Access Your Scores and Feedback

In the EdX Demo course, check the Open Ended Console page for the scores to the peer and AI assessments you've completed. Step-by-step instructions are provided below.

**Note** If you have technical problems when you use open response assessments, look through the discussions on the edX101 Help forum to see if other users have seen similar issues. If no discussion already exists, you can create a new discussion. (For Category, select edX101.)

INSTRUCTIONS: ACCESS SCORES AND FEEDBACK

You'll access your scores for your responses to the AI and peer assessment problems through the Open Ended Console page.

1. In the EdX Demo course, click the Open Ended Panel tab at the top of the page.

2. On the Open Ended Console page, click Problems You Have Submitted.

3. On the Open Ended Problems page, check the Status column to see whether your responses have been graded.
4. When grading for a problem has been finished, click the name of a problem in the list to see your score for that problem. When you click the name of the problem, the problem opens in the courseware.

For both AI and peer assessments, the score appears below your response in an abbreviated version of the rubric. For peer assessments, you can also see the written feedback that your response received from different graders.

**Graded AI Assessment**

![Graded AI Assessment](image)

**Graded Peer Assessment**

![Graded Peer Assessment](image)
If you want to see the full rubric for either an AI or peer assessment, click **Toggle Full Rubric**.

**Note** For a peer assessment, if you haven’t yet graded enough problems to see your score, you receive a message that lets you know how many problems you still need to grade.
For more information about grading peer assessments, see Peer Assessments.

Resubmitting a Response

Some open response assessments allow multiple attempts. For these problems, a New Submission button appears below your original response.

If you want to answer the question again, click New Submission to clear your former response, and click OK in the dialog box that appears.

You can then enter a new response for the problem.
Action: Try a Combined Assessment

In this unit, you'll respond to an open response assessment problem that uses all three types of assessments. Follow the instructions below to access and respond to a problem in the EdX Demo course.

**Note** If you have technical problems when you use open response assessments, look through the discussions on the edX101 Help forum to see if other users have seen similar issues. If no discussion already exists, you can create a new discussion. (For Category, select edX101.)

**Instructions: Try a Combined Assessment**

1. In the EdX Demo course, go to the Combined Open-Ended Assessment unit.
2. Submit a response to the "Combined Open Ended Response" problem.
3. Perform a self assessment on your response, and then click Next Step to move your response to the AI assessment.
4. Wait to receive the AI assessment score for your response. (Note that you can't start peer grading until you receive this score.) You can access this score by going to the unit that contains the problem in the courseware, or through the Problems I Have Submitted option on the Open Ended Console page. For more information, see Action: Access Scores and Feedback.
5. After you receive your AI assessment score, click Next Step to move to the peer assessment.
6. When you can start peer grading, grade several other students' responses.
7. Wait to receive your peer assessment score. You can check for your score by going to the unit that contains the problem in the courseware, or through the Problems I Have Submitted option on the Open Ended Console page. For more information, see Action: Access Scores and Feedback.
Action: Create an ORA Problem in Your Practice Course

Now that you've become more familiar with open response assessments, you'll create a new open response assessment in your practice course. The open response assessment you'll create is the "Combined Open-Ended Assessment" problem from the EdX Demo course. Step-by-step instructions are provided below.

Note If you have technical problems when you use open response assessments, look through the discussions on the edX101 Help forum to see if other users have seen similar issues. If no discussion already exists, you can create a new discussion. (For Category, select edX101.)

Instructions: Create an Open Response Assessment Problem

Creating an open response assessment is a multi-step process. Each of these steps is described in detail below.

1. Add the Advanced component to your course.
2. Create the component for the open response assessment.
3. Add the question.
4. Add the rubric.
5. Set the assessment type and scoring.
6. Set the problem name.
7. Set other options.
8. Save the problem.
9. Add the peer grading interface (for peer assessments only).
10. Test the problem.

1. Add the Advanced Component to Your Course

By default, when you create a new component in Studio, you see the following options.
To create an ORA, you must first add the Advanced component to your course. To do this, follow these steps.

1. On the **Settings** menu, click **Advanced Settings**.

2. On the **Advanced Settings** page, locate the **Manual Policy Definition** section, and then locate the `advanced_modules` policy key (this key is at the top of the list).

3. Under **Policy Value**, place your cursor between the brackets, and then type the following exactly as shown. Make sure to include the quotation marks, but do not include any spaces.

   "peergrading","combinedopenended"

4. At the bottom of the page, click **Save Changes**.

   The page refreshes automatically. At the top of the page, you see the following notification.

   The text in the **Policy Value** field now appears as follows.

5. Return to the unit where you want to add the ORA. The list of possible components now contains an Advanced option.
2. Create the Component

1. In Studio, open the unit where you want to create the ORA.
2. Under Add New Component, click Advanced, and then click Open Response Assessment.
3. In the problem component that appears, click Edit, and then click OK in the dialog box that appears.
4. The component editor opens. The component editor contains a sample question ("prompt"), rubric, and assessment type specification. You'll replace this sample content with the content for your problem.

3. Add the Question

Note In this example, we’ll use the default prompt text. You don’t have to make any changes to it.

1. In the component editor, locate the [prompt] tags.
If you were creating an open response assessment with your own question, you would replace the sample text between the [prompt] tags with the text of your question. When you replace the sample text, make sure you follow these guidelines to avoid common formatting mistakes.

- Leave the [prompt] tags in place.
- Enclose all text in HTML tags.

For this exercise, you won’t make any changes to the sample text. You can go on to the next step.

4. Add the Rubric

1. In the component editor, locate the [rubric] tags. (The sample rubric is long, so you’ll have to scroll down to locate the second tag.)
2. Replace the sample rubric with the text below. Make sure to do the following.

- Include the beginning and ending [rubric] tags.
- Precede the categories with a plus (+) sign.
- Precede the options with a minus (-) sign.
- List the option that scores zero points first, followed by the option that scores one point, and so on.

[rubric]

+ Writing Applications
- The essay loses focus, has little information or supporting details, and the organization makes it difficult to follow.
- The essay presents a mostly unified theme, includes sufficient information to convey the theme, and is generally organized well.

+ Language Conventions
- The essay demonstrates a reasonable command of proper spelling and grammar.
- The essay demonstrates superior command of proper spelling and grammar.

[rubric]

5. Set the Assessment Type and Scoring

To set the assessment type and scoring for your open response assessment, you’ll enter code that specifies the type and order of assessments to use along with the scoring thresholds for each assessment. The code uses the following format.

[tasks]
(Type 1), ({min-max}Type 2), ({min-max}Type 3)
[tasks]

- The [tasks] tags surround the code.
- Type 1, Type 2, and Type 3 are the names of the types of assessments. Assessments run in the order in which they’re listed.
- min is the point value the response must receive in the previous assessment to move to this assessment. Note that you do not define a scoring threshold for the first assessment, because there is no required previous assessment.
- max is the maximum point value for the assessment. The maximum score is the second number in the pair of numbers for each assessment after the first assessment.

For example, a problem might contain the following code.

[tasks]
(Self), {{5-7}Peer), {{4-7}AI)
[tasks]

The problem that includes this code has the following characteristics.
- The problem has a self assessment, a peer assessment, and then an AI assessment.
- The maximum score for the problem is 7.
- To advance to the peer assessment, the response must have a self assessment score of 5 or greater.
- To advance to the AI assessment, the response must have a peer assessment score of 4 or greater.

**Set the Type and Scoring**

In the problem you're creating, the student must give herself 2 out of 2 points in the self assessment before the peer assessment can start. Her response must then score at least 1 out of 2 points before it can move on to the AI assessment.

1. In the component editor, locate the `[tasks]` tags.

![Image of component editor]

2. Replace the sample code with the following code.

```plaintext
[ tasks ]
(Self), (4-12)AI, (9-12)Peer
[ tasks ]
```
6. Set the Problem Name

Next, you'll change the name of the problem. The name of the problem appears as a heading above the problem in the courseware.

The name of the problem also appears in the list of problems on the Staff Grading page. You'll learn more about this page later in this lesson.

To change the name:

1. In the upper-right corner of the component editor, click Settings.
2. In the Display Name field, replace Open Response Assessment with Combined Open Ended Response.
7. Set Other Options

If you want to change the problem settings, which include the number of responses a student has to peer grade and whether students can upload files as part of their response, you can specify these options on the **Settings** tab.

Open response assessments include the following settings.

- **Allow “overgrading” of peer submissions**: This setting applies only to peer grading. If all of the responses for a question have been graded, the instructor can allow additional students to grade responses that were previously graded. This can be helpful if an instructor feels that peer grading has helped students learn, or if some students haven’t...
graded the required number of responses yet, but all available responses have been graded.

- **Allow File Uploads**: This setting specifies whether a student can upload a file, such as an image file or a code file, as a response. Files can be of any type.

- **Disable Quality Filter**: This setting applies to peer grading and AI grading. When the quality filter is disabled (when this value is set to True), Studio allows submissions that are of "poor quality" (such as responses that are very short or that have many spelling or grammatical errors) to be peer graded. For example, you may disable the quality filter if you want students to include URLs to external content—otherwise Studio sees a URL, which may contain a long string of seemingly random characters, as a misspelled word. When the quality filter is enabled (when this value is set to False), Studio does not allow poor-quality submissions to be peer graded.

- **Display Name**: This name appears in two places in the LMS: in the course ribbon at the top of the page and above the exercise.

- **Graded**: This setting specifies whether the problem counts toward a student's grade. By default, if a subsection is set as a graded assignment, each problem in that subsection is graded. However, if a subsection is set as a graded assignment, and you want this problem to be a "test" problem that doesn't count toward a student's grade, you can change this setting to False.

- **Maximum Attempts**: This setting specifies the number of times the student can try to answer the problem. Note that each time a student answers a problem, the student’s response is graded separately. If a student submits two responses to a peer-assessed problem (for example, by using the New Submission button after her first response receives a bad grade or because she wants to change her original response), and the problem requires three peer graders, three separate peer graders will have to grade each of the student's two responses. We thus recommend keeping the maximum number of attempts for each question low.

- **Maximum Peer Grading Calibrations**: This setting applies only to peer grading. You can set the maximum number of responses a student has to "practice grade" before the student can start grading other students' responses. The default value is 6, but you can set this value to any number from 1 to 20. This value must be greater than or equal to the value set for Minimum Peer Grading Calibrations.

- **Minimum Peer Grading Calibrations**: This setting applies only to peer grading. You can set the minimum number of responses a student has to "practice grade" before the student can start grading other students' responses. The default value is 3, but you can set this value to any number from 1 to 20. This value must be less than or equal to the value set for Maximum Peer Grading Calibrations.

- **Peer Graders per Response**: This setting applies only to peer grading. This setting specifies the number of times a response must be graded before the score and feedback are available to the student who submitted the response.

- **Peer Track Changes**: This setting is new and still under development. This setting applies only to peer grading. When this setting is enabled (set to True), peer graders can make inline changes to the responses they’re grading. These changes are visible to the student who submitted the response, along with the rubric and comments for the problem.
• **Problem Weight**: This setting specifies the number of points the problem is worth. By default, each problem is worth one point.

• **Required Peer Grading**: This setting specifies the number of responses that each student who submits a response has to grade before the student receives a grade for her response. This value can be the same as the value for the **Peer Graders per Response** setting, but we recommend that you set this value higher than the **Peer Graders per Response** setting to make sure that every student’s work is graded. (If no responses remain to be graded, but a student still needs to grade responses, you can set the **Allow “overgrading” of peer submissions** setting to allow more students to grade previously graded responses.)

8. **Save the Problem**

• After you have created the prompt and the rubric, set the assessment type and scoring, changed the name of the problem, and specified any additional settings, click **Save**.

The component appears in Studio. In the upper right corner, you can see the type of assessments that you have set for this problem.
9. Add the Peer Grading Interface (for peer assessments only)

You can add just one peer grading interface for the whole course, or you can add a separate peer grading interface for each individual problem. For more information, see "The Peer Grading Interface" in Peer Assessments.
Add a Single Peer Grading Interface for the Course

When you add just one peer grading interface for the entire course, we recommend that you create that peer grading interface in its own section so that students can find it easily. Students will be able to access all the ORA problems for the course through this peer grading interface.

1. Create a new section, subsection, and unit. You can use any names that you want. One course used "Peer Grading Interface" for all three.

2. Under Add New Component in the new unit, click Advanced, and then click Peer Grading Interface.

   A new Peer Grading Interface component appears.

3. To see the peer grading interface in the course, set the visibility of the unit to Public, and then click View Live.

   The following page opens.

   When students submit responses for peer assessments in your course, the names of the problems appear in this unit.
Add the Peer Grading Interface to an Individual Problem

When you add a peer grading interface for an individual problem, you need to make sure to add the identifier for the problem to that peer grading interface. If you don't add the identifier, the interface will show all of the peer assessments in the course.

Note that the peer grading interface doesn't have to appear under the problem you want it to be associated with. As long as you've added the identifier of the problem, the peer grading interface will be associated with the problem, even if you include the peer grading interface in a later unit (for example, if you want the problem to be due after a week).

1. Open the unit that contains the ORA.
2. If the visibility of the unit is set to Public, click View Live. If the visibility is set to Private, click Preview. The unit opens in the LMS in a new tab. You'll need to make sure you're in Staff view rather than Student view.
3. Scroll down to the bottom of the ORA, and then click Staff Debug Info.

4. In the image that opens, locate the string of alphanumeric characters to the right of the word location. Press CTRL+C to copy this string, starting with i4x.

5. Switch back to the unit in Studio. If the visibility of the unit is set to Public, change the visibility to Private.
6. Scroll to the bottom of the unit, click Advanced under Add New Component, and then click Peer Grading Interface.
7. On the Peer Grading Interface component that opens, click Edit.
8. In the Peer Grading Interface component editor, click **Settings**.

9. In the **Link to Problem Location** field, paste the string of alphanumeric characters that you copied in step 4. Then, change the **Show Single Problem** setting to **True**.

10. Click **Save** to close the component editor.
10. Test the Problem

Now that you've added your problem to Studio, you're ready to try it out. Go ahead and type a response to your problem.

1. In Studio, open the unit that contains your ORA problem.

2. Under Unit Settings, change the Visibility setting to Public, and then click View Live.

When you click View Live, the unit opens in the LMS in a new tab.

3. In the LMS, locate your ORA question, and then type your response in the Response field under the question.

Note that when you view your ORA problem in the LMS as an instructor, you see the following message below the problem. This message never appears to students.
4. Test the problem to make sure that it works as expected.

To test your open response assessment, you may want to sign into your course as a student, using an account that's different from the account that you use as an instructor.

- If you want to keep your course open as an instructor when you sign in as a student, either open a window in Incognito Mode in Firefox or Chrome or use a different browser to access your course. For example, if you used Firefox to create the course, use Chrome when you sign in as a student.

- If you don't need to keep your course open, sign out of your course, and then sign back in using a different account. Note that if you do this, you can't make changes to your course without signing out and signing back in as an instructor.
**Action: Grade Student Responses**

Now that you've created an open response assessment problem, you'll learn to grade student responses. You'll grade responses through the **Staff Grading** page. Step-by-step instructions are provided below.

*Note* If you have technical problems when you use open response assessments, look through the discussions on the edX101 Help forum to see if other users have seen similar issues. If no discussion already exists, you can create a new discussion. (For Category, select edX101.)

**The Staff Grading Page**

You'll grade student responses to both AI assessments and peer assessments from the **Staff Grading** page in the LMS. Take a moment to familiarize yourself with the features of this page.

When a response is available for you to grade, a yellow exclamation mark appears next to **Open Ended Panel** at the top of the screen.

To access the **Staff Grading** page, click **Open Ended Panel**.

When the **Open Ended Console** page opens, click **Staff Grading**. Notice the **New submissions to grade** notification.
When the **Staff Grading** page opens, information about your open response assessment appears in several columns.

<table>
<thead>
<tr>
<th>Problem Name</th>
<th>Graded</th>
<th>Available to Grade</th>
<th>Required</th>
<th>Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combined Open Ended Response</td>
<td>51</td>
<td>994</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>Peer-Graded Essay</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Humanities Question -- Machine Assessed</td>
<td>110</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

- **Problem Name**: The name of the problem. Click the name of the problem to open it.
  
  Note that even if a problem appears in your course, it does not appear under **Problem Name** on the **Staff Grading** page until at least one response to the problem has been submitted and is available to grade.

- **Graded**: The number of responses for that problem that you have already graded.

- **Available to grade**: The total number of ungraded student submissions.
  
  Note that even if the AI algorithm has graded all available responses, you can still grade the responses that the algorithm designates as low-confidence responses by clicking the problem name in the list.

- **Required**: The number of responses remaining to be graded to train the algorithm for AI or to calibrate the responses for peer grading. Note that if your open response assessment calls for both AI and peer assessment, the 20 responses that you grade for the peer assessment count toward the 100 responses for the AI assessment.

  In the example above, the instructor has graded 51 responses for the “Combined Open Ended Response” question, but must grade 49 more to train the AI algorithm.

- **Progress**: This column provides a visual indication of your progress through the grading process.

**Grading Responses**

You're already familiar with performing peer assessments as a student, which you did in the Peer Assessment unit. Grading student responses as an instructor is very similar.

1. Under **Problem Name**, click the **Combined Open Ended Response** problem.
When the problem opens, the information about the number of responses that are still available to grade, that have been graded, and that an instructor is required to grade appears under the problem name. You can also find out about the AI algorithm's error rate. The error rate is a calculation of the difference between the scores that AI algorithm provides and the scores that the instructor provides.

2. In the rubric below the response, grade the response the same way you graded other assessments in this lesson, selecting the option that best describes the response.

3. If applicable, add additional feedback.
   - You can provide comments for the student in the Written Feedback field.
   - If you do not feel that you can grade the response (for example, you’re a member of course staff but you would rather have the instructor grade the response), you can click Skip to skip it.
If the response contains inappropriate content, you can select the Flag as inappropriate content for later review check box. Flagged content is accessed on the Staff Grading page. If necessary, course staff can ban a student from peer grading.

4. When you are done grading the response, click Submit.

When your course is running, another response opens automatically after you grade the first response, and a message appears at the top of the page.

After you’ve graded all responses for this problem, No more submissions to grade appears on the page.
Click Back to problem list to return to the list of problems. You can also wait for a few minutes and click Re-check for submissions to see if any other students have submitted responses.
Best Practices

Open response assessments are complex, and creating, grading, and introducing them to your students can present challenges. Below are some reminders and best practices that we hope you'll find useful.

Remember, you can use any or all types of ORAs in your course. If you are using AI assessments for the first time, we encourage you to also include self assessments, peer assessments, or both so that you can compare the accuracy of the AI assessment scoring with your students' own assessments.

If you want to test AI assessments before you run them in your live course, we suggest the following testing plan (created by Donna Kidwell of the University of Texas at Austin):

1. Create a practice course that contains an AI assessment.
2. Invite a number of graduate students or other students to respond to the assessment.
3. Grade 100 responses so that the AI assessment can create a model.
4. Invite new students to respond to the assessment, and test the AI assessment model.

We also suggest you use the following best practices when you use an ORA.

- Make sure that the question is clear.
- Design a specific rubric.
- Consider whether you want the problem to count toward the student's grade, or if you'd rather use the problem as an in-course learning exercise.
- Think about which grader type (self, peer, machine learning) is the best for your question and why.
- Determine what the students will learn from the problem and evaluation as they are currently structured.
- Explain how evaluation will work to students, and show them samples of good, adequate, and bad answers to similar questions, with explanations.
- Introduce the idea and the purpose of self and peer assessment to your students in a lesson, e-mail, or video.
- Make sure to continue communicating with students during the times when the course includes an open response assessment problem.
- Review a sample of the responses and provide feedback to the class in the next week's e-mail or update.

Suggestions

As we've mentioned, our learning sciences team is continually making improvements to open response assessments. If you have any suggestions, please contact us at edx101@edx.org.
Discussion: Open Response Assessments Lesson

We hope you've found this beta version of the Open Response Assessment lesson useful and informative. Since edX101 is a work in progress, we would love to hear your feedback on the lesson. In the discussion space below, please let us know if you think some important information is missing, if other information is superfluous, or if you have any other questions or comments on this lesson. We're continually working to make edX101 better for you.

(Discussion space provided)
Building, Running, and Ending
Building

In this subsection, you will learn the basics of how to enter your videos, exercises, and worksheets into Studio.

This subsection is under construction and will be updated soon.
Running

Overview

You have four venues for communication with your students.

1. **Outbound: E-mail.**

---

*CS 184.1x: Foundations of Computer Graphics*

Welcome to Week 5

Welcome to week 5 in CS 184.1x. I hope you are doing well with unit 2 of the course. A reminder that you should submit homework 2 by Monday December 3, at 11:59pm Pacific Standard Time (07:59am Dec 4 GMT).

The next part of the course (unit 3) is on ray tracing, which is one of the most interesting topics in computer graphics, and allows creation of some of the most beautiful images. Those of you who have finished homework 2 have likely already qualified for a certificate.
The purpose of e-mail is to reach out to students and remind them about your course. It can provide a feeling of connection with the course staff. It is not a good place for essential information because it can easily get lost, forgotten, or deleted.

2. **Outbound: Announcements on the landing page.**

The announcements page is the first thing a student sees upon opening the course website. Newer announcements get placed above older announcements, so as a student scrolls down on the page, she sees a record of all the past announcements. The announcements page is a good place for information a student needs to see that particular week of the course.

3. **Inbound: The discussion forum.**

You will mainly write to the discussion forum in response to student posts. It is a good place for very specific issues do not need to be broadcasted to the entire class. For instance, in one course, students were using the discussion forum to pirate the recommended textbook; the professors wrote in asking them not to do this.
Summary of Outbound Communications

In this lesson, we will discuss your outbound communications. In the next lesson, we will discuss your inbound communications.

Here is a summary of outbound communications to prepare for the first week of your course:

- a reminder e-mail to be sent 5-10 days before the start date (optional),
- a welcome e-mail to be sent the first day (very important), and
- a welcome announcement on the landing page (very important).

These will be discussed in greater detail in the next 2 units.

After that, we will discuss weekly e-mails and announcements.
Welcome Email

You will need to send a welcome e-mail to your students on the first day of class. The e-mail should ideally be sent as soon as possible after the first week’s material has been released.

The welcome e-mail serves several purposes.

- It reminds students about the course. Students often sign up for a course months in advance, so they may have forgotten about it.
- It provides a link that students can click to get started.
- It generates excitement about the course and creates a friendly atmosphere.

When writing your welcome e-mail, it is useful to keep in mind that the landing page of your course will have a welcome announcement. It is best to avoid having too much redundancy between the welcome e-mail and the welcome announcement because your student is very likely to click on the link and go directly to the welcome announcement after reading the welcome e-mail. The announcement is a more logical place than the welcome email detailed instructions and information about the course because it will stay on the announcements page for the entirety of the course.

Examples

CS 188.1x Artificial Intelligence

Dear [student’s name],

Welcome to the launch of CS188x, which begins today! To get started, visit

http://www.edx.org/ai

where all of this week's materials are now live. This first week, you will learn about the goals of artificial intelligence, how current AI matches up against Hollywood portrayals, and what you can expect from this course. There is no homework in the first week, but if you want to check your preparedness for the course, there are two optional items: a short math self-diagnostic and a Python mini-refresher (or introduction).

We hope you enjoy your first week of class!

Dan, Pieter, and the CS188x Course Team
Notice that the e-mail is short and upbeat. It provides a link, and some information about the current week but does not provide broad information about the course (e.g. the grading rubric, details about the project, etc.). This is because all such information will be available on the website.

Some course teams decide to send out an e-mail the week before the launch date to give students time to prepare for the course. This is particularly useful if the course requires special software. Berkeley's CS 169.1x required students to set up and configure a virtual machine for their programming assignments. The welcome e-mail, which was sent several days in advance, is included below.

CS 169.1x Software as a Service

Dear [student’s name],

This course, which begins on Monday, is an opinionated path through the bewildering array of methodologies, languages, tools, and artifact types that collectively make up software engineering. The goal is to instill good software habits: "testability, software architecture, modularity, and reusability" while providing the gratification of building a working deployed artifact in the cloud in just 6 weeks. We teach Agile Development in the context of building and deploying a Software-as-a-Service application implemented using Ruby on Rails.

Videos:
The lectures are recorded live on campus at the University of California, Berkeley and then broken into several short segments for the online version of the course. At the end of most of the segments, there is a multiple-choice question to the audience; in the beginning of the next segment, the answer choices are discussed. The total is about 2 to 2.5 hours of video a week.

There are three important differences between the online course and the course given on-site at UC Berkeley. First, the UC Berkeley students are required to do a long-term programming project. Second, for the questions asked during the lecture by the professor, the Berkeley students use colored cards to vote on the correct answer. You will be given the opportunity to answer the questions on your own and check your answers before seeing the discussion. Finally, the on-site course is a 10-week course which has been divided into two shorter courses for the online version. You may hear an occasional reference to projects or cards, even though most have been removed.

Homework:
There are 4 homework assignments; all involve a substantial amount of programming. We will also give three 45-minute quizzes. You can only take each quiz once, but you can submit homework as many times as you
Each time you submit, you will receive feedback from the automatic grader.

Grading Breakdown:
Each of the 4 homework assignments will be worth 10% of your final grade.

- Quiz 0 and Quiz 3 will be worth 10% each.
- Quiz 1 and Quiz 2 will be worth 20% each.

Assignments turned in between one minute and one week late will receive a penalty of 25%, e.g. an assignment that would have scored 100 points would only receive 75 points if it was late.

Assignments turned in more than a week late will receive no credit, but will still receive feedback from the auto-grader.

Workload:
In terms of workload, 50% of the students in a previous course spent less than 6 hours a week on the course, 40% spent 6 to 12 hours a week, and 10% spent more than 12 hours a week.

Required software:
To do homework 1, you will need to run a virtual machine image on either a local or cloud virtual machine.

- To run the virtual machine on your own computer, you will need to use VirtualBox (instructions here). Please randomly pick a download site; if you run into problems, try a different download site.

- To work from the cloud, you will need to use Amazon Web Services Free Tier for Elastic Cloud Computing (EC2) and Simple Storage Service (S3) (instructions here).

In your homework assignments, you will create and modify a SaaS application called "RottenPotatoes". In the setup screencast for the Virtual Machine the instructor uses the contents of the directory "\Documents\rottenpotatoes". The current version of the Machine Image does not contain this directory, so the commands run in the video will not work. This difference will not prevent you from doing any of the projects for the course. We are working to find a new method to distribute this directory, and will post an update on the course info page when it is available.

We hope that you enjoy the class! At the end of the course we'll send a survey to collect your feedback on the material.

Good luck!

Armando and Dave

One nice feature of this e-mail is that it has personality ("an opinionated path through the bewildering array of methodologies..."). It also provides information about how the videos were recorded. For this course, this extra information was necessary because it was not possible to cut out all of the references to campus-specific details. The extra information about the videos was intended to prevent the online student from being confused when faced with a video mentioning projects or colored cards. This information isn't important for the students to be able to refer back to, so it is appropriate for a welcome e-mail.

This e-mail also discusses grading and assignments. This is generally not recommended because it is more difficult to make adjustments on the fly; once the e-mail has been sent, it is
impossible to change. Further, the grading rubric is already available on the site. However, it
does allow the student to see this extra information without extra mouse-clicks.
Welcome Announcement

Most of your students will be taking an online course for the first time and have no idea how to get started. The welcome announcement is the first thing a student will see when entering the course, and should therefore achieve the following goals.

- Create a friendly atmosphere.
- Let students know what to expect from the course. Many students will not know what exactly an online course is and will be wondering whether there will be graded homework assignments and avenues for questions and discussion.
- If you are not using traditional 60 or 90 minute lectures, you should tell your students what to expect and explain the rationale. Students will probably be expecting two or three hour-long videos per week and need to know that they should instead expect a sequence of short videos interleaved with exercises.
- Tell students how to get started. Students will not know to click on "Courseware". They will also not know to click on the right arrow after they are done looking at the first unit.

Examples

PH 207x Health in Numbers: Quantitative Methods in Clinical and Public Health Research

Welcome to the first edX offering of PH207x, "Health in Numbers: Quantitative Methods in Clinical and Public Health Research"! There are tens of thousands of students enrolled in this course and each of you will cover the same material and be held to the same level of rigor as the students taking the same on-campus course at the Harvard School of Public Health.

We're pleased to be able to announce that we'll be able to offer free access to Prof. Pagano's "Principles of Biostatistics" through our friends at Cengage.
You can all access an electronic version of the book using the "Textbook" tab above. If you'd rather purchase a hard copy, you can get yours here.

We hope you are eager to get started with the course. To begin this week's work, we recommend you complete the following three steps.

**Step 1:**
Download and review the syllabus. After you are done reading the syllabus, please review the course calendar and schedule. You can find these under Course Handouts, to the right.

**Step 2:**
Familiarize yourself with the course by going through the course overview. The course overview, as well as the lectures, problem sets, and labs are all under the Courseware tab at the top left of the page.

**Step 3:**
Begin your first learning sequence! Click on Courseware at the top left, select "Week 1", then the first sequence, "Basic Biostatistics: Data and Graphics".

Finally, we hope you will use the discussion area to ask questions, help one another, and to report bugs as they come up.

Good luck and welcome to the course!

Here is a link to the [online textbook](#) for PH207x.

**3.091x Introduction to Solid State Chemistry**

(In this course, the main page was opened to students before any of the course material was posted as a way of giving students time to browse around.)

I am pleased to welcome you on behalf of the entire course team to the first edX offering of 3.091x: Introduction to Solid State Chemistry. The next 13 weeks will be a learning experience that is just as rigorous as the on-campus MIT course. You will be watching the in-class lectures as short video segments as well as learning from additional
screencasts made specifically for the online environment. We have also created new questions between video segments that help you check that you have understood the concepts in the videos. The homework problems are similar to those used here at MIT and will require you to master the material presented in the videos, handouts, and text.

Please take some time to familiarize yourself with the edX platform by completing the following steps. These steps will explain both the course expectations, and how to use the edX courseware.

Step 1: Review the course expectations and course schedule by downloading the Syllabus and viewing the Course Calendar, Course-At-A-Glance (including the reading assignments), Collaboration Guidelines and the Discussion Forum Guidelines. These items are found on the right side of this page under the heading Course Handouts.

Step 2: Learn how the course website works and try out the basic features you will utilize at the Overview section under the Courseware tab.

On October 15 at 3 pm Eastern Standard Time, the first week of materials will go live and you can also start working on the first set of homework problems. You will notice that many of the videos were produced during the live lecture at MIT. This means that students may pass in view of the camera from time-to-time.

We expect that you will have questions. Please use the discussion forum to ask any questions you have related to the course and also to report any bugs or system issues you might have.

We are looking forward to an exciting and productive course. Thank you for joining us on this journey.

Michael and the 3.091x Course Team
The Forum

8. Discussion Forum

We will provide a discussion forum on our website for all students of 6.002x. You may use these forums to discuss course concepts, problem solving approaches, interesting references, or anything else that may be of interest. You may just use it to ask questions. Please observe the appropriate online etiquette and be courteous to your fellow students when using the discussion forum. The forum is moderated by course staff.

Course staff will be available at periodic timeslots to answer questions on the discussion forum. A schedule of the timeslots will be posted in a separate document under the Course Info tab.

The discussion forum is a critical component of an online course. Here, students ask questions, help each other and comment on various components of the course.

There are many ways that students can search for postings. Student posts can be sorted by date, votes, number of comments, or by categories.

- By date:

A student can sort the posts so that the most recent ones are on the top.

- By number of votes:

A "vote" is a way for students to agree with an existing comment or question by using the "plus button." A search by vote would allow the most upvoted questions to float to the top and give you a sense of the most pressing issues being discussed on the forum.

- By number of comments:

Next to each post header is a number indicating the number of comments made in response to the original posting.
• By category:

Participants can navigate the different categories of topics via the drop-down menu in the upper left hand corner. These categories are customizable for each course.

On the instructor end, the forum also allows you to "Approve" of a particular post, which will appear as tagged as such. This way, students can tell when the professor or TA has approved an answer to a question.

We encourage you to spend a significant amount of time on the discussion forum, as it is a great place to communicate with students. After a few weeks of the course, forums tend to become self-supporting in terms of peer learning. In the initial weeks, however, it is critical that you and a TA actively monitor the forum and create its culture. Even after students begin to support each other, there will always be questions that only course staff can answer. No issues should go completely unaddressed - even if it is simply to acknowledge an issue that can only be fixed the next time the course is offered.

**Forum Moderators**

Dedicated forum moderators are the key to effective online communities: They keep the discussions productive and relay important information (bugs, errata, student confusion or interest with particular topics) to the rest of the course staff. These team members should be hired and integrated with the rest of the course staff at least one week before the course launch. Successful moderators have the following qualities.

Good or great content knowledge: stellar students from previous years often make good moderators.

Solid communication skills: positive, consistent, and effective communication with students, other moderators, and course staff.

Time: Keeping up-to-date with a large MOOC forum requires at least 5 hours per week for reading posts, replying to or editing posts, and communicating with the other moderators and course staff.

Enthusiasm: It is the best predictor of forum moderator excellence! Undergraduates who have taken your course or online students from previous runs of your course often make good moderators.
Moderator privileges include editing and deleting posts, closing threads, and direct communication with the rest of the course staff. A document with more detailed information for forum moderators is available here.

Forum posts/comments from course staff and paid moderators have a blue "STAFF" banner/icon signifying their status:

Exemplary participants in the course can also be promoted to "Community TA" status. Community TAs are student volunteers who are given moderator privileges and green banners/icons in the forums. Forum moderators should keep an eye out for these highly engaged and helpful participants and invite them to become community moderators a few weeks after launch. We recommend having a brief voice or video chat with potential community TAs before promoting them to make sure they understand the responsibilities involved.

You and your course staff are highly encouraged to engage frequently with students in the forums. These discussions are excellent sources of feedback, ideas for future iterations of the course, and rewarding community development.
The Wiki

Every course has a wiki. How the wiki will be used will be defined by how you initially seed the wiki with content, by how you introduce it at the beginning of the course, and by how students want to use it. It was originally designed as a first-pass at allowing students to collaboratively take sequence notes, in an experiment to see whether this could evolve into an interactive, open, on-line textbook, as well as to see how students interact with it. Common uses include:

- On-line errata for the course. Useful to students for avoiding bugs, and to instructors for fixing them.
- Suggestions for a second run
- Bugs in the platform
- Links to student-created resources
- User-editable course handouts (e.g. discussion forum policies, instructions for specific pieces of software, etc.)

If you wish to encourage specific uses, it is good to seed the wiki with content, and to point students to how you would like the wiki used. For instance, you can create a document in the wiki called "Transcript errata" for your students to fill with errors they find in the transcripts, and then mention it in the introductory lesson of the course.
Weekly E-Mail and Announcements

Sending a weekly e-mail and posting a weekly announcement provides a human connection for your students and shows them that you are actively engaged with the course. Weekly e-mails and updates can include the following information.

- Answers to common questions.
- Acknowledgment of students who actively post on the forums.
- Relevant current events (for example, if one of the thinkers behind a topic you are covering wins a Nobel Prize, that would be worth mentioning).
- A survey (don't do this too often). If you do a survey, you can discuss the results of the survey.
- Information about general performance on homework assignments and exams.
- Other interesting course related data.
- You can talk about interesting student posts or discussions.
- Interesting tidbits about course materials.

You can also post video announcements on your landing page from time to time. To see an example of a video announcement, click here.

Examples of Weekly E-Mails

**CS 188.1x Artificial Intelligence**

![CS188x](image)

Dear [student’s name],

Week 2 of CS188x: Artificial Intelligence is now live! To access the course materials, visit:

http://www.edx.org/ai

Now that you have seen some of the things that AI can (and can't) do, it's time to start learning about the underlying techniques. This week's topic is search. You will learn about agents that plan ahead, single-agent search problems, and general-purpose algorithms that solve them efficiently.
This week's timeline is as follows:

Lectures 2 and 3: Live now
Homework 1: Live now
Project 1: Releasing on Thursday 10/4

As a general timeline, primary lecture videos and homeworks will go live on Mondays. Projects and secondary video processing, which includes captioning, may release later in the week.

As you work through the homeworks and projects, we encourage you to continue taking advantage of the forums. They are there to help connect with the staff and with your classmates, so don't be shy about getting involved (though, as always, please do avoid spoilers in your posts). We've already had some great threads, so thanks to all those who are already contributing!

Happy searching,

Dan, Pieter, and the CS188x Course Team

CS 188.1x Artificial Intelligence

Dear [student’s name],

Week 6 of CS188x: Artificial Intelligence is now live! To access the course materials, visit:

http://www.edx.org/ai

THIS WEEK: We're now embarking on the last week of new technical material! Last week, we saw how to plan when our actions are uncertain, using a known probability distribution over possible outcomes to weigh consequences. This week, these ideas culminate in reinforcement learning, where not only are our actions uncertain in the sense that they have multiple possible outcomes, but we do not even know what those outcomes are until we try them. As a result, our agents must experiment and learn, trading off between exploration (which gains new knowledge) and exploitation (which makes use of current knowledge). In the process of scaling up to large state spaces, we'll see one of the most common techniques in machine learning: feature-based representations.

This week's timeline is as follows:

Homework 4: Released last week (due Saturday 11/3)
Lectures 10 and 11: Live now
Homework 5: Live now (due Wednesday 11/7)
Project 3: Live now (due Sunday 11/11)

As in past weeks, secondary video processing, which includes captioning, will release a few days after the original video releases.
COMMUNITY SPOTLIGHT: We're excited to see such a large number of insightful forum contributions from so many students in the course, including many thoughtful contributions from rikie, chasmosaurus, and Biki23, as well as our now-larger group of community TAs: ichaleynbin, BayAreaRon, kevinsallee, PoorCoco, rdh, CarbonTwelve, and vulcronos. We also want congratulate all participants in the P1 mini-contest! We received a lot of very strong submissions. Special congratulations to the top three: ksuszka (first place), and CarbonTwelve and nataraja (tied for second)! This post lists the top twenty-one and expands on some of the ideas that went into the top-scoring submissions.

Happy (reinforcement) learning!

Examples of Weekly Announcements

Here are links for announcements pages for some edX courses.

- BerkeleyX CS 188.1x, Artificial Intelligence
- BerkeleyX CS 169.1x, Software as a Service
- BerkeleyX CS 184x, Foundations of Computer Graphics
- HarvardX CS 50x, Introduction to Computer Science I
- MITx 3.091x, Introduction to Solid State Chemistry
- MITx 6.002x, Circuits and Electronics
- MITx 6.00x, Introduction to Computer Science and Programming
Ending
Certificates of Mastery

Students who successfully pass an edX course receive a Certificate of Mastery. These certificates are automatically issued by edX. However, we will need the following items from you:

- The preferred spelling of your name,
- Your title,
- A high resolution scan of your signature, and
- Certificate issue date. The issue date must be after the last due date. Note that if extend a due dates beyond the certificate issue date, you will need to choose a new issue date and notify edX.
Appendix
Course Readiness Checklists

Below is a list of interactive checklists useful for keeping track of your course launch-readiness.

- Course Catalog checklist (.pdf)
- Course Info checklist (.pdf)
- Exercises checklist (.pdf)
- Schedule checklist (.pdf)
- Video checklist (.pdf)
Sample HTML Templates

Simple HTML Template
The Simple HTML template creates the following page.

An edX Haiku
Start an online course
And teach in your pajamas
Nobody will know

(Image courtesy of Julia Zimmerman, jwzimmer1990@gmail.com.)
Simple HTML Template

<h1>An edX Haiku</h1>
<p>
Start an online course<br/>
And teach in your pajamas<br/>
Nobody will know  
</p>
<p><img alt="" src="/static/haiku.jpg"></p>
Dynamic HTML Schedule

The Dynamic HTML Schedule template creates the following page.

<table>
<thead>
<tr>
<th>Wk of</th>
<th>Topic</th>
<th>Read</th>
<th>Learning Sequence</th>
<th>Slides</th>
<th>HW/Q</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/22</td>
<td>Topic 1</td>
<td>Ch. 1</td>
<td>L1: Title</td>
<td>L1</td>
<td></td>
<td>11/04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>L2: Title</td>
<td>L2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10/29</td>
<td>Topic 2</td>
<td>Ch. 2</td>
<td>L3: Title</td>
<td>L3</td>
<td>Quiz 1</td>
<td>11/11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>L4: Title</td>
<td>L4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/05</td>
<td>Topic 3</td>
<td>Ch. 3</td>
<td>L5: Title</td>
<td>L5</td>
<td>HW 2</td>
<td>11/18</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>L6: Title</td>
<td>L6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/12</td>
<td>Topic 4</td>
<td>Ch. 4</td>
<td>L7: Title</td>
<td>L7</td>
<td>Quiz 2</td>
<td>11/25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>L8: Title</td>
<td>L8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/19</td>
<td>Topic 5</td>
<td>Ch. 5</td>
<td>L9: Title</td>
<td>L9</td>
<td>HW 3</td>
<td>12/02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>L10: Title</td>
<td>L10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11/26</td>
<td>Topic 6</td>
<td>Ch. 6</td>
<td>L11: Title</td>
<td>L11</td>
<td>HW 4</td>
<td>12/09</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>L12: Title</td>
<td>L12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Dynamic HTML Schedule Template

```html
<div class="syllabus">
<table style="width: 100%">
  <col width="10%">
  <col width="15%">
  <col width="10%">
  <col width="30%">
  <col width="10%">
  <col width="15%">
  <col width="10%">

  <!-- Headings -->
  <thead>
    <td class="day"> Wk of </td>
    <td class="topic"> Topic </td>
    <td class="reading"> Read </td>
    <td class="video"> Lecture Sequence </td>
    <td class="slides"> Slides </td>
    <td class="assignment"> HW/Q </td>
  </thead>
</div>
```

edX101: How to Create an edX Course
### Week 1

<table>
<thead>
<tr>
<th>Day</th>
<th>Topic</th>
<th>Reading</th>
<th>Video</th>
<th>Slides</th>
<th>Assignment</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/22</td>
<td>Topic 1</td>
<td>Ch. 1</td>
<td>L1: Title</td>
<td>L1</td>
<td>HW 1</td>
<td>11/04</td>
</tr>
<tr>
<td>10/29</td>
<td>Topic 2</td>
<td>Ch. 2</td>
<td>L3: Title</td>
<td>L3</td>
<td>Quiz 1</td>
<td>11/11</td>
</tr>
</tbody>
</table>

### Week 2

<table>
<thead>
<tr>
<th>Day</th>
<th>Topic</th>
<th>Reading</th>
<th>Video</th>
<th>Slides</th>
<th>Assignment</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/05</td>
<td>Topic 3</td>
<td>Ch. 3</td>
<td>L5: Title</td>
<td>L5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**edX101: How to Create an edX Course**
<table>
<thead>
<tr>
<th>Week 3 Row 2</th>
<th>Week 4 Row 1</th>
<th>Week 4 Row 2</th>
<th>Week 5 Row 1</th>
<th>Week 5 Row 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/18</td>
<td>11/12</td>
<td>11/19</td>
<td>11/19</td>
<td>11/19</td>
</tr>
<tr>
<td>HW 2</td>
<td>Quiz 2</td>
<td>HW 3</td>
<td>HW 3</td>
<td>L10: Title</td>
</tr>
<tr>
<td></td>
<td>Quiz 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L6: Title</td>
<td>L7: Title</td>
<td>L8: Title</td>
<td>L9: Title</td>
<td>L10: Title</td>
</tr>
<tr>
<td>L6</td>
<td>L8</td>
<td>L8</td>
<td>L9</td>
<td>L10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L6: Title</td>
<td>L7: Title</td>
<td>L8: Title</td>
<td>L9: Title</td>
<td>L10: Title</td>
</tr>
<tr>
<td>L6</td>
<td>L8</td>
<td>L8</td>
<td>L9</td>
<td>L10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Quiz 2</td>
<td>Quiz 2</td>
<td>Quiz 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Quiz 2</td>
<td>Quiz 2</td>
<td>Quiz 2</td>
<td></td>
</tr>
<tr>
<td>L8: Title</td>
<td>L8: Title</td>
<td>L8: Title</td>
<td>L8: Title</td>
<td>L10: Title</td>
</tr>
<tr>
<td>L8</td>
<td>L8</td>
<td>L8</td>
<td>L8</td>
<td>L10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Day</td>
<td>Topic</td>
<td>Reading</td>
<td>Video</td>
<td>Slides</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>11/26</td>
<td>Topic 6</td>
<td>Ch. 6</td>
<td>L11: Title</td>
<td>L11</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>L12: Title</td>
<td>L12</td>
</tr>
</tbody>
</table>

---

Week 6 Row 1

Week 6 Row 2

---

edX101: How to Create an edX Course
HTML Syllabus

The HTML Syllabus template creates the following page.

Syllabus

- WELCOME TO THE COURSE!
- PREREQUISITES
- WORKLOAD
- GRADING
- COURSE SCHEDULE

Note that you can expand and collapse each section.

Syllabus

- WELCOME TO THE COURSE!

- PREREQUISITES
  Programming proficiency in an object-oriented programming language such as Java, C, C++, Python, or Ruby is required. We will teach the basics of Ruby at a very accelerated pace that assumes thorough familiarity with OOP inheritance, static/class vs. instance methods and attributes, recursion, hash tables/mash maps, list comprehensions, higher-order functions, lambda expressions. This course is NOT a good first course in programming. Basic Unix command-line skills are helpful; we will provide a cheat sheet.

- WORKLOAD

- GRADING

- COURSE SCHEDULE
  You can view the Fall schedule for CS169.1x as a PDF, or as a Google Calendar.

Release schedule:
- All lessons are released on Mondays.
- All homework assignments are released on Wednesdays, and are due on Sunday of the following week.
- All quizzes are released on Fridays and due on Sunday of the following week (9 days later).

<table>
<thead>
<tr>
<th>Week of</th>
<th>Topic</th>
<th>Reading</th>
<th>Slides</th>
<th>Lecture Sequence</th>
<th>Assignment</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mo 9/24</td>
<td>Software Engineering and Skill Architecture</td>
<td>Ch. 1</td>
<td>Lecture 1</td>
<td>Lecture 1 Course Introduction</td>
<td>Quiz 0</td>
<td>9/7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ch 2</td>
<td>Lecture 2</td>
<td>Lecture 2.3D and Cloud Computing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mo 10/1</td>
<td>Ruby</td>
<td>Ch 3</td>
<td>Lecture 3</td>
<td>Lecture 3: Introduction to Ruby</td>
<td>HW 1</td>
<td>10/16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Lecture 4</td>
<td>Lecture 4: Ruby Continued</td>
<td>Quiz 1</td>
<td>10/14</td>
</tr>
</tbody>
</table>
**HTML Syllabus Template**

```html
<html>

<script language="JavaScript" type="text/javascript">
if (document.getElementById) {
  document.writeln("<style type="text/css">!--
  document.writeln('.texter {display:none} @media print {.texter
  {display:block;}}')
  document.writeln('--></style>")
}

function openClose(theID) {
  if (document.getElementById(theID).style.display == "block")
  {
    document.getElementById(theID).style.display = "none"
  }
  else
  {
    document.getElementById(theID).style.display = "block"
  }
}

function switchStyle(theID) {
  if (document.getElementById(theID).className == 'subsection_1')
  {
    document.getElementById(theID).className = "subsection_2";
  }
  else
  {
    document.getElementById(theID).className = "subsection_1";
  }
}

function expandCollapseAll(theID) {
  subsList=['b1','b2'];
  compsList=['a1','a2'];
  if (document.getElementById('all').className == 'expandAll')
  {
    document.getElementById('all').className = "collapseAll";
    document.getElementById('all').innerHTML="<img
      src="/c4x/edX/Art101/asset/collapse-all.png" title='Collapse All'/><span
      style='font-size:.7em'>Collapse all</span>";
    for (var i = 0; i < compsList.length; i++) {
      var id = compsList[i];
      document.getElementById(id).style.display = "block";
    }
    for (var i = 0; i < subsList.length; i++) {
      var id = subsList[i];
      document.getElementById(id).className = "subsection_2";
    }
  }
  else{
    document.getElementById('all').className = "expandAll";
    document.getElementById('all').innerHTML="<img
      src="/c4x/edX/Art101/asset/expand-all.png" title='Expand All'/><span
      style='font-size:.7em'>Expand all</span>";
    for (var i = 0; i < compsList.length; i++) {
      var id = compsList[i];
      document.getElementById(id).style.display = "none";
    }
    for (var i = 0; i < subsList.length; i++) {
  
```

edx101: How to Create an edX Course
Syllabus

Welcome to the Course!

Welcome to Software Engineering for Software as a Service! This course, which begins on Monday, is an opinionated path through the bewildering array of methodologies, languages, tools, and artifact types that collectively make up software engineering. The goal is to instill good software habits: “testability, software architecture, modularity, and reusability” while providing the gratification of building a working deployed artifact in the cloud in just 6 weeks. We teach Agile Development in the context of building and deploying a Software-as-a-Service application implemented using Ruby on Rails.

Prerequisites

Programming proficiency in an object-oriented programming language such as Java, C#, C++, Python, or Ruby is required. We will teach the basics of Ruby at a very accelerated pace that assumes thorough familiarity with OOP inheritance, static/class vs. instance methods and attributes, recursion, hash tables/hash maps, list comprehensions, higher-order functions, lambda expressions. This course is NOT a good first course in programming. Basic Unix command-line skills are helpful; we will provide a cheat sheet.
<h2>Workload</h2>

In a previous version of the course,

50% of the students spent 6 hours or less per week on the course
40% spent 6 to 12 hours a week
10% spent more than 12 hours a week

<h2>Grading</h2>

By the end of this course, you will have completed 4 homework assignments, all of which involve a substantial amount of programming. Each assignment can be submitted multiple times without penalty, so you can use feedback from the automatic grader to revise your code. Note that your most recent score will be recorded as your grade.

Homework assignments turned in between one minute and one week late will receive a penalty of 25%. An assignment that would have scored 100 points would only receive 75 points if it is submitted 2 days late. Assignments turned in more than a week late will receive no points, but will still receive feedback from the autograder.

In addition to the homework assignments, there will be 4 quizzes. You will have only one chance to take each quiz, with the exception of Quiz 0 (the diagnostic quiz). You may take Quiz 0 as many times as you want. All quizzes must be turned in by the end of the day on the due date (midnight of your local time). Late quizzes will receive 0 points.

Each of the 4 homework assignments will be worth 10% of the final grade.
The diagnostic quiz (Quiz 0) is worth 10% of the final grade.
Quiz 1, Quiz 2, and Quiz 3 are worth 20%, 20% and 10% of the final grade, respectively.
To earn a certificate for this class you must achieve a score of at least 50% overall.
<h2>Course Schedule</h2>

You can view the Fall 2012 schedule for CS169.1x as a PDF, or as a Google Calendar.

Release schedule:

- All lessons are released on Mondays.
- All homework assignments are released on Wednesdays, and are due on Sunday of the following week.
- All quizzes are released on Fridays and due on Sunday of the following week (9 days later).

<table>
<thead>
<tr>
<th>Week of</th>
<th>Topic</th>
<th>Reading</th>
<th>Slides</th>
<th>Lecture Sequence</th>
<th>Assignment</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mo 9/24</td>
<td>Software Engineering and SaaS Architecture</td>
<td>Ch. 1</td>
<td>Lecture 1: Course Introduction.pdf</td>
<td>Lecture 1: Course Introduction</td>
<td>Quiz 0</td>
<td>10/7</td>
</tr>
</tbody>
</table>

Ch. 2
<table>
<thead>
<tr>
<th>Day</th>
<th>Topic</th>
<th>Reading</th>
<th>Slides</th>
<th>Video</th>
<th>Assignment</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mo 10/1</td>
<td>Ruby</td>
<td>Ch. 3</td>
<td>Lecture 3: Introduction to Ruby</td>
<td>Lecture 3: Introduction to Ruby</td>
<td>HW 1</td>
<td>10/16</td>
</tr>
<tr>
<td>Mo 10/8</td>
<td>Rails</td>
<td>Ch. 4</td>
<td>Lecture 5: SaaS Architecture &amp; intro to Rails</td>
<td>Lecture 5: SaaS Architecture &amp; intro to Rails</td>
<td>HW 2</td>
<td>10/21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

edX101: How to Create an edX Course
<table>
<thead>
<tr>
<th>Day</th>
<th>Topic</th>
<th>Reading</th>
<th>Slides</th>
<th>Video</th>
<th>Assignment</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mo 10/15</td>
<td>Behavior Driven Development</td>
<td>Ch. 5</td>
<td><a href="/static/handouts/slides/169xLecture8.pdf">Lecture 8</a></td>
<td><a href="/course/courseware/Week_4x/L8/">Lecture 8: Behavior Driven Design</a></td>
<td>HW 3</td>
<td>10/28</td>
</tr>
<tr>
<td>Mo 10/22</td>
<td>Test Driven Development</td>
<td>Ch. 6</td>
<td><a href="/static/handouts/slides/169xLecture10.pdf">Lecture 10</a></td>
<td><a href="/course/courseware/Week_5x/L10/">Lecture 10: Test Driven Development</a></td>
<td>HW 4</td>
<td>11/7</td>
</tr>
</tbody>
</table>

This table outlines the schedule for the course, including the topics, readings, and assignments for each day. The course covers a variety of topics related to software development, including Behavior Driven Development and Test Driven Development.
<table>
<thead>
<tr>
<th>Day</th>
<th>Topic</th>
<th>Reading</th>
<th>Slides</th>
<th>Video</th>
<th>Assignment</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mo 10/29</td>
<td>Conclusion</td>
<td>N/A</td>
<td>N/A</td>
<td>Farewell Video</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Long Course Description

The Long Course Description template creates the following page.

In November 2012, the New York Times declared 2012 to be "The Year of the MOOC." The excitement has yet to die down: everyone has been talking about the future of learning and the transformative power of this new medium. Students from all over the world have written into the course forums with awe and gratitude.

EdX101 is designed specifically to help you, the course author, harness the intrinsic strengths of this new vehicle for learning while walking you through the mechanics of creating your own course. By the time you have completed edX101, you will also have completed the entire process of planning, building, and running your own course. Along the way, you will have the opportunity to interact with other faculty and course staff using the edX101 discussion forum, and to share ideas and collaborate using the wiki. Unlike most edX courses, this course has no exams, grades, or fixed schedule.

Note To register for edX101, you will have to create an account for edge.edX.org (edX's domain for experimental online courses) even if you already have an edX.org account. To do so, click Sign up. Once you have successfully signed up, you will need to come back to this page (you cannot get to it by clicking the Find Courses button).

Long Course Description Template

<!-- For regular text, you just need to use paragraph (<p>) tags. -->

<!-- If you want to include a link, copy everything between the blue <a> tags below (including the <a> tags). Replace the green text with the URL of your link, and replace the purple text with the text that you want to appear in the body of your long course description. -->

<!-- The <em> tags create italic text. The <strong> tags create bold text. -->

<p>In November 2012, the New York Times declared 2012 to be "<a href="http://www.nytimes.com/2012/11/04/education/edlife/massive-open-online-courses-are-multiplying-at-a-rapid-pace.html" target="_blank">The Year of the MOOC</a>." The excitement has yet to die down: everyone has been talking about the future of learning and the transformative power of this new medium. Students from all over the world have written into the course forums with awe and gratitude.</p>

<p>EdX101 is designed specifically to help you, the course author, harness the intrinsic strengths of this new vehicle for learning while walking you through the mechanics of creating your own course. By the time you have completed edX101, you will also have completed the entire process of planning, building, and running your own course. Along the way, you will have the opportunity to interact with other faculty and course staff using the edX101 discussion forum, and to share ideas and collaborate using the wiki. Unlike most edX courses, this course has no exams, grades, or fixed schedule.</p>
<p><em><strong>Note</strong></em> To register for edX101, you will have to create an account for edge.edX.org (edX's domain for experimental online courses) even if you already have an edX.org account. To do so, click <strong>Sign up</strong>. Once you have successfully signed up, you will need to come back to this page (you cannot get to it by clicking the <strong>Find Courses</strong> button).</p>
Course Overview

If your course is hosted on Edge, you can use the Course Overview template to create content for the Introducing Your Course field on the Schedule & Details page in Studio.

Course Overview Template

<!-- In this template, you'll replace bold placeholder text with your own information. You don't have to make any other changes to the tags or code. -->

<!-- The text between the h2 tags in each section tells you the name of that section. -->

<!-- When you're done adding your information, copy and paste everything from this document into the Course Overview field under Introducing Your Course. -->

<!-- The following section contains the long description for your course. Replace the bold text with your information.-->

<section class="about">
  <h2>About This Course</h2>
  <p>Include your long course description here. The long course description should contain 150-400 words.</p>
  <p>This is paragraph 2 of the long course description. Add more paragraphs as needed. Make sure to enclose them in paragraph tags.</p>
  <p>If you want to include an e-mail address in your long course description, use the format <a href="mailto:course_name@organization.com">course_name@organization.com</a>.</p>
</section>

<!-- The following section contains your prerequisites. Replace the bold text with your information.-->

<section class="prerequisites">
  <h2>Prerequisites</h2>
  <p>Add information about course prerequisites here.</p>
</section>

<!-- The following section contains your course staff pictures and biographies. -->

<section class="course-staff">
  <h2>Course Staff</h2>

  <!-- Add the information for the first staff member below, between the opening and closing article tags. -->
  <!-- Replace /static/images/pl-faculty.png with the URL for the picture of your first staff member. You obtain the URL when you upload the picture to the Files & Uploads page. For example, for a picture in an edX course whose course number is PC101x, the URL would resemble /c4x/edX/PC101x/asset/StaffMemberName.png. -->

edX101: How to Create an edX Course
<!-- Replace Staff Member #1 and Biography of instructor/staff member #1 with the name and biographical information for your first staff member. -->
<article class="teacher">
  <div class="teacher-image">
    <img src="/static/images/pl-faculty.png" align="left" style="margin:0 20px 0">
  </div>
  <h3>Staff Member #1</h3>
  <p>Biography of instructor/staff member #1</p>
</article>

<!-- Add the information for the second staff member below, between the opening and closing article tags. Replace the bold placeholder text with your staff member's information. -->
<article class="teacher">
  <div class="teacher-image">
    <img src="/static/images/pl-faculty.png" align="left" style="margin:0 20px 0">
  </div>
  <h3>Staff Member #2</h3>
  <p>Biography of instructor/staff member #2</p>
</article>

<!-- For any additional staff members, copy and paste sets of tags from <article class="teacher"> to </article> (including the article tags), then replace the placeholder text inside the set of tags with a staff member's information. Make sure that you paste the sets of tags before the closing section tag below. -->

</section>

<!-- The following section contains the frequently asked questions for your course. -->
<section class="faq">
  <section class="responses">
    <h2>Frequently Asked Questions</h2>
    <article class="response">
      <h3>Do I need to buy a textbook?</h3>
      <p>No, a free online version of Chemistry: Principles, Patterns, and Applications, First Edition by Bruce Averill and Patricia Eldredge will be available, though you can purchase a printed version (published by FlatWorld Knowledge) if you’d like.</p>
    </article>
    <article class="response">
      <h3>Question #2</h3>
      <p>Your answer would be displayed here.</p>
    </article>
    <!-- For any additional questions, copy and paste sets of tags from <article class="response"> to </article> (including the article tags), replacing the

edX101: How to Create an edX Course
placeholder text with your questions and answers. Make sure that you paste
the sets of tags before the closing section tag below. -->

</section>
</section>
XML Documentation

Option Response (Drop-Down List)

Description
The Option Response input type allows the student to choose from a collection of answer options, presented as a drop-down list.

XML tags
optionresponse with optioninput sub element

Example Problem
Option Response is most similar to ________.
- Multiple Choice
- String Response
- Numerical Response
- External Response
- Image Response

**Answer:** Multiple Choice

**Explanation**

Like Option Response, Multiple Choice also allows students to select from a variety of pre-written responses.

**Code**

```xml
<problem>
  <p><b>Example Problem</b></p>
  <p>Option Response is most similar to _________.</p>

  <optionresponse>
    <optioninput
      options="('Multiple Choice','String Response','Numerical Response','External Response','Image Response')"
      correct="Multiple Choice"/>
  </optionresponse>

  <solution>
    <div class="detailed-solution">
      <p>Explanation</p>
      <p>Like Option Response, Multiple Choice also allows students to select from a variety of pre-written responses.</p>
    </div>
  </solution>
</problem>
```

**Template**

```xml
<problem>

  <optionresponse>
    <optioninput
      options="('A','B')"
      correct="A"/>
  </optionresponse>

  <solution>
    <div class="detailed-solution">
      </div>
  </solution>
</problem>
```

edX101: How to Create an edX Course
## XML Elements and Attributes

**Element: `<optionresponse>`**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- none --</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Subelement**

<table>
<thead>
<tr>
<th>Subelement</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>All standard HTML tags</td>
<td>Display text relevant to the problem</td>
<td></td>
</tr>
</tbody>
</table>

**Element: `<optioninput>`**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>options</td>
<td>A list of options that students can choose from</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The attribute must be defined with double quotes, and the values in the list with single quotes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>There must not be a space between the separating commas and the single-quote start of the option.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The answers as displayed to students cannot contain any quotes.</td>
<td></td>
</tr>
</tbody>
</table>

| correct    | The option from the list of options that the grader will take as correct. | In order for students to get credit, this option must be typed exactly the same as the corresponding item in the 'options' list. |

**Subelement**

<table>
<thead>
<tr>
<th>Subelement</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- none --</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Multiple Choice Response (Drop-Down List)

Description

The Multiple Choice Response input type allows the student to select one choice from a collection of answer choices, presented as a list of radio buttons.

Notes

When creating a problem from a Multiple Choice Response type, you have to set up the response options (choices) correctly in order for grading to work properly. In particular, as stated by the answer to the following question, you should mark exactly one choice as correct.

In order for the correct answer to properly be displayed when Show Answer is clicked, you have to wrap the answer options in text tags in XML, as in the template below.

XML tag

multiplechoiceresponse with choicegroup sub element

Example Problem

How many response choices should you mark as correct when creating a problem from a Multiple Choice Response type?

a. Any number
b. Exactly one
c. Two
d. Zero

Answer: Exactly one

Explanation

You should mark exactly one choice as correct when setting up a Multiple Choice Response input, otherwise scoring and grading will not work properly.

In the Simple Editor, the incorrect choices for a Multiple Choice Response are identified by a pair of parentheses with a space in between, as follows: ( ). The correct choice is marked by filling in the parentheses with an x, as follows: (x).

In the Advanced Editor, this property is controlled by an attribute. Incorrect <choice> elements are identified by setting the attribute correct="false", and the correct <choice> should have correct="true".
**Code**

```xml
<problem>
  <p>How many response choices should you mark as correct when creating a problem from a Multiple Choice Response type?</p>
  <multiplechoiceresponse>
    <choicegroup type="MultipleChoice">
      <choice correct="false" name="any"><text>Any number</text></choice>
      <choice correct="true" name="one"><text>Exactly one</text></choice>
      <choice correct="false" name="two"><text>Two</text></choice>
      <choice correct="false" name="zero"><text>Zero</text></choice>
    </choicegroup>
  </multiplechoiceresponse>
  <solution>
    <div class="detailed-solution">
      <p>Explanation</p>
      <p>You should mark exactly one choice as correct when setting up a Multiple Choice Response input, otherwise scoring and grading will not work properly.</p>
      <p>In the Simple Editor, the incorrect choices for a Multiple Choice Response are identified by a pair of parentheses with a space in between, as follows: <b>( )</b>. The correct choice is marked by filling in the parentheses with an x, as follows: <b>(x)</b>. </p>
      <p>In the Advanced Editor, this property is controlled by an attribute. Incorrect <code>&lt;choice&gt;</code> elements are identified by setting the attribute <code>correct="false"</code>, and the correct <code>&lt;choice&gt;</code> should have <code>correct="true"</code>.</p>
    </div>
  </solution>
</problem>
```

**Template**

```xml
<problem>
  <multiplechoiceresponse>
    <choicegroup type="MultipleChoice">
      <choice correct="false" name="a"><text>A</text></choice>
      <choice correct="true" name="b"><text>B</text></choice>
    </choicegroup>
  </multiplechoiceresponse>
  <solution>
    <div class="detailed-solution">
    </div>
  </solution>
</problem>
```
## XML Elements and Attributes

### Element: `<multiplechoiceresponse>`

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- none --</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subelement</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>All standard HTML tags</td>
<td>Display text relevant to the problem</td>
<td></td>
</tr>
<tr>
<td><code>&lt;choicegroup&gt;</code></td>
<td>A method to take input from students. See description below.</td>
<td></td>
</tr>
</tbody>
</table>

### Element: `<choicegroup>`

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Has to be set to &quot;MultipleChoice&quot;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subelement</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;choice&gt;</code></td>
<td>The option from the list of options that the grader will take as correct.</td>
<td>A description of an option to be displayed to students. Details below.</td>
</tr>
</tbody>
</table>

### Element: `<choice>`

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>(Optional) A unique name that the backend will use to refer to this option.</td>
<td></td>
</tr>
<tr>
<td>Correct</td>
<td>Has to be set to either &quot;true&quot;, if this option is the correct answer, or &quot;false&quot;.</td>
<td>Exactly one choice subelement in the choicegroup element for a should have correct=&quot;true&quot; in order for computing point scores and grading to work properly.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subelement</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;text&gt;</td>
<td>Answer choices have to be wrapped in a &lt;text&gt;tag to ensure the correct answer is appropriately displayed when Show Answer is clicked.</td>
<td></td>
</tr>
</tbody>
</table>
Checkbox Response

Description

The Checkbox Response input type allows the student to select a set of choices from a list of possible answers.

A Checkbox Response question has exactly one correct set of answers. All the choices marked as `correct="true"` have to be selected for the submitted answer (i.e. the response) to be considered correct. In particular, the response of no boxes checked off could be the single correct response, although this use is discouraged.

XML tags

`checkboxgroup` sub element of `choiceresponse`

Example Problem

*How many correct responses can a Checkbox Response question have?*

- Zero
- One
- Two or more
- Nobody knows
- Somebody might know :)  

**Answer:** Two or more and Somebody might know

Explanation

Each Checkbox Response question has exactly one response that is evaluated as correct. This is the response consisting of checked-off boxes exactly for those choices that are marked as `correct` in the underlying response input in the problem.

Code

```xml
<problem>
  <startouttext/>
  <p>How many correct responses can a Checkbox Response question have?</p>
  <endouttext/>

  <choiceresponse>
    <checkboxgroup>
      <choice correct="false"><text>Zero</text></choice>
    </checkboxgroup>
  </choiceresponse>
</problem>
```
<choice correct="true"><text>One</text></choice>
<choice correct="false"><text>Two or more</text></choice>
<choice correct="false"><text>Nobody knows</text></choice>
<choice correct="true"><text>Somebody might know :)</text></choice>
</checkboxgroup>
</choiceresponse>

<solution>
  Each Checkbox Response question has exactly one response that is evaluated as correct. This is the response consisting of checked-off boxes exactly for those choices that are marked as correct in the underlying response input in the problem.
</solution>
</problem>

## Template

<problem>
<choiceresponse>
<checkboxgroup>
<choice correct="false"><text>Zero</text></choice>
<choice correct="true"><text>One</text></choice>
</checkboxgroup>
</choiceresponse>

<solution>
</solution>
</problem>

## XML Elements and Attributes

### Element: `<choiceresponse>`

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- none --</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Subelement: `<checkboxgroup>`

A method to take input from students. See description below.

### Element: `<checkboxgroup>`

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- none --</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subelement</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;checkboxgroup&gt;</code></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
<choice>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>correct</td>
<td>Has to be set to either &quot;true&quot;, if this option is one of the correct answers, or &quot;false&quot;, if this option is not one of the correct answers.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subelement</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;text&gt;</td>
<td>Answer choices have to be wrapped in a &lt;text&gt;tag to ensure the correct answer is appropriately displayed when Show Answer is clicked.</td>
<td></td>
</tr>
</tbody>
</table>
String Response

Description

The String Response input type provides an input box in which the student can enter a line of text, which is then checked against a specified expected answer.

XML tag

stringresponse

Example Problem

What is the name of this problem? (What response type is this?)

(text edit box is available to type and submit answer)

Answer: String Response

Explanation

The name of this problem is "String Response," written without the punctuation marks.

Capitalization does not matter, so "string response" and "STRING RESPONSE" would also be correct answers.

Putting quotation marks around the answer, however, would be marked as incorrect.

Code

<problem>
  <p><b>Example Problem</b></p>
  <p>What is the name of this unit? (What response type is this?)</p>
  <stringresponse answer="String Response" type="ci">
    <textline size="20"/>
  </stringresponse>
  <solution>
    <div class="detailed-solution">
      <p>Explanation</p>
      <p>The name of this unit is "String Response," written without the punctuation.</p>
      <p>Arbitrary capitalization is accepted.</p>
    </div>
  </solution>
</problem>
Template

<problem>

<stringresponse answer="REPLACE_THIS" type="ci">
  <textline size="20"/>
</stringresponse>

<solution>
  <div class="detailed-solution">
    
  </div>
</solution>

</problem>

XML Elements and Attributes

Element: <stringresponse>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>(Optional) Set this to &quot;ci&quot; if the student response should be graded case-insensitively. The default is to take case into consideration when grading.</td>
<td></td>
</tr>
<tr>
<td>answer</td>
<td>The string that students need to enter in order to get credit.</td>
<td></td>
</tr>
</tbody>
</table>

Subelement: <textline>

<table>
<thead>
<tr>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used to accept student input. See description below.</td>
<td></td>
</tr>
</tbody>
</table>

Element: <textline>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>math</td>
<td>(Optional) If set to any nonempty value, a math preview will display beneath the textbox showing well-formatted math corresponding to student input</td>
<td></td>
</tr>
<tr>
<td>Subelement</td>
<td>Description</td>
<td>Notes</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>size</td>
<td>(Optional) Defines the size in character widths of the input box as it is displayed to students.</td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>(Optional) If true, the textbox will be hidden from students.</td>
<td></td>
</tr>
</tbody>
</table>
Numerical Response

Description

The Numerical Response input type accepts a line of text input from the student and evaluates the input for correctness based on its numerical value. The input is allowed to be a number or a mathematical expression in a fixed syntax.

XML tags

optionresponse with optioninput sub element

Example Problem

What base is the decimal numeral system in?
Answer: 10

What is the value of the standard gravity constant $g$, measured in m/s$^2$? Give your answer to at least two decimal places.
Answer: 9.80665

What is the distance in the plane between the points $(\pi, 0)$ and $(0, e)$? You can type math.
Answer: 4.15435440231

Explanation

The decimal numerical system is base ten.

The standard gravity constant is defined to be precisely 9.80665 m/s2. This is 9.80 to two decimal places. Entering 9.8 also works.

By the distance formula, the distance between two points in the plane is the square root of the sum of the squares of the differences of each coordinate. Even though an exact numerical value is checked in this case, the easiest way to enter this answer is to type sqrt(pi^2+e^2) into the editor. Other answers like sqrt((pi-0)^2+(0-e)^2) also work.

Code

<problem>
  <p><b>Example Problem</b></p>
  
  <p>What base is the decimal numeral system in?</p>
  <numericalresponse answer="10">
    <textline />
  </numericalresponse>
</problem>
What is the value of the standard gravity constant \( g \), measured in m/s\(^2\)? Give your answer to at least two decimal places.

```xml
<numericalresponse answer="9.80665">
  <responseparam type="tolerance" default="0.01" />
</numericalresponse>
```

<!-- Use python script spacing. The following should not be indented! -->
```xml
<script type="loncapa/python">
computed_response = math.sqrt(math.fsum([math.pow(math.pi,2),
math.pow(math.e,2)]))
</script>
```

What is the distance in the plane between the points \((\pi, 0)\) and \((0, e)\)?

You can type math.

```xml
<numericalresponse answer="computed_response">
  <responseparam type="tolerance" default="0.0001" />
</numericalresponse>
```

```xml
<p>What is the distance in the plane between the points \((\pi, 0)\) and \((0, e)\)?
You can type math.</p>
```

```xml
<p>By the distance formula, the distance between two points in the plane is
the square root of the sum of the squares of the differences of each coordinate.
</p>
```

```xml
<p>Even though an exact numerical value is checked in this case, the easiest way to enter this answer is to type
<code>sqrt(pi^2+e^2)</code> into the editor. Other answers like <code>sqrt((pi-0)^2+(0-e)^2)</code> also work.</p>
```

```xml
</solution>
</problem>
```

**Template**

**Exact values:**

```xml
<numericalresponse answer="10">
  <textline />
</numericalresponse>
```

```xml
<solution>
  <div class="detailed-solution">
    <p>Explanation</p>
  </div>
</solution>
```

edX101: How to Create an edX Course
XML Elements and Attributes

Element: <numericalresponse>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>answer</td>
<td>A value to which student input must be equivalent. Note that this expression can be expressed in terms of a variable that is computed in a script provided in the problem by preceding the appropriate variable name with a dollar sign. Note that any numeric expression provided by the student will be automatically simplified on the grader's backend.</td>
</tr>
</tbody>
</table>

Subelement | Description |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;responseparam&gt;</td>
<td>Specifies a tolerance on the accepted values of a number. See description below.</td>
</tr>
<tr>
<td>&lt;textline&gt;</td>
<td>Used to accept student input. See description below.</td>
</tr>
</tbody>
</table>

Element: <responseparam>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>Has to be set to &quot;tolerance&quot; to signal that this parameter defines a tolerance for a numeric input</td>
</tr>
<tr>
<td>default</td>
<td>Set to either a number or a percentage, defining how different from the provided answer a student answer can be while still getting full credit</td>
</tr>
<tr>
<td>Subelement</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>-- none --</td>
<td></td>
</tr>
</tbody>
</table>

**Element: <textline>**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>math</td>
<td>(Optional) If set to any nonempty value, a math preview will display beneath the textbox showing well-formatted math corresponding to student input</td>
<td></td>
</tr>
<tr>
<td>size</td>
<td>(Optional) Defines the size in character widths of the input box as it is displayed to students.</td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>(Optional) If true, the textbox will be hidden from students.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subelement</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- none --</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Formula Response

Description

The Formula Response input type accepts a line of text representing a mathematical expression from the student and evaluates the input for equivalence to a mathematical expression provided by the grader. Correctness is based on numerical sampling of the symbolic expressions.

The answer is correct if both the student-provided response and the grader’s mathematical expression are equivalent to specified numerical tolerance, over a specified range of values for each variable.

XML tags

formularesponse with responseparam and textline sub elements

Example Problem

Write an expression for the product of R_1, R_2, and the inverse of R_3.
Answer: (R_1*R_2)/R_3

Let c denote the speed of light. What is the relativistic energy E of an object of mass m?
Answer: m*c^2

Let x be a variable, and let n be an arbitrary constant. What is the derivative of xn?
Answer: n*x^(n-1)

Explanation

Use standard arithmetic operation symbols and indicate multiplication explicitly.
Use the symbol ^ to raise to a power.
Use parentheses to specify order of operations.

Code

<problem>
  <p><b>Example Problem</b></p>
  <p>This is a short introduction to the Formula Response editor.</p>
</problem>
<p>Write an expression for the product of $R_1$, $R_2$, and the inverse of $R_3$.</p>
<formularesponse type="ci" samples="R_1,R_2,R_3@1,2,3:3,4,5#10"
answer="VoVi">
  <responseparam type="tolerance" default="0.00001"/>
  <textline size="40" math="1" />
</formularesponse>

<p>Let $c$ denote the speed of light. What is the relativistic energy $E$ of an object of mass $m$?</p>
<script type="loncapa/python">
VoVi = "(R_1*R_2)/R_3"
</script>
<formularesponse type="cs" samples="m,c@1,2:3,4#10" answer="m*c^2">
  <responseparam type="tolerance" default="0.00001"/>
  <text>$E =$</text> <textline size="40" math="1" />
</formularesponse>

<p>Let $x$ be a variable, and let $n$ be an arbitrary constant. What is the derivative of $x^n$?</p>
<script type="loncapa/python">
derivative = "n*x^(n-1)"
</script>
<formularesponse type="ci" samples="x,n@1,2:3,4#10" answer="$derivative">
  <responseparam type="tolerance" default="0.00001"/>
  <textline size="40" math="1" />
</formularesponse>

<solution>
  <div class="detailed-solution">
    <p><strong>Explanation</strong></p>
    <p>Use standard arithmetic operation symbols and indicate multiplication explicitly.</p>
    <p>Use the symbol <tt>^</tt> to raise to a power.</p>
    <p>Use parentheses to specify order of operations.</p>
  </div>
</solution>

**Template**

<problem>
  <script type="loncapa/python">
answer_value = "n*x^(n-1)"
  </script>
  <formularesponse type="ci" samples="x,n@1,2:3,4#10" answer="$answer_value">
    <responseparam type="tolerance" default="0.00001"/>
    <textline size="40" math="1" />
  </formularesponse>
  <solution>
    <div class="detailed-solution">
    </div>
  </solution>
</problem>
XML Elements and Attributes

Element: `<formularesponse>`

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>Set to either &quot;cs&quot; - Case sensitive -- the default, or &quot;ci&quot; - Case insensitive, capitalization does not matter in variable names</td>
<td></td>
</tr>
<tr>
<td>answer</td>
<td>An expression to which the student provided response must be equivalent. Note that this expression can be expressed in terms of a variable that is computed in a script provided with the problem by preceding the appropriate variable name with a dollar sign.</td>
<td>Multiplication must be explicitly stated with an asterisk. Students also must enter their input with multiplication as an asterisk. The phrase &quot;mc^2&quot; will be interpreted as a single value represented by the variable named &quot;mc&quot; raised to the second power.</td>
</tr>
</tbody>
</table>
samples

This attribute is one of the more complicated attributes in the EDXML tag system. It contains four related comma-delineated lists, separated by special delineators. The lists are in the format

<variables>@<lower_bounds>:<upper_bound>#<num_samples>

variables - a set of variables that are allowed as student input
lower_bounds - for every variable defined in variables, a lower bound on the numerical tests to use for that variable
upper_bounds - for every variable defined in variables, an upper bound on the numerical tests to use for that variable
num_samples - how many times to test the expression

<table>
<thead>
<tr>
<th>Subelement</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;responseparam&gt;</td>
<td>Defines an upper bound on the variance of the numerical methods (tolerance) used to approximate a test for equality. See description below.</td>
<td>Failure to include a tolerance leaves expressions vulnerable to unavoidable rounding errors during sampling, causing some student input to be graded as incorrect, even if it is algebraically equivalent to the grader's expression.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Element: &lt;responseparam&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
</tr>
<tr>
<td>type</td>
</tr>
</tbody>
</table>
default

Set to either a number or a percentage, defining how different from the provided answer a student answer can be while still getting full credit.

<table>
<thead>
<tr>
<th>Subelement</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- none --</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Element: `<textline>`**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>math</td>
<td>(Optional) If set to any nonempty value, a math preview will display beneath the textbox showing well-formatted math corresponding to student input.</td>
<td></td>
</tr>
<tr>
<td>size</td>
<td>(Optional) Defines the size in character widths of the input box as it is displayed to students.</td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>(Optional) If true, the textbox will be hidden from students.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subelement</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- none --</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Image Response

Description

The Image Response input type presents an image and accepts clicks on the image as an answer.

Images can be uploaded to the courseware Assets directory. Response clicks are marked as correct if they are within a certain specified sub rectangle of the image canvas.

XML tags

imageresponse with specified src and rectangle attributes

Example Problem

You are given three shapes. Click on the triangle.

Pressing check after clicking within the triangle displays a green check mark.

Pressing check without having properly clicked within the triangle causes a red x to be displayed.

Code

<problem>
  <p><b>Example Problem</b></p>
  <startouttext/>
  <p>You are given three shapes. Click on the triangle.</p>
  <imageresponse>
    <imageinput src="/c4x/edX/edX101/asset/threeshares.png" width="220" height="150" rectangle="(80,40)-(130,90)" />
  </imageresponse>
  <startouttext/>
  <p>*Note: Firefox is currently not supported</p>
  <endouttext/>
</problem>
Template

<problem>
    <imageresponse>
        <imageinput src="Path_to_Imag..." width="220" height="150"
          rectangle="(80,40)-(130,90)" />
    </imageresponse>
</problem>

XML Elements and Attributes

Element:  <imageresponse>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- none --</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Subelement | Description | Notes |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;imageinput&gt;</td>
<td>Used to get input based off of an image. See description below.</td>
<td></td>
</tr>
</tbody>
</table>

Element:  <imageinput>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>src</td>
<td>The url from which the image should be loaded.</td>
<td></td>
</tr>
<tr>
<td>height</td>
<td>The height in pixels that the image should be displayed with.</td>
<td></td>
</tr>
<tr>
<td>width</td>
<td>The width in pixels that the image should be displayed with.</td>
<td></td>
</tr>
</tbody>
</table>
rectangle

<table>
<thead>
<tr>
<th>Subelement</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- none --</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Custom Response: Script Evaluation

Description

A Custom Response input type accepts one or more lines of text input from the student. The response is evaluated for correctness using an embedded python script.

XML tags

customresponse, textinput sub element with optional correct_answer attribute, and python script.

Example Problem

This question consists of two parts.
First, enter two integers which sum to 10.
Answer: see Explanation below

Now enter two (finite) decimals which sum to 20.
Answer: see Explanation below

Explanation

For the first part, any two numbers of the form $n$ and $10-n$, where $n$ is any integer, will work. One possible answer would be the pair 0 and 10.

For the second part, any pair $x$ and $20-x$ will work, where $x$ is any real number with a finite decimal representation. Both inputs have to be entered either in standard decimal notation or in scientific exponential notation. One possible answer would be the pair 0.5 and 19.5. Another way to write this would be $5e-1$ and $1.95e1$.

Code

```python
<problem>
  <p><b>Example Problem</b></p>
  <script type="loncapa/python">
    def test_add_to_ten(expect,ans):
      try:
        ...
  </script>
</problem>
```
This question consists of two parts. First, enter two integers which sum to 10. Now enter two (finite) decimals which sum to 20.

For the first part, any two numbers of the form $n$ and $10-n$, where $n$ is any integer, will work. One possible answer would be the pair 0 and 10.

For the second part, any pair $x$ and $20-x$ will work, where $x$ is any real number with a finite decimal representation. Both inputs have to be entered either in standard decimal notation or in scientific exponential notation. One possible answer would be the pair 0.5 and 19.5. Another way to write this would be $5e-1$ and $1.95e1$.

Template

With displayed suggested correct answers:

```python
def test_add(expect, ans):
    try:
        a1 = float(ans[0])
        a2 = float(ans[1])
    except ValueError:
        a1 = 0
        a2 = 0
    return (a1 + a2) == float(expect)
</script>
<p>This question consists of two parts. First, enter two integers which sum to 10. Now enter two (finite) decimals which sum to 20.</p>
</customresponse>

<solution>
<div class="detailed-solution">
<p>Explanation</p>
<p>For the first part, any two numbers of the form $n$ and $10-n$, where $n$ is any integer, will work. One possible answer would be the pair 0 and 10.</p>
<p>For the second part, any pair $x$ and $20-x$ will work, where $x$ is any real number with a finite decimal representation. Both inputs have to be entered either in standard decimal notation or in scientific exponential notation. One possible answer would be the pair 0.5 and 19.5. Another way to write this would be $5e-1$ and $1.95e1$.</p>
</div>
</solution>
</problem>

edX101: How to Create an edX Course
return (a1+a2)== float(expect)
</script>

<p>Enter two real numbers which sum to 20: </p>
<customresponse cfn="test_add" expect="20">
  <textline size="40" correct_answer="11"><br/>
  <textline size="40" correct_answer="9"/>
</customresponse>

    <div class="detailed-solution">
    </div>
</solution>
</problem>

With no suggested correct answers:

<problem>
<script type="loncapa/python">
def test_add(expect,ans):
    a1=float(ans[0])
    a2=float(ans[1])
    return (a1+a2)== float(expect)
</script>

<p>Enter two real numbers which sum to 20: </p>
<customresponse cfn="test_add" expect="20">
  <textline size="40" />\br/>
  <textline size="40"/>
</customresponse>

    <div class="detailed-solution">
    </div>
</solution>
</problem>

**XML Elements and Attributes**

**Element:  <customresponse>**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
</table>

edX101: How to Create an edX Course
### cfnc

The function name defined in the problem script that should be used to grade this problem. The function should accept two parameters. The first parameter accepts the string defined in the 'expect' attribute, defined below. The second parameter accepts an array of strings provided by the student in the child input fields.

### expect

A string provided to the custom grading function. This gives a way to re-use common functions across multiple responses.

### Subelement Description Notes

<table>
<thead>
<tr>
<th>All standard HTML tags</th>
<th>Display text relevant to the problem</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;textline&gt;</td>
<td>Used to provide a small input field that accepts a line of text.</td>
<td></td>
</tr>
<tr>
<td>&lt;textbox&gt;</td>
<td>Used to provide a large input field (a text box) that accepts a paragraph of text.</td>
<td></td>
</tr>
</tbody>
</table>

### Element: <textline>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>size</td>
<td>The width in characters that this field should be displayed with</td>
<td></td>
</tr>
<tr>
<td>correct_answer</td>
<td>The answer to display next to the input field when Show Answer is clicked.</td>
<td></td>
</tr>
</tbody>
</table>

### Element: <textbox>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subelement</td>
<td>Description</td>
<td>Notes</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>correct_answer</td>
<td>The answer to display next to the input field when Show Answer is clicked.</td>
<td></td>
</tr>
<tr>
<td>-- none --</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Custom Response: Chemical Equations

Description
The Chemical Equation input is a specialization of the Custom Response type that allows the student to enter chemical equations as answers.

XML tags
<customresponse> with a <chemicalequationinput/> sub element

Example Problem
Some problems may ask for a particular chemical equation. Practice by writing out the following reaction in the box below.

H₂SO₄ → H⁺ + HSO₄⁻

Some tips:
- Only real element symbols are permitted.
- Subscripts are entered with plain text.
- Superscripts are indicated with a caret (^).
- The reaction arrow (→) is indicated with "->".

So, you can enter "H₂SO₄ -> H⁺ + HSO₄⁻".

Code
<pre>
<problem>
   <p><b>Example Problem</b></p>
   <startouttext/>
   <p>Some problems may ask for a particular chemical equation. Practice by writing out the following reaction in the box below.</p>
   <center>(text{H}_2text{SO}_4longrightarrow text{ H}^+ + text{ HSO}_4^-)</center>
   <br/>
   <customresponse>
      <chemicalequationinput size="50"/>
</customresponse>
</startouttext/>
</problem>
</pre>
<answer type="loncapa/python">

```python
if chemcalc.chemical_equations_equal(submission[0], 'H2SO4 -> H^+ + HSO4^-'):
    correct = ['correct']
else:
    correct = ['incorrect']

</answer>
</customresponse>

<p>Some tips:<ul><li>Only real element symbols are permitted.</li><li>Subscripts are entered with plain text.</li><li>Superscripts are indicated with a caret (^).</li><li>The reaction arrow (\(\rightarrow\)) is indicated with "->".</li></ul>So, you can enter "H2SO4 -> H^+ + HSO4^-".</p>
</problem>
Schematic Response: Circuit Schematic

Description

The Schematic Response input type provides an interactive grid on which the student can construct a schematic answer like a circuit.

XML tags

<schematicresponse> with attributes and python script

Example Problem

Make a voltage divider that splits the provided voltage evenly.

Make a high pass filter.
Explanation

A voltage divider that evenly divides the input voltage can be formed with two identically valued resistors, with the sampled voltage taken in between the two.

A simple high-pass filter without any further constraints can be formed by simply putting a resistor in series with a capacitor. The actual values of the components do not really matter in order to meet the constraints of the problem.

Code

<problem>
Make a voltage divider that splits the provided voltage evenly.
</problem>
<answer type="loncapa/python">
dc_value = "dc analysis not found"
for response in submission[0]:
    if response[0] == 'dc':
        for node in response[1:]:
            dc_value = node['output']

if dc_value == .5:
    correct = ['correct']
else:
    correct = ['incorrect']
</answer>

<answer type="loncapa/python">
ac_values = None
for response in submission[0]:
    if response[0] == 'ac':
        for node in response[1:]:
            ac_values = node['NodeA']
print "the ac analysis value:", ac_values
if ac_values == None:
    correct = ['incorrect']
elif ac_values[0][1] < ac_values[1][1]:
    correct = ['correct']
else:
    correct = ['incorrect']
</answer>

<solution>
<div class="detailed-solution">
<p>Explanation</p>
</div>
</solution>
A voltage divider that evenly divides the input voltage can be formed with two identically valued resistors, with the sampled voltage taken in between the two.

A simple high-pass filter without any further constraints can be formed by simply putting a resistor in series with a capacitor. The actual values of the components do not really matter in order to meet the constraints of the problem.
edX Glossary
edX Glossary

Below is an alphabetized list of edX terminology.

**About page**

An About page is a webpage introducing a website. Below are some examples. Note that if you are logged into one of the websites below, you may need to log out in order to avoid getting automatically redirected to your dashboard.

- edX
- twitter
- WordPress
### advanced editor interface
When you edit your course materials in Studio, you are provided with the option of using either the simple editor or the advanced editor. The advanced editor allows you to include custom HTML styling, while the simple editor provides a graphical user interface with formatting buttons.

![Advanced Editor Interface](image)

### Agile

<table>
<thead>
<tr>
<th>Agile</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Analytics</strong></td>
</tr>
<tr>
<td><strong>applet</strong></td>
</tr>
<tr>
<td><strong>assets</strong></td>
</tr>
<tr>
<td><strong>application programming interface (API)</strong></td>
</tr>
</tbody>
</table>
Assignment Type Name

In Studio, you specify assignment types such as homework, exams, exercises, etc. Every assignment of a given type contributes the same amount to the final grade. Each assignment type needs to be labeled with an Assignment Type Name.

Atlassian

Atlassian, Inc. is an Australian enterprise software company that makes products geared towards software developers and project managers. It is best known for its issue-tracking application, JIRA, and its team collaboration product, Confluence.

attribute

In computing, an attribute is a specification that defines a property of an object, element, or file. It may also refer to or set the specific value for a given instance of such. For example, in computer graphics, line objects can have attributes such as thickness (with real values), color (with descriptive values such as brown or green or values defined in a certain color model, such as RGB), dashing attributes, etc. A circle object can be defined in similar attributes plus an origin and radius.

In XML, an attribute is a markup construct consisting of a name/value pair that exists within a start-tag or empty-element tag. Markup languages, such as HTML and XML, use attributes to describe data and the formatting of data. [Source: Wikipedia].

behavior-driven development (BDD)

Software development process that uses user stories to focus on "desired behavior," or requirements set by intended users. An example of a user story is "As a <role> I want <feature> so that <benefit>"—e.g., "As an instructor I want a tool that I can use to put courses online so that I can reach a wider audience." (Source: Adapted from Wikipedia)
<table>
<thead>
<tr>
<th>beta tester</th>
<th>A staff member who makes sure that a product is error-free. In the context of edX courses, this includes doing exercises and making sure the courseware behavior is as desired, checking over videos (for example to make sure that the different speeds are labeled correctly), copy editing, and verifying accuracy of content.</th>
</tr>
</thead>
</table>
| blade | This is an edX-invented term. It refers to a custom-made interactive or other plug-in designed for embedding into a course for educational purposes. Some examples of blades are as follows:

MIT’s chemistry course used an interactive periodic table tool (all of the elements can be clicked to show more details and the coloring of the table can be set to reflect different properties of elements)

MIT’s circuits and electronics course use an interactive circuit kit, which allows students to work with a virtual pegboard, placing circuit elements like resistors and capacitors.

Harvard’s course “Justice” uses a polling tool, where students can vote on what they thought was the correct answer to an ethical question, and depending on what they answered, they would be sent to a specific discussion thread.

CS50 used their own video player, in lieu of the edX one. |
<p>| branch | Segment of code that can be checked out from a master project, downloaded into a “local dev” (local development environment), changed, and re-uploaded into the master project. |
| certificates | Most edX courses offer students who achieve a grade above a certain threshold a certificate, sometimes called a certificate of mastery. |
| change log | A change log is a log or record of changes made to a project, such as a website or software project, usually including such records as bug fixes, new features, etc. Some open source projects include a changelog as one of the top level files in their distribution. [Source: Wikipedia] |</p>
<table>
<thead>
<tr>
<th><strong>checkbox problem</strong></th>
<th>The checkbox problem is a type of question professors can ask in their course that allows the student to select one or more choices from a collection of answer choices, presented as a list of checkboxes.</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Checkbox Response" /></td>
<td>How many correct responses can a Checkbox Response question have?</td>
</tr>
<tr>
<td>- Zero</td>
<td></td>
</tr>
<tr>
<td>- One</td>
<td></td>
</tr>
<tr>
<td>- Two or more</td>
<td></td>
</tr>
<tr>
<td>- Nobody knows</td>
<td></td>
</tr>
<tr>
<td>- Somebody might know :)</td>
<td></td>
</tr>
<tr>
<td><strong>chemical equation response problem</strong></td>
<td>The Chemical Equation Response problem type is a kind of Custom Response problem type that allows the student to enter chemical equations as answers.</td>
</tr>
<tr>
<td><img src="image" alt="Chemical Equation" /></td>
<td>Fe₂O₃ + 2Al -&gt; Al₂O₃ + 2Fe</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Fe₂O₃ + 2Al → Al₂O₃ + 2Fe</td>
<td></td>
</tr>
<tr>
<td><strong>CMS</strong></td>
<td>Course Management System. The system that an instructor uses to build an online course. EdX Studio is a CMS. The counterpart to the CMS is the LMS (Learning Management System), which is what the student sees when they log in. Studio is edX’s CMS. Also CAS.</td>
</tr>
<tr>
<td><strong>COB</strong></td>
<td>Close of business. Example: Please have your status report to me by COB tomorrow.</td>
</tr>
<tr>
<td><strong>cohort</strong></td>
<td>A cohort is a group of people banded together or treated as a group.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>In HarvardX’s Greek heroes course, the students are subdivided into cohorts of about 1000 so that they can have very focused discussions.</td>
<td></td>
</tr>
<tr>
<td><strong>commit</strong></td>
<td>In software programming, to commit is a form of version control for source code. A commit refers to submitting the latest changes of the source code to the repository and making those changes part of the top-level revision history in the repository. When other users do an UPDATE or a checkout from the repository, they will receive the latest committed version, unless they specify otherwise. A commit in version control systems allows easy rollback to a previous version, even after the commit has been done. (Source: Adapted from Wikipedia)</td>
</tr>
<tr>
<td><strong>community TA</strong></td>
<td>Employed in most discussion forums to help moderate the forum and answer questions. Good community TAs are students who have been actively participating in and contributing to class discussions.</td>
</tr>
<tr>
<td>component</td>
<td>In <strong>Studio</strong>, the hierarchy of the courseware is broken down into sections, subsections, units, and components. Sections and subsections are visible in the left-hand navigation bar; units are pages that are accessed using the ribbon that appears at the top of the page when a user clicks on a subsection. Finally, components are pieces that make up units.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><img src="image1.png" alt="Diagram" /></td>
<td><img src="image2.png" alt="Diagram" /></td>
</tr>
<tr>
<td>course accordion</td>
<td>The course accordion is the left-hand navigation bar that appears in the courseware. It shows the sections and subsections for a course. The word “accordion” is intended to invoke the folding and unfolding of an accordion, because when you click on a section, the section opens up.</td>
</tr>
<tr>
<td><img src="image3.png" alt="Diagram" /></td>
<td><img src="image4.png" alt="Diagram" /></td>
</tr>
<tr>
<td>course catalog page</td>
<td>The page that contains the list of all courses offered by an online education web portal at any given time together with brief descriptions. EdX's course catalog page is <a href="https://example.com">here</a>.</td>
</tr>
<tr>
<td><strong>Course Handouts sidebar</strong></td>
<td>Area on the right side of the <strong>Course Info</strong> page that lists handouts and other materials that an instructor has posted for a class. Some handouts include a syllabus, forum guidelines, or other material that is important but that students will not need to access frequently.</td>
</tr>
<tr>
<td><strong>Course Info page</strong></td>
<td>The Course Information Page is accessible from the top navigation bar in an edX course. It is the landing page for each course and contains updates and announcements, as well as handouts for the course. Occasionally, the phrase “Course Info page” is mistakenly used to describe the <strong>About page</strong>.</td>
</tr>
<tr>
<td><strong>course ribbon</strong></td>
<td>The bar above the lesson content on edX or Edge that lists the units and components in a subsection.</td>
</tr>
<tr>
<td><strong>courseware</strong></td>
<td>The courseware page of an edX course is where the primary instructional materials in each course are accessed. Generally, it contains all lectures, homework assignments, tutorials and exams.</td>
</tr>
<tr>
<td><strong>custom pages</strong></td>
<td>Additional pages that supplement the courseware for a course. Course authors have used them to share a syllabus, calendar, handouts, and more. Custom pages appear on the navigation bar of a course along with the standard pages (including <strong>Courseware</strong> and <strong>Course Info</strong>). Also called <strong>static pages</strong>.</td>
</tr>
<tr>
<td><strong>custom response problem</strong></td>
<td>The edX platform provides instructors with the ability to write their own custom grader using a Python script. This type of problem is sometimes described as a Custom Response Problem; however, because this name is unclear, these problems have been renamed “write-your-own-grader” problems.</td>
</tr>
<tr>
<td><strong>discussion forum</strong></td>
<td>This is the page in each course where students can communicate with each other and staff by typing in questions and responding to each other. Sometimes called the Forum.</td>
</tr>
</tbody>
</table>
**Discussion component**
The edX platform allows instructors to embed discussion areas into the courseware. This means, for instance, that a course could have a discussion area immediately below a video, and the video could prompt students to comment about a specific issue that was discussed without having to leave the page and go to the discussion forum.

**Django**
A software framework, written in Python, that is designed to support the development of dynamic websites, web applications, web services and web resources. The framework aims to alleviate the overhead associated with common activities performed in web development. For example, many frameworks provide libraries for database access, templating frameworks and session management, and they often promote code reuse. (Source: Wikipedia)

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of "don't repeat yourself." (Source: Wikipedia)

**edX101**
EdX101 is edX's online course about how to create online courses. The intended audience is faculty and university administrators, but the course is used in a variety of ways, including internal training. It is also the first course ever to have been developed entirely using edX Studio.

**edX Edge**
When a course author creates a course in edX Studio (accessible at studio.edge.edX.com), the finished product appears for students to see on edX Edge (edge.edX.com). Edge is an LMS. See also Studio.

**edX Platform**

**edX Studio**
edX Studio is edX's course authoring tool. It allows instructors to build an online course. See also edX Edge.

**edXML**

**epic**
(As it relates to kanban; see also story)

**exercises**
Practice or practical exercises interspersed in edX course content to keep the learner engaged. Exercises are also an important measure of teaching effectiveness and learner comprehension.

**export**

**Fast Start**

**finger exercises**

**forum**
This is the page in each course where students can communicate with each other and staff by typing in questions and responding to each other. Also called the Discussion Forum.
<table>
<thead>
<tr>
<th>term</th>
<th>definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>forum moderator</td>
<td>Member of the course staff who periodically reviews the discussion forum, watching for inappropriate content (such as threats or cheating) and intervening as necessary.</td>
</tr>
<tr>
<td>Formula Response problem</td>
<td>The Formula Response problem type accepts a line of text representing a mathematical expression from the student and evaluates the input for equivalence to a mathematical expression provided by the grader. Correctness is based on numerical sampling of the symbolic expressions.</td>
</tr>
<tr>
<td>Getting to Done</td>
<td>Effort to help those creating courses in Studio understand, at various points in the process, exactly what needs to happen to prepare a course for release and how far along they are in that process. See Getting to Done / The Thin Red Line on the wiki.</td>
</tr>
<tr>
<td>GitHub</td>
<td>GitHub is a web-based hosting service for software development projects. Github stores software files in repositories, and enables programming teams to simultaneously work on a project without having to worry too much about writing over each other’s code by accident. In the early days (before Studio was released), edX course teams used github to write courseware. The edX open source platform code will be able to be accessed from github.</td>
</tr>
<tr>
<td>grade brackets</td>
<td><img src="image" alt="Overall Grade Range" /> Your overall grading scale for student final grades.</td>
</tr>
<tr>
<td>grading rubric</td>
<td>In education terminology, a grading or scoring rubric means “a standard of performance for a defined population”. [1] [Source: Wikipedia]</td>
</tr>
<tr>
<td>help sock</td>
<td>Area right above the page footer in Studio where links to Help documentation appear.</td>
</tr>
<tr>
<td>HipChat</td>
<td>Instant messaging system used at edX that includes chat rooms.</td>
</tr>
<tr>
<td>hypertext markup language (HTML)</td>
<td>The main markup language for creating web pages and other information that can be displayed in a web browser. [Source: Wikipedia]</td>
</tr>
<tr>
<td>HTML Component</td>
<td>In Studio an HTML component is a component whose content is limited to HTML code. An HTML component cannot contain an interactive problem, a discussion space, or the edX video player.</td>
</tr>
<tr>
<td>HTML component editor</td>
<td></td>
</tr>
</tbody>
</table>

edX101: How to Create an edX Course
### HTML view
A view in the editor that opens when a user creates a new HTML component. The default is Visual view. In HTML view, the user enters and edits raw HTML text directly.

The following is an example of HTML view. To see the same content in Visual view, see [Visual view](#).

```
<p>Text in Visual view can be formatted using the buttons above. For example:</p>
<p>This is a heading.</p>
<p>This is normal text.</p>
<p>This is bold text.</p>
<p>This line contains a link to https://example.com.</p>
<p>Paste the following code into the HTML tab to open a new window with the link in a new window.</p>

```html
<link href="https://example.com" rel="alternate" type="application/rss+xml" title="Example RSS Feed" />
```

### Image Response problem
An Image Response problem type presents an image and allows the student to click on the image as an answer.

### import
edX [Studio](#) provides course authors with the ability to write a course, export the course, and import the course into another course URL. This enables authors to create multiple customized versions of the same course.

### JSON file
File type that edX software uses to read transcripts. To turn an .srt file into an srt.sjson file, a Python script must be run.

### Leankit Kanban
A software tool for visually mapping and storyboarding projects and progress. Leankit Kanban is used to track and collaboratively manage workloads for project teams. Leankit’s Kanban tool can be modified to the work processes of a group.

### LMS - Learning Management System
This term is generally used to refer to what the student sees when they look at a course on edX. The counterpart is the CMS (Course Management System), which is the system that the instructor uses to build an online course. ([http://en.wikipedia.org/wiki/Learning_management_system](http://en.wikipedia.org/wiki/Learning_management_system))

### landing page
Page that opens when a user enters [http://studio.edge.edx.org](http://studio.edge.edx.org) into a browser. (See also soft landing page)
<table>
<thead>
<tr>
<th><strong>LaTeX</strong></th>
<th>A <strong>document markup language</strong> and <strong>document preparation system</strong> for the <strong>TeX typesetting</strong> program. The term LaTeX refers only to the language in which documents are written, not to the editor used to write those documents. In order to create a document in LaTeX, a .tex file must be created using some form of <strong>text editor</strong>. While most text editors can be used to create a LaTeX document, a <strong>number of editors</strong> have been created specifically for working with LaTeX. LaTeX is widely used in <strong>academia</strong>. [Source: Wikipedia]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LaTeX:</strong></td>
<td>The kinetic energy of an object is described by the equation $KE = \frac{1}{2} \text{ mass } \times \text{velocity}^2$.</td>
</tr>
<tr>
<td><strong>Displays:</strong></td>
<td>The kinetic energy of an object is described by the equation $KE = 1/2 \times \text{mass } \times \text{velocity}^2$.</td>
</tr>
<tr>
<td><strong>learning sequence</strong></td>
<td>Learning sequence is an edX-invented term. A learning sequence is the online counterpart of a lecture. It consists of interleaved video segments, interactives, problems, pages of text, and other educational resources. Now obsolete; use “lesson” instead.</td>
</tr>
<tr>
<td><strong>lettuce</strong></td>
<td>BDD tool that creates and runs automated tests for Python projects from plain-text descriptions of the functions or &quot;jobs to be done&quot; that the Python code should perform (e.g., &quot;Given a number, what is its factorial?&quot;). It allows non-programmers to describe the behavior of a certain system. (Source: Adapted from lettuce.it)</td>
</tr>
<tr>
<td><strong>Lighthouse</strong></td>
<td>Bug tracker that is connected to Tender (the bug and suggestion tracking system for Studio). A bug on Tender may be elevated to Lighthouse.</td>
</tr>
<tr>
<td><strong>Live mode</strong></td>
<td>One way to view your course on edX or Edge (the other is Preview mode). When you use Live mode, you see all the units of your course that are set to Public, regardless of whether the release dates have passed. You do not see content that is set to Private. If you have created a new draft of a previously published unit, you see the currently published version of the unit instead of the draft.</td>
</tr>
<tr>
<td><strong>local dev</strong></td>
<td>Abbreviation for “local development environment.” Large-scale software projects have a main code repository where all developers can access the current code base. A local development environment is a copy of the code base that an individual developer has on her own machine. When the developer writes new code, she uses her local dev to compile and view her code, make bug fixes, and add features without having to change the code in the main repository. She then makes a pull request, which is a request for her changes to be uploaded into the main repository. After another developer reviews her code, if it has no problems, the developer accepts the pull request, and her changes are merged into the main code base.</td>
</tr>
<tr>
<td><strong>location ID</strong></td>
<td>Reference code assigned by the edX software to an object. Not seen by the end user.</td>
</tr>
</tbody>
</table>
| **LON-CAPA** | E-learning platform, also known as a Course Management System (CMS) or Learning Management System (LMS). It possesses the standard features of many learning platforms (user roles, calendar, e-mail, chat rooms, blogs, resource construction, test grading, etc.), but it differs from traditional e-learning platforms in that its many web servers (in various parts of the world) can communicate with each other.

The LON-CAPA network allows participating universities and schools to create learning resources (test problems, web pages, etc.) and to share them across the network. This collaborative aspect is LON-CAPA’s most distinctive feature.

[Source: Wikipedia] |
<p>| <strong>long course description</strong> | Every edX course has an About page. The long course description is a couple of paragraphs (around 120 words) of text describing what the course is about. |</p>
<table>
<thead>
<tr>
<th><strong>Mathjax</strong></th>
<th>A system for embedding LaTeX equations into an HTML document.</th>
</tr>
</thead>
</table>
| **Mathjax:** | ![Mathjax](mathjax)
\text{E}_{I-F} = \sqrt{E_{I-I} \times E_{F-F} + 96.2(x_I - x_F)^2} |

| **meetup** | Generally, a group of people with shared interests that form offline clubs in local communities around the world using meetup.com. EdX learners have formed edX meetups. |
| **moderator** | See [forum moderator](http://example.com). |
| **mongo DB** | An open source document-oriented database system developed and supported by 10gen. It is part of the NoSQL family of database systems. Instead of storing data in tables as is done in a "classical" relational database, MongoDB stores structured data as JSON-like documents with dynamic schemas (MongoDB calls the format BSON), making the integration of data in certain types of applications easier and faster. [Source: Wikipedia] |
| **MOOC** | Massive Open Online Course. Used to describe online courses that are open to anyone and have no size limitations in terms of participation. |
| **multiple choice problem** | The multiple choice problem allows the student to select at most one choice from a collection of answer choices, presented as a list of radio buttons. |
**navigation bar**
In Studio, the navigation bar contains the logo and menus at the top of the screen.

On edX or Edge, the navigation bar is the list of tabs at the top of the screen.

**numerical response problem**
The numerical response problem type accepts a line of text input from the student and evaluates the input for correctness based on its numerical value. The input can be a number or a mathematical expression in a fixed syntax.

![Numerical Response Example](image)

**OOO**
Out Of Office

**option response problem**
The option response problem type allows the student to choose from a collection of answer options presented as a drop-down list.

![Option Response Example](image)

**page chrome**
Basic structure or framework of a page, including navigation bars, text fields, and other areas that do not change from page to page.

**Pivotal Tracker**
Bug tracking system related to Lighthouse.

**Preview mode**
One way to view your course on edX or Edge (the other is Live mode). When you use Preview mode, you see all the units of your course, regardless of whether they are set to Public or Private and regardless of whether the release dates have passed. If you have created but not published a new draft of a unit, you see the draft of the content instead of the currently published version.

Using Preview mode is the only way to see content that is set to Private as a student would see it.

**private unit**
A private unit is a unit whose visibility settings have been set to Private. These settings determine whether units are visible to students.
<table>
<thead>
<tr>
<th><strong>Problem component</strong></th>
<th>Most courses include homework or exercise problems. Problem component refers to each such problem you create in your course. There are different kinds of Problem components, such as multiple choice, numerical response, etc.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Production (server)</strong></td>
<td>Server that hosts the websites that are world-viewable. Content starts in Sandbox, where it is developed. It then moves to Staging for final testing, and is only “pushed” to Production (or Prod) when it is believed to be ready for the public.</td>
</tr>
<tr>
<td><strong>Production Stage</strong></td>
<td></td>
</tr>
<tr>
<td><strong>program manager</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Progress page</strong></td>
<td>Every edX course has a navigation bar across the top listing the course pages. All courses have the following pages: Courseware, Course Info, Wiki, Discussion, and Progress. (Some courses have additional pages as well). The progress page contains information about a student’s grades.</td>
</tr>
<tr>
<td><strong>prototyping</strong></td>
<td></td>
</tr>
<tr>
<td><strong>public unit</strong></td>
<td>A public unit is a unit whose visibility settings have been sent to Public. These settings determine whether units are visible to students.</td>
</tr>
<tr>
<td><strong>pull request</strong></td>
<td></td>
</tr>
<tr>
<td><strong>push</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Python</strong></td>
<td>A general-purpose, high-level computer programming language that is free and open source software. [Source: Wikipedia] Almost all of the edX platform is written in Python.</td>
</tr>
<tr>
<td><strong>QA</strong></td>
<td>Quality Assurance. Generally used in the context of &quot;QA testing,&quot; i.e. checking course materials for errors.</td>
</tr>
<tr>
<td><strong>radio button</strong></td>
<td>A radio button or option button is a type of graphical user interface element that allows the user to choose only one of a predefined set of options.</td>
</tr>
</tbody>
</table>

---

**edX101: How to Create an edX Course**
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>refactoring</td>
<td>A disciplined technique for restructuring an existing body of code, altering its internal structure without changing its external behavior, undertaken in order to improve some of the nonfunctional attributes of the software. Advantages include improved code readability and reduced complexity to improve the maintainability of the source code, as well as a more expressive internal architecture or object model to improve extensibility. (Source: Wikipedia)</td>
</tr>
<tr>
<td>relative directory</td>
<td></td>
</tr>
<tr>
<td>repository/repo</td>
<td>Online storage space for a hierarchy and files that are associated with a GitHub project.</td>
</tr>
<tr>
<td>research and evaluation</td>
<td></td>
</tr>
<tr>
<td>rubric</td>
<td>Short for &quot;scoring rubric.&quot;</td>
</tr>
<tr>
<td>In education terminology, scoring rubric means &quot;a standard of performance for a defined population&quot;. A scoring rubric is an attempt to communicate expectations of quality around a task. In many cases, scoring rubrics are used to delineate consistent criteria for grading. Because the criteria are public, a scoring rubric allows teachers and students alike to evaluate criteria, which can be complex and subjective. A scoring rubric can also provide a basis for self-evaluation, reflection, and peer review. It is aimed at accurate and fair assessment, fostering understanding, and indicating a way to proceed with subsequent learning/teaching. (Source: Wikipedia)</td>
<td></td>
</tr>
<tr>
<td>sandbox</td>
<td>A environment used for testing software or content before release. A sandbox server is a faithful copy of the live edX system, but with newer content or software loaded onto it. You may hear people talk about “pushing to sandbox” meaning to load pre-release content or software onto a sandbox server. A safe execution environment for untrusted snippets of code in a production environment. Content starts in Sandbox, where it is developed. It then moves to Staging for final testing, and is only “pushed” to Production (or Prod) when it is believed to be ready for the public.</td>
</tr>
<tr>
<td>schematics response problem</td>
<td>The Schematic Response problem type allows the student to construct a schematic answer (such as an electronics circuit) on an interactive grid.</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>screenshot</td>
<td>An <em>image</em> taken by the <em>computer</em> user to record the visible items displayed on the <em>monitor</em>, television, or other visual output device. [Source: Wikipedia]</td>
</tr>
<tr>
<td>Scrum</td>
<td><em>Scrum</em> is a way for teams to work together to develop a product. Product development, using Scrum, occurs in small pieces, with each piece building upon previously created pieces. Building products one small piece at a time encourages creativity and enables teams to respond to feedback and change, to build exactly and only what is needed. [Source: Scrum.org]</td>
</tr>
<tr>
<td>section</td>
<td>In <em>Studio</em>, the hierarchy of the courseware is broken down into sections, subsections, units, and components. Sections are visible in the left-hand navigation bar; they are pieces that make up courses.</td>
</tr>
<tr>
<td>sequentials</td>
<td>Studio divides the hierarchy of the LMS into sections, subsections, units, and components. In the XML, sections are chapters, subsections are sequentials, units are verticals, and components are either videos, HTMLs, problems, or discussions.</td>
</tr>
<tr>
<td><strong>settings</strong></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td></td>
</tr>
<tr>
<td><img src="image.png" alt="image" /></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>simple editor interface</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>One of two of Studio’s editor interfaces. The Simple Editor interface allows you to edit problems visually, without having to work in XML.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>SLA</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>A <a href="#">Service-Level Agreement</a> is a part of a <a href="#">service contract</a> where a service is formally defined.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>soft landing page</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>(See also <a href="#">landing page</a>)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>SPOCs</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Small Private Online Courses. The opposite of a Massive Open Online Course (MOOC). edX works with a number of community colleges to customize existing MOOCs and run small private versions on campus.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>sprint</strong></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>SRT file</strong></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>Staging</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Server that hosts the test versions of the websites that will eventually be released to the public. Content starts in Sandbox, where it is developed. It then moves to Staging for final testing, and is only “pushed” to Production (or Prod) when it is believed to be ready for the public.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>static pages</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional pages that supplement the courseware for a course. Course authors have used them to share a syllabus, calendar, handouts, and more. Static pages appear on the navigation bar of a course along with the standard pages (including <a href="#">Courseware</a> and <a href="#">Course Info</a>). Also called <a href="#">custom pages</a>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>story</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>(As it relates to kanban; see also <a href="#">epic</a>)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Studio</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>See edX Studio.</td>
</tr>
<tr>
<td>string response problem</td>
</tr>
<tr>
<td>--------------------------</td>
</tr>
<tr>
<td>subsection</td>
</tr>
<tr>
<td>syllabus</td>
</tr>
<tr>
<td>short course description</td>
</tr>
<tr>
<td>transcripts</td>
</tr>
<tr>
<td>unit</td>
</tr>
<tr>
<td>URL</td>
</tr>
<tr>
<td>UT-GMT, UTC, UT1</td>
</tr>
</tbody>
</table>
**versioning**

**Software versioning** is the process of assigning either unique **version names** or unique **version numbers** to unique states of **computer software**. Within a given version number category (major, minor), these numbers are generally assigned in increasing order and correspond to new developments in the software. At a fine-grained level, **revision control** is often used for keeping track of incrementally different versions of electronic information, whether or not this information is actually computer software. [Source: Wikipedia]

**vertical**

Studio divides the hierarchy of the LMS into sections, subsections, units, and components. In the XML, sections are chapters, subsections are sequentials, units are verticals, and components are either videos, HTMLs, problems, or discussions.

<table>
<thead>
<tr>
<th>Sections</th>
<th>Units</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subsections</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**video**

Video is one type of Media Production. It involves pre-production, production and post-production.

**Video component**

Your course will include video snippets of your lectures or tutorials. Each one is called a Video component.
**Visual view**

Visual view is the default view for the editor that opens when a user creates a new HTML component in Studio. In Visual view, users can enter formatted text without adding HTML tags. The other available view is [HTML view](#).

The following is an example of Visual view. To see the same content in HTML view, see [HTML view](#).

```
<table>
<thead>
<tr>
<th><strong>Text in Visual view can be formatted using the buttons above. For example:</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>THIS IS A HEADING.</strong></td>
</tr>
<tr>
<td>This is normal text.</td>
</tr>
<tr>
<td><strong>This is bold text.</strong></td>
</tr>
<tr>
<td>This line contains a link that opens in a new window.</td>
</tr>
<tr>
<td>* This is an item in a bulleted list.</td>
</tr>
</tbody>
</table>
```

**wiki**

A wiki is a website which allows its users to add, modify, or delete its content via a web browser usually using a simplified markup language or a rich-text editor. Wikis serve many different purposes, such as knowledge management and notetaking.

Each edX course features its own wiki, which can be accessed from the top navigation bar. Students can use the wiki to share links, notes, and other helpful information with each other. They can also use it to create a running list of typos and errors in the course.

**wireframe**

**XBlock**

edX’s component architecture for writing courseware components. Third parties can create components as small isolated web applications that can run within the edX LMS.
Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. [Source: Wikipedia]