
Building and Running an edX Course

Release

May 07, 2014

Contents

Read Me

The edX *Building a Course with edX Studio* documentation is created using [RST](#) files and [Sphinx](#). You, the user community, can help update and revise this documentation project on GitHub:

https://github.com/edx/edx-platform/tree/master/docs/course_authors/source

To suggest a revision, fork the project, make changes in your fork, and submit a pull request back to the original project: this is known as the [GitHub Flow](#). All pull requests need approval from edX. For more information, contact edX at docs@edx.org.

Change Log

2.1 May, 2014

Date	Change
05/07/14	Updated the <i>Discussions</i> chapter to include a topic on closing discussions.
05/06/14	Expanded the <i>Grade and Answer Data</i> chapter to include a topic on interpreting the score histograms for problems.
05/02/14	Updated <i>LTI Component</i> to reflect changes to the Studio UI. Updated <i>Drag and Drop Problem</i> information

2.2 April, 2014

Date	Change
05/02/14	Update <i>IFrame Tool</i> documentation
04/28/14	Updated <i>Show or Hide the Course Wiki Page</i> to include note about Wiki content being available after you hide the Wiki page.
04/26/14	Updated label information; added XML information to <i>Problem with Adaptive Hint</i>
04/24/14	Expanded the <i>Grade and Answer Data</i> chapter to include topics on interpreting the grade reports and student progress page. Updated the <i>Beta Testing a Course</i> section to reflect feature changes.
04/23/14	Updated the <i>Working with HTML Components</i> chapter to reflect changes to the HTML component editor. Reorganized information about problems into <i>Creating Exercises and Tools</i> section.
04/23/14	Added more information about collecting language and location data from students to <i>Student Data</i> .
04/22/14	Updated the <i>Bulk Email</i> section with information about the dashboard option to opt out of course email. In <i>Discussions</i> , corrected the steps to “Create Discussion Categories”. Updated the <i>Enrollment</i> section to reflect feature changes.
04/16/14	Updated “Transcripts in Additional Languages” in <i>Working with Video Components</i> . In support of new features, added the following sections to <i>Multiple Choice Problem</i> problems: <ul style="list-style-type: none"> • <i>Shuffle Answers in a Multiple Choice Problem</i> • <i>Targeted Feedback in a Multiple Choice Problem</i> • <i>Answer Pools in a Multiple Choice Problem</i>
04/15/14	Updated <i>Testing Your Course</i> to include a section on how to <i>Switch Between Studio and Your Live Course</i> .
04/11/14	Expanded the <i>Grade and Answer Data</i> section to include a topic on interpreting the Student Answer Distribution report.
04/08/14	Updated the chapter <i>Working with HTML Components</i> to reflect the new HTML editor.
04/07/14	Expanded the <i>Course Data, Enrollment</i> , and
04/03/14	Updated the <i>Adding Pages to a Course</i> chapter to reflect ability to <i>Show or Hide the Course Wiki Page</i> .
04/02/14	Reorganized the sections <i>Building a Course</i> and <i>Creating Course Content</i> to better reflect the workflow of building a new course.
04/01/14	Update the <i>Establishing a Grading Policy</i> chapter to emphasize that grading is applied to subsections only. Updated the <i>Releasing Your Course</i> section to include <i>Course Launching Activities</i> .

2.3 March, 2014

Date	Change
03/31/14	Expanded the <i>Grade and Answer Data</i> chapter to include the new <i>Student Answer Distribution</i> section.
03/27/14	Updated the <i>Adding Pages to a Course</i> section to reflect feature changes.
03/27/14	Updated the section on <i>Beta Testing a Course</i> to include the new “batch add” feature.
03/19/14	Updated the sections on <i>Beta Testing a Course</i> , <i>Discussions</i> , <i>Grade and Answer Data</i> , and <i>Student Data</i> with changes to the new Instructor Dashboard.
03/17/14	Reorganized this document into major sections: <ul style="list-style-type: none"> • <i>Getting Started</i> • <i>Building a Course</i> • <i>Creating Course Content</i> • <i>Creating Exercises and Tools</i> • <i>Releasing Your Course</i> • <i>Running Your Course</i> • <i>Information for Your Students</i>
03/10/14	Added information about setting up your course summary page to <ul style="list-style-type: none"> • <i>The Course Start Date</i> • <i>The Course End Date</i> • <i>Add a Course Image</i> • <i>Add a Course Introduction Video</i> • <i>Describe Your Course</i>

2.4 February, 2014

Date	Change
02/25/14	Updated <i>Adding Files to a Course</i> section to include new External URL feature. Updated <i>Add a Link to a File</i> and <i>Add an Image to an HTML Component</i> to specify you must use the file’s Embed URL.
02/24/14	Created new chapter, <i>Getting Started with edX</i> . Updated <i>Add a Course Introduction Video</i> section.
02/21/14	Added the <i>Beta Testing a Course</i> chapter.
02/19/14	Updated <i>Import LaTeX Code into an HTML Component</i> to reflect new workflow and UI change for creating Latex HTML components.
02/18/14	Included several enhancements to the chapter <i>Establishing a Grading Policy</i>
02/14/14	Added <i>Additional Transcripts</i> section to <i>Working with Video Components</i> ; updated <i>Advanced Options</i> . Added the <i>Course Data</i> , <i>Staffing</i> , and <i>Enrollment</i> chapters.
02/11/14	Added <i>Gene Explorer Tool</i> and updated <i>Periodic Table Tool</i> and <i>Molecule Editor Tool</i> .
02/07/14	Added section on <i>Full Screen Image Tool</i> .
02/06/14	Added <i>Periodic Table Tool</i> and <i>Molecule Editor Tool</i>
02/05/14	Added section <i>Set the Advertised Start Date</i> .
02/04/14	Added the <i>Student Data</i> and <i>Grade and Answer Data</i> chapters. Added <i>Multiple Choice</i> and <i>Numerical Input Problem</i> and <i>Protex Protein Builder Tool</i> .

2.5 January, 2014

Date	Change
01/29/2014	Added the chapter <i>Google Instant Hangout Tool</i> .
01/24/2014	Added the <i>Discussions</i> and <i>Guidance for Discussion Moderators</i> chapters. Added more detailed instructions to <i>Zooming Image Tool</i>
01/21/2014	Added information about accessibility in the topic <i>Adding Textbooks</i> .
01/14/2014	Added info about scoring (<i>Access Scores and Feedback</i>) and due dates in <i>Open Response Assessment Problems</i> .
01/13/2014	Extensive updates to <i>Organizing Your Course Content</i> and <i>Working with HTML Components</i> .
01/08/2014	Updated <i>Adding Files to a Course</i> to reflect addition of sorting to the Files & Uploads page. Updated <i>Set Important Dates for Your Course</i> to reflect change to default course start date to 2029.
01/07/2014	Updated <i>Text Input Problem</i> with info about multiple strings. Added info about template to <i>Checkbox Problem</i> .
01/06/2014	Created <i>Custom JavaScript Problem</i>
01/06/2014	Created <i>Zooming Image Tool</i>
01/01/2014	Updated the chapters <i>Organizing Your Course Content</i> and <i>Testing Your Course</i> to reflect changes in the Course Outline design.

2.6 December, 2013

Date	Change
12/20/2013	Made <i>Open Response Assessments for Students</i> into template that instructors can customize.
12/19/2013	Created “Tools” topic. (Note 4/10/14: Topic merged into <i>Creating Exercises and Tools</i> .)
12/18/2013	Updated documentation about video player options in <i>Working with Video Components</i> .
12/13/2013	Created <i>LTI Component</i> . Created <i>Open Response Assessments for Students</i> .
12/12/2013	Added the edX <i>Glossary</i> .
12/11/2013	Added the chapter <i>Guidelines for Creating Accessible Content</i> .
12/10/2013	Added note about number of responses in “Available to Grade” column in <i>Open Response Assessment Problems</i> . Added <i>A Brief Introduction to MathJax in Studio</i> .
12/09/2013	Created <i>A Brief Introduction to MathJax in Studio</i> .
12/05/2013	Complete revision of edX Studio documentation and integration of edX101 content.

Getting Started

3.1 Getting Started with edX

The following sections provide an introduction to edX and instructions for getting started on edX websites:

- *edX.org and edX Edge*
- *Register Your Account*
- *edX101 and the edX Demo Course*
- *Reset Your Password*

This information is intended for course staff. You may want to include the details about the registration process and password policies in your communications with your prospective students.

Also see *Enroll Students in a Course* for information about enrolling students in a course, and for an example of the email students receive.

3.1.1 edX.org and edX Edge

When you want to explore edX courses, you can register on [edX.org](#), [edX Edge](#), or both. These sites are visually and functionally the same, but the content and purpose are different.

- [edX.org](#) hosts massive open online courses (MOOCs) from edX institutional partners. To publish courses on [edX.org](#), you must have an agreement with edX and specific approval from your university. Courses on [edX.org](#) are publicly listed in the edX course catalog and are open to students from around the world.
- [EdX Edge](#) is edX's more private site. Courses on Edge are not published on [edX.org](#). Any member of a partner course team can create and publish courses, including test courses, on Edge without receiving approval from edX or an affiliated university. However, Edge does not have a course catalog, and courses cannot be found through search engines such as Google. Only students whom you explicitly invite or who have the URL for your course can participate in your course on Edge.
- [EdX Edge](#) also hosts small private online courses (SPOCs).

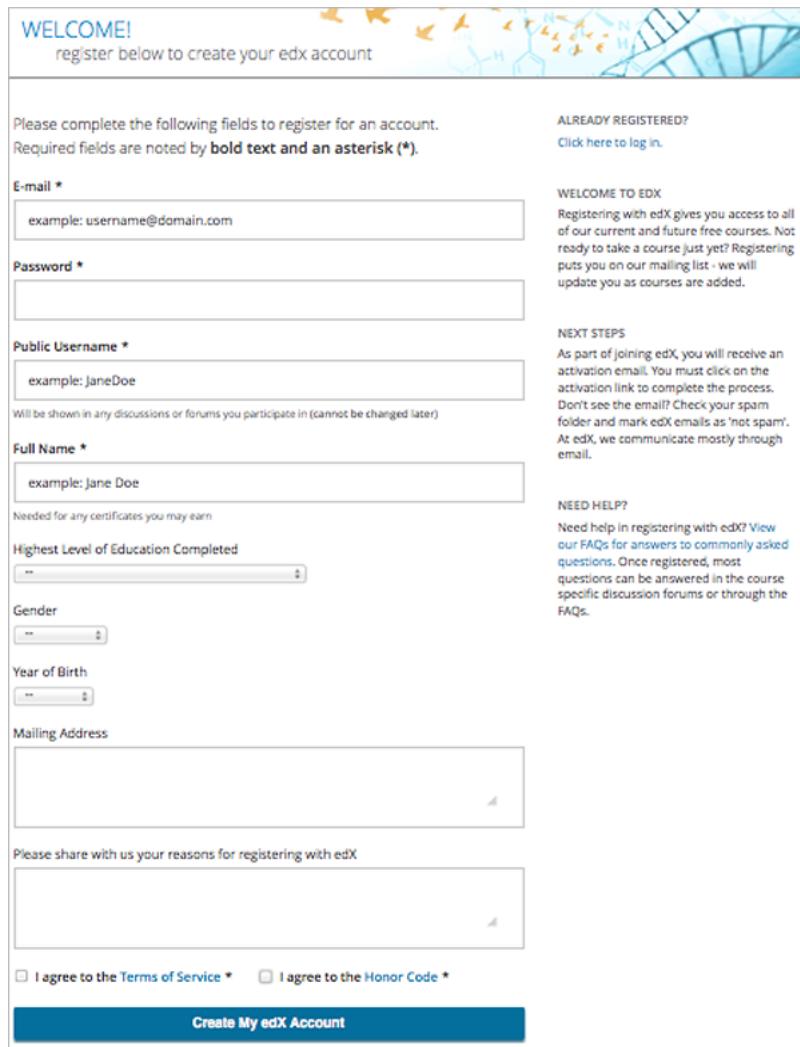
Note: All course data and accounts on [Edge](#) and [edX.org](#) are separate. If you want to use both [edX.org](#) and Edge, you must go through the registration process on both sites.

3.1.2 Register Your Account

To get started, you register your account on [edX.org](#) and Edge.

1. On the registration page, enter your account information.

- For edX.org, go to the registration page:



The screenshot shows the 'WELCOME!' registration page for edX.org. The top features a decorative banner with DNA helixes and birds. The left side contains input fields for E-mail, Password, Public Username, Full Name, Highest Level of Education Completed, Gender, Year of Birth, and Mailing Address. The right side includes links for 'ALREADY REGISTERED?' and 'NEXT STEPS' (with instructions about activation emails), and a 'NEED HELP?' section with a link to FAQs. At the bottom, there's a text area for sharing reasons for registering, two checkboxes for agreeing to Terms of Service and Honor Code, and a large blue 'Create My edX Account' button.

WELCOME!

register below to create your edX account

Please complete the following fields to register for an account.
Required fields are noted by **bold text** and an asterisk (*).

E-mail *

example: username@domain.com

ALREADY REGISTERED?
[Click here to log in.](#)

WELCOME TO EDX
Registering with edX gives you access to all of our current and future free courses. Not ready to take a course just yet? Registering puts you on our mailing list - we will update you as courses are added.

Public Username *

example: JaneDoe

Will be shown in any discussions or forums you participate in (cannot be changed later)

Full Name *

example: Jane Doe

Needed for any certificates you may earn

Highest Level of Education Completed

Gender

Year of Birth

Mailing Address

Please share with us your reasons for registering with edX

I agree to the [Terms of Service](#) * I agree to the [Honor Code](#) *

Create My edX Account

- For Edge, go to <https://edge.edx.org> and click **Register**:

Sign Up for edX

E-mail *

e.g. yourname@domain.com

Password *

.....

Public Username *

e.g. yourname (shown on forums)

Full Name *

e.g. Your Name (for certificates)

Ed. Completed **Gender** **Year of birth**

-- -- --

Mailing address

Goals in signing up for edX

I agree to the Terms of Service*

I agree to the Honor Code*

3.1. Getting Started with edX

Make sure to check both **I agree to the Terms of Service** and **I agree to the Honor Code**.

Note: Students will see your **Public Username**, not your **Full Name**.

If you are at an edX consortium university, you should use your institutional e-mail address.

Your password can be any string.

- When you complete the form, click **Create my edX Account**.

After you submit the registration, you receive an activation email message. The email content is:

Thank you for signing up for edX!

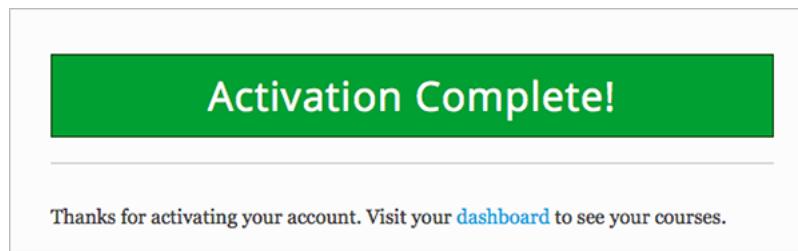
To activate your account, please copy and paste this address into your web browser's address bar:

<https://courses.edx.org/activate/unique-registration-code>

If you didn't request this, you don't need to do anything;
you won't receive any more email from us.

Please do not reply to this e-mail; if you require assistance,
check the help section of the edX web site.

- Click the link in the e-mail to complete the activation. When you see the following page, your account has been activated:



3.1.3 edX101 and the edX Demo Course

EdX has provided [edX101](#) and the [edX Demo course](#) to help familiarize you with taking and creating edX courses.

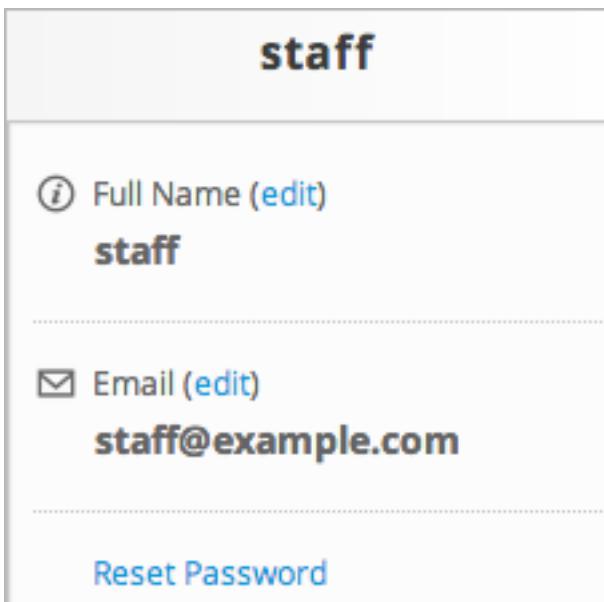
- [edX101](#), available on [Edge](#), is an example of a course built with Studio. It is a self-paced walk through of planning, building, and running your own online course.
- The [edX Demo course](#), available on [edX.org](#), allows new students to explore and learn how to take an edX course. We recommend that you become familiar with the student's experience of an edX course before you begin building your first course.

Note: You may want to include information about the edX Demo Course in your course materials, and recommend that new students take the edX Demo Course before proceeding with your course.

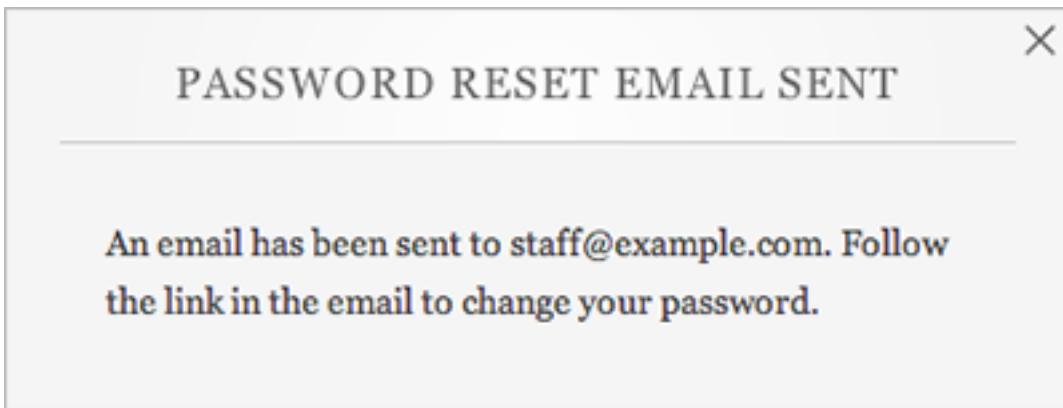
3.1.4 Reset Your Password

The process to reset your password on edX.org and Edge is the same.

1. On [edx.org](#) or [edge.edx.org](#), go to your Dashboard.
2. In the account information pane in the upper left corner, click **Reset Password**.



A dialog box opens confirming that a message has been sent to your email address.



- When you receive the following e-mail message, click the link in the message:

You're receiving this e-mail because you requested a password reset for your user account at edx.org.

Please go to the following page and choose a new password:

https://edx.org/password_reset_confirm/unique-code/

If you didn't request this change, you can disregard this email - we have not yet reset your password.

Thanks for using our site!

The edX Team

- When the following dialog box opens, enter your new password in both fields, and then click **Change My Password**:

RESET YOUR EDX PASSWORD

Please enter your new password twice so we can verify you typed it in correctly.
Required fields are noted by **bold text and an asterisk (*)**.

Your New Password *

Your New Password Again *

Change My Password

Note: Your password can be any string.

After you click **Change My Password**, your password is reset for edX.org or edge.edX.org. You must use the new password when you log in.

3.2 Getting Started with Studio

3.2.1 Overview

This chapter describes the tools you use to build an edX course, and how to create your first course:

- *What is Studio?*
- *Use Studio on Edge*
- *Create Your First Course*
- *View Your Course on Edge*
- *View Your Course on edX.org*

If you are using an instance of Open edX, some specifics in this chapter may not apply.

3.2.2 What is Studio?

Studio is the edX tool you use to build your courses.

You use Studio to create course content, problems, videos, and other resources for students.

With Studio, you can also manage your schedule and course team, set grading policies, publish your course, and more.

You use Studio directly through your browser. You do not need any additional software.

3.2.3 Use Studio on Edge

You can use **Studio** on Edge to build your own courses.

Go to: <https://studio.edge.edx.org>. Sign in with the account you created on Edge.

You must then request access to create courses:

1. Click the + sign to expand the field labeled **Becoming a Course Creator in Studio**.
2. Click **Request the Ability to Create Courses**.

EdX then evaluates your request. When course creation permissions are granted, you receive an email message.

3.2.4 Create Your First Course

When you receive notice that you can create courses, log in to Studio.

You see the following page:

My Courses

Welcome, MikeSmith!

You currently aren't associated with any Studio Courses.

Are you staff on an existing Studio course? You will need to be added to the course in Studio by the course creator. Please get in touch with the course creator or administrator for the specific course you are helping to author.

Create Your First Course Your new course is just a click away! + Create Your First Course

Need help? If you are new to Studio and having trouble getting started, there are a few things that may be of help:
Get started by reading Studio's Documentation
Request help with Studio

1. Click **Create Your First Course**.
2. Enter course information as needed and click **Create**.

Create a New Course

Course Name *
e.g. Introduction to Computer Science
The public display name for your course.

Organization *
e.g. UniversityX or OrganizationX
The name of the organization sponsoring the course. Note: This is part of your course URL, so no spaces or special characters are allowed. This cannot be changed, but you can set a different display name in Advanced Settings later.

Course Number *
e.g. CS101
The unique number that identifies your course within your organization. Note: This is part of your course URL, so no spaces or special characters are allowed and it cannot be changed.

Course Run *
e.g. 2014_T1
The term in which your course will run. Note: This is part of your course URL, so no spaces or special characters are allowed and it cannot be changed.

CREATE **CANCEL**

Note: Enter new course information carefully. This information becomes part of the URL for your course. To change the URL after the course is created, you must contact edX through the Help site (<http://help.edge.edx.org>). Additionally, because this information becomes part of your course URL, the total number of characters in the following three fields must be 65 or fewer.

-
- For **Course Name**, enter the title of your course. For example, the name may be “Sets, Maps and Symmetry Groups”. Use title capitalization for the course title.

- For **Organization**, enter the name of your university. Do not include whitespace or special characters.
- For **Course Number**, enter both a subject abbreviation and a number. For example, for public health course number 207, enter **PH207**. For math course 101x, enter **Math101x**. 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”.

3. Click **Save**.

Your new course opens to the **Course Outline** page. Because you haven't created any content yet, this page is empty. In your browser's address bar, notice that the URL of your course includes the course organization, number, and course run.

The rest of this documentation describes how you now build and run your course. But first, let's view your empty course on Edge.

3.2.5 View Your Course on Edge

You can now view the course you just created, even though it doesn't have any content.

In the Course Outline in Studio, click **View Live**. The course opens on Edge.

You can also go directly to [Edge](#). Log in if prompted. You see the course you just created listed:

The screenshot shows the Edge user interface. On the left, there is a sidebar with the user's name, MikeSmith, and options to edit full name and email, and to reset password. The main area is titled "CURRENT COURSES". It lists a single course: "UNH 101 Introduction". To the right of the course title is the text "Course Started - Jan 01, 1970" and links for "Email Settings" and "Unregister". Below the course title is a blue "View Course" button.

You can view the course and see that there is no content yet.

To build your course, keep reading this document.

3.2.6 View Your Course on edX.org

If your organization has an agreement with edX and you have specific approval, you can create a course on edX. To see the content of your course as students will see it, you must register for your course. You can then view it on the LMS.

1. Create your course and add content.
2. 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.
3. Click the blue **Register** button to register for your course. After you register for your course, it opens in the LMS.
4. To continue working on your course, in your browser, switch back to the tab that shows Studio. You will still be on the **Course Outline** page.

3.3 Glossary

3.3.1 A

About Page

The course page that provides potential students with a course summary, prerequisites, a course video and image, and important dates.

See *The Course Summary Page* for more information.

Advanced Editor

An XML-only editor in a Problem component that allows you to can create and edit any type of problem.

For more information, see *The Advanced Editor*.

Assignment Type

The category of graded student work, such as homework, exams, and exercises.

For more information, see *Establishing a Grading Policy*.

3.3.2 C

Certificate

A document issued to a registered student who successfully completes a course. Not all edX courses offer certificates, and not all students register as certificate candidates.

Checkbox Problem

A problem that prompts the student to select one or more options from a list of possible answers. For more information, see *Checkbox Problem*.

Chemical Equation Response Problem

A problem that allows the student to enter chemical equations as answers. For more information, see *Chemical Equation Problem*.

Circuit Schematic Builder Problem

A problem that allows the student to construct a schematic answer (such as an electronics circuit) on an interactive grid.

See *Circuit Schematic Builder Problem* for more information.

Cohort

A group of students that participate in the class together. A cohort can have forum discussions apart from the rest of the students.

Component

The part of a unit that contains your actual course content. A unit can contain one or more components. For more information, see *Components*.

Course Accordion

The left-hand navigation bar in the courseware that shows the sections and subsections for a course. The word “accordion” is intended to evoke the folding and unfolding of an accordion, because when you click a section, the section expands to show subsections.

Course Catalog

The page that lists all courses offered in the edX learning management system.

Course Handouts

Course handouts are files you make available to students in the Course Info page.

See *Add Course Handouts* for more information.

Course Info Page

The page that opens first every time students access your course. You can post announcements on the Course Info page. The Course Handouts sidebar appears in the right pane of this page.

Course Run

The term in which your course takes place. You set the course run when you create your course. For more information, see *Create a New Course*.

Courseware The page where students access the primary instructional materials for your course. Sections, subsections, units and components are all accessed from the Courseware page.

Custom Response Problem

A custom response problem evaluates text responses from students using an embedded Python script. These problems are also called “write-your-own-grader” problems. For more information, see *Write-Your-Own-Grader Problem*.

3.3.3 D

Discussion Forum

The page where students can communicate with peers and staff by typing in questions and responding to each other.

See *Working with Discussion Components* for more information.

Discussion Component

Forums that course staff can add directly to units. For example, a Video component can be followed by a Discussion component so that students can discuss the video content without having to leave the page.

See *Working with Discussion Components* for more information.

Dropdown Problem

A problem that asks students to choose from a collection of answer options, presented as a drop-down list. For more information, see *Dropdown Problem*.

3.3.4 E

edX101

edX’s online course about how to create online courses. The intended audience is faculty and university administrators It is also the first course ever to have been developed entirely using edX Studio.

edX Edge

A less restricted site than edX.org. While only consortium members can create and post content on edX.org, any users with course creator permissions can create courses with Studio on studio.edge.edx.org, then view the courses on the learning management system at edge.edx.org.

edX Studio

The edX tool you use to build your courses.

See *What is Studio?* for more information.

Exercises

Practice or practical problems interspersed in edX course content to keep the learner engaged. Exercises are also an important measure of teaching effectiveness and learner comprehension.

3.3.5 G

Grade Range

Thresholds that specify how numerical scores are associated with grades, and the score required to pass a course.

See *Set the Grade Range* for more information.

Grading Rubric

List of the items that a student's response should cover in an open response assessment.

See *Rubrics* for more information.

3.3.6 H

HTML Component

The component where you add and format text for your course. An HTML component can contain text, lists, links and images.

See *Working with HTML Components* for more information.

3.3.7 I

Image Mapped Input Problem

A problem that presents an image and accepts clicks on the image as an answer.

See *Image Mapped Input Problem* for more information.

Import

A tool in edX Studio that loads a new course into your existing course. When you use the Import tool, Studio replaces all of your existing course content with the content from the imported course.

See *Import a Course* for more information.

3.3.8 L

LaTeX

A document markup language and document preparation system for the TeX typesetting program.

In edX Studio, you can *Import LaTeX Code into an HTML Component*.

You can also create a *Problem Written in LaTeX*.

Learning Management System (LMS)

The platform that students use to view courses.

Live Mode

A view that allows course staff to review all public units as students see them, regardless of the release dates of the section and subsection that contain the units.

See *Switch Between Studio and Your Live Course* for more information.

3.3.9 M

Math Expression Input Problem

A problem that requires students to enter a mathematical expression as text, such as $e=m*c^2$.

See *Math Response Formatting for Students* for more information.

MathJax

A LaTeX-like language you use to write equations. Studio uses MathJax to render text input such as x^2 and $\sqrt{x^2-4}$ as “beautiful math.”

See *A Brief Introduction to MathJax in Studio* for more information.

Multiple Choice Problem

A problem that asks students to select one answer from a list of options.

See *Multiple Choice Problem* for more information.

3.3.10 N

Numerical Input Problem

A problem that asks students to enter numbers or specific and relatively simple mathematical expressions.

See *Numerical Input* for more information.

3.3.11 P

Pages

Pages that supplement the courseware for a course. Each page appears in your course’s navigation bar.

See *Adding Pages to a Course* for more information.

Preview Mode

A view that allows you to see all the units of your course as students see them, regardless of whether the visibility of each unit is set to Public or Private and regardless of whether the release dates have passed.

See *Preview Your Course* for more information.

Private Unit

A unit whose **Visibility** option is set to Private so that students cannot see the unit, even if it is located in a subsection that has been released.

See *Public and Private Units* for more information.

Problem Component

A component that allows you to add interactive, automatically graded exercises to your course content. You can create many different types of problems.

See *Working with Problem Components* for more information.

Progress Page

The page in the learning management system that shows students their scores on graded assignments in the course.

Public Unit

A unit whose **Visibility** option is set to Public so that the unit is visible to students, if the subsection that contains the unit has been released.

See *Public and Private Units* for more information.

3.3.12 R

Rubric

List of the items that a student's response should cover in an open response assessment.

See *Rubrics* for more information.

3.3.13 S

Section

The topmost category in your course. A section can represent a time period in your course, or another organizing principle.

See *Sections* for more information.

Simple Editor

The graphical user interface in a Problem component that contains formatting buttons and is available for some problem types. For more information, see *The Studio View of a Problem*.

Subsection

A division that represents a topic in your course, or another organizing principle. Subsections are found inside sections and contain units. Subsections can also be called “lessons.”

See *Subsections* for more information.

Short Course Description

The description of your course that appears on the edX [Course List](#) page.

See *Describe Your Course* for more information.

3.3.14 T

Text Input Problem

A problem that asks the student to enter a line of text, which is then checked against a specified expected answer.

See *Text Input Problem* for more information.

Transcript

A printed version of the content of a video. You can make video transcripts available to students.

See *Working with Video Components* for more information.

3.3.15 V

Video Component

A component that you can use to add recorded videos to your course.

See *Working with Video Components* for more information.

3.3.16 W

Wiki

The page in each edX course that allows students as well as course staff to add, modify, or delete content.

Students can use the wiki to share links, notes, and other helpful information with each other.

3.3.17 X

XBlock

EdX's component architecture for writing courseware components.

Third parties can create components as web applications that can run within the edX learning management system.

Building a Course

4.1 Creating a New Course

4.1.1 Overview

This chapter describes how to create and set up your course:

- *Create a New Course*
- *Edit Your Course*
- *Use the Course Checklist*
- *Add Course Team Members*

You can also *Export a Course* and *Import a Course* through Studio. You can do this when you need to edit the course in XML.

4.1.2 Create a New Course

1. Log in to Studio.
2. Click **New Course**.
3. Enter course information as needed and click **Create**.

Create a New Course

Course Name *
e.g. Introduction to Computer Science
The public display name for your course.

Organization *
e.g. UniversityX or OrganizationX
The name of the organization sponsoring the course. Note: This is part of your course URL, so no spaces or special characters are allowed. This cannot be changed, but you can set a different display name in Advanced Settings later.

Course Number *
e.g. CS101
The unique number that identifies your course within your organization. Note: This is part of your course URL, so no spaces or special characters are allowed and it cannot be changed.

Course Run *
e.g. 2014_T1
The term in which your course will run. Note: This is part of your course URL, so no spaces or special characters are allowed and it cannot be changed.

CREATE **CANCEL**

Note: Enter new course information carefully. This information becomes part of the URL for your course. To change the URL after the course is created, you must contact edX through the Help site (<http://help.edge.edx.org>). Additionally, because this information becomes part of your course URL, the total number of characters in the following four fields must be 65 or fewer.

- For **Course Name**, enter the title of your course. For example, the name may be “Sets, Maps and Symmetry Groups”. Use title capitalization for the course title.
- For **Organization**, enter the name of your university. Do not include whitespace or special characters.
- For **Course Number**, enter both a subject abbreviation and a number. For example, for public health course number 207, enter **PH207**. For math course 101x, enter **Math101x**. 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”.*

- For **Course Run**, enter the term your course will run. Do not include whitespace or special characters.

The Course Run date you enter does not affect the default **Course Start Date**. See *Set Important Dates for Your Course* for more information.

4. Click **Save**.

You then see the empty Course Outline.

4.1.3 Edit Your Course

When you create a new course, the course opens in Studio automatically and you can begin editing.

If you come back to Studio later, your courses are listed on the Studio login page.

Welcome, MikeSmith!

Here are all of the courses you currently have access to in Studio:

[Introduction](#)

UNH / 101 / 2014_Fall

Need help?

If you are new to Studio and having trouble getting started, there are a few things that may be of help:

[Get started by reading Studio's Documentation](#)

[Request help with Studio](#)

To open the course, click the course name. You go to the Course Outline.

4.1.4 Use the Course Checklist

You can use a Course Checklist within Studio to help you work through the tasks of building a course.

Categories of tasks in the Course Checklist include:

- Getting Started with Studio
- Draft a Rough Course Outline
- Explore edX's Support Tools
- Draft Your Course About Page

From the **Tools** menu, select **Checklists**.

Course Checklists

Getting Started With Studio Tasks Completed: 0 / 4

- Add Course Team Members** Grant your collaborators permission to edit your course so you can work together. [Edit Course Team](#)
- Set Important Dates for Your Course** Establish your course's student enrollment and launch dates on the Schedule and Details page.
- Draft Your Course's Grading Policy** Set up your assignment types and grading policy even if you haven't created all your assignments.
- Explore the Other Studio Checklists** Discover other available course authoring tools, and find help when you need it.

Draft a Rough Course Outline Tasks Completed: 0 / 7

As shown above for the **Add Course Team Members** task, if you hover over a task, a button is displayed that takes you to the page to complete that task.

You can expand and collapse sections of this page as needed.

You can check tasks as you complete them. Studio saves your changes automatically. Other course staff can see your changes.

4.1.5 Add Course Team Members

Course team members are users who help you build your course.

Only a team member with Admin access can add or remove course team members, or grant Admin access to other team members.

Other course team members can edit the course and perform all tasks except adding and removing other new team members and granting Admin access.

Note: Any course team member can delete content created by other team members.

All course team members must be registered with Studio and have an active account.

To add a course team member:

1. Ensure you have Admin access.
2. Ensure that the new team member has registered with Studio.
3. From the **Settings** menu, select **Course Team**.
4. Click **Add a New Team Member**.
5. Enter the new team member's email address, then click **ADD USER**.

4.2 Exporting and Importing a Course

You can *Export a Course* and *Import a Course* through Studio.

4.2.1 Export a Course

There are several reasons you may want to export your course:

- To save your work
- To edit the XML in your course directly
- To create a backup copy of your course, which you can import if you want to revert the course back to a previous state
- To create a copy of your course that you can later import into another course instance and customize
- To share with another instructor for another class

When you export your course, Studio creates a **.tar.gz** file that includes the following course data:

- Course content (all Sections, Subsections, and Units)
- Course structure
- Individual problems
- Pages
- Course assets

- Course settings

The following data is not exported with your course:

- User data
- Course team data
- Discussion data
- Certificates

To export a course:

1. From the **Tools** menu, select **Export**.
2. Click **Export Course Content**.

When the export completes you can then access the .tar.gz file on your computer.

4.2.2 Import a Course

Warning: Content of the imported course replaces all the content of this course. **You cannot undo a course import.** We recommend that you first export the current course, so you have a backup copy of it.

There are several reasons you may want to import a course:

- To run a new version of an existing course
- To replace an existing course
- To load a course you developed outside of Studio

The course that you import must be in a .tar.gz file (that is, a .tar file compressed with GNU Zip). This .tar.gz file must contain a course.xml file in a course data directory. The tar.gz file must have the same name as the course data directory. It may also contain other files.

If your course uses legacy layout structures, you may not be able to edit the course in Studio, although it will probably appear correctly on Edge. To make sure that your course is completely editable, ensure that all of your material is embedded in a unit.

The import process has five stages. During the first two stages, you must stay on the Course Import page. You can leave this page after the Unpacking stage has completed. We recommend, however, that you don't make important changes to your course until the import operation has completed.

To import a course:

1. From the **Tools** menu, select **Import**.
2. Click **Choose a File to Import**.
3. Locate the file that you want, and then click **Open**.
4. Click **Replace my course with the one above**.

4.3 Setting up the Student View

4.3.1 Overview

This chapter describes how you set up your course to be displayed in the course summary page and in a student's dashboard. The information you configure for your course is important for prospective and current students to understand.

See:

- *The Course Summary Page*
- *The Student Dashboard*
- *Set Important Dates for Your Course*
- *The Course Start Date*
- *Set the Advertised Start Date*
- *The Course End Date*
- *Describe Your Course*
- *Add a Course Image*
- *Add a Course Introduction Video*
- *Set Course Requirements*
- *A Template For Your Course Overview*

4.3.2 The Course Summary Page

The following image shows an example course summary page. Students can see the course summary page before registering, and may decide to register based on the content of the page. You configure the contents of this page in Studio, as described in this chapter:



Justice

Justice is an introduction to moral and political philosophy, including discussion of contemporary dilemmas and controversies.

About this Course

Justice is a critical analysis of classical and contemporary theories of justice, including discussion of present-day applications. Topics include affirmative action, income distribution, same-sex marriage, the role of markets, debates about rights (human rights and property rights), arguments for and against equality, dilemmas of loyalty in public and private life. The course invites students to subject their own views on these controversies to critical examination.

The principal readings for the course are texts by Aristotle, John Locke, Immanuel Kant, John Stuart Mill, and John Rawls. Other assigned readings include writings by contemporary philosophers, court cases, and articles about political controversies that raise philosophical questions.

Closed Captioning is available in Chinese, German, Portuguese, and Spanish.

Before your course starts, try the new edX Demo where you can explore the fun, interactive learning environment and virtual labs. [Learn more](#).

School: [HarvardX](#)

Course Code: ER22.1x

Classes Start: 8 Apr 2014

Course Length: 12 weeks

Estimated effort: From 2:00 hours a week to a lifetime

Prerequisites:

None

[Register for ER22.1x](#)



4.3.3 The Student Dashboard

If a student registers for your course, the course is then listed on the dashboard, with the course image. From the dashboard, a student can open a course that has started. If the course has not started, the student can see the start date, as explained in this chapter.



MITx

6.00.2x Introduction to Computational Thinking and Data Science

[View Course](#) [Email Settings](#) [Unregister](#)



MITx

6.SFMx Street Fighting Mathematics

[Email Settings](#) [Unregister](#)

4.3.4 Set Important Dates for Your Course

You must set dates and times for enrollment and for the course.

In Studio, from the **Settings** menu, select **Schedule and Details**.

Course Schedule

Course Start Date

12/31/2029

First day the course begins

Course Start Time

(EST)

19:00

Course End Date

MM/DD/YYYY

Last day your course is active

Course End Time

(EST)

HH:MM

Enrollment Start Date

MM/DD/YYYY

First day students can enroll

Enrollment Start Time

(EST)

HH:MM

Enrollment End Date

MM/DD/YYYY

Last day students can enroll

Enrollment End Time

(EST)

HH:MM

Follow the on-screen text to enter the course and enrollment schedule.

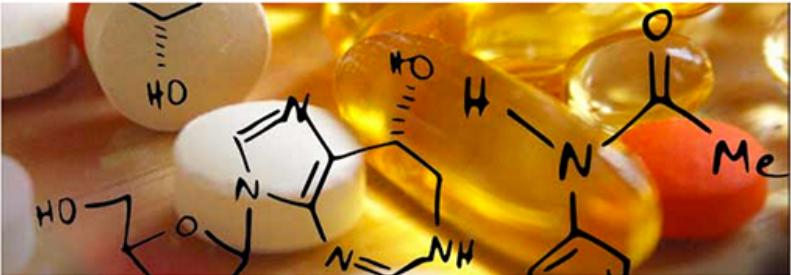
Note: The Time fields on this page reflect the current time zone in your browser, depending on your geography. Course start times for students are shown as UTC.

4.3.5 The Course Start Date

Note: The default **Course Start Date** is set far into the future, to **01/01/2030 GMT**. This is to ensure that your course does not start before you intend it to. You must change the course start date to the date you want students to begin using the course.

Students see the course start date on their dashboards and on the course summary page.

The following example shows the course start date on the course summary page:



Watch the Course Intro Video



Medicinal Chemistry: The Molecular Basis of Drug Discovery

This course explores how to bring a drug from concept to market, and how a drug's chemical structure relates to its biological function.

About this Course

This course explores how to bring a drug from concept to market, and how a drug's chemical structure relates to its biological function. The course opens with an introduction to the drug approval process. This introduction combines the social, economic, and ethical aspects of drug discovery. Topics include how diseases are selected for treatment, the role of animal testing, and the costs of various discovery phases. The course then focuses on the scientific side of drug discovery. Topics include how drugs interact with biological molecules, drug absorption and elimination, and the discovery of weakly active molecules and their optimization into viable drugs.

School: DavidsonX
Course Code: 001x
Classes Start: 10 Mar 2014
Course Length: 7 weeks
Estimated effort: 6 to 8 hours per week

Prerequisites:
Students should be able to identify organic chemistry functional groups and read line-angle chemical structures. Students should also know the parts of a cell and be comfortable working...
[see more...](#)

Register for 001x
and choose your student track

 |  |  | 

Note: For courses on edX.org, you must communicate the course start date to your edX Program Manager, to ensure the date is accurate on the course summary page.

In the dashboard, if the course has not yet started, students see the start date as in the following example:



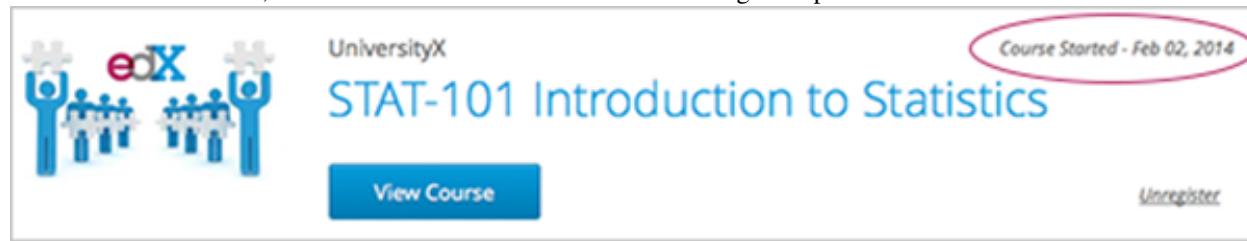
MITx

6.SFMx Street-Fighting Math

Course Starts - Mar 25, 2014

[Unregister](#)

If the course has started, students see the start date as in the following example:



edX

UniversityX

STAT-101 Introduction to Statistics

View Course

Course Started - Feb 02, 2014

[Unregister](#)

30

Chapter 4. Building a Course

4.3.6 Set the Advertised Start Date

You can set an advertised start date for your course that is different than the course start date you set in the **Schedule and Details** page. You may want to do this if there is uncertainty about the exact start date. For example, you could advertise the start date as **Spring, 2014**.

To set an advertised start date:

1. From the **Settings** menu, select **Advanced Settings**.
2. Find the policy key **advertised_start**. The default value is **null**.
3. Enter the date you want as an advertised start date. You can use any string, enclosed in double quotation marks. If you format the string as a date (for example, as 02/01/2014), the value is parsed and presented to students as a date.

Policy Key:	Policy Value:
advertised_start	"Spring 2014"

4. Click **Save Changes** at the bottom of the page.

The start date shown on the student's dashboard is now the value of the `advertised_start` policy key:

If you do not change the default course start date (01/01/2030), and the `advertised_start` policy value is `null`, then the student dashboard does not list a start date for the course. Students just see that the course has not yet started:

4.3.7 The Course End Date

When your course is completed, students see the course end date on their dashboards.

Note: For courses on [edX.org](#), you must communicate the course end date to your edX Program Manager, to ensure the date is accurate on the course summary page.

If grades and certificates are not yet issued, or if students enroll in an archived course after it has ended, the course appears in the dashboard as in the following example:

The screenshot shows the course summary page for CS169.2x Software as a Service. In the top left corner is a purple cloud icon containing three interlocking gears and the letters "SaaS". To its right is the text "BerkeleyX". In the top right corner is a red oval containing the text "Course Completed - Dec 17, 2012". The main title "CS169.2x Software as a Service" is displayed prominently. Below the title is a message in a box: "Final course details are being wrapped up at this time. Your final standing will be available shortly." At the bottom are two buttons: "View Archived Course" and "Unregister".

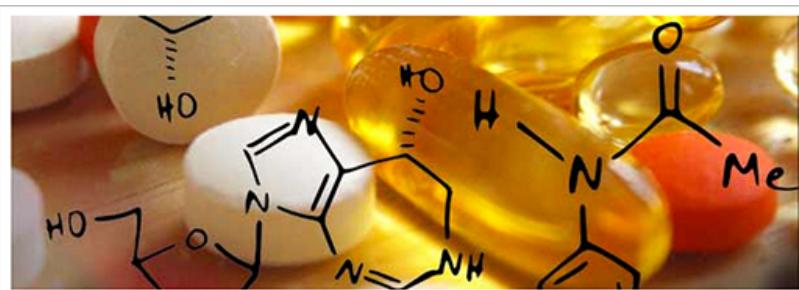
If grades are complete and certificates are issued, students see the course, the end date, and the message as in the following example:

The screenshot shows the course summary page for 6.00.1x Introduction to Computer Science and Programming. In the top left corner is a dark grey rectangular image showing some Python code. To its right is the text "MITx". In the top right corner is a red oval containing the text "Course Completed - Jan 01, 2014". The main title "6.00.1x Introduction to Computer Science and Programming" is displayed prominently. Below the title is a message in a box: "Your final grade: 55%." followed by a blue button "Download Your Certificate (PDF)". At the bottom are two buttons: "View Archived Course" and "Unregister".

4.3.8 Describe Your Course

On Edge, students that you explicitly invite see the description of your course on the course summary page.

For example, the course description is circled in the following course summary page:



[Watch the Course Intro Video](#)



Medicinal Chemistry: The Molecular Basis of Drug Discovery

This course explores how to bring a drug from concept to market, and how a drug's chemical structure relates to its biological function.

About this Course

This course explores how to bring a drug from concept to market, and how a drug's chemical structure relates to its biological function. The course opens with an introduction to the drug approval process. This introduction combines the social, economic, and ethical aspects of drug discovery. Topics include how diseases are selected for treatment, the role of animal testing, and the costs of various discovery phases. The course then focuses on the scientific side of drug discovery. Topics include how drugs interact with biological molecules, drug absorption and elimination, and the discovery of weakly active molecules and their optimization into viable drugs.

School: [DavidsonX](#)

Course Code: [001x](#)

Classes Start: [10 Mar 2014](#)

Course Length: [7 weeks](#)

Estimated effort: [6 to 8 hours per week](#)

Prerequisites:

Students should be able to identify organic chemistry functional groups and read line-angle chemical structures. Students should also know the parts of a cell and be comfortable working...

[see more...](#)

[Register for 001x
and choose your student track](#)



Note: For courses on edX.org, you must communicate the course description to your edX Program Manager, to ensure the content is accurate on the course summary page.

1. From the **Settings** menu, select **Schedule & Details**.
2. Scroll down to the **Introducing Your Course** section, then locate the **Course Overview** field.

Introducing Your Course

Information for prospective students

Course Overview

```
1 <section class="about">
2   <h2>About This Course</h2>
3   <p>Include your long course description here. The long course description should contain 150-400 words.
4   </p>
5   <p>This is paragraph 2 of the long course description. Add more paragraphs as needed. Make sure to
6   enclose them in paragraph tags.</p>
7 </section>
8 <section class="prerequisites">
9   <h2>Prerequisites</h2>
10  <p>Add information about course prerequisites here.</p>
11 </section>
12 <section class="course-staff">
13   <h2>Course Staff</h2>
14   <article class="teacher">
15     <div class="teacher-image">
16       
18     </div>
19   <h3>Staff Member #1</h3>
```

Introductions, prerequisites, FAQs that are used on [your course summary page](#) (formatted in HTML)

3. Overwrite the content as needed for your course, following the directions in the boilerplate text. Do not edit HTML tags. For a template that includes placeholders, see *A Template For Your Course Overview*.

Note: There is no save button. Studio automatically saves your changes.

4. Click **your course summary page** in the text beneath the field to test how the description will appear to students.

4.3.9 Add a Course Image

The course image that you add in Studio appears on the student dashboard.

On Edge, the image also appears on the course summary page.

In the following example, the course image that was added in Studio is circled in the student dashboard:

The screenshot shows the student dashboard for the course "001x Medicinal Chemistry" offered by "DavidsonX". The course image, a complex chemical structure, is highlighted with a red circle. The dashboard includes the course title, start date ("Course Starts - Mar 10, 2014"), and two buttons: "Challenge Yourself!" and "Take this course as an ID-verified student.". There is also a link to "Unregister".

On edX.org, the course image you add in Studio does not appear on the course summary page automatically. You must work directly with your edX Program Manager to set up the course summary page.

The course image should be a minimum of 660 pixels in width by 240 pixels in height, and in .JPG or .PNG format.

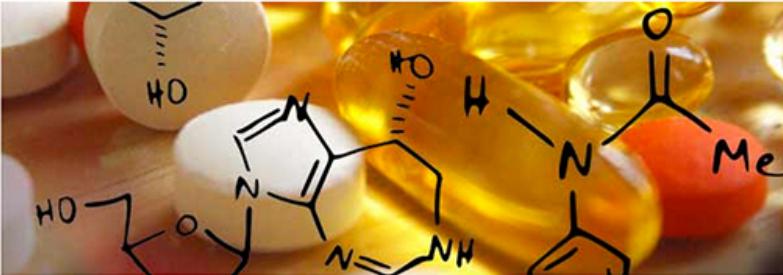
1. From the **Settings** menu, select **Schedule & Details**.
2. Scroll down to the **Course Image** section.
3. To select an image from your computer, click **Upload Course Image**, then follow the prompts to find and upload your image.
4. View your dashboard to test how the image will appear to students.

4.3.10 Add a Course Introduction Video

On Edge, the course introduction video appears on the course summary page that students see.

Note: On edX.org, you work directly with your Program Manager to set up the course video in the summary page.

In the following example, the course video is circled in the course summary page:



Watch the Course Intro Video


School: DavidsonX

Course Code: 001x

Classes Start: 10 Mar 2014

Course Length: 7 weeks

Estimated effort: 6 to 8 hours per week

Prerequisites:

Students should be able to identify organic chemistry functional groups and read line-angle chemical structures. Students should also know the parts of a cell and be comfortable working...

[see more...](#)

Register for 001x
and choose your student track

Medicinal Chemistry: The Molecular Basis of Drug Discovery

This course explores how to bring a drug from concept to market, and how a drug's chemical structure relates to its biological function.

About this Course

This course explores how to bring a drug from concept to market, and how a drug's chemical structure relates to its biological function. The course opens with an introduction to the drug approval process. This introduction combines the social, economic, and ethical aspects of drug discovery. Topics include how diseases are selected for treatment, the role of animal testing, and the costs of various discovery phases. The course then focuses on the scientific side of drug discovery. Topics include how drugs interact with biological molecules, drug absorption and elimination, and the discovery of weakly active molecules and their optimization into viable drugs.

The course video should excite and entice potential students to register, and reveal some of the personality the instructors bring to the course.

The video should answer these key questions:

- Who is teaching the course?
- What university or college is the course affiliated with?
- What topics and concepts are covered in your course?
- Why should a learner register for your course?

The video should deliver your message as concisely as possible and have a run time of less than 2 minutes.

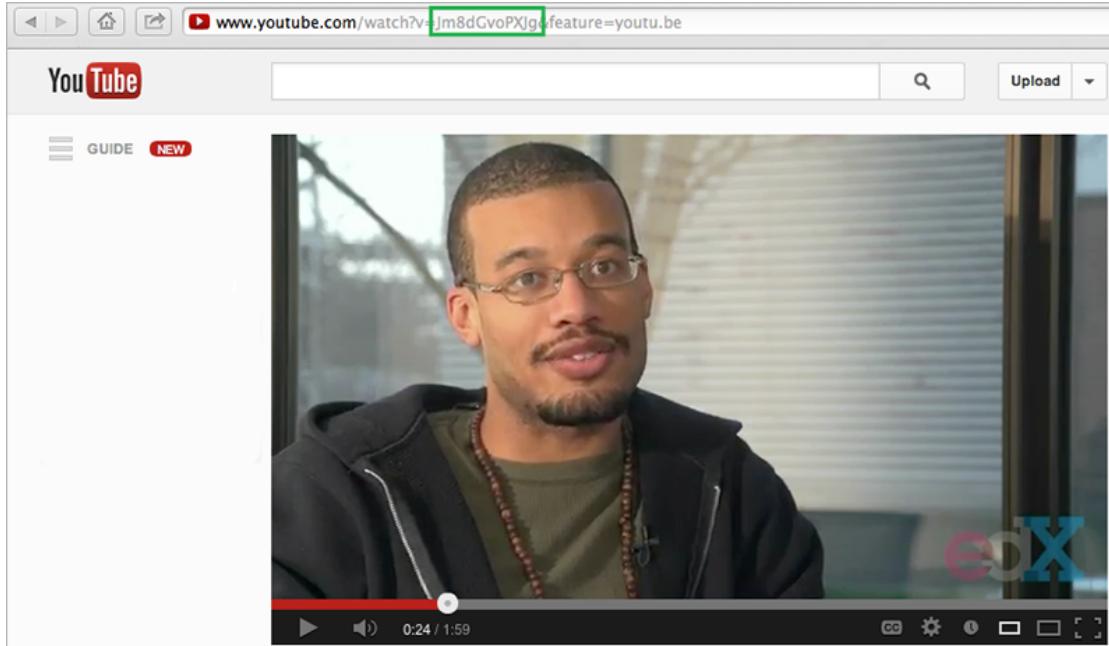
Ensure your course introduction video follows the same *Compression Specifications* and *Video Formats* guidelines as course content videos.

4.3. Setting up the Student View

35

To add a course introduction video:

1. Upload the course video to YouTube. Make note of the code that appears between `watch?v=` and `&feature` in the URL. This code appears in the green box below.



2. From the **Settings** menu, select **Schedule & Details**.
3. Scroll down to the **Course Introduction Video** section.
4. In the field below the video box, enter the YouTube video ID (the code you copied in step 1). When you add the code, the video automatically loads in the video box. Studio automatically saves your changes.
5. View your course summary page to test how the video will appear to students.

4.3.11 Set Course Requirements

The estimated Effort per Week appears at the bottom of the course summary page.

1. From the **Settings** menu, select **Schedule & Details**.
2. Scroll down to the **Requirements** section.
3. In the **Hours of Effort per Week** field, enter the number of hours you expect students to work on this course each week.
4. View your course summary page to test how the requirements will appear to students.

4.3.12 A Template For Your Course Overview

Replace the placeholders in the following template 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>  
<section>  
<section class="prerequisites">  
    <h2>Prerequisites</h2>  
    <p>Add information about class prerequisites here.</p>  
</section>  
<section class="course-staff">  
    <h2>Course Staff</h2>  
    <article class="teacher">  
        <div class="teacher-image">  
            <!-- Replace the path below with the path to your faculty image. -->  
              
        </div>  
        <h3>Staff Member</h3>  
        <p>Biography of instructor/staff member</p>  
    </article>  
    <article class="teacher">  
        <div class="teacher-image">  
              
        </div>  
        <h3>Staff Member Name</h3>  
        <p>Biography of instructor/staff member</p>  
    </article>  
</section>  
<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>Answer 2.</p>  
        </article>  
    </section>  
</section>  
  
<!--Paragraph: <p>CONTENT GOES IN HERE</p> -->  
<!--Line break: <br/> -->  
<!--Hyperlink: <a href="URL">LINK TEXT</a> -->  
<!--Email hyperlink: <a href="mailto:EMAIL@ADDRESS.COM">LINK TEXT</a> -->  
<!--Bold text: <b>TEXT</b> -->  
<!--Italic text: <i>TEXT</i> -->
```

4.4 Establishing a Grading Policy

4.4.1 Overview

Establishing a grading policy takes several steps. You must:

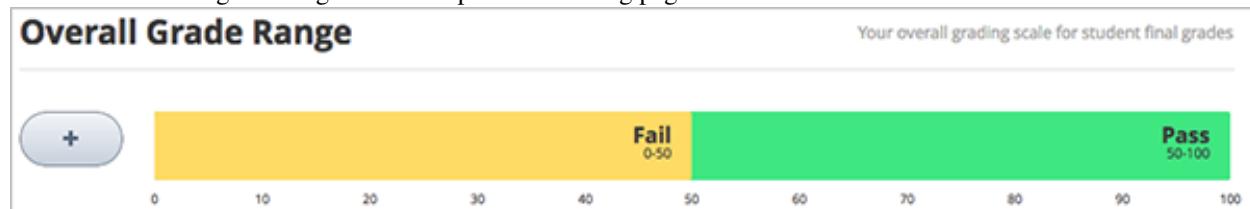
1. Set the Grade Range
2. Set the Grace Period
3. Configure the Assignment Types
4. Set the Assignment Type for Graded Subsections
5. The Student View of Grades

4.4.2 Set the Grade Range

You must set the grade range for the course. For example, your course can be pass/fail, or can have letter grades A through F.

To set the grade range, from the **Settings** menu, select **Grading**.

The control for the grade range is at the top of the Grading page.



The above example shows that you have a pass/fail grade range, with a score of 50 as the cutoff. This is the default setting used when you create a course.

You use the grade range control to change these settings:

- To add a grade in the range, click the + icon.
A new grade is added to the range between the existing grades. For example, if you add a grade in the default setting, the grade range changes to **F** (0 to 50), **B** (50 to 75), and **A** (75 to 100):



- To change the score range, hover the mouse over the line dividing two grades, click and drag the line left or right.

You can see the range numbers of the two grades adjacent to the line change. Release the mouse button when the line is where you want it.

- To change the name of the grade, double-click the current name of the grade to select it, and then start typing the name of the new grade. For example, if the original name of the grade is “Pass”, you can double-click “Pass” and then type “Excellent” to replace the name.

You cannot change **F** or **Fail**.

- To remove a grade, hover the mouse over the grade.

A **remove** link appears above the grade. Click the link.

You cannot remove F or A.

After you make any changes to the grade range, you must click **Save Changes** at the bottom of the page.

4.4.3 Set the Grace Period

You can set a grace period that extends homework due dates for your students.

Note: The grace period applies to the whole course; you cannot set a grace period for individual assignments.

In the Grading page, under **Grading Rules & Policies**, enter a value in the **Grace Period on Deadline** field. Enter the value in Hours:Minutes format.

4.4.4 Configure the Assignment Types

You must create assignment types for your course and determine the weight of the student's total grade for each assignment type.

For example, you may have:

- 10 homework assignments, worth a total of 50% of the grade;
- A midterm exam, worth a total of 20% of the grade;
- A final exam, worth 30% of the grade.

By default, a new course you create has four assignment types:

- Homework
- Lab
- Midterm Exam
- Final Exam

You can use these assignment types, modify or remove them, and create new assignment types.

To create a new assignment type, in the bottom of the Grading page, click **New Assignment Type**, then configure the fields described below.

Assignment Type Fields

You configure the following fields for each assignment type:

- **Assignment Type Name:**

The general category of the assignment. This name will be visible to students.

Note: All assignments of a particular type count the same toward the weight of that category. As a result, a homework assignment that contains 10 problems is worth the same percentage of a student's grade as a homework assignment that contains 20 problems.

- **Abbreviation:**

This is the short name that appears next to an assignment on a student's **Progress** tab.

- **Weight of Total Grade:**

The assignments of this type together account for the percent value set in **Weight of Total Grade**.

The total weight of all assignment types must equal 100.

Note: Do not include the percent sign (%) in this field.

- **Total Number:**

The number of assignments of this type that you plan to include in your course.

- **Number of Droppable**

The number of assignments of this type that the grader will drop. The grader will drop the lowest-scored assignments first.

4.4.5 Set the Assignment Type for Graded Subsections

After you configure assignment types, as you are organizing your course, you set the assignment type for subsections that contain problems that are to be graded.

Each subsection that contains problems to be graded can include only one assignment type. See *Subsections* for more information.

Note: You can only set assignment types and due dates at the subsection level. You cannot set assignment types or due dates for entire sections or for individual units within subsections. Additionally, you can designate a subsection as one, and only one, of the assignment types you configured.

See *Subsections* for general instructions on configuring a subsection. See *Set the Grading Policy* for instructions on designating a subsection as a graded assignment.

Within a graded subsection, you create problems of the type designated for that subsection. You cannot \neq not mix problems of different assignment types in the same subsection.

For example, if you want to create a homework assignment and a lab for a specific topic, create two subsections. Set one subsection as the Homework assignment type and the other as the Lab assignment type. Both subsections can contain other content as well as the actual homework or lab problems.

Note: You can create problems in Studio without specifying that the subsection is an assignment type. However, such problems do not count toward a student's grade.

See *Working with Problem Components* for instructions on creating problems.

Set the Grading Policy

You can designate a subsection as one of the assignment types that you specified in the grading policy.

You set the grading policy for the subsection from two places:

- The course outline
- The subsection page

From the course outline, click the checkmark next to the subsection. Then select a grading policy from the popup menu:

From the subsection page, click the text next to the **Graded as** label, then select a grading policy from the popup menu:

See *Establishing a Grading Policy* for more information.

Set the Due Date

For subsections that contain graded problems, you can set a due date. Students must complete the problems in the subsection before the due date to get credit. All problems in a subsection have the same due date.

1. From the subsection page, click **SET A DUE DATE**. The Due Day and Due Time fields appear.
2. Place the cursor in the Due Date field, and pick a day from the popup calendar.
3. Place the cursor in the Due Time field and pick a time.

Note: When you set a due date, keep in mind that students will be in different time zones. By default, the time zone appears as UTC, not the student's local time. If you tell your students an assignment is due at 5:00 PM, make sure to specify that the time is 5:00 PM UTC and point them to a time converter.

You can also *Set the Grace Period* for your assignments to cover any misunderstandings about time. For example, some classes have set a grace period of 1 day, 6 hours, and 1 minute. The grace period applies to all assignments.

Students see the due date in the course accordion under the subsection title. For example:

Week 2

Lecture 3 - Simple Algorithms 
Lecture Sequence

Lecture 4 - Functions 
Lecture Sequence

General Problem Set Tips

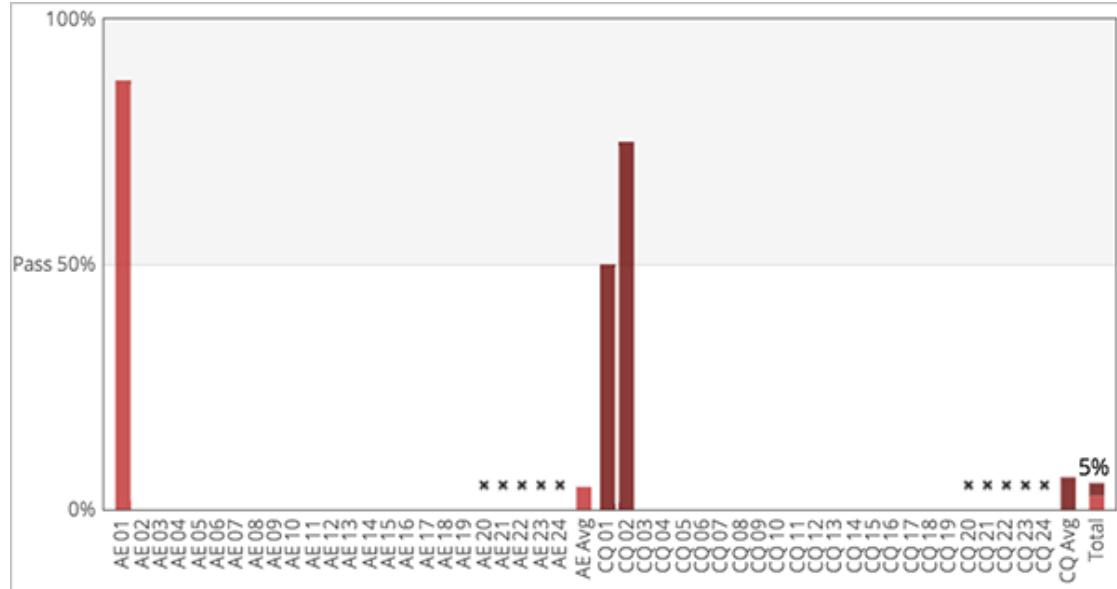
Problem Set 1
Problem Set due Oct 28, 2013 at 02:00 UTC 

Problem Set 2
Problem Set due Nov 03, 2013 at 23:30 UTC 

There are several problems in this subsection, all due November 3, 2013 at 23:30 UTC.

4.4.6 The Student View of Grades

Once a grading policy is in place, students can view both their problem scores and the percent completed and current grade in the **Progress** tab for the course.



Each item in the X axis of the chart is for a graded subsection. Graded problems in units are not broken out in the chart; the score from each problem in the subsection is added to that vertical bar.

Graded subsections are grouped in the chart by assignment type rather than listed in chronological order. For example, all homework exercises are grouped together, followed by labs, then exams.

Note: The x for an assignment in the Progress tab indicates that the assignment grade is currently dropped. You configure how many assignments are dropped when you *Configure the Assignment Types*.

Creating Course Content

5.1 Guidelines for Creating Accessible Content

EdX strives to create an innovative, online-learning platform that promotes accessibility for everyone, including learners with disabilities. We intend for these guidelines to help the course teams understand the importance of considering accessibility when designing courses and provide guidance so that they can serve the widest possible audience. *Accessibility* in online instruction refers to the degree to which information and activities are available to all students equally, regardless of physical or other disabilities.

Our guidance is based on international standards and principles for web accessibility (W3C WCAG 2.0) and universal design (usable by all, to the greatest extent possible, without the need for adaptation or specialized design). Instructors who build courses based on these principles promote the opportunity to create an inclusive experience that considers the diverse set of learning styles and needs of all learners—including learners with disabilities, learners who speak English as a second language, learners with technical issues such as low bandwidth internet or no access to audio, and learners with age-related capability issues. For purposes of these guidelines, we have assumed that end users will be equipped with the appropriate adaptive technology and compatible software.

Occasionally, unanticipated accessibility barriers will arise. To supplement the accessibility you can achieve within the edX platform, we recommend that you engage the resources available at your institution to support learners with disabilities. Most institutions offer disability support services and information technology resources that provide accessibility advice and support. These professionals are trained in making disability accommodation decisions and can advise you on what accommodations may be appropriate in light of the goals of the course and the inherent instructional methodologies employed.

As technology and accessibility improvements are constantly emerging, we plan to update these guidelines periodically.

See the following sections for more information:

- *Supporting Learners with Diverse Needs*
- *Accessibility Best Practices*
- *Conclusion*

5.1.1 Supporting Learners with Diverse Needs

Almost one-fifth of the world's population has some kind of disability. Online courses can reduce many barriers to education for these learners by providing access to courses from any location, at any time, and through the use of assistive technologies.

EdX is dedicated to creating a platform that is not only itself accessible, but also enables course creators to create accessible content. If you encounter platform issues that you believe may affect your ability to provide accessible course content, please contact us at accessibility@edx.org. We also welcome your comments and questions.

In the following sections, we outline guidelines for creating and delivering course content that allows students to use built-in accessibility functionality (such as magnification and zoom features), assistive technologies, and alternative formats. These practices consider learners such as the following:

- Blind learners who use a screen reader, which reads page text aloud, or a Braille display device, which renders page text in Braille.
- Low-vision learners who use screen magnification software to enlarge text and other onscreen content.
- Learners with vision impairments, such as difficulty seeing in low-light conditions, who modify their browser or operating system to change background colors and text settings to make text easier to read.
- Learners with learning disabilities, such as dyslexia, who use text-to-speech technology that reads page content aloud.
- Physically disabled learners who use switching devices, voice recognition software, or eye-gaze activated technology instead of a standard mouse or keyboard to control their computer.
- Learners who modify their operating system settings to make the mouse or keyboard easier to use.
- Learners with hearing impairments who cannot access audio content and need the equivalent information in an alternative format, such as captions or a transcript.

We highly recommend that you implement the best practices shared with you in this document and other widely available resources (some of which are referred to herein). As mentioned above, if you cannot easily address these barriers, we recommend that you consult with any disability-related resources at your institution (Disability Services, Assistive Technology, or Accessibility). While your ability to support students in the MOOC context may be different from supporting on-campus students, we encourage you to develop a plan to respond to students who inform you of accessibility barriers to learning. However, given the large numbers of learners enrolling in many of the courses, you will quickly see how important it is to address accessibility concerns when creating a course.

5.1.2 Accessibility Best Practices

- *Best Practices for Universal Design for Learning*
- *Best Practices for Readability*
- *Best Practices for Accessible PDFs*
- *Best Practices for Custom Content Types*
- *Best Practices for Describing Images*
- *Best Practices for Accessible Media*
- *Best Practices for HTML Markup*

Best Practices for Universal Design for Learning

Universal Design for Learning focuses on delivering courses in a format so that as many of your learners as possible can successfully interact with the learning resources and activities you provide them, without compromising on pedagogic rigor and quality.

The principles of Universal Design for Learning can be summarized as:

1. Present information and content in different ways.

2. Differentiate the ways that students can express what they know.
3. Stimulate interest and motivation for learning.

Instructors can apply these principles in course design by:

- Designing resources and activities that can be accessed by learners in different ways (for example, providing text that allows a student to enlarge it or change color, a diagram with an equivalent text description, or a video with audio and text captions).
- Providing multiple ways for learners to engage with information and demonstrate their knowledge. This is particularly important when developing exercises and assessments.
- Identifying activities that require specific sensory or physical capability, such as activities that require color identification, for which accommodating the specific accessibility needs of students will be difficult or impossible. In these cases, consider whether there is a pedagogical justification for these requirements. If there is a justification, consider communicating these requirements to prospective students in the course description and establish a plan for responding to students who encounter barriers that can be addressed without fundamental alteration. If there is no justification for the requirements, we recommend that you modify the learning activities to be more flexible and broadly accessible.

Resources

- Delivering Accessible Digital Learning (JISC Techdis) provides a useful overview of an inclusive approach to course design: <http://www.jisctechdis.ac.uk/techdis/resources/accessiblecontent>
- The National Center on Universal Design for Learning provides a helpful overview on Universal Design for Learning: <http://www.udlcenter.org/implementation/postsecondary>

Best Practices for Readability

EdX courses have a global and diverse audience. Learners will be better positioned to access the concepts of your content if it is written in clear, straightforward language and the content is well structured. Use appropriate terminology to your subject area, but keep it as clear and unambiguous as possible to help learners who:

- Are not native English speakers; or
- Have a disability that affects reading, such as dyslexia or a visual impairment.

To produce content that is more readable by all students:

- Make the names of elements such as course sections, subsections, units, components, and discussion topics descriptive and easy to skim by putting the important information first in the name. These names are used in navigation menus, page headings, and section headings and are signposts that help learners navigate your course and read course content. “Front-loading” menus and headings particularly helps screen reader users, who can more quickly assess the relevance of a link or heading.
- When creating written learning resources, break text into sections by using HTML elements, such as headings, paragraphs, and lists. Long blocks of unbroken text are a barrier to most readers. Segmented content is more inviting and is easier to navigate and search. See *Best Practices for HTML Markup* for guidance on creating accessible HTML.
- Avoid jargon. If unfamiliar words or phrases are relevant to the subject, explain them when they are first used, and include a glossary with your course materials. When using an abbreviation or acronym, write out the phrase the first time it appears: for example, “National Aeronautics and Space Administration (NASA).”
- Use link text that clearly explains the link destination (for example, “Review the Course Syllabus”). Avoid using constructs like “You can review the Course Syllabus here.” For links that point to documents rather than web pages, include the document type in the link (e.g., “Course Syllabus (PDF)”).

Resources

- The Center for Plain Language provides detailed resources on writing clearly and concisely, in language appropriate for your content and target audience: <http://centerforplainlanguage.org/about-plain-language/checklist/>

Best Practices for Accessible PDFs

PDF is a common format for course materials, including textbooks supplied by publishers. However, converting materials to PDFs can create accessibility barriers, particularly for learners with visual impairments. To improve the accessibility of your PDFs, review the guidance below about preparing documents for conversion, using Adobe Acrobat Professional, and working with third-party suppliers.

Converting Microsoft Office documents to PDF

The teaching materials that you will convert to PDFs may use different formats—for example, your syllabus may be in Word, your presentation slides in PowerPoint, and your textbooks in publisher-supplied PDF. Use the tools available in the applicable software to create well-structured source documents. This early step helps minimize issues that may be difficult or impossible to address later in the conversion process.

Preparing Word documents

- Keep formatting simple. Use headings, paragraphs, lists, images, and captions, and tables for tabular data. Don't add unnecessary indents, rules, columns, blank lines, and typographic variation. The simpler the formatting, the easier it will be to make an accessible PDF document.
- Use styles for formatting your text, such as Normal, Heading 1, and Heading 2, rather than manually formatting text using bold and indents. Add alternative text to images (see *Best Practices for Describing Images*) using Word's picture formatting options.

Preparing PowerPoint documents

- To help make your content accessible and comprehensible to learners who use screen reading software, start in Outline view and include all of your content as text. Add design elements and images after completing the outline, and use PowerPoint's picture formatting options to include detailed descriptions of images that convey information. Avoid adding animations or transitions, as they will not be saved with the PDF format.
- Use the Home > Drawing > Arrange > Selection Pane option to view the reading order of objects on each slide. If the reading order is not logical, reorder the objects.
- Use the Home > Slides > Reset option to give each slide a unique and informative title. The title can be hidden if preferred.
- Identify column headers for any data table using PowerPoint's table formatting options (Tables > Table Options > Header Row), and ensure that each header contains informative text describing the data in that column.

Preparing Excel spreadsheets

- Use a unique and informative title for each worksheet tab.
- Include text alternatives for images (see *Best Practices for Describing Images*) using Excel's picture formatting options.
- Identify column headers using Excel's table formatting options (Table > Table Options > Header Row), and include in each header cell informative text describing the data in that column.
- Do not use blank cells for formatting.
- Use descriptive link text rather than URLs in data cells.

Converting Word, PowerPoint, and Excel documents to PDF

To generate PDFs from Microsoft Office documents, use the **Save as PDF** option. Make sure the **Document Structure Tags for Accessibility** option is selected (consult your software documentation for more details). Note that PDFs generated from Windows versions of Office will be more accessible than those generated from Mac OS.

Working with third-party supplied PDFs

When you control the creation of a PDF, you have greater control over the document's accessibility. If you use PDFs provided by third parties, including textbooks supplied by publishers, the document's accessibility may be unknown.

Asking the right questions about accessible PDFs

Where possible, ask the supplier of the PDF if the PDF is accessible. If it isn't, ask whether the supplier can provide an accessible version. Questions to ask include:

- Can screen readers read the document text?
- Do images in the document include text descriptions?
- Are all tables, charts, and math provided in an accessible format?
- Does all media include text equivalents?
- Does the document have navigational aids, such as a table of contents, index, headings, and bookmarks?

Updating PDFs for accessibility

You may need to update your existing teaching materials in PDF format to improve accessibility. This might include PDFs that were:

- Created by scanning a hard-copy document;
- Generated from a document that was not created with accessibility in mind; or
- Generated by a process that does not preserve source accessibility information.

In such cases, you need special software, such as Adobe Acrobat Professional, to enhance the accessibility of the PDF. PDFs that are created from scanned documents require a preliminary Optical Character Recognition (OCR) step to generate a text version of the document. The procedure checks documents for accessibility barriers, adds properties and tags for document structure, sets the document's language, and adds alternative text for images.

Resources

- Microsoft provides detailed guidance on generating accessible PDFs from Microsoft Office applications, including Word, Excel, and PowerPoint: <http://office.microsoft.com/en-gb/word-help/create-accessible-pdfs-HA102478227.aspx>
- Adobe provides a detailed accessibility PDF repair workflow using Acrobat XI: <http://www.adobe.com/content/dam/Adobe/en/accessibility/products/acrobat/pdfs/acrobat-xi-pdf-accessibility-repair-workflow.pdf>
- Adobe Accessibility (Adobe) is a comprehensive collection of resources on PDF authoring and repair, using Adobe's products: <http://www.adobe.com/accessibility.html>
- PDF Accessibility (University of Washington) provides a step-by-step guide to creating accessible PDFs from different sources and using different applications: <http://www.washington.edu/accessibility/pdf/>
- PDF Accessibility (WebAIM) provides a detailed and illustrated guide on creating accessible PDFs: <http://webaim.org/techniques/acrobat/>
- The National Center of Disability and Access to Education has a collection of one- page "cheat sheets" on accessible document authoring: <http://ncdae.org/resources/cheatsheets/>

- The Accessible Digital Office Document (ADOD) Project provides guidance on creating accessible Office documents: <http://adod.idrc.ocad.ca/>

Best Practices for Custom Content Types

Using different content types can significantly add to the learning experience. We discuss below how to design several custom content types to be accessible to students with disabilities.

Information graphics (charts, diagrams, illustrations)

Although images can be helpful for communicating concepts and information, they present challenges for people with visual impairments. For example, a chart that requires color perception or a diagram with tiny labels and annotations will likely be difficult to comprehend for learners with color blindness or low vision. All images present a barrier to learners who are blind.

The following are best practices for making information graphics accessible to visually impaired students:

- Avoid using only color to distinguish important features of the image. For example, on a line graph, use a different symbol as well as color to distinguish the data elements.
- Whenever possible, use an image format, such as SVG, that supports scaling. Consider providing a high-resolution version of complex graphics that have small but essential details.
- Provide a text alternative that describes the information in the graphic. For charts and graphs, a text alternative could be a table displaying the same data. See *Best Practices for Describing Images* for details about providing text alternatives for images.

Math content

Math in online courses has been challenging to deliver in a way that is accessible to people with vision impairments. Instructors frequently create images of equations rather than including text equations. Math images cannot be modified by people who need a high-contrast display and cannot be read by screen reader software. EdX uses MathJax to render math content in a format that is clear, readable, and accessible to people who use screen readers. MathJax works together with math notation, like LaTeX and MathML, to render mathematical equations as text instead of images. We recommend that you use MathJax to display your math content. You can learn more about using MathJax in the MathJax documentation on accessibility (see the link in “Resources” below). We will update these guidelines as improvements to MathJax are developed.

Simulations and interactive modules

Simulations, including animated or gamified content, can enhance the learning experience. In particular, they benefit learners who may have difficulty acquiring knowledge from reading and processing textual content alone. However, simulations can also present some groups of learners with difficulties. To minimize barriers, consider the intended learning outcome of the simulation. Is it to reinforce understanding that can also come from textual content or a video lecture, or is it to convey new knowledge that other course resources can't cover? Providing alternative resources will help mitigate the impact of any barriers.

Although you can design simulations to avoid many accessibility barriers, some barriers, particularly in simulations supplied by third parties, may be difficult or impossible to address for technical or pedagogic reasons. Understanding the nature of these barriers can help you provide workarounds for learners who are affected. Keep in mind that attempted workarounds for simulations supplied by third parties may require the supplier's consent if copyrighted material is involved.

Consider the following questions when creating simulations, keeping in mind that as the course instructor, you enjoy considerable freedom in selecting course objectives and outcomes. Additionally, if the visual components of a simulation are so central to your course design, providing alternate text description and other accommodations may not be practical or feasible:

- Does the simulation require vision to understand? If so, provide text describing the concepts that the simulation conveys.
- Is the mouse necessary to operate the simulation? If so, provide text describing the concepts that the simulation conveys.
- Does the simulation include flashing or flickering content that could trigger seizures? If so and this content is critical to the nature of the simulation:
 - do not require learners to use the simulation for a required assessment activity; and
 - provide a warning that the simulation contains flickering or flashing content.

As best practices continue to emerge in this area, we will update these guidelines.

Online exercises and assessments

For activities and assessments, consider difficulties students may have in completing an activity and consider using multiple assessment options, keeping in mind that some of the end users have disabilities. Focus on activities that allow students to complete the activity and submit their work without difficulties.

Some students take longer to read information and input responses, such as students with visual or mobility impairments and students who need time to comprehend the information. If an exercise has a time limit, consider whether it's long enough to allow students to respond. Advanced planning may help cut down on the number of students requesting time extensions.

Some online exercise question types may be difficult for students who have vision or mobility impairments. For example:

- Exercises requiring fine hand-eye coordination, such as image mapped input or drag and drop exercises, may present difficulties to students who have limited mobility. Consider alternatives that do not require fine motor skills, unless, of course, such skills are necessary for effective participation in the course. For example, for a drag-and-drop exercise mapping atoms to compounds, provide a checkbox or multiple-choice exercise.
- Highly visual stimuli, such as word clouds, may not be accessible to students who have visual impairments. Provide a text alternative that conveys the same information, such as an ordered list of words in the word cloud.

Third-party content

When including links to third-party content in your course, be mindful as to the accessibility of such third party resources, which may not be readily accessible to learners with disabilities. We recommend that you test any links prior to sharing them with users.

You can use the eReader tool or *Adding Files to a Course* to incorporate third-party textbooks and other publications in PDF format into your course. You can also incorporate such materials into your course in HTML format. See *Best Practices for Accessible PDFs* for guidance on working with third- party supplied PDFs, and *Best Practices for HTML Markup* for guidance on creating accessible HTML.

Resources

- Effective Practices for Description of Science Content within Digital Talking Books, from the National Center for Accessible Media, provides best practices for describing graphs, charts, diagrams, and illustrations: http://ncam.wgbh.org/experience_learn/educational_media/stemdx

- The University of Washington's DO-IT project provides guidance on creating accessible math content: <http://www.washington.edu/doit/Faculty/articles?465>
- AccessSTEM provides guidance on creating accessible science, technology, engineering and math educational content: <http://www.washington.edu/doit/Stem/>
- The National Center on Educational Outcomes (NCEO) provides Principles and Characteristics of Inclusive Assessment and Accountability Systems: <http://www.cehd.umn.edu/nceo/onlinepubs/Synthesis40.html>
- MathJax provides guidance on creating accessible pages with the display engine: <http://www.mathjax.org/resources/articles-and-presentations/accessible-pages-with-mathjax/>

Best Practices for Describing Images

Pictures, diagrams, maps, charts, and icons can present information very effectively. However, some visually impaired students, including people who use screen reader software, need text alternatives to understand the information conveyed by these images. The text alternative for an image depends on the image's context and purpose, and may not be a straight description of the image's visual characteristics.

Use the following guidelines when you include images in your course:

- Provide a short text description that conveys the purpose of the image, unless the image conveys a concept or is the only source for the information it presents, in which case a long text description is appropriate. Note that you don't need to provide a long description if the information appears elsewhere on the page. For example, you don't need to describe a chart if the same data appears as text in a data table.
 - For a representative image, such as a photograph of Ponte Vecchio, a short description could be "Photo of Ponte Vecchio." If the photograph's purpose is to provide detailed information about the location, the long description should be more specific: "Photo of Ponte Vecchio showing its three stone arches and the Arno River."
 - For a chart, diagram, or illustration, the short description might be "Diagram of Ponte Vecchio." The long description should include the details conveyed visually, such as dimensions and materials used.
 - For a map, a short description might be "Map showing location of Ponte Vecchio." If the map is intended to provide directions to the bridge, the long description should provide text directions.
 - For icons, the short description should be the equivalent to the information that the icon provides. For example, for a Course Syllabus link containing a PDF icon, the text equivalent for the icon would be "PDF," which would be read as "Course Syllabus PDF."
 - For an image that serves primarily as a link to another web page, the short description should describe the link's destination, not the image. For example, an image of a question mark that serves as a link to a Help page should be described as "help," not "question mark."
 - Images that don't provide information don't need text descriptions. For example, a PDF icon that is followed by link text reading "Course Syllabus (PDF)" does not need a description. Another example is a banner graphic whose function is purely aesthetic.
- Include the short description in the alt attribute of the HTML image element, as follows (see *Add an Image to an HTML Component* for more information about adding images):

```

```
- Include an empty alt attribute for non-informative images. When image elements do not include an alt attribute, screen reader software may skip the image, announce the image filename, or, in the case of a linked image, announce the link URL. An empty alt attribute tells screen reader software to skip the image.

```

```

- Consider using a caption to display long descriptions so that the information is available to all users. In the following example, the image element includes the short description as the alt attribute and the paragraph element includes the long description.

```
<p>Photo of Ponte Vecchio showing its three stone arches and the Arno river</p>
```

- Alternatively, provide long descriptions by creating an additional unit or downloadable file that contains the descriptive text and providing a link to the unit or file below the image.

```
 <p><a href="description.html">Description of Ponte Vecchio Diagram</a></p>
```

Resources

- A decision tree for choosing appropriate alternative text for images (Dey Alexander):
<http://www.4syllables.com.au/2010/12/text-alternatives-decision-tree/>
- General guidance on appropriate use of alternative text for images (WebAim):
<http://webaim.org/techniques/alttext/>
- HTML5: A more detailed description of techniques for providing useful alternative text for images:
<http://dev.w3.org/html5/alt-techniques/>
- The DIAGRAM Center, established by the US Department of Education (Office of Special Education Programs), provides guidance on ways to make it easier, faster, and more cost effective to create and use accessible images:
<http://www.diagramcenter.org/webinars.html>

Best Practices for Accessible Media

Media-based course materials help convey concepts and bring course information to life. We require all edX courses to use videos with interactive, screen-reader-accessible transcripts. This built-in universal design mechanism helps enhance your course's accessibility. When you create your course, you need to factor in time and resources for creating these transcripts.

Audio transcription

Audio transcripts are essential for presenting audible content to students who can't hear and are helpful to students who are not native English speakers. Synchronized transcripts allow students who can't hear to follow along with the video and navigate to a specific section of the video by clicking the transcript text. Additionally, all students can use transcripts of media-based learning materials for study and review.

A transcript starts with a text version of the video's spoken content. If you created your video using a script, you have a great start on creating the transcript. Just review the recorded video and update the script as needed. Otherwise, you'll need to transcribe the video yourself or engage someone to do it. There are many companies that will create timed video transcripts (i.e., transcripts that synchronize the text with the video using time codes) for a fee.

The edX platform supports the use of transcripts in .srt format. When you integrate a video file into the platform, you should also upload the .srt file of the timed transcript for such video. See *Working with Video Components* for details on how to add timed transcripts.

Video description

When creating video segments, consider how to convey information to learners who can't see. For many topics, you can fully cover concepts in the spoken presentation. If practical, you might also describe visual information, for example, by speaking as you are writing on a tablet.

Downloadable transcripts

For both audio and video transcripts, consider including a text file that students can download and review using tools such as word processing, screen reader, or literacy software. The downloadable transcript should be text only, without time codes.

Resources

- Accessible Digital Media Guidelines provides detailed advice on creating online video and audio with accessibility in mind: http://ncam.wgbh.org/invent_build/web_multimedia/accessible-digital-media-guide

Best Practices for HTML Markup

HTML is the best format for creating accessible content. It is well supported and adaptable across browsers and devices, the information in the markup helps assistive technologies, such as screen reader software, provide information and functionality to people with vision impairments.

To make it easier for our course teams to create content with good HTML markup, we are working to make all templates in edX Studio conform to the best practices set forth below. In the interim, we recommend that you manually add the appropriate HTML tagging. Depending on the type of component you are adding to your course in edX Studio, the raw HTML data will be available either automatically or by selecting the “Advanced Editor” or “HTML” views.

Keep the following guidelines in mind when you create HTML content:

- Use HTML to describe your content’s *meaning* rather than its *appearance*. A phrase marked as a level 1 heading (`<h1>`) clearly indicates the topic of the page, while a phrase marked as bold text (`<bold>` or ``) may be a heading or may just be text that the instructor wants to emphasize. A group of items marked up as a list are related in the code, without relying on visual cues such as bullets and indents. Coding meaning into content is particularly useful for students using screen readers, which, for example, can read through headings or announce the number of items in a list.
- Use HTML heading levels in sequential order to represent the structure of the document. Well-structured headings help students navigate a page and find what they are looking for.
- Use HTML list elements to group related items and make content easier to skim and read. HTML offers three kinds of lists:
 1. Unordered lists, where each item is marked with a bullet.
 2. Ordered lists, where each item is listed with a number.
 3. Definition lists, where each item is represented using term and description pairs (like a dictionary).
- Use table elements to mark up data sets—that is, information that works best in a grid format—with descriptive rows and columns. Mark up row and column headers using the `<th>` element so screen readers can effectively describe the content in the table.

Resources

- Creating Semantic Structure provides guidance on reflecting the semantic structure of a web page in the underlying markup (WebAIM): <http://webaim.org/techniques/semanticstructure/>
- Creating Accessible Tables provides specific guidance on creating data tables with the appropriate semantic structure so that screen readers can correctly present the information (WebAIM): <http://webaim.org/techniques/tables/data>

5.1.3 Conclusion

At edX, the heart of our mission is to provide global access to higher-level learning with only a computer and the Internet. We have designed a platform that enables course creators to reach thousands of learners, some of whom will lack the typical backgrounds and resources of resident students taking traditional courses on college campuses. We hope that these guidelines prove useful to you as you work with your institution's disability support services and information technology resources to comply with applicable accessibility laws. As we are all on this learning venture together, we encourage you to share your thoughts with us at accessibility@edx.org.

5.2 Adding Files to a Course

5.2.1 Overview

To use images in your course content, or to use other documents such as a syllabus, you must add the files to your course.

- *Add a File*
- *File URLs*
- *Sort Files*
- *Find Files*
- *Lock a File*
- *Delete a File*

5.2.2 Add a File

You can add files that you want students to access in the course. After you add a file, you must link to it from a component, a course update, or in the course handouts. A file is only visible to students if you create a link to it.

Note: Because the file name becomes part of the URL, students can see the name of the file when they open it. Avoid using file names such as AnswerKey.pdf.

Warning: If you upload a file with the same name as an existing course file, the original file is overwritten without warning.

To add files:

1. From the **Content** menu, select **Files & Uploads**.
2. Click **Upload New File**.
3. In the **Upload New File** dialog box, click **Choose File**.
4. In the **Open** dialog box, select one more files that you want to upload, then click **Open**.
5. To add more file, click **Load Another File** and repeat the previous step.
6. To close the dialog box, click the **x** in the top right corner.

When you close the dialog box, the new files appear on the **Files & Uploads** page.

5.2.3 File URLs

In the Files & Uploads page, each file has has an **Embed URL** and an **External URL**:

Preview	Name	Date Added	Embed URL	External URL
	first_course.png	Feb 25, 2014 at 16:18 UTC	/static/first_course.png	mhoeber.m.sandbox.ed:
	display-name.png	Feb 25, 2014 at 16:18 UTC	/static/display-name.png	mhoeber.m.sandbox.ed:

- You use the **Embed URL** to link to the file or image from a component, a course update, or a course handout.
- You use the **External URL** to reference the file or image from outside of your course. The external URL does not work if you lock the file unless the person accessing the URL is enrolled in the course.

Warning: You cannot use the External URL as the reference to a file or image from within your course.

You can double click a value in the **Embed URL** or **External URL** column to select the value, then copy it.

5.2.4 Sort Files

By default, files are sorted by the **Date Added** column, with the most recently added first.

Alternatively, can also sort the list by the **Name** column by clicking on the column header.

For either the the **Date Added** or **Name** column, you can switch the sort order from descending to ascending, and back, by clicking the column header a second time.

The current sort order is shown at the top of the file list, and the active sort column header is underlined:

Preview	Name	Date Added	Embed URL	External URL
	first_course.png	Feb 25, 2014 at 16:18 UTC	/static/first_course.png	mhoeber.m.sandbox.ed:
	display-name.png	Feb 25, 2014 at 16:18 UTC	/static/display-name.png	mhoeber.m.sandbox.ed:

5.2.5 Find Files

The **Files & Uploads** page lists up to 50 files. If your course has more than 50 files, additional files are listed on other pages.

The range of the files listed on the page, and the total number of files, are shown at the top of the page.

You can navigate through the pages listing files in two ways:

- Use the < and > buttons at the top and bottom of the list to navigate to the previous and next pages.
- At the bottom of the page, enter the page number to skip to, then tab out of the field:



5.2.6 Lock a File

By default, anyone can access a file you upload if they know the URL, even people not enrolled in your class.

To ensure that those not in your class cannot view the file, click the lock icon.

Note: The external URL does not work if you lock the file.

5.2.7 Delete a File

To delete a file, click the x icon next to the file. You are prompted to confirm the deletion.

Warning: If you have links to a file you delete, those links will be broken. Ensure you change those links before deleting the file.

5.3 Adding Course Updates and Handouts

You add course updates and handouts in Studio.

Students see the course updates and handouts in the **Course Info** tab in your course:

Courseware Course Info Discussion Wiki Progress Open Ended Panel Course Handouts Instructor Staff view

Course Updates & News

SEPTEMBER 30, 2013

Welcome to edX101-beta!

This self-paced course is designed to walk you through the process of planning, building, and running your own online course. There are no grades for this course, and you can start and stop at any time. We encourage you to take all the time you need to learn about adapting your traditional course to an online environment.

As you've undoubtedly noticed if you've been around for a while, we're continually updating edX101 to better meet your needs. To see what changed in the most recent release, check out the [New Features](#) tab at the top of the page. (When you're done, click the [Course Info](#) tab to come back here, or click the [Courseware](#) tab to see the lessons in the course.)

Check out the Stable Marriage Problem homework assignment, added September 30!

Course Handouts

- Overview of Creating an Online Course
- edX101 - June 2013 Archive
- Suggestions and Bugs
- edX Edge and Studio Help
- Course Readiness Checklists
- Studio Documentation (pdf)
- Printable, searchable version of edX101

5.3.1 Add a Course Update

You add updates to notify students of exams, changes in the course schedule, or anything else of a more urgent nature.

To add a course update:

1. From the **Content** menu, select **Updates**.
2. Click **New Update**.
3. Enter your update in the HTML editor that opens.

Note: You must enter the update in HTML.

4. Click **Save**.

5.3.2 Add Course Handouts

You can add course handouts that are visible to students on the **Course Info** page. To add an uploaded file to the course handouts, you will need its URL.

Note: You must *Adding Files to a Course* before you can add them as course handouts.

1. From the **Content** menu, select **Updates**.
2. In the **Course Handouts** panel, click **Edit**.
3. Edit the HTML to add links to the files you uploaded. See *Add a Link in an HTML Component* for more information.
4. Click **Save**.

5.4 Adding Textbooks

You can add PDF textbooks for your course.

Note: Do not use image files (for example, .PNG files) as textbooks for your course, as they are not accessible to screen readers. Review the *Best Practices for Accessible PDFs* for more information.

Each textbook that you add is displayed to students as a tab in the course navigation bar.

It's recommended that you upload a separate PDF file for each chapter of your textbook.

When students open the textbook tab in the course, they can navigate the textbook by chapter:

The screenshot shows a navigation bar with tabs: Courseware, Course Info, Discussion, Advice & Schedule, H24H, Sourcebook (which is highlighted), Progress, My Notes, FAQ, and Discussion Board Guidelines. Below the navigation bar, there is a sidebar with a red box around the 'Title Page' section, which lists 'Homeric Iliad: Scroll I' through 'Homeric Iliad: Scroll VI'. To the right of the sidebar, the main content area shows the title 'HOMERIC ILIAD SCROLL I', author information ('Translated by Samuel Butler' and 'Revised by Soo-Young Kim, Kelly McCray, Gregory Nagy, and Timothy Power'), and a sample text excerpt: '[1] Anger [mēnis], goddess, sing it, of Achilles son of Peleus - 2 disastrous [ouλomenē] anger that made countless pains [algea] for the Achaeans, 3 and many steadfast lives [psūkhai] it drove down to Hādēs, 4 heroes' lives, but their bodies it made prizes for dogs'.

To add a textbook:

1. From the **Content** menu, select **Textbooks**.
2. Click **New Textbook**. The following screen opens:

The screenshot shows a form for adding a new textbook. The 'Textbook Name *' field contains 'Introduction to Cookie Baking'. Below it is a note: 'provide the title/name of the text book as you would like your students to see it'. The 'Chapter Name *' field contains 'Chapter 1'. To its right is the 'Chapter Asset *' field, which has 'path/to/introductionToCookieBaking-CH1.pdf' and an 'Upload PDF' button. Below the asset field is a note: 'upload a PDF file or provide the path to a Studio asset file'. At the bottom of the form are 'SAVE' and 'CANCEL' buttons, and a '+ Add a Chapter' button.

3. Enter the **Textbook Name**.
4. Enter the first **Chapter Name**.
5. To upload a PDF file from your computer, click **Upload PDF**. Follow the prompts to upload your file.
6. To add more chapters, click **+Add a Chapter** and repeat steps 3 and 4.
7. Click **Save**.

5.5 Adding Pages to a Course

5.5.1 Overview

By default, your course has the following pages:

- Courseware
- Course Info
- Discussion
- Wiki
- Progress

You cannot rename, reorder, or remove these pages.

You can add pages to your course. Each page appears in your course's navigation bar.

For example, the following navigation bar includes the default pages as well as the **Course Schedule** and **Supplements & Instructor's Blog** pages.



You can create other pages for the grading policy, course slides, or any other purpose. More examples of pages you can add are:

- A Google calendar, by embedding the code for it.
- A dynamic HTML calendar, using the template in *Code for Dynamic HTML Schedule*.
- An instant hangout. See *Google Instant Hangout Tool* for more information.

See:

- [Add a Page](#)
- [Show or Hide the Course Wiki Page](#)
- [Reorder Pages](#)
- [Delete a Page](#)
- [Code for Dynamic HTML Schedule](#)

5.5.2 Add a Page

1. In Studio, from the **Content** menu, select **Pages**.

Pages

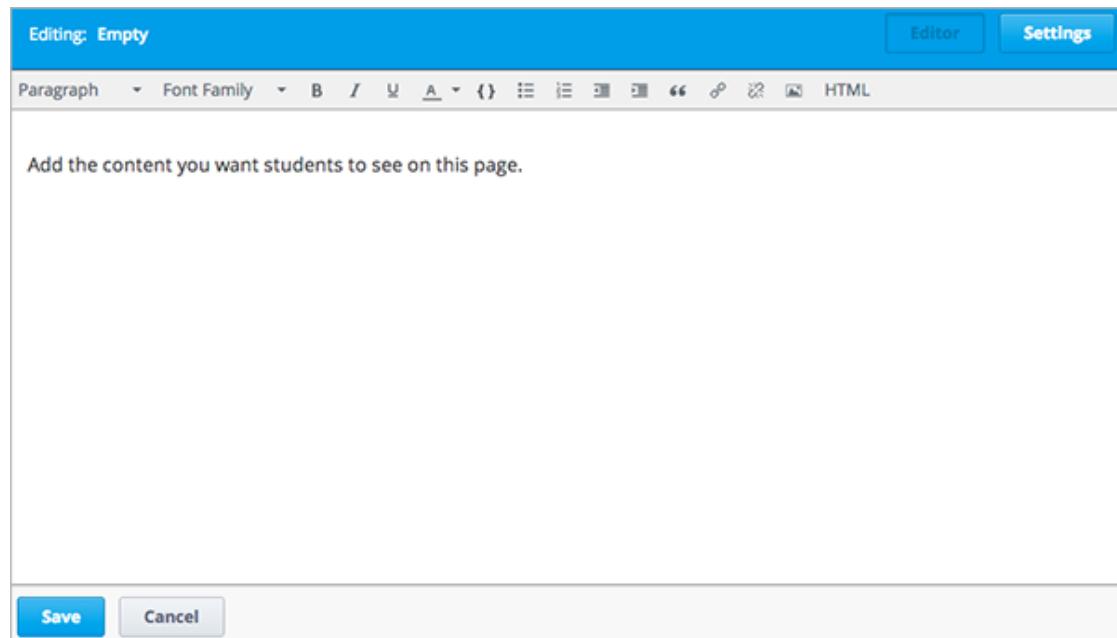
Courseware	
Course Info	
Discussion	⋮
Wiki	👁️ ⋮
Progress	⋮

2. Click **Add a New Page**. A page with the title **Empty** is added to the list:

Pages

Courseware	
Course Info	
Discussion	⋮
Wiki	👁️ ⋮
Progress	⋮
Empty	☒ Edit ⌂ ⋮

3. Click **Edit**. The HTML editor opens.



4. Enter text for your page. See *Options for Editing HTML Components* for more information about using the editor.
5. Click **Settings** to edit the **Display Name**. The display name is the name of the page visible to students in the course.
6. Click **Save**.

The new page is immediately available to students, if the course has started.

5.5.3 Show or Hide the Course Wiki Page

By default, your course includes a Wiki page. Students and course staff can use the Wiki to post content and comment on others' content.

If you do not want to use the Wiki in your course, you can hide the page.

The eye icon in the Wiki object indicates that the Wiki page is visible in your course:



Click the eye icon to hide the Wiki page. The icon changes:



Click it again to make the Wiki page visible.

Note: Content remains in the Wiki when you hide the page. For example, if a student bookmarks a Wiki topic, then you hide the Wiki page, the student can still use the bookmark to access that Wiki topic. All content that was previously posted in the Wiki remains available after you hide the Wiki page, and any students logged in to edX can access the content if they know the URL.

5.5.4 Reorder Pages

You can reorder pages in your course by dragging and dropping the pages to different locations.

To move a page, hover over the element handle on the right side of the page row until the mouse pointer changes to a four-headed arrow. Then, click and drag the page to the location that you want.

Note: You cannot reorder the Courseware, Course Info, Discussion, Wiki, and Progress pages that your course includes by default

5.5.5 Delete a Page

To delete a page that you previously added, click the trash can icon in the row for the page. You are prompted to confirm the deletion.

5.5.6 Code for Dynamic HTML Schedule

You can use the following code in a page to provide a dynamic HTML schedule in your course.

```
<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>
        <td class="due"> Due </td>
    </thead>

    <!-- Week 1 Row 1 -->
    <tr class="first">
        <td class="day">10/22</td>
        <td class="topic">Topic 1</td>
        <td class="reading">Ch. 1</td>
        <td class="video"><a href="#">L1: Title</a></td>
        <td class="slides"><a href="#">L1</a></td>
        <td class="assignment"><a href="#">HW 1</a></td>
        <td class="due">11/04</td>
    </tr>

    <!-- Week 1 Row 2 -->
    <tr>
        <td class="day"> </td>
```

```
<td class="topic"></td>
<td class="reading"></td>
<td class="video"><a href="#">L2: Title</a></td>
<td class="slides"><a href="#">L2</a></td>
<td class="assignment"> </td>
<td class="due"> </td>
</tr>

<tr> <td class="week_separator" colspan=7> <hr/> </td> </tr>

<!-- Week 2 Row 1 -->
<tr>
    <td class="day">10/29</td>
    <td class="topic">Topic 2</td>
    <td class="reading">Ch. 2</td>
    <td class="video"> <a href="#">L3: Title</a></td>
    <td class="slides"><a href="#">L3</a></td>
    <td class="assignment"><a href="#">Quiz 1</a></td>
    <td class="due">11/11</td>
</tr>

<!-- Week 2 Row 2 -->
<tr>
    <td class="day"></td>
    <td class="topic"></td>
    <td class="reading"></td>
    <td class="video"><a href="#">L4: Title</a></td>
    <td class="slides"><a href="#">L4</a> </td>
    <td class="assignment"></td>
    <td class="due"></td>
</tr>
<tr> <td class="week_separator" colspan=7> <hr/> </td> </tr>

<!-- Week 3 Row 1 -->
<tr>
    <td class="day">11/05</td>
    <td class="topic">Topic 3</td>
    <td class="reading">Ch. 3</td>
    <td class="video"><a href="#">L5: Title</a></td>
    <td class="slides"><a href="#">L5 </a></td>
    <td class="assignment"><a href="#">HW 2</a></td>
    <td class="due">11/18 </td>
</tr>

<!-- Week 3 Row 2 -->
<tr>
    <td class="day"> </td>
    <td class="topic"> </td>
    <td class="reading"></td>
    <td class="video"><a href="#">L6: Title</a></td>
    <td class="slides"><a href="#">L6 </a></td>
    <td class="video"></td>
    <td class="assignment"></td>
    <td class="due"></td>
</tr>
<tr> <td class="week_separator" colspan=7> <hr/> </td> </tr>

<!-- Week 4 Row 1 -->
```

```
<tr>
    <td class="day">11/12</td>
    <td class="topic">Topic 4</td>
    <td class="reading">Ch. 4</td>
    <td class="video"><!--<a href="#">L7: Title</a>--> L7: Title</td>
    <td class="slides"><!--<a href="#">L7</a>-->L7</td>
    <td class="assignment"><!--<a href="#">Quiz 2</a>-->Quiz 2</td>
    <td class="due"> 11/25 </td>
</tr>

<!-- Week 4 Row 2 -->
<tr>
    <td class="day"></td>
    <td class="topic"></td>
    <td class="reading"></td>
    <td class="video"><!--<a href="#">L8: Title</a>-->L8: Title</td>
    <td class="slides"><!--<a href="#">L8</a>-->L8</td>
    <td class="assignment"></td>
    <td class="due"></td>
</tr>
<tr> <td class="week_separator" colspan=7> <hr/> </td> </tr>

<!-- Week 5 Row 1 -->
<tr>
    <td class="day">11/19</td>
    <td class="topic">Topic 5</td>
    <td class="reading">Ch. 5</td>
    <td class="video"><!--<a href="#">L9: Title</a>-->L9: Title</td>
    <td class="slides"><!--<a href="#">L9</a>-->L9</td>
    <td class="assignment"><!--<a href="#">HW 3</a>-->HW 3</td>
    <td class="due"> 12/02 </td>
</tr>

<!-- Week 5 Row 2 -->
<tr>
    <td class="day"></td>
    <td class="topic"></td>
    <td class="reading"></td>
    <td class="video"><!--<a href="#">L10: Title</a>-->L10: Title</td>
    <td class="slides"><!--<a href="#">L10</a>-->L10 </td>
    <td class="assignment"></td>
    <td class="due"></td>
</tr>
<tr> <td class="week_separator" colspan=7> <hr/> </td> </tr>

<!-- Week 6 Row 1 -->
<tr>
    <td class="day">11/26</td>
    <td class="topic">Topic 6</td>
    <td class="reading">Ch. 6</td>
    <td class="video"><!--<a href="#">L11: Title</a>-->L11: Title </td>
    <td class="slides"><!--<a href="#">L11</a>-->L11</td>
    <td class="assignment"><!--<a href="#">HW 4</a>-->HW 4</td>
    <td class="due">12/09</td>
</tr>

<!-- Week 6 Row 2 -->
<tr>
```

```
<td class="day"> </td>
<td class="topic"> </td>
<td class="reading"></td>
<td class="video"><!--<a href="#">L12: Title</a>-->L12: Title</td>
<td class="slides"><!--<a href="#">L12</a>-->L12</td>
<td class="assignment"></td>
<td class="due"> </td>
</tr>

</table>
</div>
```

5.6 Organizing Your Course Content

5.6.1 How a Course is Organized

You organize your course in the following hierarchy:

- *Sections*, which contain
- *Subsections*, which contain
- *Units*, which contain
- *Components*, which contain your actual course content.

Studio provides you with flexibility when organizing your course. A common course model is for sections to correspond to weeks or chapters, and for subsections to correspond to lessons.

Note: We recommend that you review *Guidelines for Creating Accessible Content* before developing content for your course.

The Course Outline

In Studio, you view your course organization through the course outline.

To open the course outline, on the **Content** menu, click **Outline**.

The following image shows the different elements of a course outline:

The screenshot shows the 'Course Outline' section of the edX platform. It displays two main sections: 'Section 1' and 'Section 2'. Each section contains a list of units and subsections. A red bracket labeled 'Units' points to the list of items under each section header. Another red bracket labeled 'Subsections' points to the expandable arrows next to the unit names.

Section 1:

- Introduction: A Tour of EdX
- What Does an edX Course Look Like?
- Studio Tutorial
- Advanced Studio Tutorial (optional)
 - Roadmap for the Advanced Studio Tutorial
 - Add a Link to a Course Unit
 - Add an Image to a Unit
 - Add an Advanced Problem: Image Mapped Input
 - Specify Video Settings
 - Add Transcripts to Videos
 - Seed a Discussion Space
 - Create a Mini-Course
- Homework 1

Section 2:

- Structuring Your Course
- Your Course Info
- Your About Page

The following image shows the way this course content appears in the LMS:

The screenshot shows the course content as it appears in the LMS. On the left, there is a sidebar with 'Sections' listed: 'Introduction: A Tour of EdX', 'What Does an edX Course Look Like?', 'Studio Tutorial', 'Advanced Studio Tutorial (optional)', and 'Homework 1'. A red bracket labeled 'Sections' points to this sidebar. On the right, there is a main content area titled 'Roadmap for the Advanced Studio Tutorial' with a sub-section titled 'Subsections'. A red bracket labeled 'Units' points to the title of this section. The main content area contains a list of instructions and links:

Roadmap for the Advanced Studio Tutorial
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.

5.6.2 Sections

A section is the topmost category in your course. A section can represent a time period in your course, a chapter, or another organizing principle. A section contains one or more subsections.

To create a section:

1. In the course outline, click **New Section**.
2. In the field that opens at the top of the outline, enter the new section name.
3. Click **Save**.

The new, empty section appears at the bottom of the course outline. You can now add subsections to the section.

Whether or not students see the new section depends on the release date.

See *Publishing Your Course* for more information.

5.6.3 Subsections

Sections are divided into subsections, which in turn contain one or more units. A subsection may represent a topic in your course, or another organizing principle. Subsections are sometimes called “lessons” or “learning sequences.”

To create a subsection:

1. On the **Course Outline** page, under the name of the section that you want, click **New Subsection**.
2. In the field that opens at the bottom of the section, enter the new subsection name.
3. Click **Save**.

The new, empty subsection appears at the bottom of the section. You can then add units to the subsection.

Edit a Subsection

You add and delete subsections from the **Course Outline** page. You edit a subsection’s settings on the page for that subsection. To open the page for a subsection, click the name of the subsection on the **Course Outline** page.

On the page for the subsection, you can see all the units in the subsection.

The screenshot shows the 'Subsection Settings' interface for a subsection named 'Lesson 2 - Let's Get Interactive!'. The left panel lists four units: 'Lesson 2 - Let's Get Interactive!', 'An Interactive Reference Table', 'Zooming Diagrams', and 'Electronic Sound Experiment'. Each unit has a trash icon and a three-dot menu icon. Below the units is a button labeled '+ New Unit'. The right panel displays subsection settings, including 'Release Day' (01/01/2014) and 'Release Time (UTC)' (05:00). It also shows 'Graded as: NOT GRADED' and a 'SET A DUE DATE' link. At the bottom are 'Preview Drafts' and 'View Live' buttons.

You can also do the following.

- Change the subsection name
- Add or delete a unit
- Set the subsection release date and time
- Set the subsection to be an assignment type
- Set a due date for the exercises in the subsection (if you set the assignment type of the subsection)
- Preview a draft of the subsection
- View the live version of your course

Whether students can see a subsection depends on its release date. For more information, see *Publishing Your Course*.

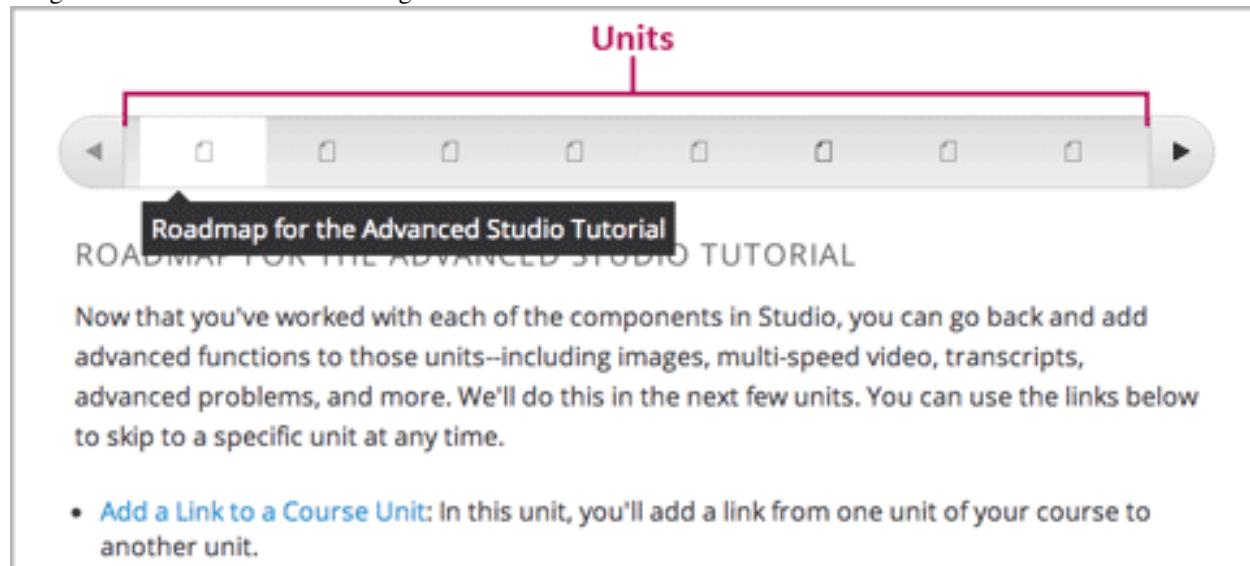
For more information about setting assignment types, see *Establishing a Grading Policy*.

For more information about viewing drafts or the current version of your course, see *Preview Your Course* and *Switch Between Studio and Your Live Course*.

5.6.4 Units

Subsections are divided into units. A unit, in turn, contains one or more components, such as HTML content, problems, and videos.

In the LMS, each unit in the subsection appears as a link on the course ribbon at the top of the page. The following image shows a subsection that has eight units:



The titles in the pop-up dialog, in white text on a black background, are the display names of the components in that unit.

You can create a unit from the course outline or the subsection page. To create a unit, click **New Unit** within the subsection that you want.

When you create a new unit, a page opens for that unit. On the unit page, you can do the following.

- Enter the unit name that you want in the **Display Name** field. Note that students do not see the unit name.
- Create components in the unit.

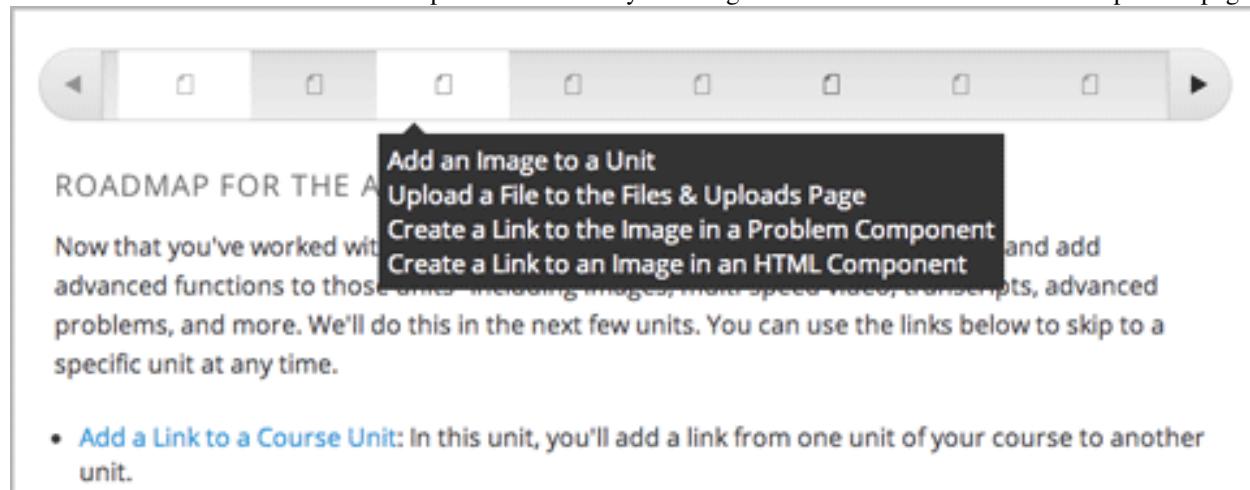
- Change the visibility of the unit. By default, the unit visibility is set to **Private**, so students will not be able to see the unit. Unless you want students to see the unit immediately, leave this setting as-is. For more information, see *Publishing Your Course*.
- Preview a draft of the unit.
- If a unit has been published, view the live version of the unit.

After you create a new unit, you can create components in the unit.

Warning: Studio does not have versioning and does not automatically update your browser between refreshes. Versioning is planned for future releases, but, in the meantime, only one author should edit a unit, in one browser, on only one tab. If a unit is open for editing in multiple browser sessions, the session that saves last will overwrite any previously saved content without displaying a warning. Also, older browser sessions can overwrite more recent content, so you should refresh your browser before you start working every time you work with a private unit or edit a draft of a public unit.

5.6.5 Components

A component is the part of a unit that contains your actual course content. A unit can contain one or more components. A student can view the name of all components in a unit by hovering over the unit in the ribbon at the top of the page.



The screenshot shows a portion of the edX Studio interface. At the top, there's a horizontal ribbon with several small square icons. Below the ribbon, the text "ROADMAP FOR THE A" is visible. A tooltip is overlaid on the text, containing the following items:

- Add an Image to a Unit
- Upload a File to the Files & Uploads Page
- Create a Link to the Image in a Problem Component
- Create a Link to an Image in an HTML Component

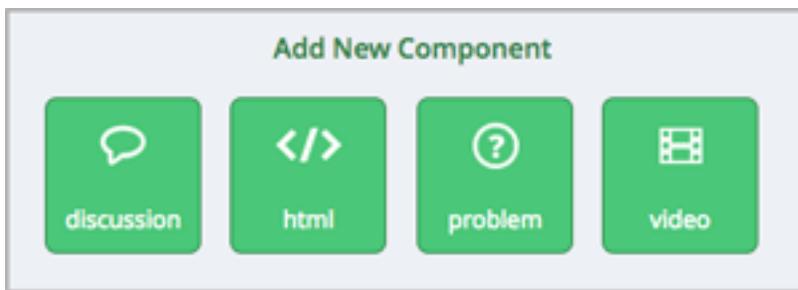
Below the tooltip, the main text continues: "Now that you've worked with advanced functions to those units—including images, multi-speed video, transcripts, advanced problems, and more. We'll do this in the next few units. You can use the links below to skip to a specific unit at any time." At the bottom of the visible area, there's a bulleted list:

- [Add a Link to a Course Unit](#): In this unit, you'll add a link from one unit of your course to another unit.

By default, Studio includes four types of components.

- **Discussion components** provide discussion spaces in the body of your course. Students can explore ideas about a lesson with their peers in a discussion space.
- **HTML components** allow you to add text, images, and some types of learning tools to your course. Content in HTML components is formatted as HTML.
- **Problem components** enable you to add many different types of exercises and problems to your course, from simple multiple choice problems to complex circuit schematic exercises.
- **Video components** contain the videos that you want to include in your course.

To add a component to the unit, click the component type that you want under **Add New Component**.



For more information, see the documentation for the specific component type that you want:

- *Working with Discussion Components*
- *Working with HTML Components*
- *Working with Problem Components*
- *Working with Video Components*

Edit a Component

You can edit a component only if the unit that contains it is Private, or if you are editing a new draft of the unit. For more information, see *Public and Private Units*.

When you can edit the component, the Edit icon and the Display Name appear in the component header:

The screenshot shows a video player interface. At the top left, the display name "Building a Computer Memory Element" is circled in red. At the top right, there is an "EDIT" button, also circled in red. Below the video frame, the title "BUILDING A COMPUTER MEMORY ELEMENT" is visible. The video frame itself contains a hand-drawn diagram of a memory cell with the text "Building a memory element ...", "Ⓐ First attempt", and a play button icon. To the right of the video frame, there is a transcript in blue text:

SPEAKER 1: OK, let's dive right in and try to build this abstract memory cell that I just showed you earlier. OK, so here's my memory cell. It has an input dIN, output dOUT. And it has a store signal, OK? And recall the property. Its property is that when the store signal is high, the

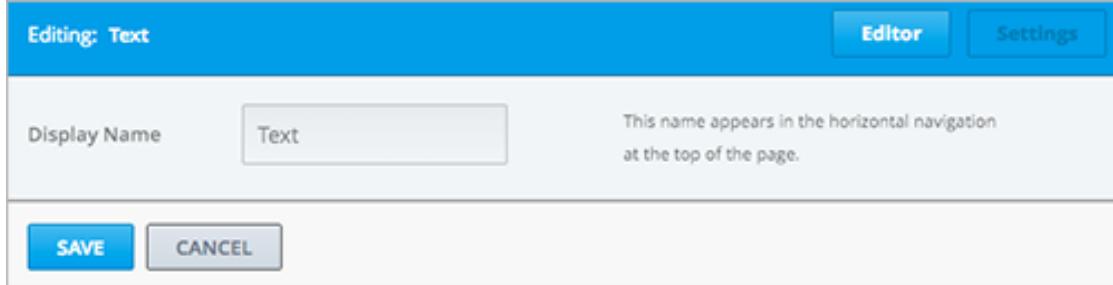
Click the Edit icon. Then follow instructions for the type of component you are editing.

Set the Display Name for a Component

Each component has a display name that shows in the component header when you can edit the component. The display name also shows to students when they hover the mouse pointer over the unit icon in the course accordion.

To set the display name for a component:

1. Edit the component.
2. Click **Settings**.
3. Edit the **Display Name** field.



4. Click **Save**.

Different types of components have different fields in the Settings dialog box, but all have the **Display Name** field.

Duplicate a Component

You can duplicate a component only if the unit that contains it is Private, or if you are editing a new draft of the unit. For more information, see *Public and Private Units*.

When you duplicate a component, a new copy of that component is added directly beneath the first component. You can then modify the duplicate. In many cases, duplicating a component and editing a copy is a faster way to create new content.

When you can duplicate the component, the Duplicate icon appears in the component header:

Building a Computer Memory Element   

BUILDING A COMPUTER MEMORY ELEMENT



Building a memory element ...

(A) First attempt

* 1

SPEAKER 1: OK, let's dive right in and try to build this abstract memory cell that I just showed you earlier. OK, so here's my memory cell. It has an input dIN, output dOUT. And it has a store signal, OK? And recall the property. Its property is that when the store signal is high, the

0:00 / 2:11 SPEED 1.0x HD cc

Click the Duplicate icon. Then follow instructions for the type of component you are editing.

Delete a Component

You can delete a component only if the unit that contains it is Private, or if you are editing a new draft of the unit. For more information, see *Public and Private Units*.

Note: Be sure you want to delete the component. You can not undo the deletion.

When you can delete the component, the Delete icon appears in the component header:

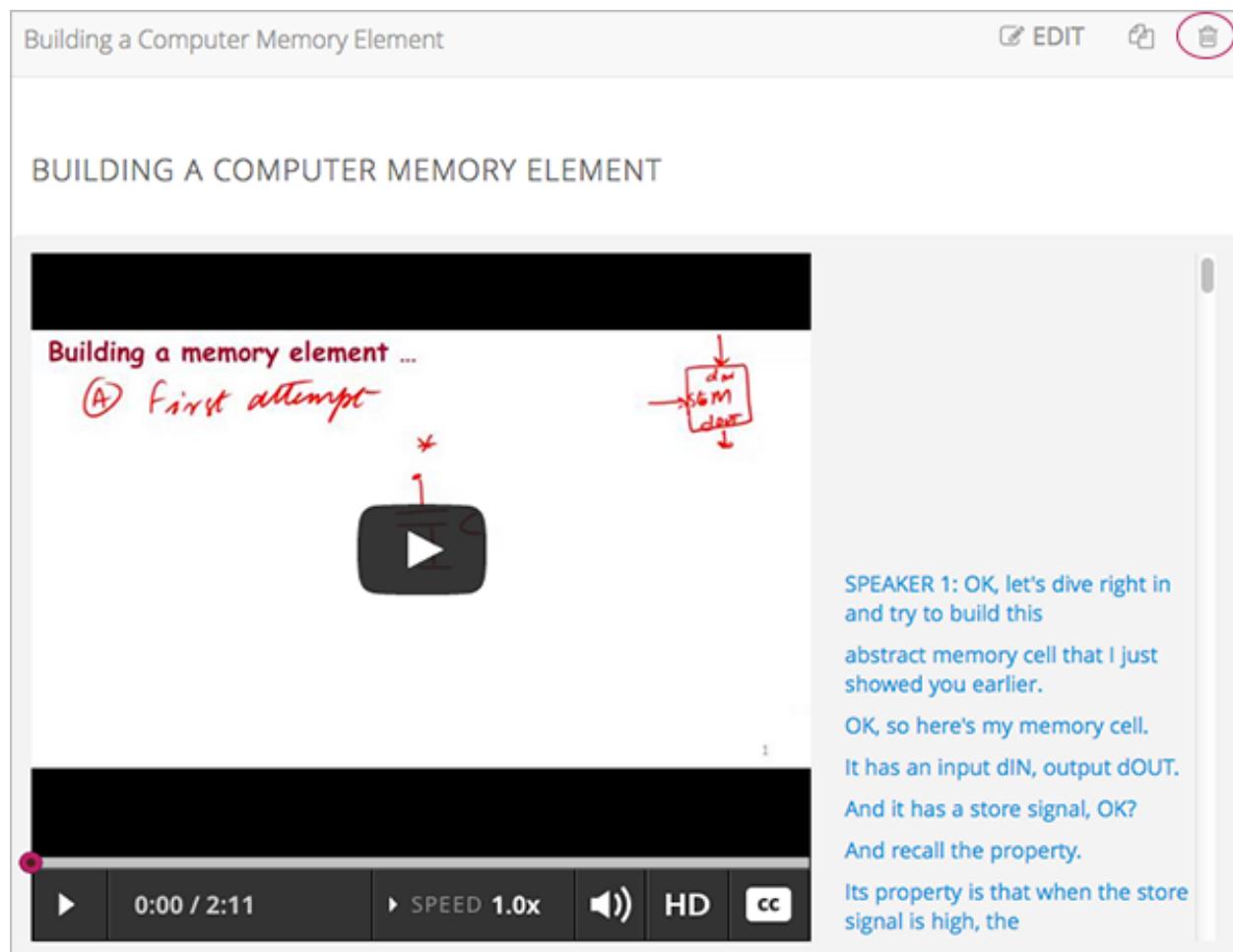
Building a Computer Memory Element

BUILDING A COMPUTER MEMORY ELEMENT

Building a memory element ...
Ⓐ First attempt

* 1

SPEAKER 1: OK, let's dive right in and try to build this abstract memory cell that I just showed you earlier. OK, so here's my memory cell. It has an input dIN, output dOUT. And it has a store signal, OK? And recall the property. Its property is that when the store signal is high, the



Click the Delete icon. Then follow instructions for the type of component you are editing.

5.6.6 Reorganize Your Course

You can reorganize your course by dragging and dropping sections, subsections, units, and components. You can move elements on the **Course Outline** page or on an individual unit page.

To move an element, hover over the element handle on the right side of the screen until the mouse pointer changes to a four-headed arrow. Then, click and drag the element to the location that you want.

▼ Introduction: A Tour of EdX

Release date: Nov 17, 2012 at 17:37 UTC

- ▶ What Does an edX Course Look Like? **Lesson** ✓ ⏪ ⏴
- ▶ Studio Tutorial ✓ ⏪ ⏴
- ▼ Advanced Studio Tutorial (optional)
 - Roadmap for the Advanced Studio Tutorial ⏪ ⏴
 - Add a Link to a Course Unit ⏪ ⏴
 - Add an Image to a Unit **Add** ⏪ ⏴
 - Add an Advanced Problem: Image Mapped Input ⏪ ⏴
 - Specify Video Settings ⏪ ⏴
 - Add Transcripts to Videos ⏪ ⏴
 - Seed a Discussion Space ⏪ ⏴
 - Create a Mini-Course ⏪ ⏴
- + New Unit

▶ Homework 1

+ New Subsection

When you move an element, a blue line indicates where the element will land when you release the mouse.

The screenshot shows a list of components in a Studio Tutorial interface:

- Studio Tutorial
- Advanced Studio Tutorial (optional)
 - Roadmap for the Advanced Studio Tutorial
 - Add a Link to a Course Unit (highlighted with a blue border)
 - Add an Image to a Unit
- Add an Advanced Problem: Image Mapped Input
- Specify Video Settings

5.7 Working with HTML Components

5.7.1 HTML Component Overview

HTML, or HyperText Markup Language, is the standard markup language used to create web pages. Web browsers present HTML code in a more readable format. When students see text and images in your course, they are seeing HTML code that is formatted and presented by their browsers. For more information about HTML, see [Wikipedia](#).

HTML components are the basic building blocks of your course content. You use HTML components to add and format text, links, images, and more. You can choose to create HTML components directly in HTML code, or using a Visual editor that hides the HTML code details, as described below.

For more information, see the following topics:

- *Options for Editing HTML Components*
- *The Visual Editor*
- *The Raw HTML Editor*
- *HTML Component Templates*
- *Create an HTML Component*
- *Add a Link in an HTML Component*
- *Add an Image to an HTML Component*
- *Import LaTeX Code into an HTML Component*

Note: Review *Organizing Your Course Content* and *Best Practices for HTML Markup* before you start working with HTML components.

To add an instant hangout to an HTML component, see *Google Instant Hangout Tool*.

5.7.2 Options for Editing HTML Components

You can work with HTML in two ways:

- *The Visual Editor*

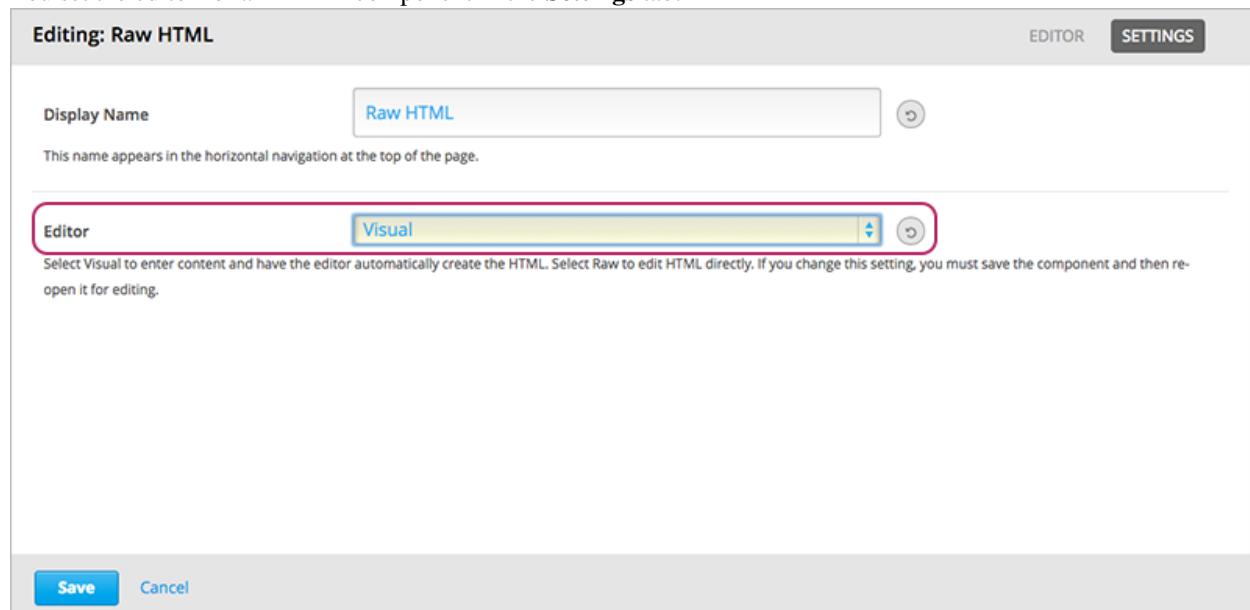
With the Visual editor you can create, edit, and format content in a word processing-like interface, without using HTML code directly. With the Visual editor, you can more easily format your content, and add links and images. The Visual editor provides access to HTML code so you can make small changes to formatting, if required. However, the HTML view in the Visual editor does not provide the detailed control you can get with the Raw HTML editor, and does not support custom formatting or scripts.

- *The Raw HTML Editor*

With the Raw HTML Editor, you work directly with HTML code. If you need to use custom formatting or scripts in your content, you should use the Raw HTML Editor.

Set the Editor for an HTML Component

You set the editor for an HTML component in the **Settings** tab:



Select **Visual** or **Raw**. When you change the editor, you must click **Save** and re-open the component to begin using the new editor.

Warning: If you work with content in the Raw HTML editor, then switch to the Visual editor, you may lose custom HTML that you created. Therefore, it is recommended that you start by using the Visual editor, then switch to the Raw HTML editor when you need to create custom HTML.

5.7.3 The Visual Editor

The Visual editor provides a “what you see is what you get” (WYSIWYG) interface that allows you to format text by clicking the formatting buttons at the top of the editor.

The following image shows call-outs for the editing options and is followed by descriptions.

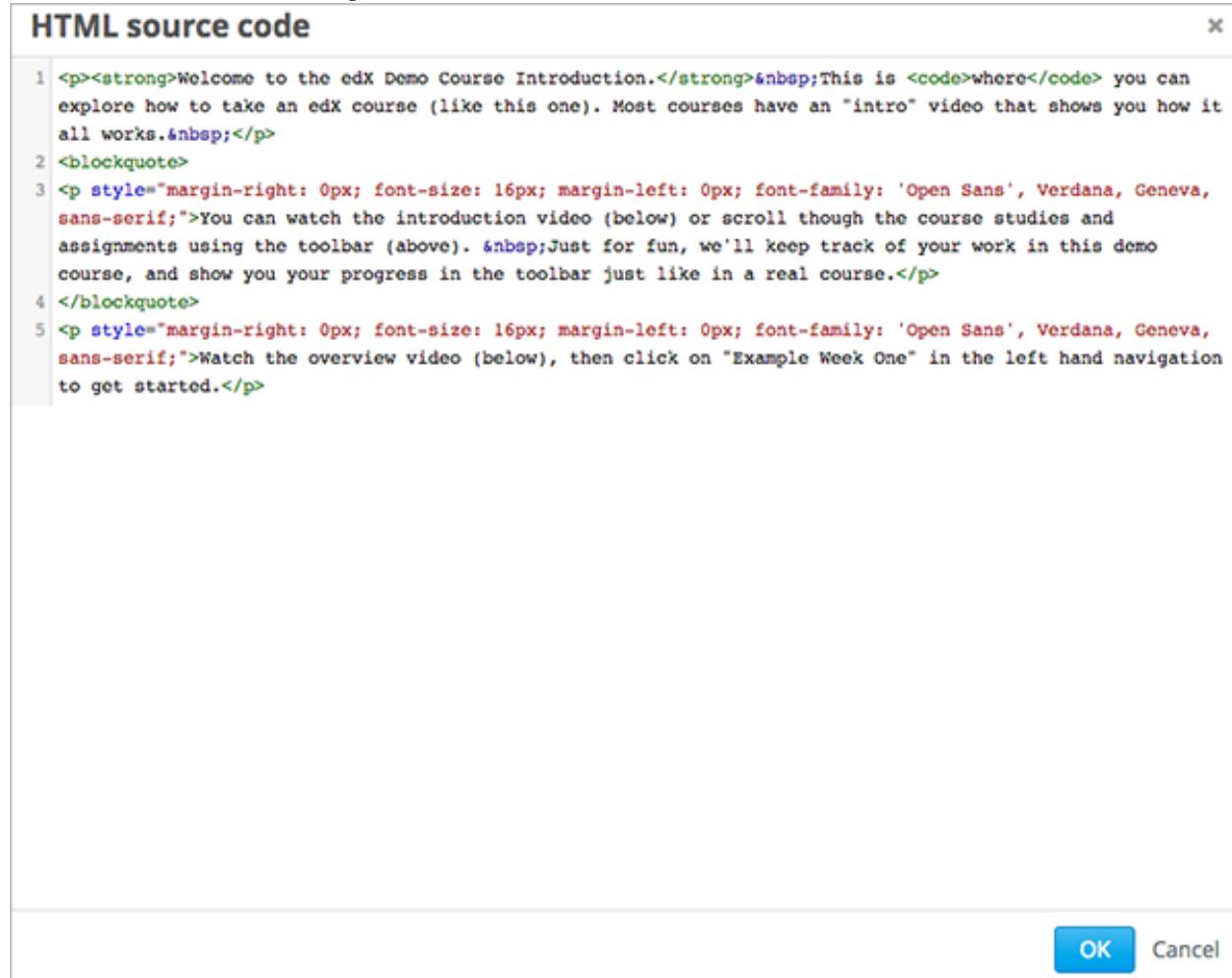
The image shows the top portion of the CKEditor interface. On the left, the title bar reads "Editing: Blank HTML Page". To the right are two buttons: "Editor" and "Settings". Below the title bar is a toolbar with several icons: "Paragraph" (dropdown), "Font Family" (dropdown), "B" (bold), "I" (italic), "U" (underline), "A" (dropdown), "{}" (code view), "≡" (list view), "☰" (grid view), "“”" (double quotes), "♂" (male symbol), "♀" (female symbol), "☒" (checkbox), "☒" (radio button), and "HTML" (dropdown). At the bottom of the toolbar is a color palette with 15 numbered squares, each containing a different shade of pink or purple.

1. Choose a formatting style for the selected paragraph, such as heading 1, heading 2, or paragraph.
 2. Choose a font family for selected text, such as Arial, Courier New, or Times New Roman.
 3. Format selected text in bold.
 4. Format selected text in italics.
 5. Underline selected text.
 6. Apply a color to the selected text.
 7. Format selected text as code.
 8. Create a bulleted list.
 9. Create a numbered list.
 10. Decrease and increased the indentation of the selected paragraph.
 11. Format the selected paragraph as a blockquote.

12. Create a link from the selected text. See *Add a Link in an HTML Component*.
13. Delete the current link.
14. Insert an image at the cursor. See *Add an Image to an HTML Component*.
15. Work with HTML source code, described below.

Work with HTML code in the Visual editor

To work with HTML source code for the content you create in the Visual editor, click **HTML** in the editor toolbar. The HTML source code editor opens:



The screenshot shows a modal dialog box titled "HTML source code". Inside the dialog, there is a text area containing five numbered lines of HTML code. Lines 1 and 2 are part of a paragraph. Line 3 is a blockquote. Lines 4 and 5 are part of another paragraph. The code describes the edX Demo Course Introduction, mentioning an "intro" video and course studies. At the bottom right of the dialog are two buttons: "OK" and "Cancel".

```
1 <p><strong>Welcome to the edX Demo Course Introduction.</strong>&nbsp;This is <code>where</code> you can  
explore how to take an edX course (like this one). Most courses have an "intro" video that shows you how it  
all works.&nbsp;</p>  
2 <blockquote>  
3 <p style="margin-right: 0px; font-size: 16px; margin-left: 0px; font-family: 'Open Sans', Verdana, Geneva,  
sans-serif;">You can watch the introduction video (below) or scroll though the course studies and  
assignments using the toolbar (above). &nbsp;Just for fun, we'll keep track of your work in this demo  
course, and show you your progress in the toolbar just like in a real course.</p>  
4 </blockquote>  
5 <p style="margin-right: 0px; font-size: 16px; margin-left: 0px; font-family: 'Open Sans', Verdana, Geneva,  
sans-serif;">Watch the overview video (below), then click on "Example Week One" in the left hand navigation  
to get started.</p>
```

Edit the HTML code as needed.

You should not add custom styles or scripts in the HTML code view in the Visual editor. Use the Raw HTML editor instead.

Click **OK** to close the source code editor and apply your changes in the Visual editor. The Visual editor then attempts to ensure the underlying HTML code is valid; for example, if you do not close a paragraph tag, the editor will close it for you.

Warning: Clicking **OK** in the source code editor does not save your changes to the HTML component. You return to the component editor, where your changes are applied. You must then also click **Save** to save your changes and close the component. If you click **Cancel**, the changes you made in the HTML source code are lost.

5.7.4 The Raw HTML Editor

When you select the Raw editor for the HTML component, you edit your content in a text editor:

The screenshot shows a modal window titled "Editing: Raw HTML". At the top right are two tabs: "EDITOR" (which is selected) and "SETTINGS". The main area contains a code editor with the following content:

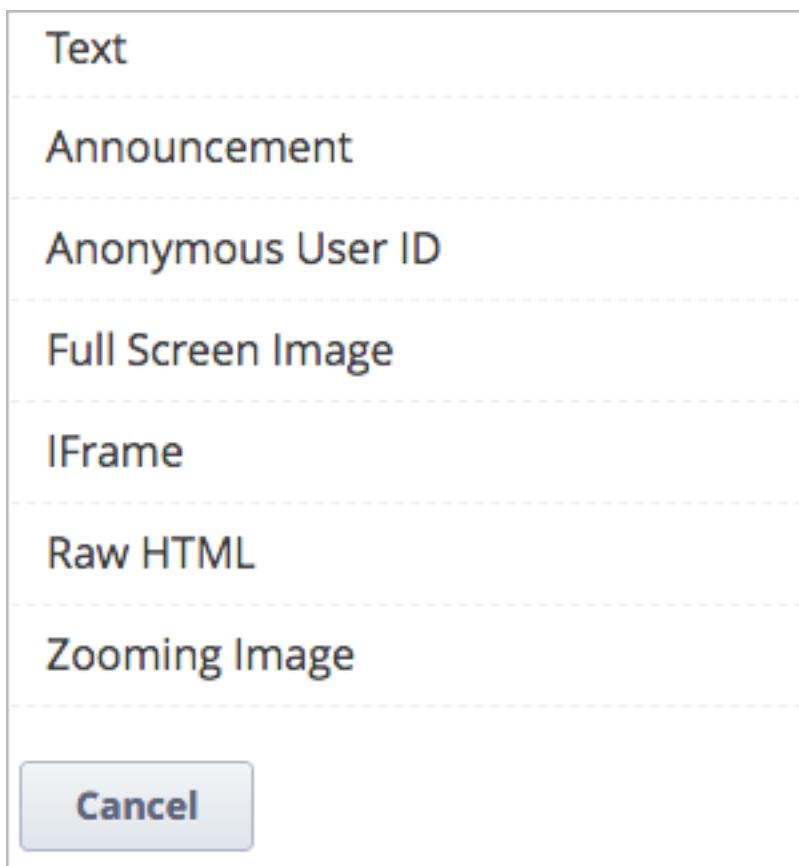
```
1 <p>This template is similar to the Text template. The only difference is  
2 that this template opens in the Raw HTML editor rather than in the Visual  
3 editor.</p>  
4  
5 <p>The Raw HTML editor saves your HTML exactly as you enter it.  
6 You can switch to the Visual editor by clicking the Settings tab and  
7 changing the Editor setting to Visual. Note, however, that some of your  
8 HTML may be modified when you save the component if you switch to the  
9 Visual editor.</p>  
10
```

At the bottom left are "Save" and "Cancel" buttons.

You must enter valid HTML. The Raw HTML editor does not validate your HTML code. Therefore you should thoroughly test the HTML content in your course.

5.7.5 HTML Component Templates

When you create a new HTML component, you select from a list of templates:



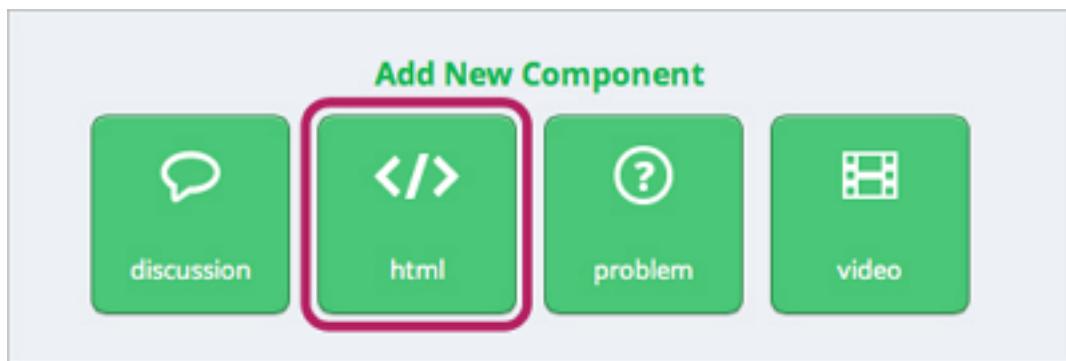
The Raw HTML template is set to use the Raw HTML editor. All other templates use the Visual editor.

For any HTML component, you can change the editor, regardless of the template used to create the component. See Set the Editor for an HTML Component.

5.7.6 Create an HTML Component

To create an HTML component:

1. Under Add New Component, click **html**.



2. Select the template.

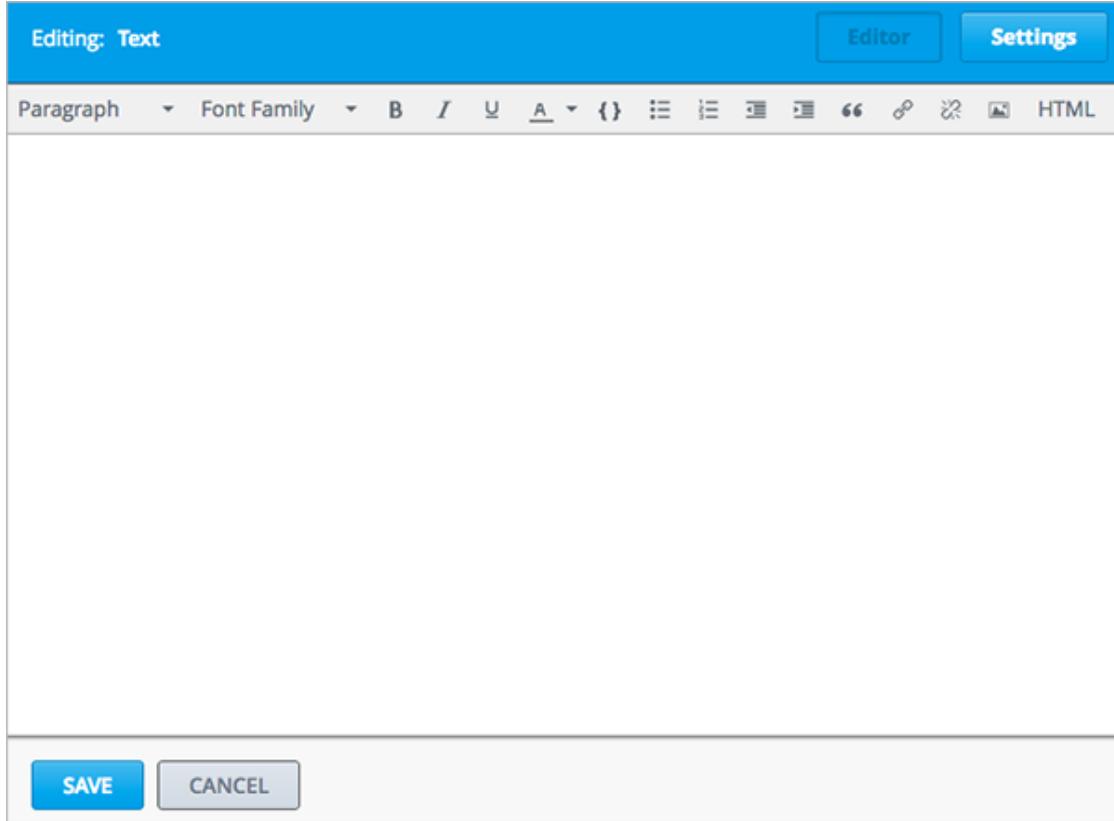
The rest of these instructions assume you selected **Text**, which creates an empty component with the Visual editor selected.

An empty component appears at the bottom of the unit.



3. In the component, click **Edit**.

The HTML component opens in the Visual editor.



4. Enter and format your content. You can *Work with HTML code in the Visual editor* if needed.
5. Enter a display name (the name that you want students to see). To do this, click **Settings** in the upper-right corner of the component editor, and then enter text in the **Display Name** field.

To return to the text editor, click **Editor** in the upper-right corner.

6. Click **Save** to save the HTML component.

When using the Visual editor, you can also:

- *Add a Link in an HTML Component*
- *Add an Image to an HTML Component*
- *Import LaTeX Code into an HTML Component*

5.7.7 Add a Link in an HTML Component

When using the Visual editor, to add a link to a website, course unit, or file in an HTML component, you work with the **Insert link** dialog box.



You can:

- *Add a Link to a Website*
- *Add a Link to a Course Unit*
- *Add a Link to a File*

Add a Link to a Website

To add a link to a website:

1. Select the text that you want to make into the link.
2. Click the link icon in the toolbar.
3. In the **Insert link** dialog box, enter the URL of the website that you want in the **URL** field.



4. If you want the link to open in a new window, click the drop-down arrow next to the **Target** field, and then select **New Window**. If not, you can leave the default value.
5. Click **OK**.
6. Save the HTML component and test the link.

Add a Link to a Course Unit

You can add a link to a course unit in an HTML component.

1. Obtain the unit identifier of the unit you're linking to. To do this, open the unit page in Studio, and copy the unit ID from the **Unit Identifier** field under **Unit Location** in the right pane.

The screenshot shows the course navigation bar at the top with 'Course Home' and 'Week 1'. Below it is a 'BIOLOGY DEMONSTRATION' section containing a video player. The video player displays a person in lab attire working with red liquid in flasks. The video controls show '0:00 / 2:10', 'SPEED 1.0x', 'HD', and 'cc'. To the right of the video is a transcript:

Hi. My name is Ryan. Today, I'm going to introduce you to the model organism, *Saccharomyces cerevisiae*, or more commonly known as the budding yeast. In this video I'm going to demonstrate for you how geneticists grow and replicate yeast, just like Professor Lander

On the right side of the screen is the 'Unit Settings' panel. It includes fields for 'Display Name' (Video Demonstrations), 'Visibility' (Public), and 'Unit Identifier' (f0e6d90842c44cc7a50fd1a18a7dd982). The 'Unit Location' sidebar lists various sections: Getting Started, Working with Videos, Videos on edX, Video Demonstrations (which is highlighted in yellow), Video Presentation Styles, and Interactive Questions.

2. Open the HTML component where you want to add the link.
3. Select the text that you want to make into the link.
4. Click the link icon in the toolbar.
5. In the **Insert link** dialog box, enter the following in the **URL** field.

/jump_to_id/<unit identifier>

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 (/).



6. If you want the link to open in a new window, click the drop-down arrow next to the **Target** field, and then select **New Window**. If not, you can leave the default value.

7. Click **Insert**.
8. Save the HTML component and test the link.

Add a Link to a File

You can add a link in an HTML component to any file you've uploaded for the course. For more information about uploading files, see *Adding Files to a Course*.

1. On the **Files & Uploads** page, copy the **Embed URL** of the file.

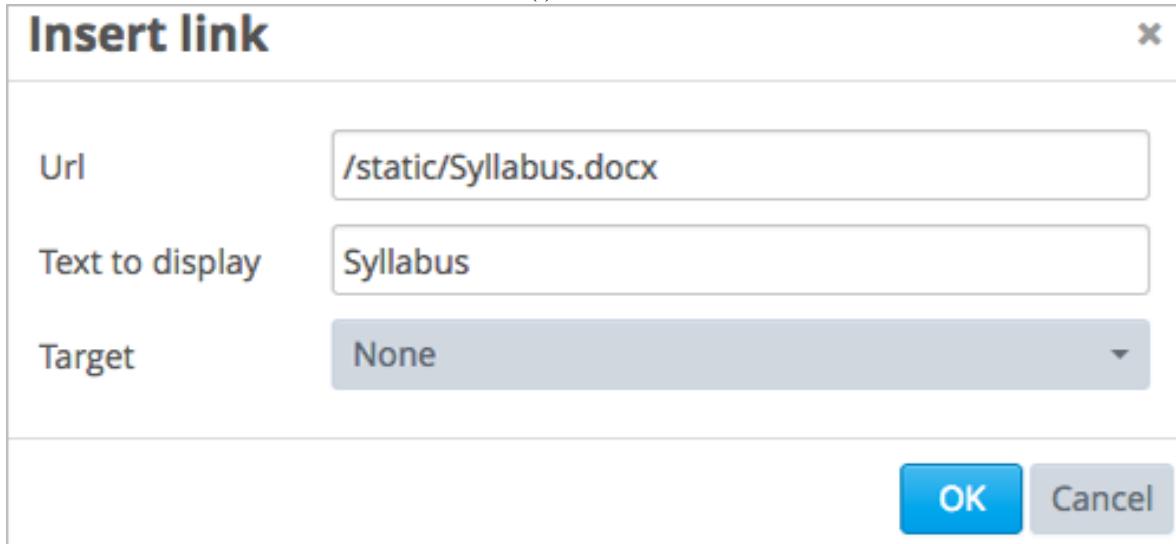
Preview	Name	Date Added	Embed URL	External URL
	Syllabus.docx	Mar 06, 2014 at 17:54 UTC	/static/Syllabus.docx	courses.stage.edx.org/c4x/m

Note: You must use the **Embed URL** to link to the file, not the **External URL**.

2. Select the text that you want to make into the link.
3. Click the link icon in the toolbar.
4. In the **Insert link** dialog box, enter the following in the **URL** field.

/static/FileName.type

Make sure to include both forward slashes (/).



5. If you want the link to open in a new window, click the drop-down arrow next to the **Target** field, and then select **New Window**. If not, you can leave the default value.
6. Click **Insert**.
7. Save the HTML component and test the link.

5.7.8 Add an Image to an HTML Component

When using the Visual editor, you can add any image that you have uploaded for the course to an HTML component. For more information about uploading images, see *Adding Files to a Course*.

Note: Review *Best Practices for Describing Images* before you add images to HTML components.

To add an image, you'll need the URL of the image that you uploaded to the course. You'll then create a link to the image in the HTML component.

1. On the **Files & Uploads** page, copy the **Embed URL** of the image that you want.

The screenshot shows a table with columns: Preview, Name, Date Added, Embed URL, and External URL. Two rows are visible:

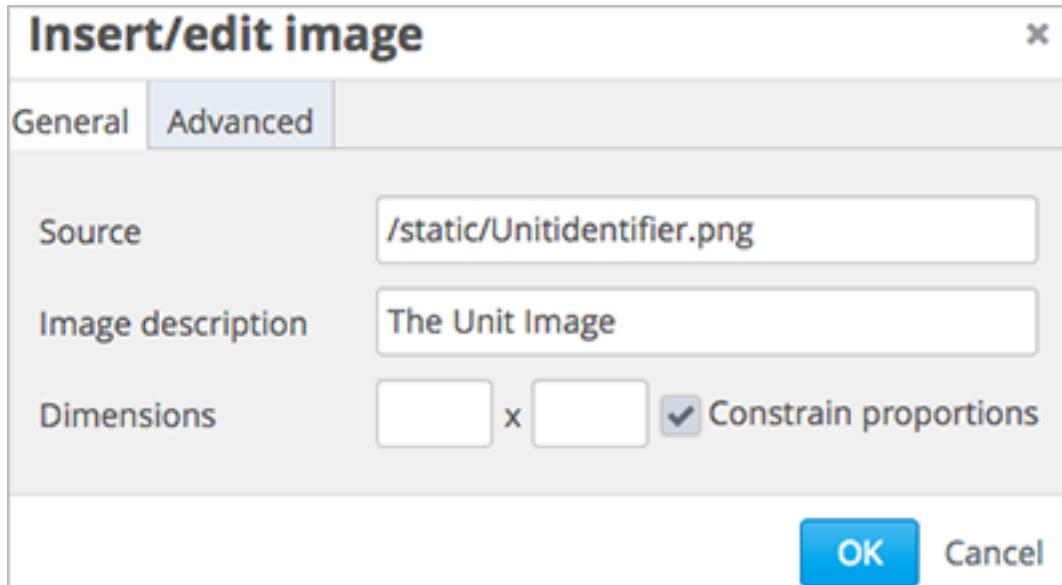
Preview	Name	Date Added	Embed URL	External URL
	first_course.png	Feb 25, 2014 at 16:18 UTC	/static/first_course.png	mhoeber.m.sandbox.ed:
	display-name.png	Feb 25, 2014 at 16:18 UTC	/static/display-name.png	mhoeber.m.sandbox.ed:

Note: You must use the **Embed URL** to add the image, not the **External URL**.

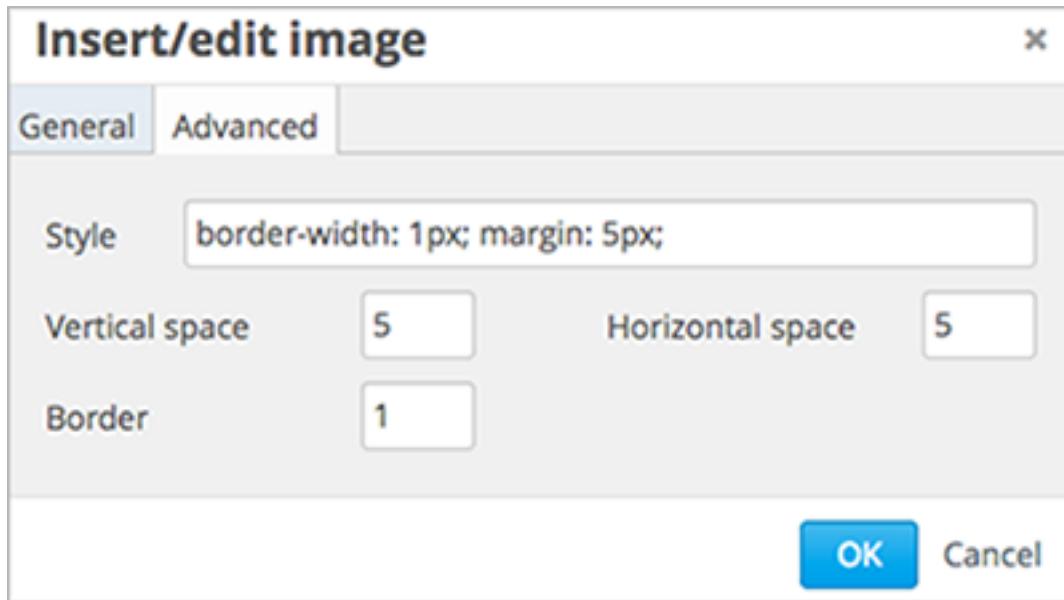
2. Click the image icon in the toolbar.
3. In the **Insert image** dialog box, enter the following in the **Source** field.

/static/FileName.type

Make sure to include both forward slashes (/).



4. Enter alternative text in the **Image description** field. This text becomes the value of the alt attribute in HTML and is required for your course to be fully accessible. See *Best Practices for Describing Images* for more information.
5. As needed, customize the image dimensions. Keep **Constrain proportions** checked to ensure the image keeps the same width and height proportions.
6. To change the spacing and border of the image, click the **Advanced** tab.



7. Enter the **Vertical space**, **Horizontal space**, and **Border** as needed. The values you enter are automatically added to the **Style** field.
8. Click **OK** to insert the image in the HTML component.
9. Save the HTML component and test the image.

5.7.9 Import LaTeX Code into an HTML Component

You can import LaTeX code into an HTML component. You might do this, for example, if you want to create “beautiful math” such as the following.

Example of E-text in LaTeX

You can write complex equations such as the following in LaTeX.

$$x = \frac{-b \pm \sqrt{b^2 - 4 * a * c}}{2a}$$

$$L' = L \sqrt{1 - \frac{v^2}{c^2}}$$

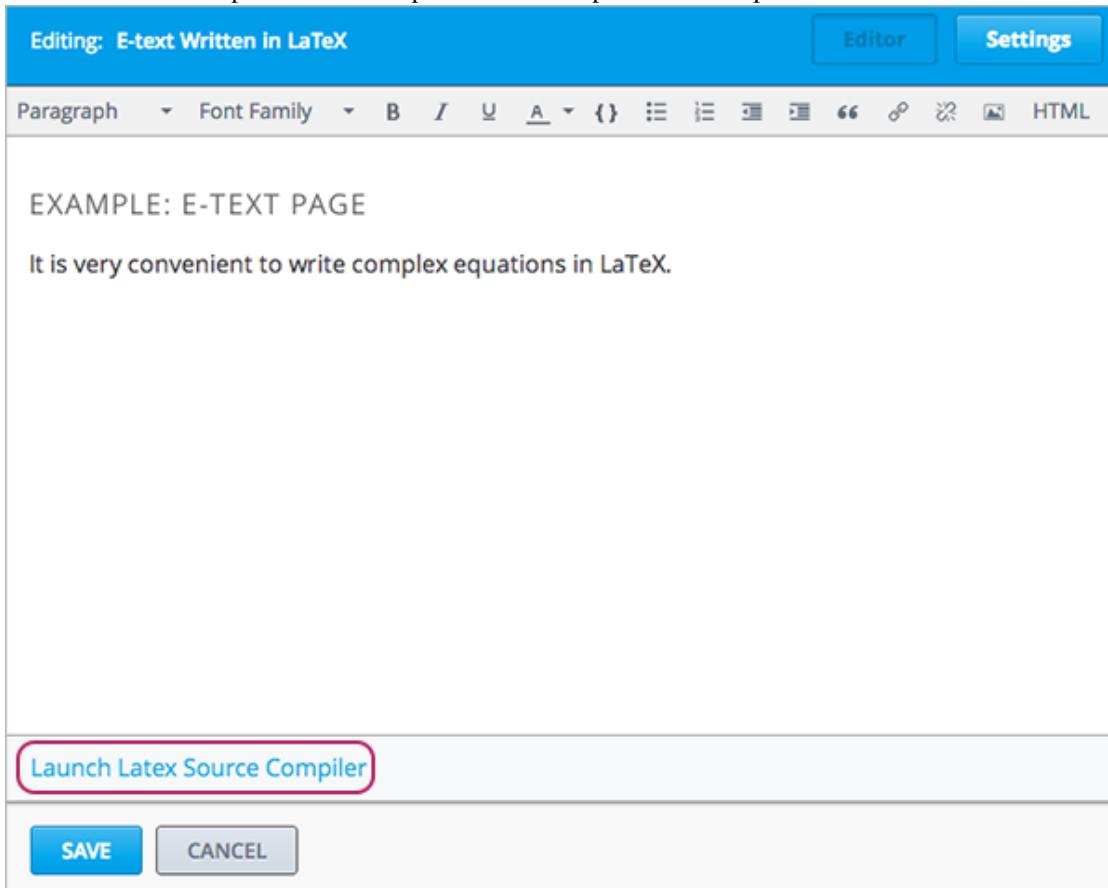
$$\lim_{x \rightarrow 0} \frac{e^x - 1}{2x} \stackrel{\left[\frac{0}{0} \right]}{=} \lim_{x \rightarrow 0} \frac{e^x}{2} = \frac{1}{2}$$

Warning: The LaTeX processor that Studio uses to convert LaTeX code to XML is a third- party tool. We recommend that you use this feature with caution. If you do use it, make sure to work with your Program Manager.

This feature is not enabled by default. To enable it, you have to change the advanced settings in your course.

To create an HTML component that contains LaTeX code:

1. Enable the policy key in your course.
 - (a) In Studio, click **Settings**, and then click **Advanced Settings**.
 - (b) On the **Advanced Settings** page, scroll down to the **use_latex_compiler** policy key.
 - (c) In the **Policy Value** field next to the **use_latex_compiler** policy key, change **false** to **true**.
 - (d) At the bottom of the page, click **Save Changes**.
 2. In the unit where you want to create the component, click **html** under **Add New Component**, and then click **E-text Written in LaTeX**. The new component is added to the unit.
 3. Click **Edit** to open the new component. The component editor opens.



4. In the component editor, click **Launch Latex Source Compiler**. The LaTeX editor opens.

The screenshot shows a component editor window titled "HIGH LEVEL SOURCE EDITING". The code area contains the following LaTeX code:

```
1 \subsection{Example of E-text in LaTeX}
2
3 You can write complex equations such as the following in LaTeX.
4
5 \begin{equation}
6     x = \frac{-b \pm \sqrt{b^2 - 4*a*c}}{2a}
7 \end{equation}
8
9 \begin{equation}
10    L' = (L) \left( \sqrt{1 - \frac{v^2}{c^2}} \right)
11 \end{equation}
12
13
14 \[
15 \lim_{x \rightarrow 0} \left( \frac{e^x - 1}{2x} \right)
16 \overset{0}{\underset{\left( \frac{0}{0} \right)}{\lim}} \left( \frac{\mathbf{H}}{2} \right) = \frac{1}{2}
17 \]
18
19
```

Below the code are several buttons: "Save & Compile to edX XML", "Save", "Refresh", and "Upload".

5. Write LaTeX code as needed. You can also upload a LaTeX file into the editor from your computer by clicking **Upload** in the bottom right corner.
6. When you have written or uploaded the LaTeX code you need, click **Save & Compile to edX XML** in the lower-left corner.

The component editor closes. You can see the way your LaTeX content looks.

The component editor has closed, and the content is now displayed in a preview window. The title is "E-text Written in LaT...". The preview shows the text "Example of E-text in LaTeX" and "It is very convenient to write complex equations in LaTeX." Below this is a box containing the equation
$$x = \frac{-b \pm \sqrt{b^2 - 4*a*c}}{2a}$$
. At the bottom of the preview window, there is a footer with the text "Seize the moment."

7. On the unit page, click **Preview** to verify that your content looks the way you want it to in the LMS. If you see errors, go back to the unit page. Click **Edit** to open the component again, and then click **Launch Latex Source Compiler** in the lower-left corner of the component editor to edit the LaTeX code.

5.8 Working with Video Components

5.8.1 Introduction to Videos

You can create a video of your lecture and add it to your course with other components—such as discussions and problems—to promote active learning. Adding a video to your course has several steps.

1. *Step 1. Create the Video.*
2. *Step 2. Create or Obtain a Video Transcript.*
3. *Step 3. Post the Video Online.*
4. *Step 4. Create a Video Component.*

Also see:

- *Additional Transcripts*
- *Steps for .sjson Files*

Note: Review *Best Practices for Accessible Media* before adding videos to your course.

5.8.2 Step 1. Create the Video

Your video can contain whatever content you want. The [Producing Videos](#) section of our [edX101](#) course has some helpful pointers for creating good video content.

Compression Specifications

When you create your video, edX recommends the following compression specs. (Note that these are recommended but not required.)

Output	Edited Files	Publish to YouTube	Publish downloadable file to AWS S3
Codec	H.264 .mp4	H.264 “main concept” .mp4	H.264 “x264” .mp4
Resolution and Frame Rate	1920x1080, progressive, 29.97 fps	1920x1080, progressive, 29.97 fps	1280x720, progressive, 29.97 fps
Aspect	1.0	1.0	1.0
Bit Rate	VBR, 2 pass	VBR, 2 pass	VBR, 2 pass
Target VBR	32 mbps	5 mbps	1 mbps
Max VBR	40 mbps	7.5 mbps	1.5 mbps
Audio	Linear AAC 48kHz / 256 kbps	AAC 44.1 / 192 kbps	AAC 44.1 / 192 kbps

Video Formats

The edX video player supports videos in .mp4, .mpeg, .ogg, and .webm format.

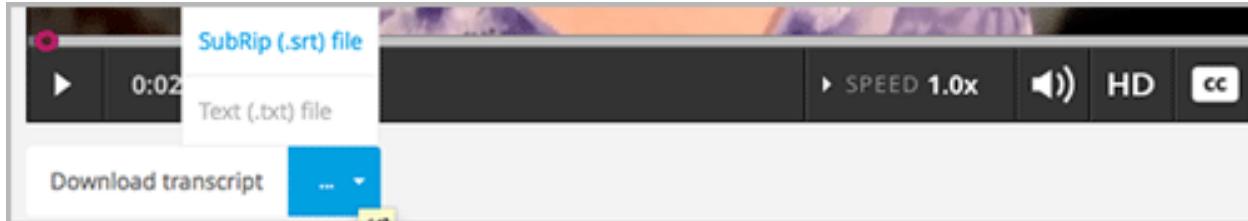
5.8.3 Step 2. Create or Obtain a Video Transcript

We strongly recommend that you associate a timed transcript with your video. Transcripts can be helpful for students whose first language isn't English, who can't watch the video, or who want to review the video's content. The transcript scrolls automatically while your video plays. When a student clicks a word in the transcript, the video opens to that word. You can also allow your students to download the transcript so that they can read it offline. You'll associate the transcript with the video when you create the Video component.

To play automatically with the video, your transcript file must be an .srt file. To create or obtain a transcript in .srt format, you can work with a company that provides captioning services. EdX works with [3Play Media](#). YouTube also provides captioning services.

In addition to your .srt file, you can provide other transcripts with your video. For example, you can provide downloadable transcripts in a text format such as .txt or .pdf, and you can provide transcripts in different languages. For more information, see *Additional Transcripts*.

If you provide transcripts for students to download, a **Download transcript** button appears under the video. Students can then select either **SubRip (.srt) file** or **Text (.txt) file** to download the .srt or .txt transcript.



Note: Some past courses have used .json files for video transcripts. If transcripts in your course uses this format, see *Steps for .json Files*. We don't recommend that you use .json files.

5.8.4 Step 3. Post the Video Online

All course videos should be posted to YouTube. By default, the edX video player accesses your YouTube videos.

Because YouTube is not available in all locations, however, 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.

After you post your video online, make sure you have the URL for the video. If you host copies of your video in more than one place, make sure you have the URL for each video location.

YouTube

After you've created your video, upload the video to [YouTube](#).

Note: YouTube only hosts videos of up to 15 minutes. If you create a 0.75-speed option, you must make sure that your 1.0-speed video segments are only 11.25 minutes long so that YouTube can host all speeds. YouTube offers paid accounts that relax this restriction.

Other Sites

You can use any video backup site that you want. However, keep in mind that the site where you post the videos may have to handle a lot of traffic.

Note: The URL for the video that you post on a third-party site must end in .mp4, .mpeg, .ogg, or .webm. EdX can't support videos that you post on sites such as Vimeo.

5.8.5 Step 4. Create a Video Component

1. Under **Add New Component**, click **Video**.
2. When the new video component appears, click **Edit**. The video editor opens to the **Basic** tab.

The screenshot shows the 'Editing: Video' dialog box. At the top, there are two tabs: 'Basic' (selected) and 'Advanced'. The 'Display Name' field contains 'Video'. Below it is a description: 'Display name for this module.' The 'Video URL' field contains 'http://youtu.be/OEoXaMPEz...'. Below it is a description: 'A YouTube URL or a link to a file hosted anywhere on the web.' A button labeled 'Add more video sources' is present. In the 'Timed Transcript' section, there is a checkbox labeled 'Timed Transcript Conflict' with the following text: 'The timed transcript file on YouTube does not appear to be the same as the timed transcript file on edX. Would you...'. At the bottom are 'SAVE' and 'CANCEL' buttons.

You'll replace the default values with your own.

3. In the **Display Name** field, enter the name you want students to see when they hover the mouse over the unit in the course ribbon. This text also appears as a header for the video.
4. In the **Video URL** field, enter the URL of the video. For example, the URL may resemble one of the following.
`http://youtu.be/OEoXaMPEzfM`
`http://www.youtube.com/watch?v=OEoXaMPEzfM`
`https://s3.amazonaws.com/edx-course-videos/edx-edx101/EDXSPCPJSP13-G030300.mp4`
5. Next to **Timed Transcript**, select an option.

- If edX already has a transcript for your video—for example, if you’re using a video from an existing course—Studio automatically finds the transcript and associates the transcript with the video.

If you want to modify the transcript, click **Download to Edit**. You can then make your changes and upload the new file by clicking **Upload New Timed Transcript**.

- If your video has a transcript on YouTube, Studio automatically finds the transcript and asks if you want to import it. To use this YouTube transcript, click **Import from YouTube**. (If you want to modify the YouTube transcript, after Studio imports the transcript, click **Download to Edit**. You can then make your changes and upload the new file by clicking **Upload New Timed Transcript**.)
- If neither edX nor YouTube has a transcript for your video, and your transcript uses the .srt format, click **Upload New Timed Transcript** to upload the transcript file from your computer.

Note:

- If your transcript uses the .json format, do not use this setting. For more information, see *Steps for .json Files*.
 - If you want to provide a transcript in a format such as .txt or .pdf, do not use this setting to upload the transcript. For more information, see *Additional Transcripts*.
-

6. Optionally, click **Advanced** to set more options for the video. For a description of each option, see the list below.
7. Click **Save**.

Advanced Options

The following options appear on the **Advanced** tab in the Video component.

- **Display Name:** The name that you want your students to see. This is the same as the **Display Name** field on the **Basic** tab.
- **Download Transcript:** The URL for the transcript file for the video. This file is usually an .srt file, but can also be a .txt or .pdf file. (For more information about .txt and .pdf files, see *Additional Transcripts*.) The URL can be an external URL, such as <http://example.org/transcript.srt>, or the URL for a file that you’ve uploaded to your **Files & Uploads** page, such as `/static/example.srt`.

This setting is related to **Transcript Download Allowed**.

- If you set **Transcript Download Allowed** to **True**, and you specify a file in the **Download Transcript** field, the file you’ve specified will be available for students to download.
- If you set **Transcript Download Allowed** to **True**, but you leave the **Download Transcript** field blank, the .srt transcript that automatically plays with the video will be available.
- **End Time:** The time, formatted as hours, minutes, and seconds (HH:MM:SS), when you want the video to end.
- **Start Time:** The time, formatted as hours, minutes, and seconds (HH:MM:SS), when you want the video to begin.
- **Transcript (primary):** The name of the .srt file from the **Timed Transcript** field on the **Basic** tab. This field is auto-populated. You don’t have to change this setting.

If your transcript uses an .json file, see *Steps for .json Files*.

- **Transcript Display:** Specifies whether you want the transcript to show by default. Students can always turn transcripts on or off while they watch the video.
- **Transcript Download Allowed:** Specifies whether you want to allow your students to download a copy of the transcript.

- **Transcript Translations:** The transcript files for any additional languages. For more information, see *Transcripts in Additional Languages*.
- **Video Download Allowed:** Specifies whether you want to allow your students to download a copy of the video.
- **Video Sources:** Additional locations where you've posted the video. This field must contain a URL that ends in .mpeg, .mp4, .ogg, or .webm.
- **YouTube ID, YouTube ID for .75x speed, YouTube ID for 1.25x speed, YouTube ID for 1.5x speed:** If you have uploaded separate videos to YouTube for different speeds of your video, enter the YouTube IDs for these videos in these fields.

5.8.6 Additional Transcripts

You can provide your students with a downloadable transcript in a format such as .txt or .pdf in addition to the .srt transcript that plays along with the video.

1. Upload the .txt or .pdf transcript to the **Files & Uploads** page or host it on an external website.
2. In the Video component, click the **Advanced** tab.
3. In the **Download Transcript** field, enter the URL for the transcript. For more information, see *Advanced Options*.

Transcripts in Additional Languages

You can provide transcripts for your video in other languages. To do this, you'll work with a third-party service to obtain an .srt transcript file for each language, and then associate the .srt file with the video in Studio.

1. After you've obtained the .srt files for additional languages, open the Video component for the video.
2. On the **Advanced** tab, scroll down to **Transcript Translations**, and then click **Add**.
3. In the drop-down list that appears, select the language for the transcript that you want to add.
An **Upload** button appears below the language.
4. Click **Upload**, browse to the .srt file for the language that you want, and then click **Open**.
5. In the **Upload translation** dialog box, click **Upload**.
6. Repeat steps 2 - 5 for any additional languages.

Note: Make sure that all your transcript file names are unique to each video and language. If you use the same transcript name in more than one Video component, the same transcript will play for each video. To avoid this problem, you could name your foreign language transcript files according to the video's file name and the transcript language.

For example, you have two videos, named video1.mp4 and video2.mp4. Each video has a Russian transcript and a Spanish transcript. You can name the transcripts for the first video video1_RU.srt and video1_ES.srt, and name the transcripts for the second video video2_RU.srt and video2_ES.srt.

When your students view the video, they can click the **CC** button at the bottom of the video player to select a language.

The screenshot shows a student's dashboard with three course cards:

- Course 1:** Title: CS50x: Crash Course in Artificial Intelligence, Course Period: Sep 24, 2012, View Courseware button.
- Course 2:** Title: Physics: Fundamentals, Quantitative Methods in Clinic..., Course Period: Oct 15, 2012, View Courseware button.
- Course 3:** Title: 3.091x Introduction to Solid State Chemistry, Course Period: Oct 15, 2012, View Courseware button.

A dropdown menu for language selection is open, showing options: English (highlighted with a pink box), Russian, and Spanish.

Below the dashboard, video player controls are visible: play/pause, time (0:36 / 2:52), speed (1.0x), volume, HD, and captions (CC).

Text on the right:

- 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

5.8.7 Steps for .sjson Files

If your course uses .sjson files, you'll upload the .sjson file for the video to the **Files & Uploads** page, and then specify the name of the .sjson file in the Video component.

Note: Only older courses that have used .sjson files in the past should use .sjson files. All new courses should use .srt files.

1. Obtain the .sjson file from a media company such as 3Play.
2. Change the name of the .sjson file to use the following format:

`subs_FILENAME.srt.sjson`

For example, if the name of your video is **Lecture1a**, the name of your .sjson file must be **subs_Lecture1a.srt.sjson**.

3. Upload the .sjson file for your video to the **Files & Uploads** page.
4. Create a new video component.
5. On the **Basic** tab, enter the name that you want students to see in the **Display Name** field.
6. In the **Video URL** field, enter the URL of the video. For example, the URL may resemble one of the following.

`http://youtu.be/OEoXaMPEzfM`

`http://www.youtube.com/watch?v=OEoXaMPEzfM`

`https://s3.amazonaws.com/edx-course-videos/edx-edx101/EDXSPCPJSP13-G030300.mp4`

7. Click the **Advanced** tab.
8. In the **Transcript (primary)** field, enter the file name of your video. Do not include `subs_` or `.sjson`. For the example in step 2, you would only enter **Lecture1a**.

9. Set the other options that you want.
10. Click **Save**.

5.9 Working with Discussion Components

5.9.1 Overview

You can add a Discussion component to a unit, to pose a question related to the Unit and give students a chance to respond and interact.

See the following topics:

- *Create a Discussion Component*
- *A Student's View of the Discussion*

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.

Also see the following chapters:

- *Discussions*
- *Guidance for Discussion Moderators*

5.9.2 Create a Discussion Component

To create a new Discussion component in an existing unit, ensure the unit is Private. For more information, see *Public and Private Units*.

1. Under **Add New Component**, click **Discussion**.
2. In the Discussion component that appears, click **Edit**.



3. When the Discussion component editor opens, follow the guidelines in the editor to fill in the **Category**, **Display Name**, and **Subcategory** fields.

Editing: Discussion

Category Videos C

A category name for the discussion. This name appears in the left pane of the discussion forum for the course.

Display Name Discussion: Welcome to edX101! C

Display name for this module

Subcategory Khan-Style Videos C

A subcategory name for the discussion. This name appears in the left pane of the discussion forum for the course.

Save Cancel

4. Click **Save**.

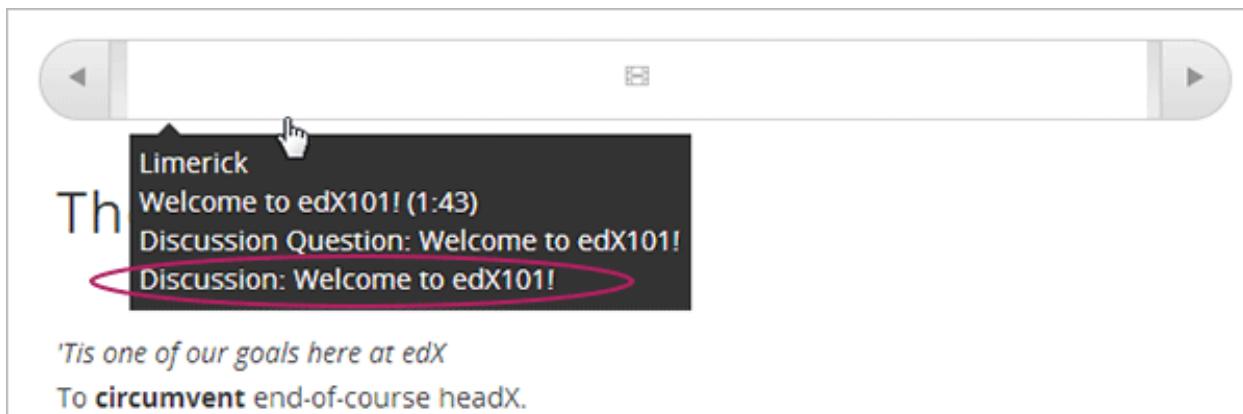
Warning: You should always use these steps to create a discussion component. If you create a discussion component by using the **Duplicate** button in Studio, both discussion components will contain the same conversations, even if users post in different discussions.

Discussion *categories* are immediately visible on the **Discussion** tab for your course when you create them, even if the unit that contains the Discussion component is set to Private. However, an individual Discussion component is not visible until the unit is published and release dates for the section and subsection have passed. See *Publishing Your Course* for more information.

Additionally, you cannot see posts, responses, or comments in a Discussion component when you are working in Studio. To see the discussion, go to the unit that contains the Discussion component in Studio, and then click **Preview** or **View Live** under **Unit Settings**.

5.9.3 A Student's View of the Discussion

For students, the display name for the Discussion component appears in the course ribbon at the top of the page:



The Discussion space appears under other components in the unit. It doesn't have a label in the body of the unit. Instead, students see "Show discussion" or "Hide discussion" on the left, and a blue **New Post** button on the right.

In the following example, the Discussion component follows Video and HTML components:

WELCOME TO EDX101! (1:43)



A video player interface showing a close-up of a man with dark hair and a mustache, wearing a dark suit jacket over a light blue shirt. He is speaking. The video player has a play button in the center, a progress bar at the bottom left indicating 0:00 / 0:00, a volume icon at 1.0x, and an HD icon at the bottom right.

Download video [here](#).

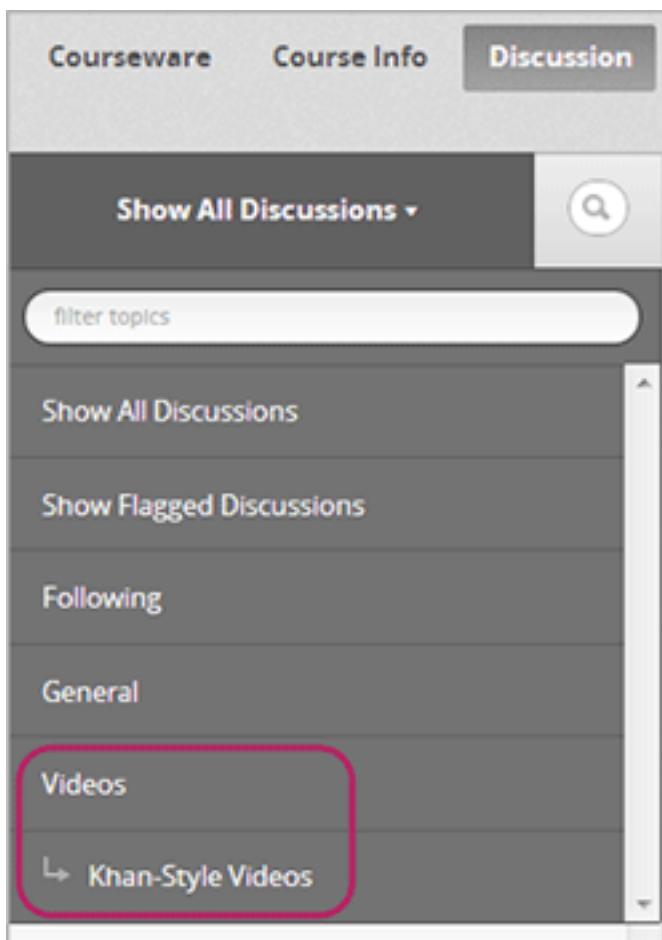
DISCUSSION QUESTION: WELCOME TO EDX101!

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.

 Show Discussion

 New Post

In the **Discussion** tab at the top of the page, students can find the category and subcategory of the discussion in the left pane.



5.10 Working with Problem Components

5.10.1 Overview of Problem Components

The problem component allows you to add interactive, automatically graded exercises to your course content. You can create many different types of problems in Studio.

All problems receive a point score, but, by default, problems do not count toward a student's grade. If you want the problems to count toward the student's grade, change the assignment type of the subsection that contains the problems.

This section covers the basics of Problem components—what they look like to you and your students, and the options that every problem component has. For more information about individual problem types, see *Creating Exercises and Tools*.

For more information, see the following topics.

- *The Student View of a Problem*
- *The Studio View of a Problem*
- *Problem Settings*
- *Modifying a Released Problem*
- *Additional Work with Problems*

- *Multiple Problems in One Component*
- *Problem Randomization*

5.10.2 The Student View of a Problem

All problems on the edX platform have several component parts.

H1P7: MERCURY ATOMS

How many atoms are in 15.3 cm^3 of mercury (at room temperature)? ← 1

6.22*10²³ ← 2

6.22 · 10^{23} ← 3

Check Save Show Answer You have used 0 of 5 submissions ← 7

4 5 6

1. **Problem text.** The problem text can contain any standard HTML formatting.
2. **Response field with the student's answer.** Students enter answers in *response fields*. The appearance of the response field depends on the type of the problem.
3. **Rendered answer.** For some problem types, Studio uses MathJax to render plain text as “beautiful math.”
4. **Check button.** The student clicks **Check** to submit a response or find out if his answer is correct. If the answer is correct, a green check mark appears. If it is incorrect, a red X appears. When the student clicks the **Check button**, Studio saves the grade and current state of the problem.
5. **Save button.** The student can click **Save** to save his current response without submitting it for a grade. This allows the student to stop working on a problem and come back to it later.
6. **Show Answer button.** This button is optional. When the student clicks **Show Answer**, the student sees both the correct answer (see 2 above) and the explanation (see 10 below). The instructor sets whether the **Show Answer** button is visible.
7. **Attempts.** The instructor may set a specific number of attempts or allow unlimited attempts.

H1P7: MERCURY ATOMS

How many atoms are in 15.3 cm³ of mercury (at room temperature)?

6.22*10²³
 6.22 · 10²³

✓ Answer: 6.22E23 ← **9**
8 10

SOLUTION:

molar volume of Hg = 14.82 cm³/ mol

moles per unit volume = (1 / 14.82) mol / cm³

of Hg atoms in 15.3 cm³ = (15.3 cm³)((1 / 14.82) mol / cm³)(6.02 × 10²³)

11 **12** = 6.22×10^{23} atoms

Reset
Hide Answer

You have used 1 of 5 submissions

8. **Feedback.** After a student clicks **Check**, all problems return a green check mark or a red X.



9. **Correct answer.** Most problems require that the instructor specify a single correct answer.
10. **Explanation.** The instructor may include an explanation that appears when a student clicks **Show Answer**.
11. **Reset button.** This button clears the student input, so that the problem looks the way it did originally.
12. **Hide Answer button.**

13

Additional Study Material

Problem Set 1
Homework due February 8, 2013

14 → ⌚

13. **Grading.** The instructor may specify whether a group of problems is graded. If a group of problems is graded, a clock icon appears for that assignment in the course accordion.
14. **Due date.** The date that the problem is due. A problem that is past due does not have a **Check** button. It also does not accept answers or provide feedback.

Note: Problems can be **open** or **closed**. Closed problems do not have a **Check** button. Students can still see questions, solutions, and revealed explanations, but they cannot check their work, submit responses, or change an earlier score.

There are also some attributes of problems that are not immediately visible. You can set these attributes in Studio.

- **Randomization.** For some problems, the instructor can specify whether a problem will use randomly generated numbers that vary from student to student.
- **Weight.** Different problems in a particular problem set may be given different weights.
- **Label.** To improve accessibility for students who have disabilities, each problem needs a descriptive label. The label typically contains part or all of the text of the question in the problem. Most templates include a space for a label. You can find example labels in the documentation for each problem or tool type.

5.10.3 The Studio View of a Problem

All problems are written in XML. However, Studio offers two interfaces for editing problem components: the Simple Editor and the Advanced Editor.

- The **Simple Editor** allows you to edit problems visually, without having to work with XML.
- The **Advanced Editor** converts the problem to edX's XML standard and allows you to edit that XML directly.

You can switch at any time from the Simple Editor to the Advanced Editor by clicking **Advanced Editor** in the top right corner of the Simple Editor interface. However, it is not possible to switch from the Advanced Editor to the Simple Editor.

The Simple Editor

Several problem templates, including multiple choice and text input problem templates, open in the Simple Editor. The following image shows a multiple choice problem in the Simple Editor.

The screenshot shows the Simple Editor interface for a "Multiple Choice" problem. At the top, there is a toolbar with numbered buttons (1-9) and icons for H1, lists, ABC, 123, and a dropdown menu. To the right of the toolbar are "Editor" and "Settings" buttons, and a "Advanced Editor" button. Below the toolbar, the question text is displayed: ">>Lateral inhibition, as was first discovered in the horseshoe crab:<<". The question asks for a property of touch sensation, referring to the ability of crabs to detect nearby predators. There are five options listed, each preceded by a radio button:

- () 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.

Below the question, there is an "[Explanation]" section containing text about lateral inhibition in horseshoe crabs. At the bottom of the editor window are "SAVE" and "CANCEL" buttons.

The Simple Editor includes a toolbar that helps you format the text of your problem. When you select text and then click the formatting buttons, the Simple Editor formats the text for you automatically. The toolbar buttons are the following:

1. Create a level 1 heading.
2. Create multiple choice options.
3. Create checkbox options.
4. Create text input options.
5. Create numerical input options.
6. Create dropdown options.
7. Create an explanation that appears when students click **Show Answer**.
8. Open the problem in the Advanced Editor.
9. Open a list of formatting hints.

The following problem templates open in the Simple Editor.

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

The Advanced Editor

The **Advanced Editor** opens a problem in XML. Templates for problems such as such as drag and drop and math expression input open directly in the Advanced Editor.

The following image shows the multiple choice problem above in the Advanced Editor instead of the Simple Editor.

The screenshot shows the 'Advanced Editor' interface for creating course content. At the top, there's a blue header bar with the title 'Editing: Lateral Inhibition' on the left, and two buttons: 'Editor' and 'Settings' on the right. Below the header is a code editor window containing XML code for a problem template. The code includes sections for the problem itself, multiple choice responses, and a solution. At the bottom of the code editor are two buttons: 'SAVE' (in a blue box) and 'CANCEL'. The XML code is as follows:

```
1 <problem>
2 <p>Lateral inhibition, as was first discovered in the horseshoe crab:</p>
3
4   <multiplechoiceresponse>
5     <choicegroup type="MultipleChoice" label="Lateral inhibition, as was first
       discovered in the horseshoe crab:">
6
7       <choice correct="false">is a property of touch sensation, referring to the
          ability of crabs to detect nearby predators.</choice>
8       <choice correct="false">is a property of hearing, referring to the ability
          of crabs to detect low frequency noises.</choice>
9       <choice correct="true">is a property of vision, referring to the ability
          of crabs eyes to enhance contrasts.</choice>
10      <choice correct="false">has to do with the ability of crabs to use sonar
          to detect fellow horseshoe crabs nearby.</choice>
11      <choice correct="false">has to do with a weighting system in the crabs
          skeleton that allows it to balance in turbulent water.</choice>
12
13   </choicegroup>
14 </multiplechoiceresponse>
15
16 <solution>
17 <div class="detailed-solution">
18 <p>Explanation</p>
```

The following problem templates open in the Advanced Editor.

- *Circuit Schematic Builder Problem* In circuit schematic problems, students create and modify circuits on an interactive grid and submit computer-generated analyses of the circuits for grading.
- *Custom JavaScript Problem* With custom JavaScript display and grading problems, you can incorporate problem types that you've created in HTML into Studio via an IFrame.
- *Drag and Drop Problem* Drag and drop problems require students to drag text or objects to a specific location on an image.
- *Image Mapped Input Problem* Image mapped input problems require students to click a specific location on an image.
- *Math Expression Input Problems* Math expression input problems require students to enter a mathematical expression as text, such as $e=m*c^2$.
- *Problem with Adaptive Hint* These problems can give students feedback or hints based on their responses. Problems with adaptive hints can be text input or multiple choice problems.
- *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.
- *Write-Your-Own-Grader Problem* Custom Python-evaluated input (also called “write-your-own-grader” problems evaluate students’ responses using an embedded Python script that you create. These problems can be any type.

5.10.4 Problem Settings

In addition to the text of the problem, problems that you create using a Problem component have the following settings. These settings appear on the **Settings** tab in the component editor.

- **Display Name**
- **Maximum Attempts**
- **Problem Weight**
- **Randomization**
- **Show Answer**

The screenshot shows the 'Multiple Choice' component editor with the 'Settings' tab selected. The interface includes a header bar with 'Editing: Multiple Choice', 'Editor', and 'Settings' buttons. Below the header are four configuration sections: 'Display Name' (set to 'Lateral Inhibition'), 'Maximum Attempts' (set to '3'), 'Problem Weight' (set to '2'), and 'Randomization' (set to 'Never'). Each section has a descriptive text and a 'Help' icon. At the bottom are 'SAVE' and 'CANCEL' buttons.

Setting	Value	Description
Display Name	Lateral Inhibition	This name appears in the horizontal navigation at the top of the page.
Maximum Attempts	3	Defines the number of times a student can try to answer this problem. If the value is not set, infinite attempts are allowed.
Problem Weight	2	Defines the number of points each problem is worth. If the value is not set, each response field in the problem is worth one point.
Randomization	Never	Defines how often inputs are

Display Name

This setting indicates the name of your problem. The display name appears as a heading over the problem in the LMS and in the course ribbon at the top of the page.

The screenshot shows a user interface for an edX course. At the top, there's a navigation bar with arrows and a search icon. Below it, a title bar displays "Display Name" and "Lateral Inhibition Disease Prevention". A pink oval highlights the title "LATERAL INHIBITION (0 points possible)". Another pink oval highlights the category "Lateral Inhibition Disease Prevention". A red arrow points from the title to the category. The main content area contains a question about lateral inhibition in horseshoe crabs, followed by a list of five multiple-choice options. At the bottom, there are "Check", "Save", and "Show Answer(s)" buttons, along with a message indicating 0 of 3 submissions have been used.

Lateral inhibition, as was first discovered in the horseshoe 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.

Check Save Show Answer(s) You have used 0 of 3 submissions

Maximum Attempts

This setting specifies the number of times a student can try to answer the problem. By default, a student has an unlimited number of attempts.

Problem Weight

Note: Studio stores scores for all problems, but scores only count toward a student's final grade if they are in a subsection that is graded.

This setting specifies the maximum number of points possible for the problem. The problem weight appears next to the problem title.

EXAMPLE DROPDOWN EXERCISE (3 points possible)

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:

Age, rounded to the nearest year:

Life stage - infant, child, and adult:

By default, each response field, or “answer space,” in a Problem component is worth one point. Any Problem component can have multiple response fields. For example, the Problem component above contains one dropdown problem that has three separate questions for students to answer, and thus has three response fields.

The following Problem component contains one text input problem, and has just one response field.

EXAMPLE TEXT INPUT EXERCISE (1 point possible)

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?

**Computing Scores**

The score that a student earns for a problem is the result of the following formula:

Score = Weight × (Correct answers / Response fields)

- **Score** is the point score that the student receives.
- **Weight** is the problem's maximum possible point score.
- **Correct answers** is the number of response fields that contain correct answers.
- **Response fields** is the total number of response fields in the problem.

Examples

The following are some examples of computing scores.

Example 1

A problem's **Weight** setting is left blank. The problem has two response fields. Because the problem has two response fields, the maximum score is 2.0 points.

If one response field contains a correct answer and the other response field contains an incorrect answer, the student's score is 1.0 out of 2 points.

Example 2

A problem's weight is set to 12. The problem has three response fields.

If a student's response includes two correct answers and one incorrect answer, the student's score is 8.0 out of 12 points.

Example 3

A problem's weight is set to 2. The problem has four response fields.

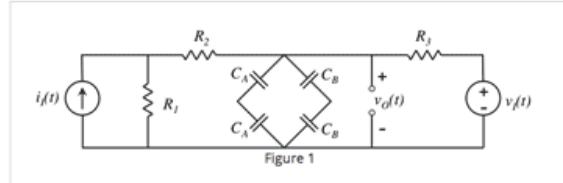
If a student's response contains one correct answer and three incorrect answers, the student's score is 0.5 out of 2 points.

Randomization

This setting specifies whether certain values in your problem change each time a different student accesses the problem, or each time a single student tries to answer the problem. For example, the highlighted values in the problem below change each time a student submits an answer to the problem.

Q1 (6 points possible)

The step response of a circuit is its response to a step input, $u(t)$. Knowing the step response of a linear circuit is extremely valuable as we can figure out the circuit's stability. In this problem we will analyze the circuit in Figure 1 for the output voltage, $v_O(t)$ given an initial condition. Then you will find the output response of two input steps—a current input, $i_I(t)$, and a voltage step, $v_I(t)$.



The element values are as follows: $R_1 = 2\text{k}\Omega$, $R_2 = 2\text{k}\Omega$, $R_3 = 6\text{k}\Omega$, $C_A = 1\mu\text{F}$, and $C_B = 0.5\mu\text{F}$.

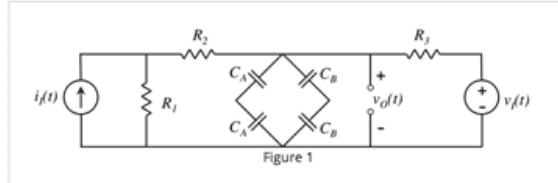
To begin we note that the capacitor bridge composed of four capacitors can be replaced with one capacitor, C_{TOT} . What is the value of C_{TOT} (in μF)?

First we consider the circuit in Figure 1 when $i_I(t) = 0\text{A}$, $v_I(t) = 0\text{V}$, and $v_O(0) = 2\text{V}$. What is the time constant, τ (in ms) of the output voltage?

What is the output voltage $v_O(t)$ at time $t = 0.25\text{ms}$?

Q1 (6 points possible)

The step response of a circuit is its response to a step input, $u(t)$. Knowing the step response of a linear circuit is extremely valuable as we can figure out the circuit's stability. In this problem we will analyze the circuit in Figure 1 for the output voltage, $v_O(t)$ given an initial condition. Then you will find the output response of two input steps—a current input, $i_I(t)$, and a voltage step, $v_I(t)$.



The element values are as follows: $R_1 = 1\text{k}\Omega$, $R_2 = 1\text{k}\Omega$, $R_3 = 2\text{k}\Omega$, $C_A = 0.5\mu\text{F}$, and $C_B = 1.5\mu\text{F}$.

To begin we note that the capacitor bridge composed of four capacitors can be replaced with one capacitor, C_{TOT} . What is the value of C_{TOT} (in μF)?

First we consider the circuit in Figure 1 when $i_I(t) = 0\text{A}$, $v_I(t) = 0\text{V}$, and $v_O(0) = 1\text{V}$. What is the time constant, τ (in ms) of the output voltage?

What is the output voltage $v_O(t)$ at time $t = 0.3\text{ms}$?

If you want to change, or “randomize,” specific values in your problem, you have to do both the following:

- Make sure that your problem contains a Python script that randomizes the values that you want.
- Enable randomization in the Problem component.

Note: Note that specifying this **Randomization** setting is different from *problem randomization*. The **Randomization** setting randomizes variables within a single problem. Problem randomization offers different problems or problem versions to different students. For more information, see *Problem Randomization*.

To enable randomization, select an option for the **Randomization** setting. This setting has four options.

Always	Students see a different version of the problem each time they click Check .
On	Students see a different version of the problem each time they click Reset .
Reset	
Never	All students see the same version of the problem. This is the default.
Per	Individual students see the same version of the problem each time they look at it, but that version is different from the version that other students see.
Student	

Show Answer

This setting defines when the problem shows the answer to the student. This setting has seven options.

Always	Always show the answer when the student clicks the Show Answer button.
An-swered	Show the answer after the student has submitted her final answer.
At-tempeted	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.
Fin-ished	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.

5.10.5 Modifying a Released Problem

Warning: Be careful when you modify problems after they have been released!

After a student submits a response to a problem, Studio stores the student's response, the score that the student received, and the maximum score for the problem. Studio updates these values when a student submits a new response to a problem. However, if an instructor changes a problem or its attributes, Studio does not automatically update existing student information for that problem.

For example, you may release a problem and specify that its answer is 3. After some students have submitted responses, you notice that the answer should be 2 instead of 3. When you update the problem with the correct answer, Studio doesn't update scores for students who answered 2 for the original problem and thus received the wrong score.

For another example, you may change the number of response fields to three. Students who submitted answers before the change have a score of 0, 1, or 2 out of 2.0 for that problem. Students who submitted answers after the change have scores of 0, 1, 2, or 3 out of 3.0 for the same problem.

If you change the weight of the problem, however, the existing scores update when you refresh the **Progress** page.

Workarounds

If you have to modify a released problem in a way that affects grading, you have two options. Note that both options require you to ask your students to go back and resubmit a problem.

- In the Problem component, increase the number of attempts for the problem. Then ask all your students to redo the problem.
- Delete the entire Problem component in Studio and create a new Problem component with the content and settings that you want. Then ask all your students to complete the new problem.

5.10.6 Additional Work with Problems

You have some further options when you work with problems. You can include more than one problem in a single problem component, or you can set up a problem that presents different versions to different students.

Multiple Problems in One Component

You may want to create a problem that has more than one response type. For example, you may want to create a numerical input problem, and then include a multiple choice question about the numerical input problem. Or, you may want a student to be able to check the answers to many problems at one time. To do this, you can include multiple problems inside a single Problem component. The problems can be different types.

To create multiple problems in one component, create a new Blank Advanced Problem component, and then add the XML for each problem in the component editor. You only need to include the XML for the problem and its answers. You don't have to include the code for other elements, such as the **Check** button.

Elements such as the **Check**, **Show Answer**, and **Reset** buttons, as well as the settings that you select for the Problem component, apply to all of the problems in that component. Thus, if you set the maximum number of attempts to 3, the student has three attempts to answer the entire set of problems in the component as a whole rather than three attempts to answer each problem individually. If a student clicks **Check**, the LMS scores all of the problems in the component at once. If a student clicks **Show Answer**, the answers for all the problems in the component appear.

Problem Randomization

You may want to present different students with different problems, or different versions of the same problem. To do this, you'll create a Problem component for each problem or version in Studio, and then edit your course outside of Studio to randomize the problem that students see.

Note that *problem randomization* is different from the **Randomization** setting in Studio. The **Randomization** setting randomizes variables within a single problem. Problem randomization offers different problems or problem versions to different students.

Note: Creating problems with versions that can be randomized requires you to export your course, edit some of your course's XML files in a text editor, and then re-import your course. We recommend that you create a backup copy of your course before you do this. We also recommend that you only edit your course files in the text editor if you're very familiar with editing XML.

Terminology

Sections, subsections, units, and components have different names in the **Course Outline** view and in the list of files that you'll see after you export your course and open the .xml files for editing. The following table lists the names of these elements in the **Course Outline** view and in a list of files.

Course Outline View	File List
Section	Chapter
Subsection	Sequential
Unit	Vertical
Component	Discussion, HTML, problem, or video

For example, when you want to find a specific section in your course, you'll look in the **Chapter** folder when you open the list of files that your course contains. To find a unit, you'll look in the **Vertical** folder.

Create Randomized Problems

1. In the unit where you want to create a randomized problem, create a separate Problem component for each version or problem that you want to randomize. For example, if you want to offer four versions or problems, you'll create four separate Problem components. Make a note of the 32-digit unit ID that appears in the **Unit Identifier** field under **Unit Location**.

2. Export your course. For information about how to do this, see *Exporting and Importing a Course*. Save the .tar.gz file that contains your course in a memorable location so that you can find it easily.
3. Locate the .tar.gz file that contains your course, and then unpack the .tar.gz file so that you can see its contents in a list of folders and files.
 - To do this on a Windows computer, you'll need to download a third-party program. For more information, see [How to Unpack a tar File in Windows](#), [How to Extract a Gz File](#), [The gzip Home Page](#), or the [Windows](#) section of the [How to Open .tar.gz Files](#) page.
 - For information about how to do this on a Mac, see the [Mac OS X](#) section of the [How to Open .tar.gz Files](#) page.
4. In the list of folders and files, open the **Vertical** folder.

Note: If your unit is not published, open the **Drafts** folder, and then open the **Vertical** folder in the **Drafts** folder.

5. In the **Vertical** folder, locate the .xml file that has the same name as the unit ID that you noted in step 1, and then open the file in a text editor such as Sublime 2. For example, if the unit ID is e461de7fe2b84ebeabe1a97683360d31, you'll open the e461de7fe2b84ebeabe1a97683360d31.xml file.

The file contains a list of all the components in the unit, together with the URL names of the components. For example, the following file contains four Problem components.

```
<vertical display_name="Test Unit">
    <problem url_name="d9d0ceb3fffc74eacb29501183e26ad6e"/>
    <problem url_name="ea66d875f4bf4a9898d8e6d2cc9f3d6f"/>
    <problem url_name="2616cd6324704f85bc315ec46521485d"/>
    <problem url_name="88987707294d4ff0ba3b86921438d0c0"/>
</vertical>
```

6. Add `<randomize> </randomize>` tags around the components for the problems that you want to randomize.

```
<vertical display_name="Test Unit">
    <randomize>
        <problem url_name="d9d0ceb3fffc74eacb29501183e26ad6e"/>
        <problem url_name="ea66d875f4bf4a9898d8e6d2cc9f3d6f"/>
        <problem url_name="2616cd6324704f85bc315ec46521485d"/>
        <problem url_name="88987707294d4ff0ba3b86921438d0c0"/>
    </randomize>
</vertical>
```

7. After you add the `<randomize> </randomize>` tags, save and close the .xml file.
8. Re-package your course as a .tar.gz file.
 - For information about how to do this on a Mac, see [How to Create a Tar GZip File from the Command Line](#).
 - For information about how to do this on a Windows computer, see [How to Make a .tar.gz on Windows](#).
9. In Studio, re-import your course.

Note:

- Once you've implemented randomization, you can only see one of the versions or problems in Studio. You can edit that single problem directly in Studio, but to edit any of the other problems, you'll have to export your course, edit the problems in a text editor, and then re-import the course. This is true for instructors as well as course teams.

- A .csv file for student responses contains the responses to each of the problems in the problem bank.
-

Creating Exercises and Tools

6.1 Creating Exercises and Tools

6.1.1 Introduction to Exercises and Tools

Studio allows you to create a wide variety of exercises and tools for your course. Many of these exercises and tools have templates in Studio so that you can create them easily. In addition, individual course teams frequently create exercises that don't have templates in Studio. We're striving to make these tools available to all our course teams as well, and we have instructions for creating some of them in this section.

Depending on the exercise or tool, you'll use an HTML, Problem, or Advanced component. The page for each individual exercise or tool contains an example of each exercise or tool, together with all the files, code, and step-by-step instructions that you need to create the exercise or tool.

Note: Problems must include labels for accessibility. The label generally includes the text of the main question in your problem. Instructions for adding labels appear in the page for each individual problem.

6.1.2 General Exercises and Tools

Instructions [Collapse 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 [Hide Annotations](#)

This exercise first appeared in Horodot'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-kethnonoi], the guardians of mortal humans. |124 They guard acts of justice [dikai] 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 [basileion].

QUESTION 1 (2/2 points)

Annotation Exercise [Return to Annotation](#)

|124 They guard acts of justice [dikai] 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:

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.

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 never seen by mortals. Their 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 values as "dikai" occasionally occur even in Homeric poetry, as I have argued in my book *The Invention of Justice in Homer*). The reason for this is that Homeric heroes are always mortal, while the heroes of the Golden Age are in an immortally 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.

[Reset](#) [Hide Answer\(s\)](#)

Annotation Problem

POLL QUESTION

All things being equal, should the government use public funds to invest in museums rather than sports arenas?

<input checked="" type="radio"/> Yes		5780 (63.6%)
<input type="radio"/> No		3308 (36.4%)

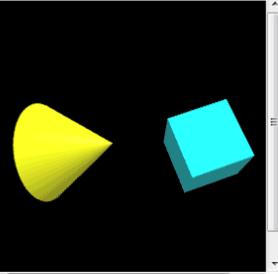
You responded "Yes" to the poll question. But what would you say to the following: What if public officials expect that more people will make use of and enjoy a sports arena? Should the government still use public funds to invest in museums instead?

Post your answer in the discussion section below, or view the discussion of those who responded "No" to the poll question [here](#).

Conditional Module

JAVASCRIPT INPUT EXAMPLE (1/1 points)

In the image below, click the cone.



[Reset](#) You have used 4 of 10 submissions

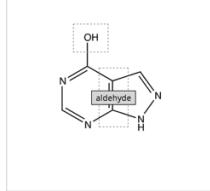
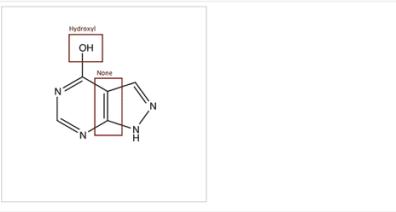
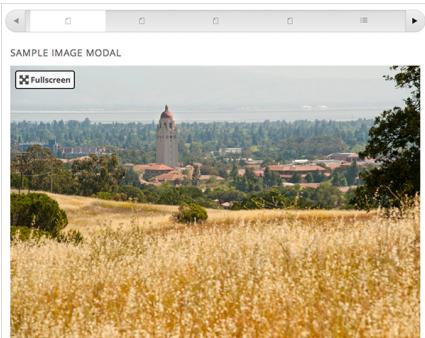
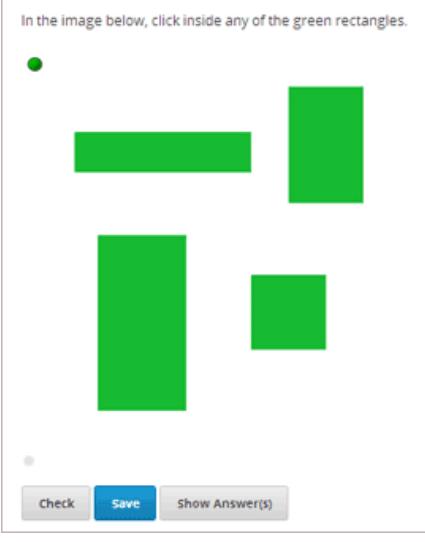
6.1. Creating Exercises and Tools

Annotation problems 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.

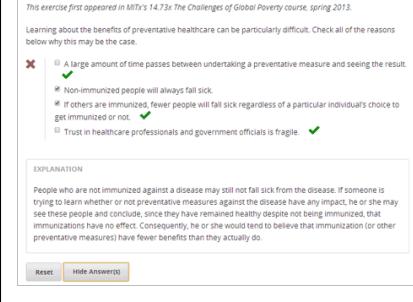
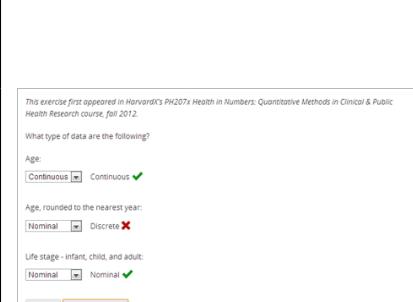
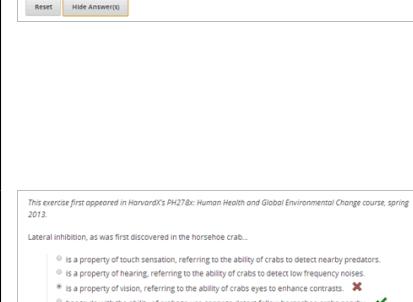
You can create a conditional module to control versions of content that groups of students see. For example, students who answer "Yes" to a poll question then see a different block of text from the students who answer "No" to that question.

Custom JavaScript

6.1.3 Image-Based Exercises and Tools

<p>The following drag and drop exercise first appeared in MIT's 7.000: Introduction to Biology - The Secret of Life course in March 2013.</p> <p>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.</p>  <p><input type="button"/> methyl <input type="button"/> hydroxyl <input type="button"/> amino <input type="button"/> carboxyl <input type="button"/> aldehyde <input type="button"/> phospho <input type="button"/></p>   <p>In the image below, click inside any of the green rectangles.</p>  <p><input type="button"/> Check <input type="button"/> Save <input type="button"/> Show Answer(s)</p>	<p>Drag and Drop Problem</p>	<p>In drag and drop problems, students respond to a question by dragging text or objects to a specific location on an image.</p>
<p>6.1. Creating Exercises and Tools</p> 	<p>Full Screen Image Tool</p>	<p>The Full Screen Image tool allows a student to enlarge an image in the whole browser window. This is useful when the image contains a large amount of detail and text that is easier to view in context when enlarged.</p>
	<p>Image Mapped</p>	<p>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.</p>

6.1.4 Multiple Choice Exercises and Tools

 <p>This exercise first appeared in Harvard's PH207x <i>Health in Numbers: Quantitative Methods in Clinical & Public Health Research</i> course, fall 2012.</p> <p>What type of data are the following?</p> <p>Age: <input checked="" type="checkbox"/> Continuous ✓</p> <p>Age, rounded to the nearest year: <input type="checkbox"/> Nominal ✗ Discrete ✗</p> <p>Life stage - infant, child, and adult: <input type="checkbox"/> Nominal ✗ Nominal ✓</p> <p>Reset Hide Answer(s)</p>	<p>Dropdown Problem</p>	<p>Dropdown problems allow the student to choose from a collection of answer options, presented as a dropdown list. Unlike multiple choice problems, whose answers are always visible directly below the question, dropdown problems don't show answer choices until the student clicks the dropdown arrow.</p>
 <p>This exercise first appeared in Harvard's PH278x <i>Human Health and Global Environmental Change</i> course, spring 2013.</p> <p>Lateral inhibition, as was first discovered in the horseshoe crab...</p> <p><input type="radio"/> is a property of touch sensation, referring to the ability of crabs to detect nearby predators.</p> <p><input type="radio"/> is a property of hearing, referring to the ability of crabs to detect low frequency noises.</p> <p><input checked="" type="radio"/> is a property of vision, referring to the ability of crabs eyes to enhance contrasts. ✗</p> <p><input type="radio"/> has to do with the ability of crabs to use sonar to detect fellow horseshoe crabs nearby.</p> <p><input type="radio"/> has to do with a weighting system in the crabs skeleton that allows it to balance in turbulent water.</p> <p>EXPLANATION</p> <p>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.</p> <p>Reset Hide Answer(s)</p>	<p>Multiple Choice Problem</p>	<p>In multiple choice problems, students select one option from a list of answer options. Unlike with dropdown problems, whose answer choices don't appear until the student clicks the drop-down arrow, answer choices for multiple choice problems are always visible directly below the question.</p>
 <p>THE VALUE OF PI (1/1 points)</p> <p>The numerical value of pi, rounded to two decimal points, is 3.24.</p> <p><input type="radio"/> True.</p> <p><input checked="" type="radio"/> False. The correct value is <input type="text" value="3.14"/> ✓</p> <p>Reset Hide Answer(s)</p>	<p>Multiple Choice and Numerical Input Problem</p>	<p>You can create a problem that combines a multiple choice and numerical input problems. Students not only select a response from options that you provide, but also provide more specific information, if necessary.</p>

6.1.5 STEM Exercises and Tools

CHEMICAL EQUATION PROBLEM (1 point possible)

Some problems may ask for a particular chemical equation. Practice by writing out the following reaction in the box below.

$\text{H}_2\text{SO}_4 \longrightarrow \text{H}^+ + \text{HSO}_4^-$

$\text{H}_2\text{SO}_4 \rightarrow \text{H}^+ + \text{HSO}_4^-$

$\text{H}_2\text{SO}_4 \rightarrow \text{H}^+ + \text{HSO}_4^-$

Some tips:

- Use real element symbols.
- Create subscripts by using plain text.
- Create superscripts by using a caret (^).
- Create the reaction arrow (\longrightarrow) by using " $>$ ".

SOLUTION

To create this equation, enter the following:

$\text{H}_2\text{SO}_4 \rightarrow \text{H}^+ + \text{HSO}_4^-$

Check **Save** **Hide Answer(s)** You have used 0 of 10 submissions

Chemical Equation Problem

Chemical equation problems allow the student to enter text that represents a chemical equation into a text box. The grader evaluates the student's response by using a Python script that you create and embed in the problem.

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 Vout such that

$$V_{\text{out}} \approx \frac{1}{2} V_1 + \frac{1}{6} V_2.$$

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.

Circuit Schematic Builder Problem

In circuit schematic builder problems, 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.

Gene Explorer Tool

The Gene Explorer (GeneX) simulates the transcription, splicing, processing, and translation of a small hypothetical eukaryotic gene. GeneX allows students to make specific mutations in a gene sequence, and it then calculates and displays the effects of the mutations on the mRNA and protein.

6.2 Annotation Problem

In an annotation problem, 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.

Annotation problems 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.

Instructions

[Collapse 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

[Hide Annotations](#)

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 [basileion].

QUESTION 1 (2/2 points)

Annotation Exercise

[Return to Annotation ↑](#)

| 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:

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.

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

126 the point. The heroes in the Iliad and Odyssey are still alive 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

6.2.1 Create an Annotation Problem

To create an annotation problem, you'll add the Annotation advanced component to your course, add the **Instructions** and **Guided Discussion** segments of the problem, and then the **Annotation problem** segment of the problem.

1. Add the Annotation advanced component.
 - (a) On the **Settings** menu, click **Advanced Settings**.
 - (b) 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).

Manual Policy Definition

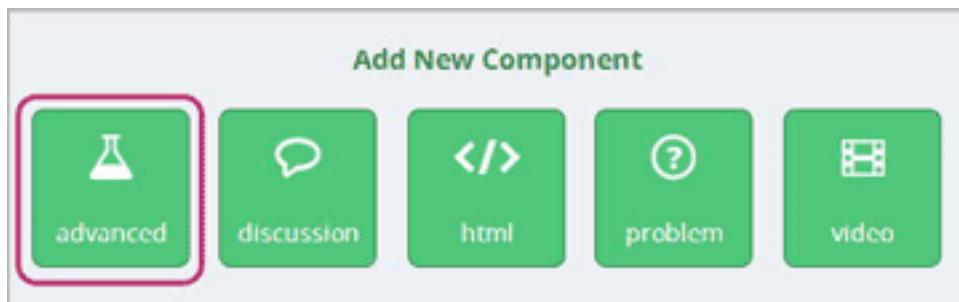
Manually Edit Course Policy Values (JSON Key / Value pairs)

Warning: Do not modify these policies unless you are familiar with their purpose.

Policy Key: advanced_modules

Policy Value: []

- (a) Under **Policy Value**, place your cursor between the brackets, and then enter the following. Make sure to include the quotation marks.
"annotatable"
- (b) At the bottom of the page, click **Save Changes**.
The page refreshes automatically. At the top of the page, you see a notification that your changes have been saved.
- (c) Return to the unit where you want to add the specialized problem. The list of possible components now contains an Advanced component.



2. Add the **Instructions** and **Guided Discussion** segments of the problem.
 1. In the unit where you want to create the problem, click **Advanced** under **Add New Component**.
 2. In the list of problem types, click **Annotation**.
 3. In the component that appears, click **Edit**.
 4. In the component editor, replace the example code with your own code.
 5. Click **Save**.
 3. Add the **Annotation problem** 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>
            <comment_prompt>PLACEHOLDER: Type your response below:</comment_prompt>
            <tag_prompt>PLACEHOLDER: In your response to this question, which tag below
do you choose?</tag_prompt>
        <options>
            <option choice="incorrect">PLACEHOLDER: Incorrect answer (to make this
option a correct or partially correct answer, change choice="incorrect"
to choice="correct" or choice="partially-correct")</option>
            <option choice="correct">PLACEHOLDER: Correct answer (to make this option
an incorrect or partially correct answer, change choice="correct" to
choice="incorrect" or choice="partially-correct")</option>
            <option choice="partially-correct">PLACEHOLDER: Partially correct answer
(to make this option a correct or partially correct answer,
change choice="partially-correct" to choice="correct" or choice="incorrect")
            </option>
        </options>
    </annotationinput>
</annotationresponse>
<solution>
    <p>PLACEHOLDER: Detailed explanation of solution</p>
</solution>
</problem>
```

4. Click **Save**.

6.3 Checkbox Problem

In checkbox problems, 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 problem correctly. Each checkbox problem must have at least one correct answer.

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.

[Reset](#)

[Hide Answer\(s\)](#)

6.3.1 Create a Checkbox Problem

You can create checkbox problems in the Simple Editor or in the Advanced Editor.

Note: All problems must include labels for accessibility. The label generally includes the text of the main question in your problem. To add a label for a common problem, surround the text of the label with angle brackets pointed toward the text (>>label text<<).

Simple Editor

1. Under **Add New Component**, click **Problem**.
2. In the **Select Problem Component Type** screen, click **Checkboxes** on the **Common Problem Types** tab.
3. In the Problem component that appears, click **Edit**.
4. In the component editor, replace the default text with the text of your problem. Enter each answer option on its own line.
5. Determine the text of the problem to use as a label, and then surround that text with two sets of angle brackets (>><<).
6. Select all the answer options, and then click the checkbox button.



When you do this, brackets appear next to each answer choice.

7. Add an **x** between the brackets for the correct answer or answers.
8. In the component editor, select the text of the explanation, and then click the explanation button to add explanation tags around the text.



9. On the **Settings** tab, specify the settings that you want.

10. Click **Save**.

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.
- [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 immunized (or other preventative measures) have fewer benefits than they actually do.

[explanation]

Advanced Editor

To create this problem in the Advanced Editor, click the **Advanced** tab in the Problem component editor, and then replace the existing code with the following code.

```
<problem>
    <p>Learning about the benefits of preventative healthcare can be particularly difficult. Check all of the reasons below why this may be the case.</p>

    <choiceresponse>
        <checkboxgroup direction="vertical" label="Check all of the reasons below why this may be the case">
            <choice correct="true"><text>A large amount of time passes between undertaking a preventative measure and seeing the result.</text></choice>
            <choice correct="false"><text>Non-immunized people will always fall sick.</text></choice>
            <choice correct="true"><text>If others are immunized, fewer people will fall sick regardless of a particular individual's choice.</text></choice>
            <choice correct="true"><text>Trust in healthcare professionals and government officials is fragile.</text></choice>
        </checkboxgroup>
    </choiceresponse>

    <solution>
        <div class="detailed-solution">
            <p>Explanation</p>
            <p>People who are not immunized against a disease may still not fall sick from the disease. If someone is immunized (or other preventative measures) have fewer benefits than they actually do.</p>
        </div>
    </solution>
</problem>
```

6.3.2 Checkbox Problem XML

Template

```
<problem>
  <p>Question text</p>

  <choiceresponse>

    <checkboxgroup direction="vertical" label="label text">
      <choice correct="false"><text>Answer option 1 (incorrect)</text></choice>
      <choice correct="true"><text>Answer option 2 (correct)</text></choice>
    </checkboxgroup>
  </choiceresponse>

  <solution>
    <div class="detailed-solution">
      <p>Solution or Explanation Heading</p>
      <p>Solution or explanation text</p>
    </div>
  </solution>

</problem>
```

Tags

- <choiceresponse> (required): Specifies that the problem contains options for students to choose from.
- <checkboxgroup> (required): Specifies that the problem is a checkbox problem.
- <choice> (required): Designates an answer option.

Tag: <choiceresponse>

Specifies that the problem contains options for students to choose from.

Attributes

(none)

Children

- <checkboxgroup>

Tag: <checkboxgroup>

Specifies that the problem is a checkbox problem.

Attributes

Attribute	Description
direction (optional)	Specifies the orientation of the list of answers. The default is vertical.
label (required)	Specifies the name of the response field.

Children

- <choice>

Tag: <choice>

Designates an answer option.

Attributes

Attribute	Description
true (at least one required)	Indicates a correct answer. For checkbox problems, one or more <choice> tags can contain a correct answer.
false (at least one required)	Indicates an incorrect answer.

Children

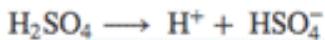
(none)

6.4 Chemical Equation Problem

The chemical equation problem type allows the student to enter text that represents a chemical equation into a text box. The system converts that text into a chemical equation below the text box. The grader evaluates the student's response by using a Python script that you create and embed in the problem.

CHEMICAL EQUATION PROBLEM (1 point possible)

Some problems may ask for a particular chemical equation. Practice by writing out the following reaction in the box below.



H2SO4 -> H⁺ + HSO4⁻

H₂SO₄ → H⁺ + HSO₄⁻

Some tips:

- Use real element symbols.
- Create subscripts by using plain text.
- Create superscripts by using a caret (^).
- Create the reaction arrow (→) by using "->".

SOLUTION

To create this equation, enter the following:

H2SO4 -> H⁺ + HSO4⁻

Check

Save

Hide Answer(s)

You have used 0 of 10 submissions

6.4.1 Create the Chemical Equation Problem

Chemical equation problems use MathJax to create formulas. For more information about using MathJax in Studio, see *A Brief Introduction to MathJax in Studio*.

To create the above chemical equation problem:

1. In the unit where you want to create the problem, click **Problem** under **Add New Component**, and then click the **Advanced** tab.
2. Click **Blank Advanced Problem**.
3. In the component that appears, click **Edit**.
4. In the component editor, paste the code from below.
5. Click **Save**.

Sample Chemical Equation Problem Code

```
<problem>
  <startouttext/>
  <p>Some problems may ask for a particular chemical equation. Practice by writing out the following</p>
  <math display="block">\text{H}_2\text{SO}_4 \rightarrow \text{H}^+ + \text{HSO}_4^-</math>

  <customresponse>
    <chemicalequationinput size="50" label="Enter the chemical equation"/>
    <answer type="loncapa/python">

if chemcalc.chemical_equations_equal(submission[0], 'H2SO4 -> H+ + HSO4-) :
    correct = ['correct']
else:
    correct = ['incorrect']

  </answer>
</customresponse>
<p>Some tips:</p>
<ul>
  <li>Use real element symbols.</li>
  <li>Create subscripts by using plain text.</li>
  <li>Create superscripts by using a caret (^).</li>
  <li>Create the reaction arrow ((\(\rightarrow\))) by using ">".</li>
</ul>

<endouttext/>

<solution>
<div class="detailed-solution">
<p>Solution</p>
<p>To create this equation, enter the following:</p>
<p>H2SO4 -> H+ + HSO4-</p>
</div>
</solution>
</problem>
```

6.4.2 Chemical Equation Problem XML

Template

```
<problem>
    <startouttext/>
    <p>Problem text</p>

    <customresponse>
        <chemicalequationinput size="NUMBER" label="LABEL TEXT"/>
        <answer type="loncapa/python">

if chemcalc.chemical_equations_equal(submission[0], 'TEXT REPRESENTING CHEMICAL EQUATION'):
    correct = ['correct']
else:
    correct = ['incorrect']

    </answer>
</customresponse>

<endouttext/>

<solution>
<div class="detailed-solution">
<p>Solution or Explanation Header</p>
<p>Solution or explanation text</p>
</div>
</solution>
</problem>
```

Tags

- `<customresponse>`: Indicates that this problem has a custom response.
- `<chemicalequationinput>`: Specifies that the answer to this problem is a chemical equation.
- `<answer type=loncapa/python>`: Contains the Python script that grades the problem.

Tag: `<customresponse>`

Indicates that this problem has a custom response. The `<customresponse>` tags must surround the `<chemicalequation>` tags.

Attributes

(none)

Children

- `<chemicalequationinput>`
- `<answer>`

Tag: `<chemicalequationinput>`

Indicates that the answer to this problem is a chemical equation and creates a response field where the student enters an answer.

Attributes

Attribute size	Description Specifies the size of the response field, in characters.
label (required)	Contains the text of the principal question in the problem.

Children

(none)

Tag: <answer>

Contains the Python script that grades the problem.

Attributes

Attribute type (required)	Description Must be “loncapa/python”.
------------------------------	--

Children

(none)

6.5 Circuit Schematic Builder Problem

In circuit schematic builder problems, 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.

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:

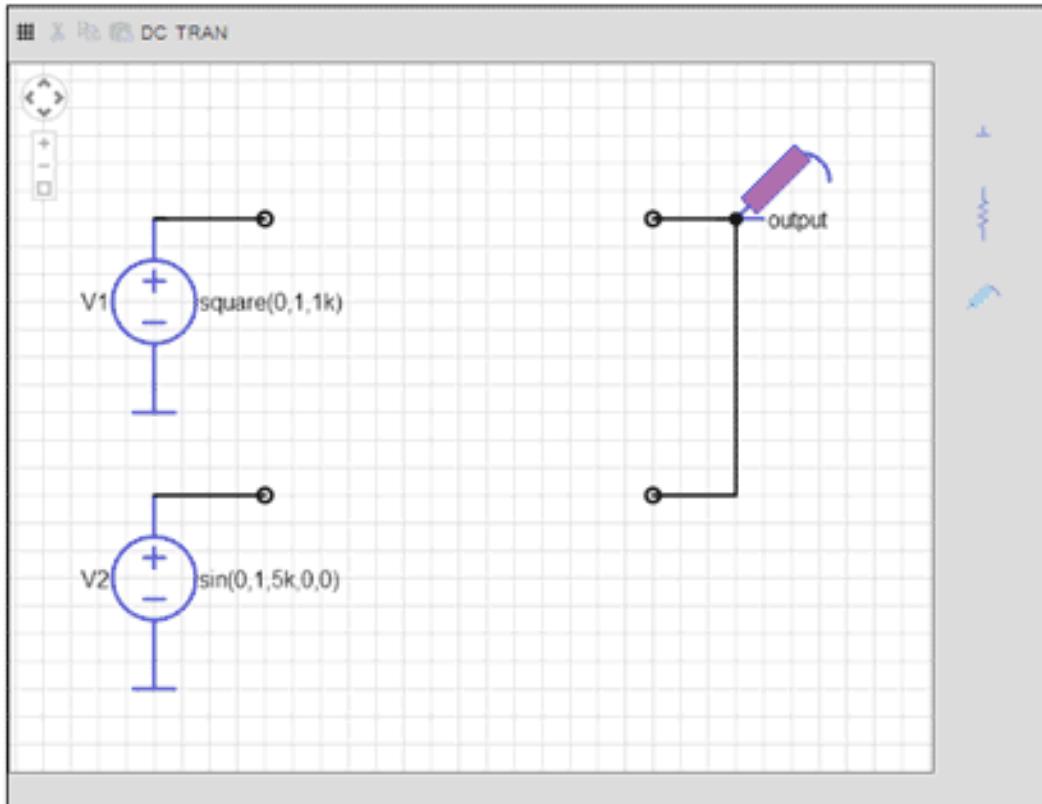
- V1 is a 1 kHz square wave that varies between 0V and +1V, and
- V2 is a 5 kHz sine wave that varies between -1V and +1V.

Please design a circuit that mixes V1 and V2 to produce Vout such that

$$V_{\text{out}} \approx \frac{1}{2} V_1 + \frac{1}{6} V_2.$$

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.



6.5.1 Create a Circuit Schematic Builder Problem

1. In the unit where you want to create the problem, 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**.

Problem Code

To create the problem in the image above, paste the following code into the Advanced Editor.

```
<problem>
    <p>Make a voltage divider that splits the provided voltage evenly.</p>
    <schematicresponse>
        <center>
            <schematic height="500" width="600" parts="g,r" analyses="dc"
initial_value="[[{"v": [168, 144, 0], "value": "dc(1)", "_json_": 0}, {"r": [296, 120, 0], "r": 1}], [{"l": "0"}]]>
        </center>
        <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>
    </schematicresponse>
    <schematicresponse>
        <p>Make a high pass filter.</p>
        <center>
            <schematic height="500" width="600" parts="g,r,s,c" analyses="ac"
submit_analyses="{"ac": [{"NodeA": 1, 9}]}>
            initial_value="[[{"v": [160, 152, 0], "name": "v1", "value": "sin(0,1,1,0,0)", "_json_": 0}, {"r": [120, 120, 0], "r": 1}], [{"l": "0"}]]>
        </center>
        <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>
    </schematicresponse>
    <solution>
        <div class="detailed-solution">
```

```
<p>Explanation</p>
<p>A voltage divider that evenly divides the input voltage can be formed with two identically valued resistors in series with the input and ground. This is a simple high-pass filter without any further constraints can be formed by simply putting a resistor between the input and ground.</p>
</div>
</solution>
</problem>
```

6.6 Conditional Module

6.6.1 Format description

The main tag of conditional module input is:

```
<conditional> ... </conditional>
```

conditional can include any number of any xmodule tags (html, video, poll, etc.) or show tags.

conditional tag

The main container for a single instance of a conditional module. The following attributes can be specified for this tag:

```
sources - location id of required modules, separated by ';'
[message | ""] - message for case, where one or more are not passed. Here you can use variable {link}
[submitted] - map to 'is_submitted' module method.
(pressing RESET button makes this function to return False.)

[correct] - map to 'is_correct' module method
[attempted] - map to 'is_attempted' module method
[poll_answer] - map to 'poll_answer' module attribute
[voted] - map to 'voted' module attribute
```

show tag

Symlink to some set of xmodules. The following attributes can be specified for this tag:

```
sources - location id of modules, separated by ','
```

6.6.2 Example

Examples of conditional depends on poll

```
<conditional sources="i4x://MITx/0.000x/poll_question/first_real_poll_seq_with_reset" poll_answer="man"
message="{link} must be answered for this to become visible.">
  <html>
    <h2>You see this because your vote value for "First question" was "man"!</h2>
  </html>
</conditional>
```

Examples of conditional depends on poll (use <show> tag)

```
<conditional sources="i4x://MITx/0.000x/poll_question/first_real_poll_seq_with_reset" poll_answer="ma  
message="{link} must be answered for this to become visible.">  
  <html>  
    <show sources="i4x://MITx/0.000x/problem/test_1; i4x://MITx/0.000x/Video/Avi_resources; i4x:  
  </html>  
</conditional>
```

Examples of conditional depends on problem

```
<conditional sources="i4x://MITx/0.000x/problem/Conditional:lec27_Q1" attempted="True">  
  <html display_name="HTML for attempted problem">You see this, cause "lec27_Q1" is attempted.</ht  
</conditional>  
<conditional sources="i4x://MITx/0.000x/problem/Conditional:lec27_Q1" attempted="False">  
  <html display_name="HTML for not attempted problem">You see this because "lec27_Q1" is not attemp  
</conditional>
```

6.7 Custom JavaScript Problem

Custom JavaScript display and grading problems (also called *custom JavaScript problems* or *JS Input problems*) allow you to create a custom problem or tool that uses JavaScript and then add the problem or tool directly into Studio. When you create a JS Input problem, Studio embeds the problem in an inline frame (IFrame) so that your students can interact with it in the LMS. You can grade your students' work using JavaScript and some basic Python, and the grading is integrated into the edX grading system.

The JS Input problem that you create must use HTML, JavaScript, and cascading style sheets (CSS). You can use any application creation tool, such as the Google Web Toolkit (GWT), to create your JS Input problem.

JAVASCRIPT INPUT EXAMPLE (1/1 points)

In the image below, click the cone.

edX

Reset You have used 4 of 10 submissions

6.7.1 Create a Custom JavaScript Display and Grading Problem

1. Create your JavaScript application, and then upload all files associated with that application to the **Files & Uploads** page.
2. In the unit where you want to create the problem, click **Problem** under **Add New Component**, and then click the **Advanced** tab.
3. Click **Custom JavaScript Display and Grading**.
4. In the component that appears, click **Edit**.
5. In the component editor, modify the example code according to your problem.

- All problems have more than one element. Most problems conform to the same-origin policy (SOP), meaning that all elements have the same protocol, host, and port. For example, http://store.company.com:81/subdirectory_1/JInputElement.html and http://store.company.com:81/subdirectory_2/JInputElement.js have the same protocol (http), host (store.company.com), and port (81).

If any elements of your problem use a different protocol, host, or port, you need to bypass the SOP. For example, <https://info.company.com/JInputElement2.html> uses a different protocol, host, and port. To bypass the SOP, change `sop="false"` in line 8 of the example code to `sop="true"`. For more information, see the same-origin policy page on the [Mozilla Developer Network](#) or on [Wikipedia](#).

6. If you want your problem to have a **Save** button, click the **Settings** tab, and then set **Maximum Attempts** to a number larger than zero.
7. Click **Save**.

Re-create the Example Problem

To re-create the example problem above, you'll need the following files.

- webGLDemo.html
- webGLDemo.js
- webGLDemo.css
- three.min.js

To download these files in a .zip archive, go to <http://files.edx.org/JInput.zip>.

Note: If you need to bypass the SOP, you'll also need the `jschannel.js` file, and your `webGLDemo.html` file will be slightly different. To download all these files in a .zip archive, go to http://files.edx.org/JInput_BypassSOP.zip.

1. Download and unpackage the files in either the `JInput.zip` file or the `JInput_BypassSOP.zip` file.
2. On the **Files & Uploads** page, upload all the files from the .zip file.
3. Create a new custom JavaScript display and grading problem component.
4. On the **Settings** tab, set **Maximum Attempts** to a number larger than zero.
5. In the problem component editor, replace the example code with the code below.
6. Click **Save**.

JavaScript Input Problem Code

```
<problem display_name="webGLDemo">
```

In the image below, click the cone.

```
<script type="loncapa/python">
import json
def vglcfn(e, ans):
    """
    par is a dictionary containing two keys, "answer" and "state"
    The value of answer is the JSON string returned by getGrade
    The value of state is the JSON string returned by getState
    """
    par = json.loads(ans)
    # We can use either the value of the answer key to grade
```

```
answer = json.loads(par["answer"])
return answer["cylinder"] and not answer["cube"]
# Or we can use the value of the state key
"""
state = json.loads(par["state"])
selectedObjects = state["selectedObjects"]
return selectedObjects["cylinder"] and not selectedObjects["cube"]
"""

</script>
<customresponse cfn="vglcfn">
    <jinput
        gradefn="WebGLDemo.getGrade"
        get_statefn="WebGLDemo.getState"
        set_statefn="WebGLDemo.setState"
        width="400"
        height="400"
        html_file="/static/webGLODemo.html"
    />
</customresponse>
</problem>
```

Note: When you create this problem, keep the following in mind.

- The webGLODemo.js file defines the three JavaScript functions (**WebGLDemo.getGrade**, **WebGLDemo.getState**, and **WebGLDemo.setState**).
 - The JavaScript input problem code uses **WebGLDemo.getGrade**, **WebGLDemo.getState**, and **WebGLDemo.setState** to grade, save, or restore a problem. These functions must be global in scope.
 - **WebGLDemo.getState** and **WebGLDemo.setState** are optional. You only have to define these functions if you want to conserve the state of the problem.
 - **Width** and **height** represent the dimensions of the IFrame that holds the application.
 - When the problem opens, the cone and the cube are both blue, or “unselected.” When you click either shape once, the shape becomes yellow, or “selected.” To unselect the shape, click it again. Continue clicking the shape to select and unselect it.
 - The response is graded as correct if the cone is selected (yellow) when the user clicks **Check**.
 - Clicking **Check** or **Save** registers the problem’s current state.
-

6.7.2 JavaScript Input Problem XML

JSInput allows problem authors to turn stand-alone HTML files into problems that can be integrated into the edX platform. Since its aim is flexibility, it can be seen as the input and client-side equivalent of **CustomResponse**.

A JSInput exercise creates an IFrame in a static HTML page, and passes the return value of author-specified functions to the enclosing response type (generally **CustomResponse**). JSInput can also store and retrieve state.

Template

The following is the basic format of a JSInput problem:

```
<problem>
    <script type="loncapa/python">
def all_true(exp, ans): return ans == "hi"
```

```

</script>
<customresponse cfn="all_true">
  <jinput gradefn="gradefn"
    height="500"
    get_statefn="getstate"
    set_statefn="setstate"
    html_file="/static/jinput.html"/>
</customresponse>
</problem>

```

The accepted attributes are:

Attribute Name	Value Type	Required	Default
html_file	URL string	Yes	None
gradefn	Function name	Yes	gradefn
set_statefn	Function name	No	None
get_statefn	Function name	No	None
height	Integer	No	500
width	Integer	No	400

Required Attributes

- **html_file**

The **html_file** attribute specifies the HTML file that the IFrame will point to. The HTML file must be located in the content directory.

The IFrame is created using the sandbox attribute. Although pop-ups, scripts, and pointer locks are allowed, the IFrame cannot access its parent's attributes.

The HTML file must contain a **gradefn** function that the JSInput file can access. To determine whether the **gradefn** function is accessible, in the console, make sure that **gradefn** returns the right thing. When JSInput uses the **gradefn** function, *gradefn* is called with *gradefn.call(obj)*, where **obj** is the object-part of **gradefn**. For example, if **gradefn** is **myprog.myfn**, JSInput calls **myprog.myfun.call(myprog)**. (This is to ensure “*this*” continues to refer to what *gradefn* expects.)

Aside from that, more or less anything goes. Note that currently there is no support for inheriting CSS or JavaScript from the parent (aside from the Chrome-only **seamless** attribute, which is set to True by default).

- **gradefn**

The **gradefn** attribute specifies the name of the function that will be called when a user clicks **Check**, and that returns the student's answer. Unless both the **get_statefn** and **set_statefn** attributes are also used, this answer is passed as a string to the enclosing response type. In the **customresponse** example above, this means **cfn** will be passed this answer as **ans**.

If the **gradefn** function throws an exception when a student attempts to submit a problem, the submission is aborted, and the student receives a generic alert. The alert can be customised by making the exception name **Waitfor Exception**; in that case, the alert message will be the exception message.

Important: To make sure the student's latest answer is passed correctly, make sure that the **gradefn** function is not asynchronous. Additionally, make sure that the function returns promptly. Currently the student has no indication that her answer is being calculated or produced.

Optional Attributes

- **set_statefn**

Sometimes a problem author will want information about a student's previous answers ("state") to be saved and reloaded. If the attribute **set_statefn** is used, the function given as its value will be passed the state as a string argument whenever there is a state, and the student returns to a problem. The function has the responsibility to then use this state appropriately.

The state that is passed is:

- The previous output of **gradefn** (i.e., the previous answer) if **get_statefn** is not defined.
- The previous output of **get_statefn** (see below) otherwise.

It is the responsibility of the iframe to do proper verification of the argument that it receives via **set_statefn**.

- **get_statefn**

Sometimes the state and the answer are quite different. For instance, a problem that involves using a javascript program that allows the student to alter a molecule may grade based on the molecule's hydrophobicity, but from the hydrophobicity it might be incapable of restoring the state. In that case, a *separate* state may be stored and loaded by **set_statefn**. Note that if **get_statefn** is defined, the answer (i.e., what is passed to the enclosing response type) will be a json string with the following format:

```
{  
    answer: '[answer string]'  
    state: '[state string]'  
}
```

The enclosing response type must then parse this as json.

- **height** and **width**

The **height** and **width** attributes are straightforward: they specify the height and width of the IFrame. Both are limited by the enclosing DOM elements, so for instance there is an implicit max-width of around 900.

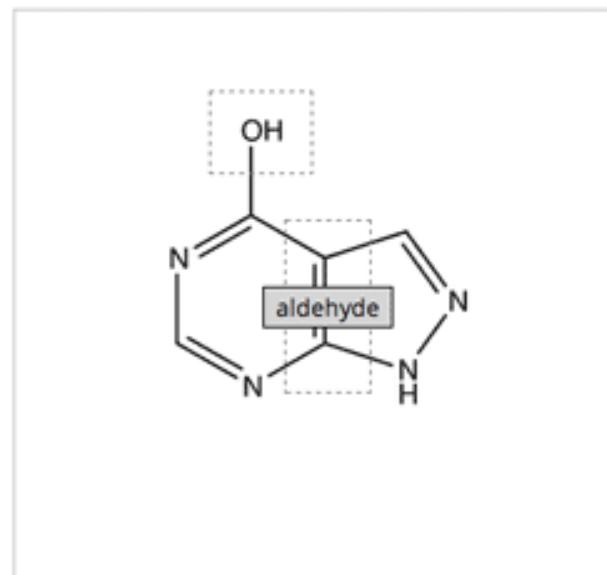
In the future, JSInput may attempt to make these dimensions match the HTML file's dimensions (up to the aforementioned limits), but currently it defaults to *500* and *400* for **height** and **width**, respectively.

6.8 Drag and Drop Problem

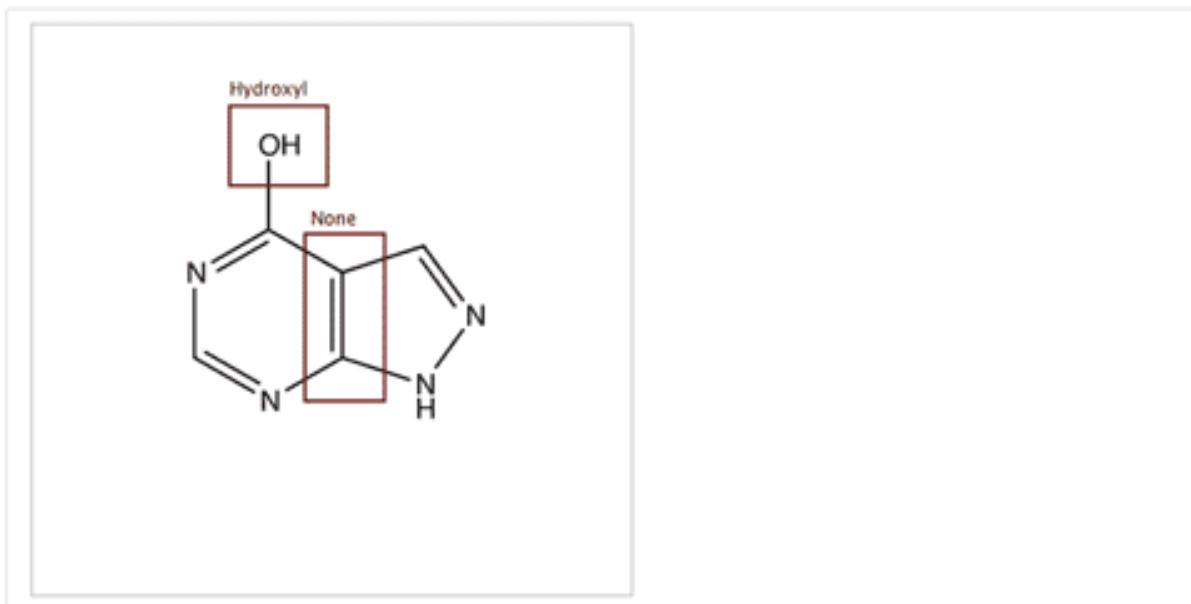
In drag and drop problems, students respond to a question by dragging text or objects to a specific location on an image.

The following drag and drop exercise first appeared in MIT's 7.00x: Introduction to Biology - The Secret of Life course in March 2013.

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.



◀ methyl hydroxyl amino carboxyl aldehyde phospho ▶



6.8.1 Create a Drag and Drop Problem

To create a simple drag and drop problem in which students drag labels onto an image, you'll upload the image that you want students to drag labels onto, and then create a Problem component.

1. On the **Files & Uploads** page, upload your image file. For more information about uploading files, see *Adding Files to a Course*.
2. In the unit where you want to create the problem, click **Problem** under **Add New Component**, and then click the **Advanced** tab.
3. Click **Drag and Drop**.
4. In the component that appears, click **Edit**.
5. In the component editor, replace the example text with the text of your problem.
6. In the `<drag_and_drop_input>` tag, replace https://studio.edx.org/c4x/edX/DemoX/asset/L9_buckets.png with the URL of your image file on the **Files & Uploads** page (for example, `/static/Image.png`).
7. For at least one `<draggable>` tag, replace the text of the **label** attribute with the text of the label you want students to drag. For example, if you want students to drag the word “Iceland” onto your image, the new tag would resemble the following:
`<draggable id="1" label="Iceland"/>`
8. Repeat the previous step for all the labels that you want to use. Make sure that the **id** attribute is different for each `<draggable>` tag.
9. Determine the coordinates and radius of the correct area on the image.
10. Under `correct_answer = {}`, add an entry for each label, using the following format. These values are in pixels:

```
'id': [[x coordinate, y coordinate], radius]
```

For example, if your image is 600 pixels wide and 400 pixels high, and you want your students to drag the Iceland label to an area in the upper-left part of the image and drag a Sweden label near the lower-right part of your image, the code would resemble the following (where 2 is the ID for the Sweden label):

```
correct-answer = {
    '1': [[50, 50], 75]
    '2': [[550, 350], 75]}
```

Note: Make sure the code contains the closing curly brace `}`.

11. Click **Save**.

Sample Drag and Drop Problem Code

To create the drag and drop problem that appears in the image above, you'll download two files from edX, upload these files to the **Files & Uploads** page, and then add the code for the problem to a Problem component.

1. Download the following files from edX:
 - Allopurinol.gif
 - AllopurinolAnswer.gif

To download both these files in a .zip archive, click <http://files.edx.org/DragAndDropProblemFiles.zip>.

2. Upload the Allopurinol.gif and AllopurinolAnswer.gif files to the **Files & Uploads** page.
3. In the unit where you want to create the problem, click **Problem** under **Add New Component**, and then click the **Advanced** tab.
4. Click **Drag and Drop**.
5. In the component that appears, click **Edit**.
6. In the component editor, replace the example code with the following code.
7. Click **Save**.

Problem Code:

```
<problem>
  <p> Allopurinol is a drug used to treat and prevent gout, a very painful form of arthritis. Once or
  <p> You are shown one of many possible molecules. On the structure of allopurinol below, identify the
  <customresponse>
    <drag_and_drop_input no_labels="true" one_per_target="true" target_outline="true" img="/static/Allopurinol.gif">
      <draggable can_reuse="true" label="methyl" id="1"/>
      <draggable can_reuse="true" label="hydroxyl" id="2"/>
      <draggable can_reuse="true" label="amino" id="3"/>
      <draggable can_reuse="true" label="carboxyl" id="4"/>
      <draggable can_reuse="true" label="aldehyde" id="5"/>
      <draggable can_reuse="true" label="phosphate" id="6"/>
      <draggable can_reuse="true" label="sulphydryl" id="7"/>
      <draggable can_reuse="true" label="phenyl" id="8"/>
      <draggable can_reuse="true" label="none" id="none"/>
    </drag_and_drop_input>
    <answer type="loncapa/python"> correct_answer = [ {'draggables': ['2'], 'targets': ['0'], 'rule': 'one_per_target'} ]
  </customresponse>
  <solution>
    
  </solution>
</problem>
```

6.8.2 Drag and Drop Problem XML**Template for Simple Problem**

```
<problem>
  <p>PROBLEM TEXT</p>
  <customresponse>
    <drag_and_drop_input img="/static/TARGET_IMAGE.png">
      <draggable id="1" label="LABEL 1"/>
      <draggable id="2" label="LABEL 2"/>
    </drag_and_drop_input>
    <answer type="loncapa/python">
correct_answer = {
    '1':      [[x, y], radius],
    '2':      [[x, y], radius]}
if draganddrop.grade(submission[0], correct_answer):
    correct = ['correct']
else:
    correct = ['incorrect']
```

```
</answer>
</customresponse>
```

Template for Advanced Problem

```
<problem>
    <customresponse>
        <text>
            <p>PROBLEM TEXT</p>
        </text>
        <drag_and_drop_input img="/static/TARGET_IMAGE.png" target_outline="true" one_per_target="true">
            <draggable id="1" label="LABEL 1" />
            <draggable id="2" label="LABEL 2" />
            <target id="A" x="NUMBER" Y="NUMBER" w="X+WIDTH" h="Y+HEIGHT"/>
            <target id="B" x="NUMBER" Y="NUMBER" w="X+WIDTH" h="Y+HEIGHT"/>
        </drag_and_drop_input>
        <answer type="loncapa/python">
correct_answer = [{{
    'draggables': ['1', '2'],
    'targets': ['A', 'B'],
    'rule': 'anyof'
}}]
if draganddrop.grade(submission[0], correct_answer):
    correct = ['correct']
else:
    correct = ['incorrect']
</answer>
</customresponse>
</problem>
```

Tags

- `<customresponse>`: Indicates that the problem is a custom response problem.
- `<drag_and_drop_input />`: Indicates the custom response problem is a drag and drop problem.
- `<draggable />`: Specifies a single object that a student will drag onto the base image.
- `<target>`: Specifies the location on the base image where a draggable must be dropped.

Tag: `<drag_and_drop_input />`

Attributes

Attribute	Description
img (required)	Relative path to an image that will be the base image. All draggables can be dragged onto it.
tar- get_outline	Specifies whether an outline (gray dashed line) should be drawn around targets (if they are specified). It can be either 'true' or 'false'. If not specified, the targets do not have outlines.
one_per_target	Specify whether to allow more than one draggable to be placed onto a single target. It can be either 'true' or 'false'. If not specified, the default value is 'true'.
no_labels (required)	default is false, in default behaviour if label is not set, label is obtained from id. If no_labels is true, labels are not automatically populated from id, and one can not set labels and obtain only icons.

Children

- <draggable>
- <target>

Tag: <draggable/>

Specifies a single draggable object in a drag and drop problem.

A draggable is what the user must drag out of the slider and drop onto the base image. After a drag operation, if the center of the draggable is located outside the rectangular dimensions of the image, it will be returned to the slider.

For the grader to work, each draggable must have a unique ID.

Attributes

Attribute	Description
id (required)	Unique identifier of the draggable object.
label (optional)	Text label that the user sees.
icon (optional)	For draggables that are images, the relative path to the image file.
can_reuse	true or false, default is false. If true, same draggable can be used multiple times.

Children

(none)

Tag: <target>

Specifies the location on the base image where a student must drop a draggable item. By design, if the center of a draggable lies within the target (i.e. in the rectangle defined by [[x, y], [x + w, y + h]], it is within the target. Otherwise, it is outside.

If you specify at least one target, and a student drops a draggable item on a location that is outside a target, the draggable item returns to the slider.

If you don't specify a target, a student can drop a draggable item anywhere on the base image.

Attributes

Attribute	Description
id (required)	Unique identifier of the target object.
x	X-coordinate on the base image where the top left corner of the target will be positioned.
y	Y-coordinate on the base image where the top left corner of the target will be positioned.
w	Width of the target, in pixels.
h	Height of the target, in pixels.

Children

(none)

For more information about how to create drag and drop problems, see [XML Format of Drag and Drop Input](#).

6.9 Dropdown Problem

Dropdown problems allow the student to choose from a collection of answer options, presented as a dropdown list. Unlike multiple choice problems, whose answers are always visible directly below the question, dropdown problems don't show answer choices until the student clicks the dropdown arrow.

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:

Continuous  Continuous 

Age, rounded to the nearest year:

Nominal  Discrete 

Life stage - infant, child, and adult:

Nominal  Nominal 

Reset

Hide Answer(s)

6.9.1 Create a Dropdown Problem

You can create dropdown problems in the Simple Editor or in the Advanced Editor.

Note: All problems must include labels for accessibility. The label generally includes the text of the main question in your problem. To add a label for a common problem, surround the text of the label with angle brackets pointed toward the text (>>label text<<).

Simple Editor

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 problem. Enter each of the possible answers on the same line, separated by commas.
5. Determine the text of the problem to use as a label, and then surround that text with two sets of angle brackets (>><<).
6. Select all the answer options, and then click the dropdown button.



When you do this, a double set of brackets ([[]]) appears and surrounds the answer options.

7. Inside the brackets, surround the correct answer with parentheses.

- In the component editor, select the text of the explanation, and then click the explanation button to add explanation tags around the text.



- On the **Settings** tab, specify the settings that you want.

10. Click **Save**.

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]]
```

Advanced Editor

To create this problem in the Advanced Editor, click the **Advanced** tab in the Problem component editor, and then replace the existing code with the following code.

Problem Code:

```
<problem>  
<p>  
    <em>This exercise first appeared in HarvardX's PH207x Health in Numbers: Quantitative Methods in C...</em>  
</p>  
<p>What type of data are the following?</p>  
<p>Age:</p>  
<optionresponse>  
    <optioninput options="('Nominal','Discrete','Continuous')" correct="Continuous" label="Age"/>  
</optionresponse>  
<p>Age, rounded to the nearest year:</p>  
<optionresponse>  
    <optioninput options="('Nominal','Discrete','Continuous')" correct="Discrete" label="Age, rounded to the nearest year"/>  
</optionresponse>  
<p>Life stage - infant, child, and adult:</p>  
<optionresponse>  
    <optioninput options="('Nominal','Discrete','Continuous')" correct="Nominal" label="Life stage"/>  
</optionresponse>  
</problem>
```

6.9.2 Dropdown Problem XML

Template

```
<problem>  
<p>  
    Problem text</p>  
<optionresponse>  
    <optioninput options="('Option 1','Option 2','Option 3')" correct="Option 2" label="label text"/>
```

```
</optionresponse>
<solution>
  <div class="detailed-solution">
    <p>Explanation or Solution Header</p>
    <p>Explanation or solution text</p>
  </div>
</solution>
</problem>

<problem>
  <p>Problem text</p>
  <optionresponse>
    options="('A','B')"
    correct="A"
    label="label text"
  </optionresponse>

  <solution>
    <div class="detailed-solution">
      <p>Explanation or Solution Header</p>
      <p>Explanation or solution text</p>
    </div>
  </solution>
</problem>
```

Tags

- `<optionresponse>` (required): Indicates that the problem is a dropdown problem.
- `<optioninput>` (required): Lists the answer options.

Tag: `<optionresponse>`

Indicates that the problem is a dropdown problem.

Attributes

(none)

Children

- `<optioninput>`

Tag: `<optioninput>`

Lists the answer options.

Attributes

Attribute	Description
options (re- quired)	Lists the answer options. The list of all answer options is surrounded by parentheses. Individual answer options are surrounded by single quotation marks (') and separated by commas (,).
correct (re- quired)	Indicates whether an answer is correct. Possible values are “true” and “false”. Only one correct attribute can be set to “true”.
label (re- quired)	Specifies the name of the response field.

Children

(none)

6.10 External Grader

6.10.1 Overview

An external grader is a service that receives student responses to a problem, processes those responses, and returns feedback and a problem grade to the edX platform. You build and deploy an external grader separately from the edX platform.

See the following sections for more information:

- *External Grader Example*
- *External Graders and XQueue*
- *The XQueue Interface*
- *Building an External Grader*
- *Create a Code Response Problem*

6.10.2 External Grader Example

An external grader is particularly useful for software programming courses where students are asked to submit complex code. The grader can run tests that you define on that code and return results to a student.

For example, you define a problem that requires students to submit Python code, and create a set of tests that an external grader can run to verify the submissions. When a student enters Python code for the problem and clicks **Check**, the code is sent to the grader for testing. If the code passes all tests, the grader returns the score and a string indicating that the solution is correct.

The screenshot shows a code editor window with the following Python code:

```
12  for month in range(12):
13      month += 1
14      tempBalance -= lowestPayment
15      tempBalance += (annualInterestRate/12) * tempBalance
16      if tempBalance < 0.1 and tempBalance > -0.1:
17          break;
18  if tempBalance > 0:
```

Below the code editor, a message says "Correct".

Under the message, it says "Test results" followed by a large empty box.

At the bottom right of the empty box, there is a link "See full output".

Below the empty box, there is a "CORRECT" status indicator and another "See full output" link.

The external grader can return a string with results, which the student can see by clicking **See full output**. This can be particularly useful when the solution is not correct and you want to return information about the failed tests. For example:

[Hide output](#)

INCORRECT

 Test Case 1

balance = 320000; annualInterestRate = 0.2

Your output:

Lowest Payment: 29157.51

*** ERROR: Your numerical answer was not within the error margin of our answer.
First error found: expected 29157.09 +/- 0.2, got 29157.51. ***

Correct output:

Lowest Payment: 29157.09

6.10.3 External Graders and XQueue

The edX Platform communicates with your external grader through XQueue. XQueue provides students' input to the grader; it then receives results from the grader and returns them to students.

Student submissions are collected in XQueue, where they remain until the grader actively retrieves, or pulls, the next submission from the queue for grading.

The external grader polls the XQueue through a RESTful interface at a regular interval. When the external grader pulls a submission, it runs the tests on it, then pushes the response back to XQueue through the RESTful interface. XQueue then delivers the response to the edX Learning Management System.

For example code of an external grader that uses Pull mode, see the [Stanford-Online repository xqueue_pull_ref](#).

External Grader Workflow

The following steps show the complete process:

1. The student either enters code or attaches a file for a problem, then clicks Check.
2. The external grader pulls the code from XQueue.
3. The external grader runs the tests that you created on the code.
4. The external grader returns the grade for the submission, as well as any results in a string, to XQueue.
5. The XQueue delivers the results to the edX Learning Management System.
6. The student sees the problem results and the grade.

The XQueue Name

Your course will use a specific XQueue name. You use this name when creating problems in edX Studio. You get this name from your edX Program Manager. As edX hosts many XQueues for different courses, it is critical that you use

the exact XQueue name in your problems, as described in the section *Create a Code Response Problem*.

6.10.4 The XQueue Interface

The student submission sent from XQueue to the grader, and the response sent from the grader to XQueue, are JSON objects, as described below.

Note: XQueue does not send the student ID to the external grader. Your grader cannot access student IDs or associate student IDs with submissions.

For the code for the XQueue interface, see the file `urls.py` in the edX XQueue repository.

Inputs to the External Grader

The grader receives student submissions as a JSON object with two keys:

- **student_response**: A string containing the student's code submission. The string comes from either input the student enters in the edX Learning Management System or a file the student attaches.
- **grader_payload**: An optional string that you can specify when creating the problem. For more information, see the section *Create a Code Response Problem*.

For example:

```
{  
    "xqueue_body":  
    "{  
        \"student_response\": \"def double(x):\n            return 2*x\\n\",  
        \"grader_payload\": \"problem_2\"\n    }"  
}
```

External Grader Responses

After running tests and recording results for a submission, the grader must return information by posting a JSON response. The JSON string contains an indication if the submission was correct, the score, and any message the tests create.

In the following example, the grader returns a JSON string that indicates the submission was correct, the score was 1, and a message:

```
{  
    "correct": true,  
    "score": 1,  
    "msg": "<p>The code passed all tests.</p>"  
}
```

6.10.5 Building an External Grader

Course staff, not edX, is responsible for building and deploying the external grader.

In addition to creating tests that are specific to the problems you use in your course, there are four areas that you must plan for when building an external grader:

- *Scale*

- *Security*
- *Reliability and Recovery*
- *Testing*

Scale

Your external grader must be able to scale to support the number of students in your course.

Keep in mind that student submissions will likely come in spikes, not in an even flow. For example, you should expect the load to be much greater than average in the hours before an exam is due. Therefore, you should verify that the external grader can process submissions from a majority of students in a short period of time.

Security

Students are submitting code that executes directly on a server that you are responsible for. It is possible that a student will submit malicious code. Your system must protect against this and ensure that the external grader runs only code that is relevant to the course problems. How you implement these protections depends on the programming language you are using and your deployment architecture. You must ensure that malicious code won't damage your server.

Reliability and Recovery

Once your course starts, many students will submit code at any possible time, and expect to see results quickly. If your external grader is prone to failure or unexpected delays, the student experience will be poor.

Therefore, you must ensure that your grader has high availability and can recover from errors. Prior to your course starting, you must have a plan to immediately notify the team responsible for operating your grader, as well as edX operations, when the grader fails. In collaboration with edX, you should develop a procedure to quickly identify the cause of failure, which can be your grader or edX's XQueue.

Contact your edX Program Manager for more information.

If you know the grader will be unavailable at a certain time for maintenance, you should *Add a Course Update*.

Testing

You should test your grader thoroughly before your course starts. Be sure to test incorrect code as well as correct code to ensure that the grader responds with appropriate scores and messages.

6.10.6 Create a Code Response Problem

You create a code response problem in edX Studio by adding a common blank problem, then editing the XML problem definition in the *The Advanced Editor*.

See *Working with Problem Components* for more information.

Following is a basic example of the XML definition of a problem that uses an external grader:

```
<problem display_name="Problem 6">
    <text>
        <p>Write a program that prints "hello world".</p>
    </text>
    <coderesponse queueName="my_course_queue">
        <textbox rows="10" cols="80" mode="python" tabsize="4"/>
    </coderesponse>
</problem>
```

```
<codeparam>
    <initial_display>
        # students please write your program here
        print ""
    </initial_display>
    <answer_display>
        print "hello world"
    </answer_display>
    <grader_payload>
        {"output": "hello world", "max_length": 2}
    </grader_payload>
</codeparam>
</coderesponse>
</problem>
```

Note the following about the XML definition:

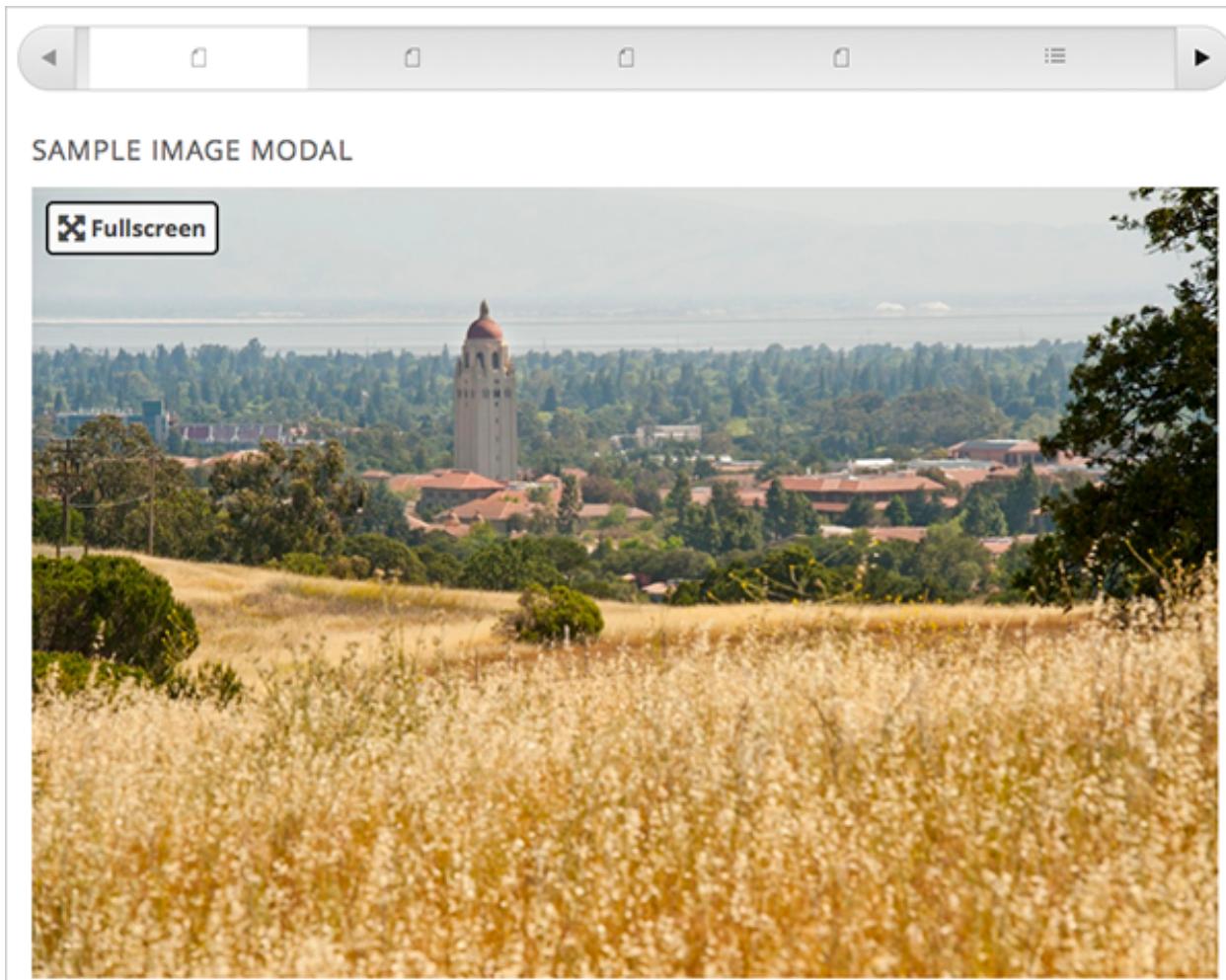
- **queuename:** The value of the queuename attribute of the `<coderesponse>` element maps to an XQueue that edX sets up for the course. You get this name from your edX Program Manager. You must use this exact name in order for the problem to communicate with the correct XQueue.
- **Input Type:** In this example, the input type is specified by the `<textbox>` element. When you use `<textbox>`, the student enters code in a browser field when viewing the course unit. The other element you can use to specify the input type is `<filesubmission>`, which enables the student to attach and submit a code file in the unit.
- **<grader_payload>:** You can use the `<grader_payload>` element to send information to the external grader in the form of a JSON object. For example, you can use `<grader_payload>` to tell the grader which tests to run for this problem.

6.11 Full Screen Image Tool

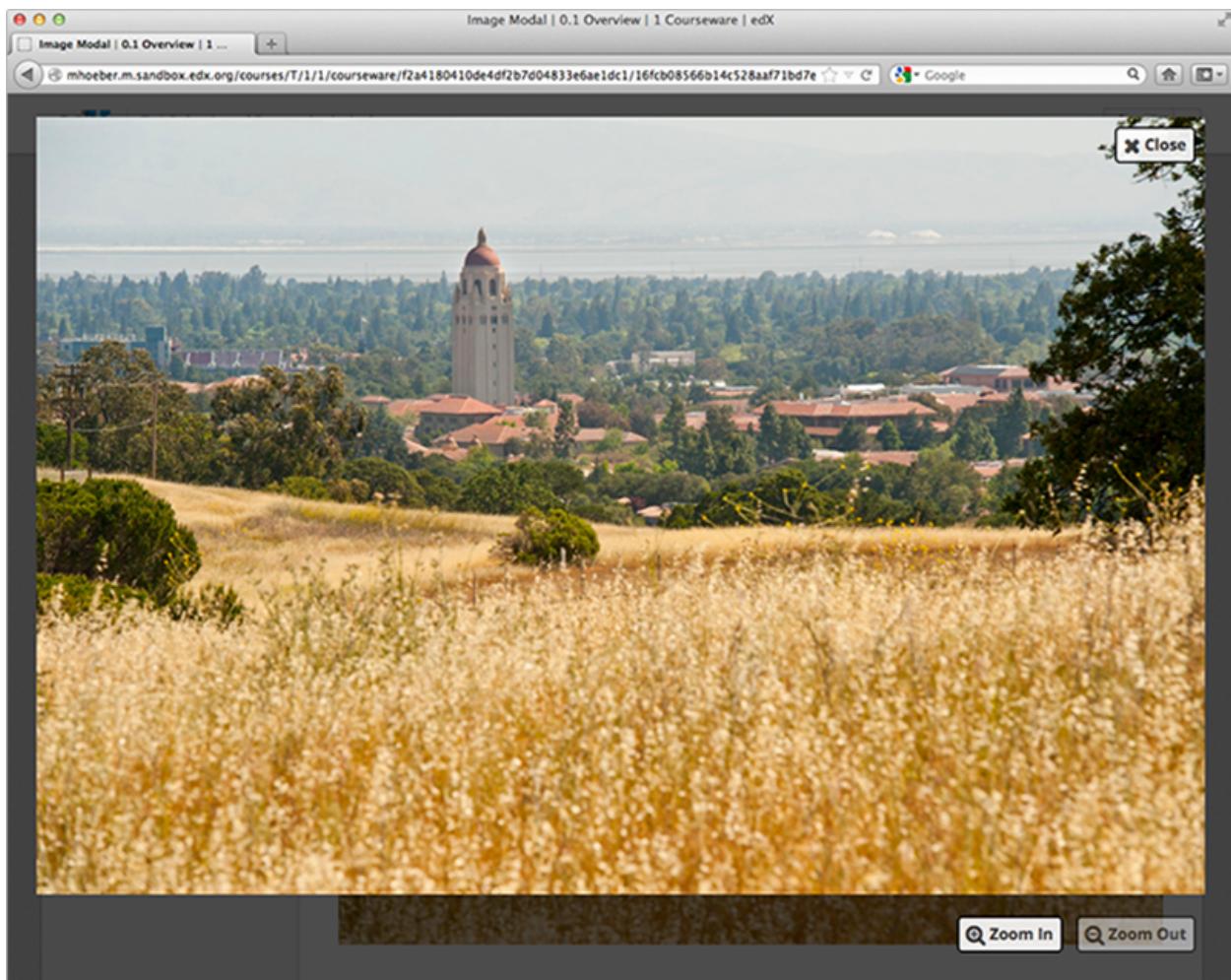
Some large images are difficult for students to view in the courseware. The full screen image tool allows students to enlarge the image, so they can see all the detail in context.

6.11.1 The Student View of a Full Screen Image

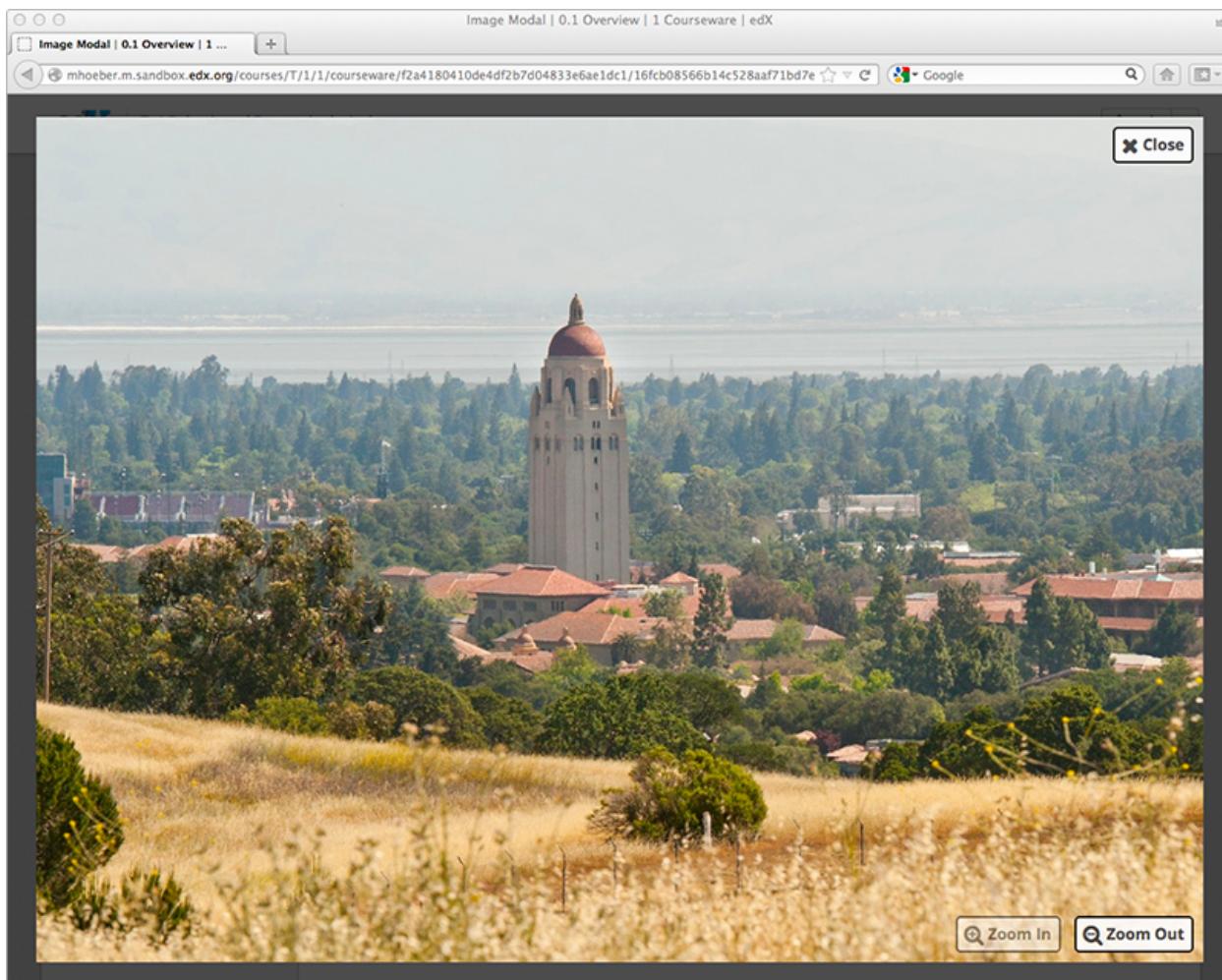
The student sees the full screen image in a unit page. When the student hovers the mouse pointer over the image, the **Fullscreen** button appears:



When the student clicks **Fullscreen**, the image opens and expands in the full browser window. The buttons **Close**, **Zoom In**, and **Zoom Out** appear:



The student can then zoom in on the image, and drag the image to view the desired part of it:



6.11.2 Create a Full Screen Image

1. Upload your image file to the **Files & Uploads** page. For more information about how to do this, see *Adding Files to a Course*.
2. Under **Add New Component**, click **html**, and then click **Full Screen Image**.
3. In the new component that appears, click **Edit**.
4. In the component editor, replace the default title, remove the instructional paragraph, and add text as needed.
5. Switch to the **HTML** tab.
6. Replace the following placeholders with your own content.
 - Replace the value of the `<a>` element's `href` attribute with the path to your image. Do not change the value of the `class` attribute. For example:
``
 - Replace the value of the `` element's `src` attribute with the path to your image. For example:
``
 - Ensure that the value of the `href` and `src` attributes are the same, and that you do not change the `class` attribute. Your sample code should look like the following:

```
<h2>Sample Image Modal</h2>
<a href="/static/Image1.jpg" class="modal-content">

</a>
```

Note: You can use this same HTML code in any HTML component, not just those components you created as full screen images.

- Click **Save** to save the HTML component.

6.12 Gene Explorer Tool

The Gene Explorer (GeneX), from the biology department at [UMB](#), simulates the transcription, splicing, processing, and translation of a small hypothetical eukaryotic gene. GeneX allows students to make specific mutations in a gene sequence, and it then calculates and displays the effects of the mutations on the mRNA and protein.

Specifically, the Gene Explorer does the following:

- Finds the promoter and terminator
- Reads the DNA sequence to produce the pre-mRNA
- Finds the splice sites
- Splices and tails the mRNA
- Finds the start codon
- Translates the mRNA

The screenshot shows the Gene Explorer tool's interface. At the top, there's a header "GENE EDITOR (1 point possible)". Below it, a "DNA: Promoter Terminator" section shows a sequence with a TATA box highlighted in green. A modal window titled "New DNA Sequence" is open, prompting the user to "Enter new DNA Sequence". The "OK" button is highlighted with a yellow box. To the right, a "pre-mRNA: Exon Int" section shows a sequence with an exon and intron highlighted in pink and yellow respectively. Below that, a "mature-mRNA and Protein (previous)" section shows the translated protein sequence: "N-Met Pro Leu Ser Asp Val Glu Arg Gly Pro-C". At the bottom, there are buttons for "Reset DNA Sequence" and "Enter New DNA Sequence", and a "Selected Base = " field.

For more information about the Gene Explorer, see [The Gene Explorer](#).

6.12.1 Gene Explorer Code

```
<problem>


Make a single base pair substitution mutation in the gene below that results in a protein that is



Note that a "single base pair substitution mutation" is when a single base is changed to another base.


<script type="loncapa/python">
def genex_grader(expect,ans):
    if ans=="CORRECT": return True
    import json
    ans=json.loads(ans)
    return ans["genex_answer"]=="CORRECT"
</script>
<customresponse cfn="genex_grader">
<editageneinput width="818" height="1000" dna_sequence="TAAGGCTATAACCGAGATTGATGCCTTGTGCGATAAGGTGTGTC">
</customresponse>
</problem>
```

In this code:

- **width** and **height** specify the dimensions of the application, in pixels.
- **genex_dna_sequence** is the default DNA sequence that appears when the problem opens.
- **dna_sequence** contains the application's state and the student's answer. This value must be the same as **genex_dna_sequence**.
- **genex_problem_number** specifies the number of the problem. This number is based on the five gene editor problems in the MITx 7.00x course—for example, if you want this problem to look like the second gene editor problem in the 7.00x course, you would set the **genex_problem_number** value to 2. The number must be 1, 2, 3, 4, or 5.

6.13 Google Instant Hangout Tool

This chapter describes how you can use instant hangouts in your course. See:

- *Overview*
- *Instant Hangouts in Your Course*
- *The Student Experience*
- *Limitations*
- *Create the Instant Hangout*

6.13.1 Overview

You can add the ability for students to participate in instant hangouts directly from your course.

With instant hangouts, students can:

- Interact through live video and voice.
- Share screens and watch videos together.
- Collaborate on documents.

For a full list of instant hangout features, see the [Google Hangouts page](#).

Note: Students who want to participate in instant hangouts must have a Google account. You should note this in your

course materials.

6.13.2 Instant Hangouts in Your Course

You can add one or more instant hangouts in your course. For example, you can add an instant hangout:

- In a page, to provide students with a hangout for the entire course. See *Adding Pages to a Course* for more information.
- In an HTML component, to provide a hangout for students working on that specific course unit. See *Working with HTML Components* for more information about creating HTML components.

An instant hangout is specific to the page it is opened from. For example, students who join a hangout from one course unit interact among themselves, while students who join a hangout from another unit interact in a different hangout.

6.13.3 The Student Experience

When you add the instant hangout to your course, a control for the hangout appears on that page. The following example shows the control in a course unit. The control shows that the student can start the hangout and be the first participant.

The screenshot shows a course navigation sidebar on the left with sections like 'Introduction', 'Example Week 1: Getting Started', 'Example Week 2: Get Interactive', 'Example Week 3: Be Social' (which is expanded to show 'Lesson 3 - Be Social', 'Homework - Find Your Study Buddy', and 'More Ways to Connect'), and 'About Exams and Certificates'. The main content area has a heading 'It is also possible to hang out with your classmates using Google Hangouts. See who's in there now, or arrange to meet people at specific times.' Below this is a box titled 'Instant Hangout' containing a red 'Start the Hangout' button and the text 'Be the first to join this discussion!'. There is also a small video camera icon.

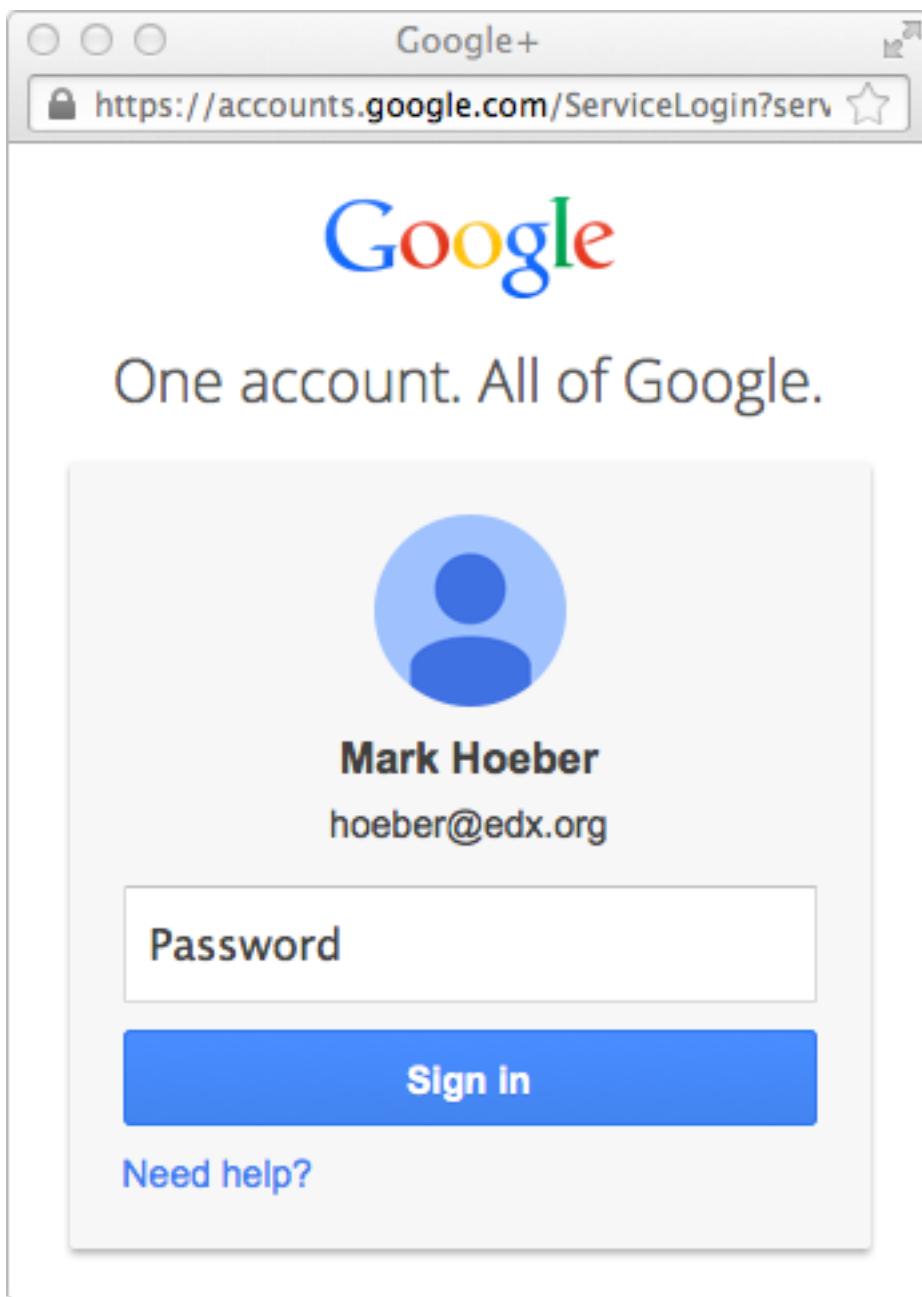
To start the hangout, the student clicks **Start the Hangout**. (After the first student clicks **Start the Hangout**, other students see a **Join the Hangout** button.)

The following example shows the control in a page when a hangout has already started. The control has a **Join the Hangout** button, and shows that one other student is already in the hangout.

The screenshot shows a horizontal navigation bar with the following items: Courseware, Course Info, Discussion, Wiki, Progress, and Instant Hangout. The 'Instant Hangout' tab is highlighted with a dark grey background and white text. Below the navigation bar, there is a message: "Join the Instant Hangout by clicking the icon below. You can use the hangout to have live video discussions with other students." Below this message is a rectangular box with rounded corners. Inside the box, the word "Instant Hangout" is displayed in bold black font. Below it is a red button with white text that says "Join the Hangout". To the left of the button is a small blue user icon. To the right of the button is a small red video camera icon. Below the button, the text "1 person discussing live" is shown next to a small blue user icon.

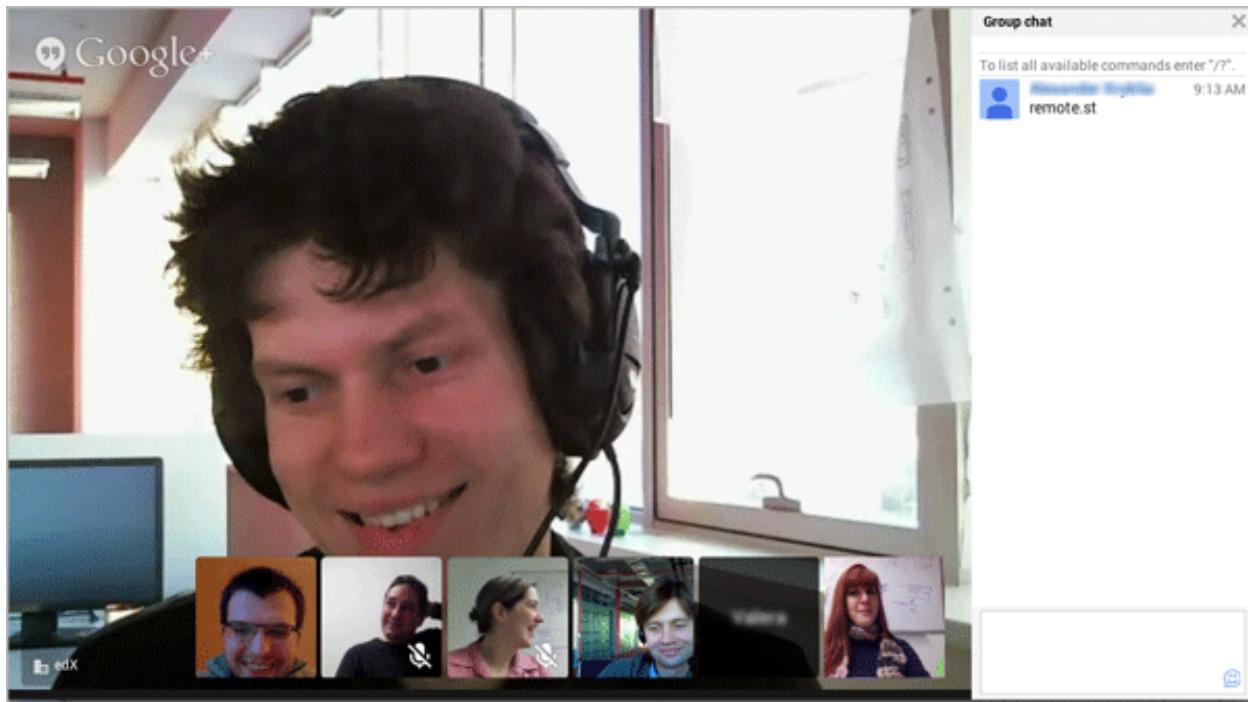
To join the hangout, the student clicks **Join the Hangout**.

If not already logged in, the student is prompted to log in to Google:



Students who do not have a Google account can create one from the login page.

After the student has logged in to Google, the hangout opens in a separate browser window:



6.13.4 Limitations

Currently, only ten students can join a single instant hangout. You should note this in your course materials.

Students in hangouts that are started from different pages in your course are counted separately. So you can have ten students participating in a hangout started from one unit, and ten other students in a hangout started from a different unit.

6.13.5 Create the Instant Hangout

To create an instant hangout in your course:

1. Get the instant hangout JavaScript file from GitHub.
2. Copy the text of this file into a text editor on your computer, and save the file as a JavaScript file (that is, when you save the file, change the extension from .txt to .js).

Note: Make sure that you copy the raw GitHub file, which does not contain formatting. Do not copy the formatted file. Any formatting will cause the JavaScript to not work correctly.

3. Upload the JavaScript file to the **Files & Uploads** page in your course. See *Adding Files to a Course* for more information.
4. In either a page or an HTML component, open the HTML editor.
5. On the first line, add the JavaScript file you uploaded in a <script> tag. For example, if your JavaScript file is named **instanthangouts-0.1.0.js**, you would enter the following:

```
<script src='/static/instanthangouts-0.1.0.js'>
```
6. After the <script> tag, add:

```
<div class='instanthangouts' />
```

7. Add any additional text and tags that you want.

For example, the complete HTML could be:

```
<p>Join an instant hangout by clicking the button below.  
You can use the hangout to have live video discussions with other students.</p>  
<script src='/static/instanthangouts-0.1.0.js'></script>  
<div class='instanthangouts' />
```

8. Test the instant hangout in your course.

Updating the JavaScript File

Google will periodically update the instant hangouts JavaScript file. To receive update notifications, go to the [instant hangouts repository page](#), and then click **Watch** in the upper-right area of the page.

To use an updated JavaScript file in your course, we recommend that you copy the JavaScript from the repository into a file that has the same name as the file that you uploaded to your course. When you upload the new file, the new file replaces the previous file.

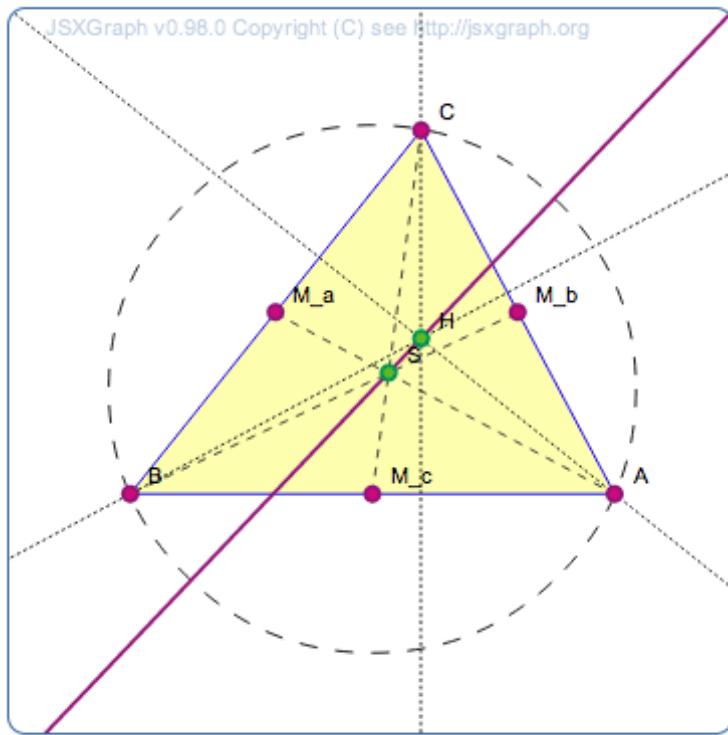
Warning: If you include version numbers in the file names of uploaded files, you will have to edit any HTML components or pages that include an instant hangout control every time that you update the JavaScript file.

6.14 IFrame Tool

An IFrame allows you to integrate ungraded exercises and tools from any Internet site into the body of your course. The IFrame appears inside an HTML component, and the exercise or tool appears inside the IFrame. IFrames can include your own tools or third-party tools.

EULER LINES

In the image below, drag any of the red points to a new location. As you drag the point, notice the way the point's position affects the triangle's Euler line.



Iframes are well-suited for tools that demonstrate a concept but that won't be graded or store student data. If you want to add a graded tool or exercise, add the tool as a *custom JavaScript problem* or an *LTI component*.

For more information about Iframes, see the Iframe specification.

6.14.1 Create an Iframe Tool

To add an exercise or tool in an Iframe, you'll create an Iframe HTML component and add the URL of the page that contains the exercise or tool to the component. You can also add text and images both before and after the Iframe.

Note: The URL of the page that contains the exercise or tool must start with https instead of http. If the URL starts with http, you have to work with the owner of that page to find out if an https version is available. Some websites do not allow their content to be embedded in Iframes.

1. Under **Add New Component**, click **html**, and then click **Iframe**.
2. In the new component that appears, click **Edit**.
3. In the toolbar in the component editor, click **HTML**.
4. In the HTML source code editor, locate the following HTML (line 7). This HTML includes the <iframe> element:

```
<p><iframe src="https://studio.edx.org/c4x/edX/DemoX/asset/eulerLineDemo.html" width="402" height="201"></iframe>
```

5. Replace the default URL in the **src** attribute (<https://studio.edx.org/c4x/edX/DemoX/asset/eulerLineDemo.html>) with the URL of the page that contains the exercise or tool. **This URL must start with https.** Make sure you don't delete the quotation marks that surround the URL.
6. Change the attributes in the IFrame element to specify any other settings that you want. For more information about these settings, see *IFrame Settings*. You can also change the text between the opening and closing <iframe> tags. A student only sees this text if the student uses a browser that does not support IFrames.
7. Click **OK** to close the HTML source code editor and return to the Visual editor.
8. In the Visual editor, replace the default text with your own text.
9. Click **Save**.

IFrame Settings

To specify settings for your IFrame, you'll add, remove, or change the attributes inside the opening <iframe> tag. The <iframe> tag **must** contain a **src** attribute that specifies the URL of the web page you want. Other attributes are optional.

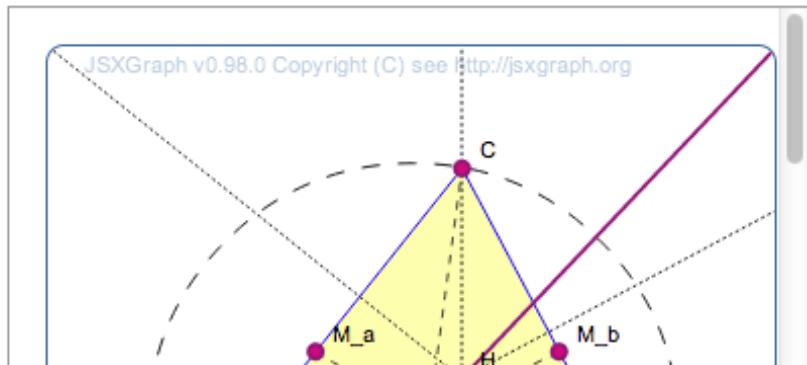
You can add these attributes in any order you want.

Attribute	Description
src (required)	Specifies the URL of the page that contains the exercise or tool.
width and height (optional)	Specify the width and height of the IFrame, in pixels or as a percentage. To specify the value in pixels, enter numerals. To specify a percentage, enter numerals followed by a percent sign. If you don't specify the width and height, the IFrame uses the dimensions that the linked page has set. These dimensions vary by website. If you change the width and height of the IFrame, the content from the linked page may be resized, or only part of the content may appear.
marginwidth and marginheight (optional)	Specify the size of the space between the edges of the IFrame and your exercise or tool, in pixels.
frameborder (optional)	Specifies whether a border appears around your IFrame. If the value is 0, no border appears. If the value is any positive number, a border appears.
scrolling (optional)	Specifies whether a scrollbar appears to help users see all of the IFrame's content if your IFrame is smaller than the exercise or tool it contains. For example, if the content in your IFrame is very tall, you can set the IFrame's height to a smaller number and add a vertical scroll bar for users, as in the first image below.

For example, compare how the different settings in each of the <iframe> elements below affect the IFrame.

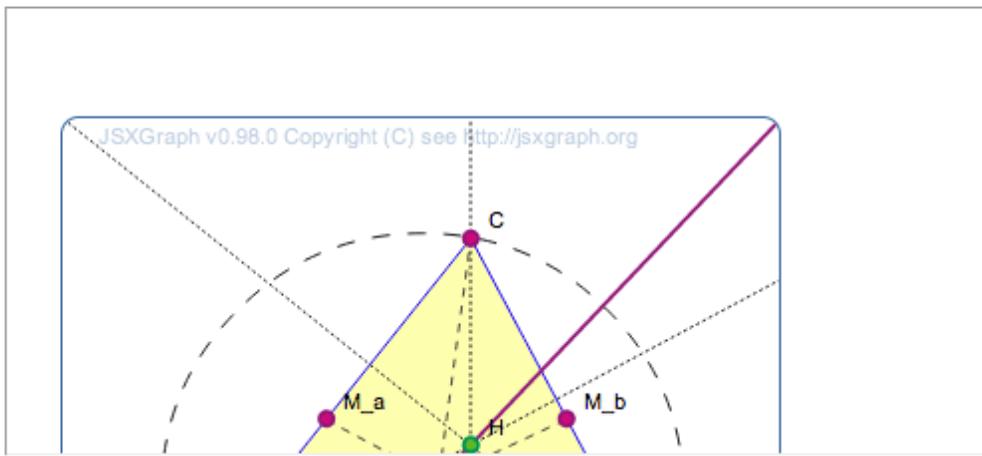
```
<p><iframe src="https://studio.edx.org/c4x/edX/DemoX/asset/eulerLineDemo.html" width="442" height="201"></iframe>
```

In the image below, drag any of the red points to a new location. As you drag the point, notice the way the point's position affects the triangle's Euler line.



<p><iframe src="https://studio.edx.org/c4x/edX/DemoX/asset/eulerLineDemo.html" width="550" height="250"></iframe>

In the image below, drag any of the red points to a new location. As you drag the point, notice the way the point's position affects the triangle's Euler line.



For more information about IFrame attributes, see the [IFrame specification](#).

6.15 Image Mapped Input Problem

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.

In the image below, click inside any of the green rectangles.

Check Save Show Answer(s)

6.15.1 Create an Image Mapped Input Problem

To create a image mapped input problem:

1. In the unit where you want to create the problem, 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**.

Problem Code:

```
<problem>
  <p><b>Example Problem</b></p>
  <startouttext>
    <p>In the image below, click the triangle.</p>
    <endouttext>
    <imageresponse>
      <imageinput src="/static/threeshapes.png" width="220" height="150" rectangle="(80,40)-(130,90)" />
    </imageresponse>
  </problem>
```

6.15.2 Image Mapped Input Problem XML

Template

```
<problem>
    <startouttext/>
        <p>In the image below, click the triangle.</p>
    <endouttext/>
        <imageresponse>
            <imageinput src="IMAGE FILE PATH" width="NUMBER" height="NUMBER" rectangle="(X-AXIS, Y-AXIS)-(X-AXIS, Y-AXIS)" />
        </imageresponse>
</problem>
```

Tags

- **<imageresponse>**: Indicates that the problem is an image mapped input problem.
- **<imageinput>**: Specifies the image file and the region in the file that the student must click.

Tag: `<imageresponse>`

Indicates that the problem is an image mapped input problem.

Attributes

(none)

Children

- `<imageinput>`

Tag: `<imageinput>`

Specifies the image file and the region in the file that the student must click.

Attributes

Attribute	Description
src (required)	The URL of the image
height (required)	The height of the image, in pixels
width (required)	The width of the image, in pixels
rectangle (required)	An attribute with four embedded values in the format (<start_width>,<start_height>)-(<end_width>,<end-height>). All coordinates start with (0,0) in the top left corner and increase in value toward the bottom right corner, very similar to the progression of reading English. The two coordinates defined form the two opposite corners of a box which a student can click inside of.

Children

(none)

6.16 LTI Component

You may have discovered or developed an external learning application that you want to add to your online course. Or, you may have a digital copy of your textbook that uses a format other than PDF. You can add external learning applications or textbooks to Studio by using a Learning Tools Interoperability (LTI) component. The LTI component is based on the [IMS Global Learning Tools Interoperability](#) version 1.1.1 specifications.

You can use an LTI component in several ways.

- **You can add external LTI content that is displayed only, such as** textbook content that doesn't require a student response.
- **You can add external LTI content that requires a student response. An** external provider will grade student responses.
- You can use the component as a placeholder for syncing with an external grading system.

For example, the following LTI component incorporates a Cerego tool that students interact with.

These activities will help you remember some of the facts you are learning about important jazz eras and artists. Your goal is to reach "memory permanence" for all items, so that the information sinks in and you're able to remember it long after you finish the course.

Your progress on this set, Jazz Facts 2, counts as 20% of your final grade. Although the recommended deadline has passed, you can still revisit Jazz Facts 1 and continue to work for credit. Links to all the sets are available on the [Practice Your Knowledge](#) page.

The screenshot shows a Cerego interface. At the top, there's a header with the Cerego logo, a progress bar indicating 47%, and links for Help and Settings. Below the header, a green banner asks, "Can you name a musician who played this instrument?". In the center, there's a large green button with a question mark. Below the button, the word "Tenor Saxophone" is displayed. At the bottom, there's a dark footer bar with a "Pause" button, a "Don't know it" button, and a green "Know it" button.

Before you create an LTI component from an external LTI provider in a unit, you need the following information.

- The **LTI ID**. This is a value that you create to refer to the external LTI provider. You should create an LTI ID that you can remember easily.

The LTI ID can contain uppercase and lowercase alphanumeric characters, as well as underscore characters (_). It can contain any number of characters. For example, you may create an LTI ID that is as simple as `test_lti_id`, or your LTI ID may be a string of numbers and letters such as `id_21441` or `book_lti_provider_from_new_york`.

- The **client key**. This value is a sequence of characters that you obtain from the LTI provider. The client key is used for authentication and can contain any number of characters. For example, your client key may be `b289378-f88d-2929-ctools.umich.edu`.

- The **client secret**. This value is a sequence of characters that you obtain from the LTI provider. The client secret is used for authentication and can contain any number of characters. For example, your client secret may be something as simple as **secret**, or it may be a string of numbers and letters such as **23746387264** or **yt4984yr8**.
- The **launch URL** (if the LTI component requires a student response that will be graded). You obtain the launch URL from the LTI provider. The launch URL is the URL that Studio sends to the external LTI provider so that the provider can send back students' grades.

6.16.1 Create an LTI Component

Creating an LTI component in your course has three steps.

1. Add LTI to the **advanced_modules** policy key.
2. Register the LTI provider.
3. Create the LTI component in an individual unit.

Step 1. Add LTI to the Advanced Modules Policy Key

1. On the **Settings** menu, click **Advanced Settings**.
2. On the **Advanced Settings** page, 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 **"lti"**. Make sure to include the quotation marks, but not the period.

The screenshot shows a user interface for managing policy keys. On the left, under 'Policy Key:', the value 'advanced_modules' is listed. To its right, under 'Policy Value:', there is a text input field containing the value '["lti"]'. The entire interface is contained within a light gray box.

Note If the **Policy Value** field already contains text, place your cursor directly after the closing quotation mark for the final item, and then enter a comma followed by **"lti"** (make sure that you include the quotation marks).

4. At the bottom of the page, click **Save Changes**.

The page refreshes automatically. At the top of the page, you see a notification that your changes have been saved.

Step 2. Register the External LTI Provider

To register the external LTI provider, you'll add the LTI ID, the client key, and the client secret in the **lti_passports** policy key.

1. On the **Advanced Settings** page, locate the **lti_passports** policy key.
 2. Under **Policy Value**, place your cursor between the brackets, and then enter the LTI ID, client key, and client secret in the following format (make sure to include the quotation marks and the colons).
- "lti_id:client_key:client_secret"**

For example, the value in the **lti_passports** field may be the following.

"test_lti_id:b289378-f88d-2929-ctools.umich.edu:secret"

If you have multiple LTI providers, separate the values with a comma. Make sure to surround each entry with quotation marks.

```
"test_lti_id:b289378-f88d-2929-ctools.umich.edu:secret",
"id_21441:b289378-f88d-2929-ctools.school.edu:23746387264",
"book_lti_provider_from_new_york:b289378-f88d-2929-ctools.company.com:yt4984yr8"
```

3. At the bottom of the page, click **Save Changes**.

The page refreshes automatically. At the top of the page, you see a notification that your changes have been saved, and you can see your entries in the **lti_passports** policy key.

Step 3. Add the LTI Component to a Unit

1. In the unit where you want to create the problem, click **Advanced** under **Add New Component**, and then click **LTI**.
2. In the component that appears, click **Edit**.
3. In the component editor, specify the settings that you want. See *LTI Component Settings* for a description of each setting.
4. Click **Save**.

6.16.2 LTI Component Settings

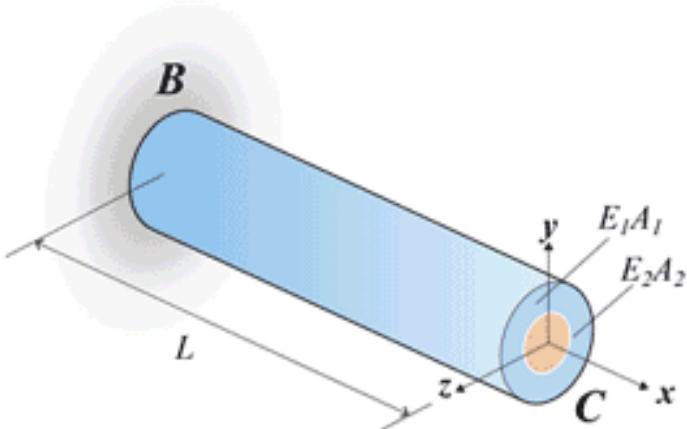
Setting	Description
Custom Parameters	<p>Enables you to add one or more custom parameters. For example, if you've added an e-book, you can set a custom parameter that opens the e-book to a specific page. You could also use a custom parameter to set the background color of the LTI component.</p> <p>Every custom parameter has a key and a value. You must add the key and value in the following format.</p> <p><code>key=value</code></p> <p>For example, a custom parameter may resemble the following.</p> <p><code>bgcolor=red</code></p> <p><code>page=144</code></p> <p>To add a custom parameter, click Add.</p>
Display Name	<p>Specifies the name of the problem. This name appears above the problem and in the course ribbon at the top of the page in the courseware. Analytics reports may also use the display name to identify this component.</p>
Hide External Tool	<p>Indicates whether you want to launch an external tool or to use this component as a placeholder for syncing with an external grading system. If you set the value to True, Studio hides the Launch button and any IFrames for this component. By default, this value is set to False.</p>
LTI ID	<p>Specifies the LTI ID for the external LTI provider. This value must be the same LTI ID that you entered on the Advanced Settings page.</p>
LTI URL	<p>Specifies the URL of the external tool that this component launches. This setting is applicable when Hide External Tool is set to False.</p>
Open in New Page	<p>Indicates whether the problem opens in a new page. If you set this value to True, the student clicks a link that opens the LTI content in a new window. If you set this value to False, the LTI content opens in an IFrame in the current page. This setting is applicable when Hide External Tool is set to False.</p>
Scored	<p>Indicates whether the LTI component receives a numerical score from the external LTI system. By default, this value is set to False.</p>
Weight	<p>Specifies the number of points possible for the problem. By default, if an external LTI provider grades the problem, the problem is worth 1 point, and a student's score can be any value between 0 and 1. This setting is applicable when Scored is set to True.</p> <p>For more information about problem weights and computing point scores, see <i>Problem Weight</i>.</p>

6.17 Math Expression Input Problems

In math expression input problems, students enter text that represents a mathematical expression into a field, and text is converted to a symbolic expression that appears below that field. Unlike numerical input problems, which only allow integers and a few select constants, math expression problems can include unknown variables and more complicated symbolic expressions.

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, $f_x(x) [\text{N/m}]$, the bar is observed to deform. The measured displacement field in the bar is $u_x(x) = ax^2 - 2aLx$, where $a = 0.05 [1/\text{m}]$ 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 $\mathcal{N}(x)$:

✓

$$12 \cdot a \cdot L \cdot E_1 \cdot A_2 \cdot \left(\frac{x}{L} - 1 \right)$$

What is the numerical value of $\mathcal{N}(x = 0)$ in Newtons?

✗

For more information about characters that students can enter, see *Math Response Formatting for Students*.

The grader uses a numerical sampling to determine whether the student's response matches the instructor-provided math expression, to a specified numerical tolerance. The instructor must specify the allowed variables in the expression as well as the range of values for each variable.

Warning: Math expression input problems cannot currently include negative numbers raised to fractional powers, such as $(-1)^{(1/2)}$. Math expression input problems can include complex numbers raised to fractional powers, or positive non-complex numbers raised to fractional powers.

When you create a math expression input problem in Studio, you'll use [MathJax](#) to change your plain text into "beautiful math." For more information about how to use MathJax in Studio, see *A Brief Introduction to MathJax in Studio*.

6.17.1 Create a Math Expression Input Problem

To create a math expression input problem:

1. In the unit where you want to create the problem, 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. To practice, you may want to use the sample problem code below.
5. Click **Save**.

Sample Problem Code

```
<problem>
    <p>Some problems may ask for a mathematical expression. Practice creating mathematical expressions</p>
    <p>Write an expression for the product of R_1, R_2, and the inverse of R_3.</p>
    <formulareponse type="ci" samples="R_1,R_2,R_3@1,2,3:3,4,5#10" answer="$VoVi">
        <responseparam type="tolerance" default="0.00001"/>
        <formulaequationinput size="40" label="Enter the equation"/>
    </formulareponse>

    <script type="loncapa/python">
VoVi = "(R_1*R_2)/R_3"
    </script>

    <p>Let <i>x</i> be a variable, and let <i>n</i> be an arbitrary constant. What is the derivative of</p>
    <script type="loncapa/python">
derivative = "n*x^(n-1)"
    </script>
    <formulareponse type="ci" samples="x,n@1,2:3,4#10" answer="$derivative">
        <responseparam type="tolerance" default="0.00001"/>
        <formulaequationinput size="40" label="Enter the equation"/>
    </formulareponse>

    <solution>
        <div class="detailed-solution">
            <p>Explanation or Solution Header</p>
            <p>Explanation or solution text</p>
        </div>
    </solution>
</problem>
```

6.17.2 Math Expression Input Problem XML

Templates

```
<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="R_1*R_2/R_3">
        <responseparam type="tolerance" default="0.00001"/>
        <formulaequationinput size="40" label="Enter the equation" />
    </formularesponse>
</problem>

<problem>
    <p>Problem text</p>
    <formularesponse type="ci" samples="VARIABLES@LOWER_BOUNDS:UPPER_BOUNDS#NUMBER_OF_SAMPLES" answer=">
        <responseparam type="tolerance" default="0.00001"/>
        <formulaequationinput size="20" label="Enter the equation" />
    </formularesponse>

<script type="loncapa/python">
PYTHON SCRIPT
</script>

<solution>
    <div class="detailed-solution">
        <p>Explanation or Solution Header</p>
        <p>Explanation or solution text</p>
    </div>
</solution>
</problem>
```

Tags

- `<formularesponse>`
- `<formulaequationinput />`
- `<responseparam>`
- `<script>`

Tag: `<formularesponse>`

Specifies that the problem is a math expression input problem. The `<formularesponse>` tag is similar to `<numericalresponse>`, but `<formularesponse>` allows unknown variables.

Attributes

type: Can be “cs” (case sensitive, the default) or “ci” (case insensitive, so that capitalization doesn’t matter in variable names).

answer: The correct answer to the problem, given as a mathematical expression. If you precede a variable name in the problem with a dollar sign (\$), you can include a script in the problem that computes the expression in terms of that variable.

samples: Specifies important information about the problem in four lists:

- **variables:** A set of variables that students can enter.

- **lower_bounds**: For every defined variable, a lower bound on the numerical tests to use for that variable.
- **upper_bounds**: For every defined variable, an upper bound on the numerical tests to use for that variable.
- **num_samples**: The number of times to test the expression.

Commas separate items inside each of the four individual lists, and the at sign (@), colon (:), and pound sign (#) characters separate the four lists. The format is the following:

```
"variables@lower_bounds:upper_bounds#num_samples
```

For example, a `<formularesponse samples="x, n@1, 2:3, 4#10">` tag that includes the **samples** attribute may look like either of the following.

```
<formularesponse samples="x, n@1, 2:3, 4#10">
<formularesponse samples="R_1, R_2, R_3@1, 2, 3:3, 4, 5#10">
```

Children

- `<formulaequationinput />`

Tag: `<formulaequationinput />`

Creates a response field where a student types an answer to the problem in plain text, as well as a second field below the response field where the student sees a typeset version of the plain text. The parser that renders the student's plain text into typeset math is the same parser that evaluates the student's response for grading.

Attributes

Attribute label (required)	Description Specifies the name of the response field.
size (optional)	Specifies the width, in characters, of the response field where students enter answers.

Children

(none)

Tag: `<responseparam>`

Used to define an upper bound on the variance of the numerical methods used to approximate a test for equality.

Attributes

Attribute default (re- quired)	Description A number or a percentage specifying how close the student and grader expressions must be. Failure to include a tolerance leaves expressions vulnerable to unavoidable rounding errors during sapling, causing some student input to be graded as incorrect, even if it is algebraically equivalent to the grader's expression.
type	“tolerance”—defines a tolerance for a number

Children

(none)

6.18 Molecule Editor Tool

Students can use the molecule editor to learn how to create molecules. The molecule editor allows students to draw molecules that follow the rules for covalent bond formation and formal charge, even if the molecules are chemically

impossible, are unstable, or do not exist in living systems. The molecule editor warns students if they try to submit a structure that is chemically impossible.

The molecule editor incorporates two tools: the JSME molecule editor created by Peter Ertl and Bruno Bienfait, and JSmol, a JavaScript-based molecular viewer from Jmol. (You don't need to download either of these tools—Studio uses them automatically.) For more information about the JSME molecule editor, see [JSME Molecule Editor](#). For more information about JSmol, see [JSmol](#).

MOLECULE EDITOR

The molecule builder makes creating and visualizing molecules easy. A chemistry professor may have you build and submit a molecule as part of an exercise.

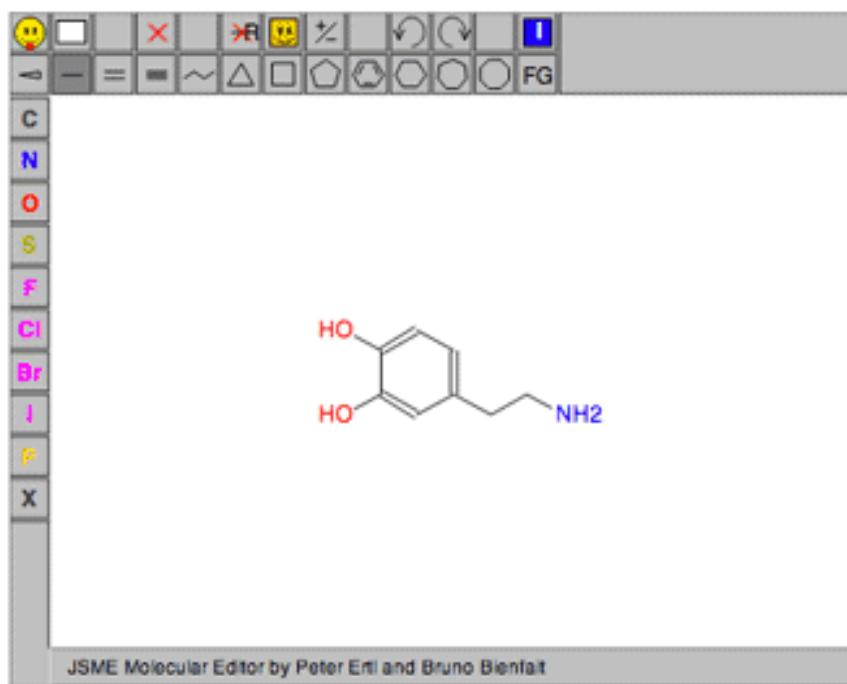
Using the Molecule Editor+ [open](#)

Are the molecules I've drawn chemically possible?+ [open](#)

MOLECULE EDITOR: DOPAMINE (1 point possible)

The dopamine molecule, as shown, cannot make ionic bonds. Edit the dopamine molecule so it can make ionic bonds.

When you are ready, click "Check." If you need to start over, click "Reset."



6.18.1 Create the Molecule Editor

To create a molecule editor, you need the following files:

- MoleculeAnswer.png
 - MoleculeEditor_HTML.png
 - dopamine.mol

To download all of these files in a .zip archive, go to <http://files.edx.org/MoleculeEditorFiles.zip>.

Note: The molecule that appears when the tool starts is a dopamine molecule. To use a different molecule, download the .mol file for that molecule from the [list of molecules](#) on the BioTopics website. Then, upload the .mol file to the **Files & Uploads** page for your course in Studio, and change “dopamine.mol” in the example code to the name of your .mol file.

To create the molecule editor that appears in the image above, you need an HTML component followed by a Problem component.

1. Upload all of the files listed above to the **Files & Uploads** page in your course.
 2. Create the HTML component.
 1. In the unit where you want to create the problem, click **HTML** under **Add New Component**, and then click **HTML**.
 2. In the component that appears, click **Edit**.
 3. In the component editor, paste the HTML component code from below.
 4. Make any changes that you want, and then click **Save**.
 3. Create the Problem component.
 1. Under the HTML component, click **Problem** under **Add New Component**, and then click **Blank Advanced Problem**.
 2. In the component that appears, click **Edit**.
 3. In the component editor, paste the Problem component code from below.
 4. Click **Save**.

Molecule Editor Code

To create the molecule editor, you need an **HTML component** and a **Problem component**.

HTML Component Code

```
<h2>Molecule Editor</h2>


The molecule editor makes creating and visualizing molecules easy. A chemistry professor may have


<div>
<script type="text/javascript">// <![CDATA[
function toggle2(showHideDiv, switchTextDiv) {
    var ele = document.getElementById(showHideDiv);
    var text = document.getElementById(switchTextDiv);
    if(ele.style.display == "block") {
        ele.style.display = "none";
        text.innerHTML = "+ open";
    } else {
        ele.style.display = "block";
        text.innerHTML = "- close";
    }
}
</script>
```

```
        }
    else {
        ele.style.display = "block";
        text.innerHTML = "- close";
    }
}
// ]]></script>
</div>
<div>
<style type="text/css"></style>
</div>
<div id="headerDiv">
<div id="titleText">Using the Molecule Editor<a id="myHeader" href="javascript:toggle2('myContent','
</div>
<div id="contentDiv">
<div id="myContent" style="display: none;">
<p>In this problem you will edit a molecule using the molecular drawing program shown below:</p>
</div>
</div>
<p>&ampnbsp</p>
<div id="headerDiv">
<div id="titleText">Are the molecules I've drawn chemically possible?<a id="IntroductionHeader" href=
</div>
<div id="contentDiv">
<div id="IntroductionContent" style="display: none;">
<p>The chemical editor you are using ensures that the structures you draw are correct in one very na
<p>If you submit a structure that includes atoms that are not possible or are beyond the scope of th
</div>
</div>
<div id="ap_listener_added">&ampnbsp</div>
```

Problem Component Code

```
<problem>
<p>The dopamine molecule, as shown, cannot make ionic bonds. Edit the dopamine molecule so it can ma
<p>When you are ready, click Check. If you need to start over, click Reset.</p>
<script type="loncapa/python">
def check1(expect, ans):
    import json
    mol_info = json.loads(ans) ["info"]
    return any(res == "Can Make Ionic Bonds" for res in mol_info)
</script>
<customresponse cfn="check1">
    <editamoleculeinput file="/static/dopamine.mol">
        </editamoleculeinput>
    </customresponse>
    <solution>
        
    </solution>
</problem>
```

Problem 2

```
<problem>
<p>The dopamine molecule, as shown, cannot make strong hydrogen bonds. Edit the dopamine molecule so
<script type="loncapa/python">
def grader_1(expect, ans):
```

```
import json
mol_info = json.loads(ans) ["info"]
return any(res == "Cannot Make Strong Hydrogen Bonds" for res in mol_info)
</script>
<customresponse cfn="grader_1">
    <editamoleculeinput file="/static/dopamine.mol">
    </editamoleculeinput>
</customresponse>
</problem>
```

Problem 3

```
<problem>
<p>The dopamine molecule has an intermediate hydrophobicity. Edit the dopamine molecule so that it is
<script type="loncapa/python">
def grader_2(expect, ans):
    import json
    mol_info = json.loads(ans) ["info"]

    hydrophobicity_index_str=mol_info[0]
    hydrophobicity_index=float(hydrophobicity_index_str[23:])
    return hydrophobicity_index > .490
</script>
<customresponse cfn="grader_2">
    <editamoleculeinput file="/static/dopamine.mol">
    </editamoleculeinput>
</customresponse>
</problem>
```

6.19 Multiple Choice Problem

In multiple choice problems, students select one option from a list of answer options. Unlike with dropdown problems, whose answer choices don't appear until the student clicks the drop-down arrow, answer choices for multiple choice problems are always visible directly below the question.

This exercise first appeared in HarvardX's PH278c: Human Health and Global Environmental Change course, spring 2013.

Lateral inhibition, as was first discovered in the horseshoe 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.

[Reset](#)

[Hide Answer\(s\)](#)

Multiple choice problems also have several advanced options, such as presenting a random set of choices to each student. For more information about these options, see *Advanced Options for Multiple Choice Problems*.

6.19.1 Create a Multiple Choice Problem

You can create multiple choice problems in the Simple Editor or in the Advanced Editor.

Note: All problems must include labels for accessibility. The label generally includes the text of the main question in your problem. To add a label for a common problem, surround the text of the label with angle brackets pointed toward the text (>>label text<<).

Simple Editor

1. Under **Add New Component**, click **Problem**.
2. In the **Select Problem Component Type** screen, click **Multiple Choice** on the **Common Problem Types** tab.
3. When the new Problem component appears, click **Edit**.
4. In the component editor, replace the sample problem text with the text of your problem. Enter each answer option on its own line.
5. Determine the text of the problem to use as a label, and then surround that text with two sets of angle brackets (>><<).
6. Select all the answer options, and then click the multiple choice button.



When you do this, the component editor adds a pair of parentheses next to each possible answer.

7. Add an “x” between the parentheses next to the correct answer.
8. In the component editor, select the text of the explanation, and then click the explanation button to add explanation tags around the text.



9. On the **Settings** tab, specify the settings that you want.

10. Click **Save**.

For the example problem above, the text in the Problem component is the following.

```
>>Lateral inhibition, as was first discovered in the horseshoe 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]

Advanced Editor

To create this problem in the Advanced Editor, click the **Advanced** tab in the Problem component editor, and then replace the existing code with the following code.

```
<problem>
<p>Lateral inhibition, as was first discovered in the horseshoe crab...</p>
<multiplechoiceresponse>
  <choicegroup type="MultipleChoice" label="Lateral inhibition, as was first discovered in the horseshoe crab">
    <choice correct="false">is a property of touch sensation, referring to the ability of crabs to detect nearby predators.
    <choice correct="false">is a property of hearing, referring to the ability of crabs to detect low frequency noises.
    <choice correct="false">is a property of vision, referring to the ability of crabs eyes to enhance contrasts.
    <choice correct="true">has to do with the ability of crabs to use sonar to detect fellow horseshoe crabs nearby.
    <choice correct="false">has to do with a weighting system in the crabs skeleton that allows it to balance in turbulent water.
  </choicegroup>
</multiplechoiceresponse>
<solution>
  <div class="detailed-solution">
```

```
<p>Explanation</p>
<p>Horseshoe crabs were essential to the discovery of lateral inhibition, a property of vision pr
</div>
</solution>
</problem>
```

6.19.2 Advanced Options for Multiple Choice Problems

Multiple choice problems have several advanced options. You can change the order of answers in the problem, include explanations that appear when a student selects a specific incorrect answer, or present a random set of choices to each student. For more information, see the following:

- *Shuffle Answers in a Multiple Choice Problem*
- *Targeted Feedback in a Multiple Choice Problem*
- *Answer Pools in a Multiple Choice Problem*

Shuffle Answers in a Multiple Choice Problem

Optionally, you can configure a multiple choice problem so that it shuffles the order of possible answers.

For example, one view of the problem could be:

The screenshot shows a mobile-style interface for a multiple-choice question. At the top, there is a navigation bar with a back arrow on the left and three horizontal dots in the center. Below the bar, the text "MULTIPLE CHOICE (1 point possible)" is displayed in bold capital letters. The question text "What Apple device competed with the portable CD player?" follows. To the left of the question is a vertical list of four options, each preceded by a radio button. The options are: "The iPad", "Napster", "The iPod", and "The vegetable peeler". Below the question and options are two buttons: a grey "Check" button on the left and a blue "Save" button on the right.

MULTIPLE CHOICE (1 point possible)

What Apple device competed with the portable CD player?

- The iPad
- Napster
- The iPod
- The vegetable peeler

Check Save

And another view of the same problem, for another student or for the same student of a subsequent view of the unit, could be:

The screenshot shows a mobile-style interface for a multiple choice exercise. At the top is a navigation bar with a back arrow on the left and a menu icon on the right. Below the bar is the question text: "MULTIPLE CHOICE (1 point possible)". The question itself is: "What Apple device competed with the portable CD player?". To the left of the question is a vertical list of four answer options, each preceded by a radio button. The options are: "Napster", "The iPad", "The vegetable peeler", and "The iPod". The "The iPod" option is highlighted with a grey background. At the bottom of the screen are two buttons: a grey "Check" button on the left and a blue "Save" button on the right.

You can also have some answers shuffled, but not others. For example, you may want to have the answer “All of the Above” fixed at the end of the list, but shuffle other answers.

You can configure the problem to shuffle answers through *The Simple Editor* or *The Advanced Editor*.

Use the Simple Editor to Shuffle Answers

You can configure the problem to shuffle answers in *The Simple Editor*.

For example, the following text defines a multiple choice problem, before shuffling is enabled. The (x) indicates the correct answer:

```
>>What Apple device competed with the portable CD player?<<
  ( ) The iPad
  ( ) Napster
  (x) The iPod
  ( ) The vegetable peeler
```

To add shuffling to this problem, add ! in the parenthesis of the first answer:

```
>>What Apple device competed with the portable CD player?<<
  (!) The iPad
  ( ) Napster
```

- (x) The iPod
- () The vegetable peeler

To fix an answer's location in the list, add @ in the parenthesis of that answer:

```
>>What Apple device competed with the portable CD player?<<
(!) The iPad
( ) Napster
(x) The iPod
( ) The vegetable peeler
(@) All of the above
```

You can combine symbols within parenthesis as necessary. For example, to show the correct answer in a fixed location, you could use:

(x@) The iPod

Use the Advanced Editor to Shuffle Answers

You can configure the problem to shuffle answers through XML in *The Advanced Editor*.

For example, the following XML defines a multiple choice problem, before shuffling is enabled:

```
<p>What Apple device competed with the portable CD player?</p>
<multiplechoiceresponse>
  <choicegroup type="MultipleChoice">
    <choice correct="false">The iPad</choice>
    <choice correct="false">Napster</choice>
    <choice correct="true">The iPod</choice>
    <choice correct="false">The vegetable peeler</choice>
  </choicegroup>
</multiplechoiceresponse>
```

To add shuffling to this problem, add `shuffle="true"` to the `<choicegroup>` element:

```
<p>What Apple device competed with the portable CD player?</p>
<multiplechoiceresponse>
  <choicegroup type="MultipleChoice" shuffle="true">
    <choice correct="false">The iPad</choice>
    <choice correct="false">Napster</choice>
    <choice correct="true">The iPod</choice>
    <choice correct="false">The vegetable peeler</choice>
  </choicegroup>
</multiplechoiceresponse>
```

To fix an answer's location in the list, add `fixed="true"` to the `choice` element for the answer:

```
<p>What Apple device competed with the portable CD player?</p>
<multiplechoiceresponse>
  <choicegroup type="MultipleChoice" shuffle="true">
    <choice correct="false">The iPad</choice>
    <choice correct="false">Napster</choice>
    <choice correct="true">The iPod</choice>
    <choice correct="false">The vegetable peeler</choice>
    <choice correct="false" fixed="true">All of the above</choice>
  </choicegroup>
</multiplechoiceresponse>
```

Targeted Feedback in a Multiple Choice Problem

You can configure a multiple choice problem so that explanations for incorrect answers are automatically shown to students. You can use these explanations to guide students towards the right answer. Therefore, targeted feedback is most useful for multiple choice problems for which students are allowed multiple attempts.

Use the Advanced Editor to Configure Targeted Feedback

You configure the problem to provide targeted feedback through XML in *The Advanced Editor*.

Follow these XML guidelines:

- Add a `targeted-feedback` attribute to the `<multiplechoiceresponse>` element, with no value:
`<multiplechoiceresponse targeted-feedback="">`
- Add a `<targetedfeedbackset>` element before the `<solution>` element.
- Within `<targetedfeedbackset>`, add one or more `<targetedfeedback>` elements.
- Within each `<targetedfeedback>` element, enter your explanation for the incorrect answer in HTML as markup described below.
- Connect the `<targetedfeedback>` element with a specific incorrect answer by using the same `explanation-id` attribute value for each.
- Use the `<solution>` element for the correct answer, with the same `explanation-id` attribute value as the correct `<choice>` element.

For example, the XML for the multiple choice problem is:

```
<p>What Apple device competed with the portable CD player?</p>
<multiplechoiceresponse targeted-feedback="">
  <choicegroup type="MultipleChoice">
    <choice correct="false" explanation-id="feedback1">The iPad</choice>
    <choice correct="false" explanation-id="feedback2">Napster</choice>
    <choice correct="true" explanation-id="correct">The iPod</choice>
    <choice correct="false" explanation-id="feedback3">The vegetable peeler</choice>
  </choicegroup>
</multiplechoiceresponse>
```

This is followed by XML that defines the targeted feedback:

```
<targetedfeedbackset>
  <targetedfeedback explanation-id="feedback1">
    <div class="detailed-targeted-feedback">
      <p>Targeted Feedback</p>
      <p>The iPad came out later and did not directly compete with portable CD players.</p>
    </div>
  </targetedfeedback>
  <targetedfeedback explanation-id="feedback2">
    <div class="detailed-targeted-feedback">
      <p>Targeted Feedback</p>
      <p>Napster was not an Apple product.</p>
    </div>
  </targetedfeedback>
  <targetedfeedback explanation-id="feedback3">
    <div class="detailed-targeted-feedback">
      <p>Targeted Feedback</p>
      <p>Vegetable peelers don't play music.</p>
    </div>
  </targetedfeedback>
</targetedfeedbackset>
```

```

</targetedfeedback>
</targetedfeedbackset>

<solution explanation-id="correct">
  <div class="detailed-solution">
    <p>The iPod directly competed with portable CD players.</p>
  </div>
</solution>

```

Answer Pools in a Multiple Choice Problem

You can configure a multiple choice problem so that a random subset of choices are shown to each student. For example, you can add 10 possible choices to the problem, and each student views a set of five choices.

The answer pool must have at least one correct answer, and can have more than one. In each set of choices shown to a student, one correct answer is included. For example, you may configure two correct answers in the set of 10. One of the two correct answers is included in each set a student views.

Use the Advanced Editor to Configure Answer Pools

You configure the problem to provide answer pools through XML in *The Advanced Editor*.

Follow these XML guidelines:

- In the `<choicegroup>` element, add the `answer-pool` attribute, with the numerical value indicating the number of possible answers in the set. For example, `<choicegroup answer-pool="4">`.
- For each correct answer, to the `<choice>` element, add an `explanation-id` attribute and value that maps to a solution. For example, `<choice correct="true" explanation-id="iPod">The iPod</choice>`.
- For each `<solution>` element, add an `explanation-id` attribute and value that maps back to a correct answer. For example, `<solution explanation-id="iPod">`.

Note: If the choices include only one correct answer, you do not have to use the `explanation-id` in either the `choice` or `<solution>` element. You do still use the `<solutionset>` element to wrap the `<solution>` element.

For example, for the following multiple choice problem, a student will see four choices, and in each set one of the choices will be one of the two correct ones. The explanation shown for the correct answer is the one with the same explanation ID.

```

<problem>
  <p>What Apple devices let you carry your digital music library in your pocket?</p>
  <multiplechoiceresponse>
    <choicegroup type="MultipleChoice" answer-pool="4">
      <choice correct="false">The iPad</choice>
      <choice correct="false">Napster</choice>
      <choice correct="true" explanation-id="iPod">The iPod</choice>
      <choice correct="false">The vegetable peeler</choice>
      <choice correct="false">The iMac</choice>
      <choice correct="true" explanation-id="iPhone">The iPhone</choice>
    </choicegroup>
  </multiplechoiceresponse>

  <solutionset>

```

```
<solution explanation-id="iPod">
<div class="detailed-solution">
    <p>Explanation</p>
    <p>Yes, the iPod is Apple's portable digital music player.</p>
</div>
</solution>
<solution explanation-id="iPhone">
<div class="detailed-solution">
    <p>Explanation</p>
    <p>In addition to being a cell phone, the iPhone can store and play your digital music.</p>
</div>
</solution>
</solutionset>
</problem>
```

6.19.3 Multiple Choice Problem XML

Template

```
<problem>
<p>Question text</p>
<multiplechoiceresponse>
    <choicegroup type="MultipleChoice" label="label text">
        <choice correct="false" name="a">Incorrect choice</choice>
        <choice correct="true" name="b">Correct choice</choice>
    </choicegroup>
</multiplechoiceresponse>

<solution>
    <div class="detailed-solution">
        <p>Explanation or solution header</p>
        <p>Explanation or solution text</p>
    </div>
</solution>
</problem>
```

Tags

- `<multiplechoiceresponse>` (required): Indicates that the problem is a multiple choice problem.
- `<choicegroup>` (required): Indicates the beginning of the list of options.
- `<choice>` (required): Lists an answer option.

Tag: `<multiplechoiceresponse>`

Indicates that the problem is a multiple choice problem.

Attributes

(none)

Children

- `<choicegroup>`
- All standard HTML tags (can be used to format text)

Tag: <choicegroup>

Indicates the beginning of the list of options.

Attributes

Attribute	Description
label (required)	Specifies the name of the response field.
type (required)	Must be set to “MultipleChoice”.

Children

- <choice>

Tag: <choice>

Lists an answer option.

Attributes

Attribute	Description
correct (at least one required)	Indicates a correct or incorrect answer. When the attribute is set to “true”, the choice is a correct answer. When the attribute is set to “false”, the choice is an incorrect answer. Only one choice can be a correct answer.
name	A unique name that the back end uses to refer to the choice.

Children

(none)

6.20 Multiple Choice and Numerical Input Problem

You can create a problem that combines a multiple choice and numerical input problems. Students not only select a response from options that you provide, but also provide more specific information, if necessary.

THE VALUE OF PI (1/1 points)

The numerical value of pi, rounded to two decimal points, is 3.24.

True.
 False. The correct value is .

Note: Currently, students can only enter numerals in the text field. Students cannot enter words or mathematical expressions.

6.20.1 Create a Multiple Choice and Numerical Input Problem

To create a multiple choice and numerical input problem:

1. In the unit where you want to create the problem, click **Problem** under **Add New Component**, and then click the **Advanced** tab.
2. Click **Blank Advanced Problem**.

3. In the component that appears, click **Edit**.
4. In the component editor, paste the code from below.
5. Replace the example problem and response options with your own text.
6. Click **Save**.

6.20.2 Multiple Choice and Numerical Input Problem Code

```
<problem>
The numerical value of pi, rounded to two decimal points, is 3.24.
<choicetextresponse>
<radiotextgroup>
<choice correct="false">True.</choice>
<choice correct="true">False. The correct value is <numtolerance_input answer="3.14"/>.</choice>
</radiotextgroup>
</choicetextresponse>
</problem>
```

6.21 Numerical Input

Numerical input problems are the simpler of the two mathematics tools that Studio offers. In these problems, students enter numbers or specific and relatively simple mathematical expressions to answer a question. The text that the students enter is converted to a symbolic expression that appears below the response field.

NUMERICAL RESPONSE

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

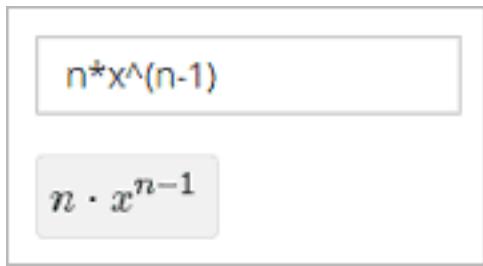
The decimal numerical system is base ten.

The standard gravity constant is defined to be precisely 9.80665 m/s^2 . 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.

Note that students' responses don't have to be exact for these problems. You can specify a margin of error, or tolerance. You can also specify a correct answer explicitly, or use a Python script. For more information, see the instructions below.

Responses for numerical input problems can include integers, fractions, and constants such as π 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, the text that the student enters is converted into mathematical symbols. The following example shows the way the system renders students' text responses in numerical input problems.



For more information about characters that students can enter, see *Math Response Formatting for Students*.

6.21.1 Create a Numerical Input Problem

You can create numerical problems in the Simple Editor or in the Advanced Editor regardless of the answer to the problem. If the text of your problem doesn't include any italics, bold formatting, or special characters, you can create the problem in the Simple Editor. If the text of your problem contains special formatting or characters, or if your problem contains a Python script, you'll use the Advanced Editor.

For example, the following example problems require the Advanced Editor.

What is the value of the standard gravity constant g , measured in m/s²? 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.



$$\sqrt{(\pi - 0)^2 + (0 - e)^2}$$

Answer: 4.15435440231

EXPLANATION

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.

You have used 2 of 5 submissions

For more information about including a Python script in your problem, see *Write-Your-Own-Grader Problem*.

Simple Editor

1. Under **Add New Component**, click **Problem**.
2. In the **Select Problem Component Type** screen, click **Numerical Input** on the **Common Problem Types** tab.
3. When the new Problem component appears, click **Edit**.
4. In the component editor, replace the sample problem text with your own text.

5. Determine the text of the problem to use as a label, and then surround that text with two sets of angle brackets (>><<).
6. Select the text of the answer, and then click the numerical input button.



When you do this, an equal sign appears next to the answer.

7. (Optional) Specify a margin of error, or tolerance. You can specify a percentage, number, or range.
 - To specify a percentage on either side of the correct answer, add **+NUMBER%** after the answer. For example, if you want to include a 2% tolerance, add **+2%**.
 - To specify a number on either side of the correct answer, add **+NUMBER** after the answer. For example, if you want to include a tolerance of 5, add **+5**.
 - To specify a range, use brackets [] or parentheses (). A bracket indicates that range includes the number next to it. A parenthesis indicates that the range does not include the number next to it. For example, if you specify **[5, 8]**, correct answers can be 5, 6, and 7, but not 8. Likewise, if you specify **(5, 8]**, correct answers can be 6, 7, and 8, but not 5.
8. In the component editor, select the text of the explanation, and then click the explanation button to add explanation tags around the text.



9. On the **Settings** tab, specify the settings that you want.
10. Click **Save**.

For the first example problem above, the text in the Problem component is the following.

```
>>What base is the decimal numeral system in?<<  
= 10  
  
[explanation]  
The decimal numeral system is base ten.  
[explanation]
```

Advanced Editor

To create this problem in the Advanced Editor, click the **Advanced** tab in the Problem component editor, and then replace the existing code with the following code.

Problem Code:

```
<problem>  
  <p><b>Example Problem</b></p>  
  
  <p>What base is the decimal numeral system in?  
    <numericalresponse answer="10">  
      <formulaequationinput label="What base is the decimal numeral system in?" />  
    </numericalresponse>  
  </p>  
  
  <p>What is the value of the standard gravity constant <i>g</i>, measured in m/s2? Give y
```

```

<numericalresponse answer="9.80665">
    <responseparam type="tolerance" default="0.01" />
    <formulaequestioninput label="Give your answer to at least two decimal places"/>
</numericalresponse>
</p>


<script type="loncapa/python">
computed_response = math.sqrt(math.fsum([math.pow(math.pi,2), math.pow(math.e,2)]))
</script>

<p>What is the distance in the plane between the points (pi, 0) and (0, e)? You can type math.<br/>
<numericalresponse answer="$computed_response">
    <responseparam type="tolerance" default="0.0001" />
    <formulaequestioninput label="What is the distance in the plane between the points (pi, 0) and (0, e)? You can type math." />
</numericalresponse>
</p>
<solution>
    <div class="detailed-solution">
        <p>Explanation</p>
        <p>The decimal numerical system is base ten.</p>
        <p>The standard gravity constant is defined to be precisely 9.80665 m/s2.</p>
        This is 9.80 to two decimal places. Entering 9.8 also works.</p>
        <p>By the distance formula, the distance between two points in the plane is<br/>
            the square root of the sum of the squares of the differences of each coordinate.<br/>
            Even though an exact numerical value is checked in this case, the easiest way to enter this answer is to type<br/>
            <code>sqrt(pi^2+e^2)</code> into the editor.<br/>
            Other answers like <code>sqrt((pi-0)^2+(0-e)^2)</code> also work.</p>
    </div>
</solution>
</problem>

```

6.21.2 Numerical Input Problem XML

Templates

The following templates represent problems with and without a decimal or percentage tolerance.

Problem with no tolerance

```

<p>TEXT OF PROBLEM
    <numericalresponse answer="ANSWER (NUMBER)">
        <formulaequestioninput label="TEXT OF PROBLEM"/>
    </numericalresponse>
</p>

<solution>
    <div class="detailed-solution">
        <p>TEXT OF SOLUTION</p>
    </div>
</solution>
</problem>

```

Problem with a decimal tolerance

```
<problem>

    <p>TEXT OF PROBLEM
    <numericalresponse answer="ANSWER (NUMBER)">
        <responseparam type="tolerance" default="NUMBER (DECIMAL, e.g., .02)" />
        <formulaequationinput label="TEXT OF PROBLEM"/>
    </numericalresponse>
</p>

    <solution>
        <div class="detailed-solution">
            <p>TEXT OF SOLUTION</p>
        </div>
    </solution>
</problem>
```

Problem with a percentage tolerance

```
<problem>

    <p>TEXT OF PROBLEM
    <numericalresponse answer="ANSWER (NUMBER)">
        <responseparam type="tolerance" default="NUMBER (PERCENTAGE, e.g., 3%)" />
        <formulaequationinput label="TEXT OF PROBLEM"/>
    </numericalresponse>
</p>

    <solution>
        <div class="detailed-solution">
            <p>TEXT OF SOLUTION</p>
        </div>
    </solution>
</problem>
```

Answer created with a script

```
<problem>

    <!-- The following uses Python script spacing. Make sure it isn't indented when you add it to the Problem XML -->
    <script type="loncapa/python">
computed_response = math.sqrt(math.fsum([math.pow(math.pi,2), math.pow(math.e,2)]))
    </script>

    <p>TEXT OF PROBLEM
    <numericalresponse answer="$computed_response">
        <responseparam type="tolerance" default="0.0001" />
        <formulaequationinput label="TEXT OF PROBLEM"/>
    </numericalresponse>
</p>

    <solution>
        <div class="detailed-solution">
```

```
<p>TEXT OF SOLUTION</p>
</div>
</solution>
</problem>
```

Tags

- <numericalresponse> (required): Specifies that the problem is a numerical input problem.
- <formulaequationinput /> (required): Provides a response field where the student enters a response.
- <responseparam> (optional): Specifies a tolerance, or margin of error, for an answer.
- <script> (optional):

Note: Some older problems use the <textline math="1" /> tag instead of the <formulaequationinput /> tag. However, the <textline math="1" /> tag has been deprecated. All new problems should use the <formulaequationinput /> tag.

Tag: <numericalresponse>

Specifies that the problem is a numerical input problem. The <numericalresponse> tag is similar to the <formularesponse> tag, but the <numericalresponse> tag does not allow unspecified variables.

Attributes

Attribute	Description
answer (required)	The correct answer to the problem, given as a mathematical expression.

Note: If you include a variable name preceded with a dollar sign (\$) in the problem, you can include a script in the problem that computes the expression in terms of that variable.

The grader evaluates the answer that you provide and the student's response in the same way. The grader also automatically simplifies any numeric expressions that you or a student provides. Answers can include simple expressions such as "0.3" and "42", or more complex expressions such as "1/3" and "sin(pi/5)".

Children

- <responseparam>
- <formulaequationinput>

Tag: * <formulaequationinput>

Creates a response field in the LMS where students enter a response.

Attributes

Attribute	Description
label (required)	Specifies the name of the response field.
size (optional)	Defines the width, in characters, of the response field in the LMS.

Children

(none)

Tag: <responseparam>

Specifies a tolerance, or margin of error, for an answer.

Attributes

Attribute type (optional) default (optional)	Description “tolerance”: Defines a tolerance for a number A number or a percentage specifying a numerical or percent tolerance.
--	---

Children

(none)

Tag: <script>

Specifies a script that the grader uses to evaluate a student’s response. A problem behaves as if all of the code in all of the script tags were in a single script tag. Specifically, any variables that are used in multiple <script> tags share a namespace and can be overridden.

As with all Python, indentation matters, even though the code is embedded in XML.

Attributes

Attribute type (required)	Description Must be set to “loncapa/python”.
------------------------------	---

Children

(none)

6.22 Open Response Assessment Problems

6.22.1 Introduction to Open Response Assessments

Warning: Open response assessments 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.

Warning: Problems can occur when you move your course from edge.edx.org to edx.org, or vice versa, if your course has the same URL in both places. To work around this problem, make sure that the URLs are different by specifying a different university or course number when you create each course. For example, if your university is edX and your course number is edX101, you can specify the course number as **edx_101** (note the underscore) on Edge and **edX101** on edX.

Open response assessments allow instructors to assess student learning through questions that may not have definite answers. Tens of thousands of students can receive feedback on written responses of varying lengths as well as files, such as computer code or images, that the students upload. Open response assessment technologies include self assessment and peer assessment. With self assessments, students learn by comparing their answers to a rubric that you create. With peer assessments, students compare their peers’ answers to the rubric.

For more information, see the following:

- *A Few Notes about Open Response Assessments*
- *Components of an Open Response Assessment*
- *Open Response Assessment Types*
- *Effective Questions*
- *Rubrics*
- *Create an Open Response Assessment Problem*
- *Grade an Open Response Assessment Problem*

- *Access Scores and Feedback*

6.22.2 A Few Notes about Open Response Assessments

Open response assessment 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).

6.22.3 Components of an Open Response Assessment

An open response assessment has three elements:

- The assessment type or types—self, peer, or artificial intelligence (AI). The type of assessment and the order in which the assessments run appears 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.

CENSORSHIP IN THE LIBRARIES

Open Response

Assessments: Self Peer AI

[Hide Question](#)

"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 if I have the right to remove that book from the shelf -- that work I abhor -- 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

In the response field below, write a persuasive essay reflecting your views on censorship in libraries. Do you believe that certain materials, such as books, music, and movies, should be removed if they are judged offensive? Support your position with convincing arguments from your own experience, observations, and/or reading.

When you've entered your response, click **Submit**. (You can also click **Save** if you need to stop working on your response for a while and come back to it later.) After you click **Submit**, the rubric for this question will appear, and you'll give yourself a score based on the rubric.

Response

[Save](#) [Submit](#)

- 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.

CENSORSHIP IN THE LIBRARIES

Open Response

Assessments: Self Peer AI

[Hide Question](#)

"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 if I have the right to remove that book from the shelf -- that work I abhor -- 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

In the response field below, write a persuasive essay reflecting your views on censorship in libraries. Do you believe that certain materials, such as books, music, and movies, should be removed if they are judged offensive? Support your position with convincing arguments from your own experience, observations, and/or reading.

When you've entered your response, click **Submit**. (You can also click **Save** if you need to stop working on your response for a while and come back to it later.) After you click **Submit**, the rubric for this question will appear, and you'll give yourself a score based on the rubric.

Response

Save Submit

- 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.)

CENSORSHIP IN THE LIBRARIES

Open Response

Assessments: Self Peer AI

[Show Question](#)

[Response](#)

Books that are offensive should not be taken off the shelf all together, but should be put in a section made for adult readers only. This way no children would see the book anyway. If an offensive book is in a school or any other place that the readers are children for the majority than it should be ok to take that book off of the shelf. All together it should be ok to remove an offensive book from a shelf but only if the book is offensive to the wrong audience.

[Rubric](#)

Select the criteria you feel best represents this submission in each category.

Writing Applications

- 0 points : The essay loses focus, has little information or supporting details, and the organization makes it difficult to follow.
- 1 points : The essay presents a mostly unified theme, includes sufficient information to convey the theme, and is generally organized well.

Language Conventions

- 0 points : The essay demonstrates a reasonable command of proper spelling and grammar.
- 1 points : The essay demonstrates superior command of proper spelling and grammar.

[Submit assessment](#)

6.22.4 Open Response 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.

Warning: In peer assessments, the **due date** that you set for the subsection that contains the ORA problem is the date by which students must not only submit their own responses, but finish grading the required number of peer responses.

You can use one or more of these assessments in any problem. 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.

6.22.5 Effective Questions

When you write your question, 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. You may also want to provide information about how to use the LMS. 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.

6.22.6 Rubrics

The same rubric is used for all three ORA types, and it can include anything that you want it to include.

In Studio, rubrics are arranged by *categories*. Each category has two or more *options*, and each option has a point value.

Options must be listed in ascending order starting at 0 points. For example, in a category with three options, the first option is worth 0 points, the second is worth 1 point, and the third is worth 2 points. The person or algorithm that grades the problem selects one value for each category.

Different categories in the same problem can have different numbers of options.

6.22.7 Create an Open Response Assessment Problem

Warning: Open response assessments are still in beta. To enable open response assessments in your course, you'll need to work with your edX program manager.

Creating an open response assessment is a multi-step process.

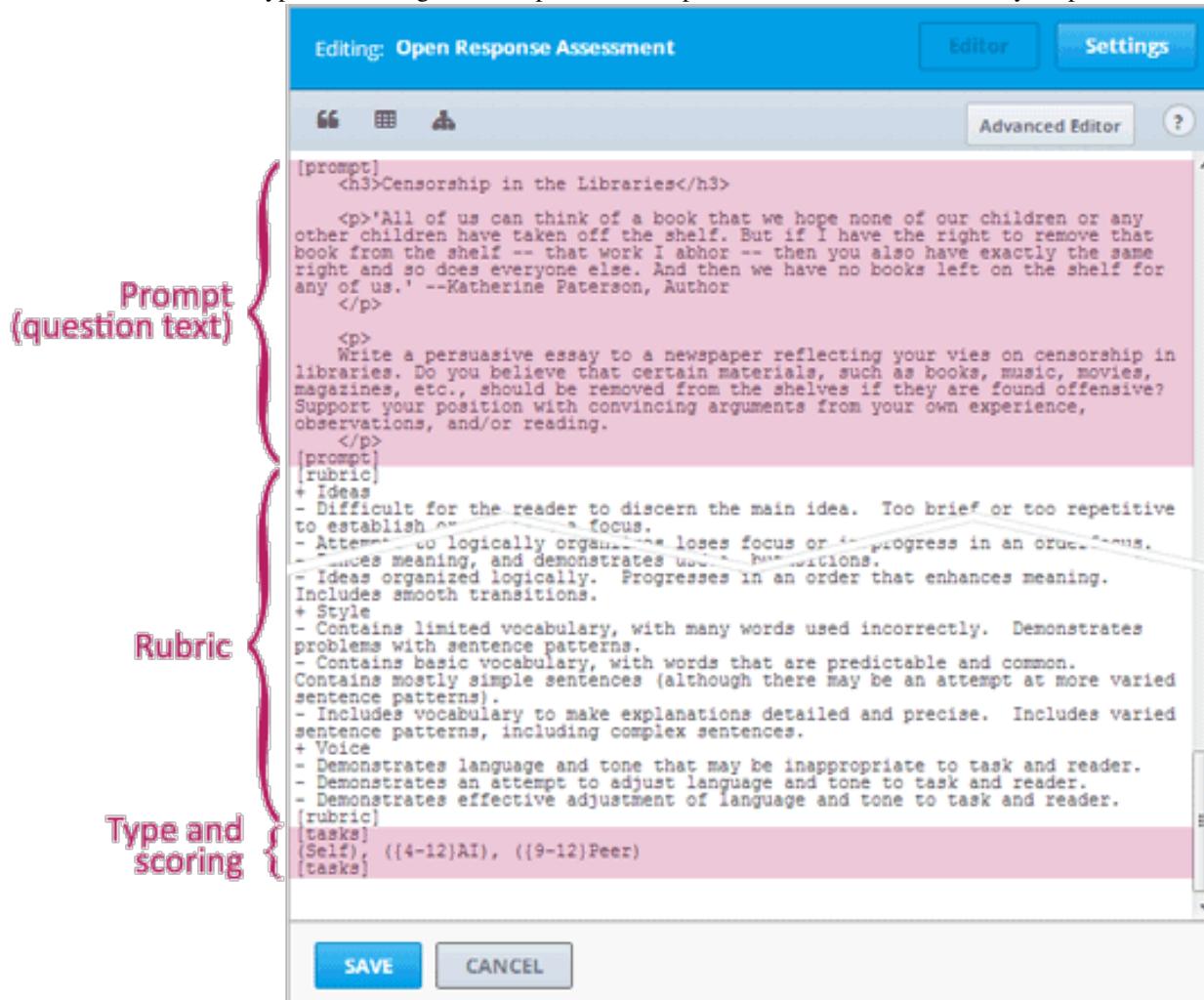
- *Step 1. Create the ORA Component*
- *Step 2. Add the Question*
- *Step 3. Add the Rubric*
- *Step 4. Set the Assessment Type and Scoring*
- *Step 5. Set the Problem Name*
- *Step 6. Set Other Options*
- *Step 7. Save the Problem*
- *Step 8. Add the Peer Grading Interface (for peer assessments only)*
- *Step 9. Test the Problem*

Each of these steps is described in detail below.

Warning: Problems can occur when you move your course from edge.edx.org to edx.org, or vice versa, if your course has the same URL in both places. To work around this problem, make sure that the URLs are different by specifying a different university or course number when you create each course. For example, if your university is edX and your course number is edX101, you can specify the course number as **edx_101** (note the underscore) on Edge and **edX101** on edX.

Step 1. Create the ORA Component

1. Modify your course's advanced settings to allow open response assessments. For more information about how to do this, contact your edX program manager.
2. In Studio, open the unit where you want to create the ORA.
3. Under **Add New Component**, click **Advanced**, and then click **Open Response Assessment**.
4. In the problem component that appears, click **Edit**, and then click **OK** in the dialog box that appears.
5. The component editor opens. The component editor contains a sample question ("prompt"), rubric, and the code for the assessment type and scoring. You'll replace this sample content with the content for your problem.



Step 2. Add the Question

1. In the component editor, locate the [prompt] tags.

The screenshot shows the 'Editing: Open Response Assessment' interface. At the top, there are 'Editor' and 'Settings' tabs. Below them are icons for text, table, and list, followed by an 'Advanced Editor' button and a help icon. The main area contains the following text:

```
[prompt]
<h3>Censorship in the Libraries</h3>
<p>'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 if I
have the right to remove that book from the shelf -- that work I
abhor -- 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
</p>
<p>
  Write a persuasive essay to a newspaper reflecting your vies 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.
</p>
[prompt]
[rubric]
+ Ideas
- Difficult for the reader to discern the main idea. Too brief or
  too repetitive to establish or maintain a focus.
```

At the bottom are 'SAVE' and 'CANCEL' buttons.

2. 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.

Step 3. Add the Rubric

Note: After you release your problem to students in your live course, make sure you don't make any changes to the rubric that affect scoring, such as adding or removing an option in a category. Changing the rubric can cause errors in live courses.*

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.)

The screenshot shows the 'Editor' tab selected in the top navigation bar. The main area contains XML code for a rubric, with some parts highlighted in purple. Below the code are two buttons: 'SAVE' and 'CANCEL'.

```
</p>
[prompt]
[rubric]
+ Ideas
- Difficult for the reader to discern the main idea. Too brief or too repetitive to establish or maintain a focus.
- Attempts a main idea. Sometimes loses focus or ineffectively displays focus.
- Presents a [REDACTED] theme or main idea, but may include minor tangents.
- vocabulary to [REDACTED] focused on topic.
- Detailed and precise.
- Includes varied sentence patterns, including complex sentences.
+ Voice
- Demonstrates language and tone that may be inappropriate to task and reader.
- Demonstrates an attempt to adjust language and tone to task and reader.
- Demonstrates effective adjustment of language and tone to task and reader.
[rubric]
[tasks]
(Self), ((4-12)AI), ((9-12)Peer)
[tasks]
```

- Replace the sample rubric with the text of your rubric. 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.

For example, your rubric might resemble the following rubric.

```
[rubric]
```

```
+ Writing Applications
- The essay loses focus, has little information or supporting details, and the organization makes it difficult to understand.
- The essay presents a mostly unified theme, includes sufficient information to convey the theme, and is well organized.

+ Language Conventions
- The essay demonstrates a reasonable command of proper spelling and grammar.
- The essay demonstrates superior command of proper spelling and grammar.
```

```
[rubric]
```

Step 4. Set the Assessment Type and Scoring

Note: After you release your problem to students in your live course, make sure you don't make any changes to the

code for the assessment type and scoring. Changing this code can cause errors in live courses.

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)
[task]
```

- 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)
[task]
```

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.

Note: If a response's score isn't high enough for the response to move to the next assessment,

Set the Assessment Type and Scoring

1. In the component editor, locate the [tasks] tags.

The screenshot shows the 'Editor' tab of the edX problem editor interface. The title bar says 'Editing: Censorship in the Libraries'. The main area contains the following code:

```
reading.  
  </p>  
[prompt]  
  
[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]  
  
[tasks]  
{Self}, ({4-12}AI), ({9-12}Peer)  
[tasks]
```

At the bottom are 'SAVE' and 'CANCEL' buttons.

1. Replace the sample code with the code for your problem.

Step 5. Set the Problem Name

Note: After you release your problem to students in your live course, make sure you don't change the name of the problem. Changing the display name when the problem is live can cause a loss of student data.

You can change the display name of a problem while you're still testing the problem. However, note that all the test responses and scores associated with the problem will be lost when you change the name. To update the problem name on the instructor dashboard, submit a new test response to the problem.

The name of the problem appears as a heading above the problem in the courseware. It also appears in the list of problems on the **Staff Grading** page.

Staff grading

INSTRUCTIONS

This is the list of problems that currently need to be graded in order to train AI grading and create calibration essays for peer grading. Each problem needs to be treated separately, and we have indicated the number of student submissions that need to be graded. You can grade more than the minimum required number of submissions--this will improve the accuracy of AI grading, though with diminishing returns. You can see the current accuracy of AI grading in the problem view.

PROBLEM LIST

Problem Name	Graded	Available to Grade	Required	Progress
Combined Open Ended Response	51	979	49	<div style="width: 10%;"><div style="width: 100%;"> </div></div>
Peer-Graded Essay	20	0	0	<div style="width: 0%;"><div style="width: 100%;"> </div></div>
Humanities Question -- Machine Assessed	119	0	0	<div style="width: 100%;"><div style="width: 100%;"> </div></div>

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 the name of your problem.

Step 6. 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, click the **Settings** tab, and then specify the options that you want.

Editing: Censorship in the Libraries

Editor Settings

Allow "overgrading" of peer submissions	False	Allow students to peer grade submissions that already have the requisite number of graders, but ONLY WHEN all submissions they are eligible to grade already have enough graders. This is intended for use when settings for 'Required Peer Grading' > 'Peer Graders per Response'
Allow File Uploads	False	Whether or not the student can submit files as a response.
Disable Quality Filter	False	If False, the Quality Filter is enabled and submissions with

SAVE **CANCEL**

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. Note Every problem must have a problem weight of at least one point. Problems that have a problem weight of zero points don’t appear on the instructor dashboard.
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
6.22. Open Response Assessment Problems	you can set the Allow “overgrading” of peer submissions setting to allow more students to grade previously graded responses.)

Step 7. 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.

Edit Delete

OPEN RESPONSE ASSESSMENT

Open Response

Assessments: Self AI Peer

[Hide](#) [Question](#)

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 if I have the right to remove that book from the shelf -- that work I abhor -- 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.

[Response](#)

Step 8. 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.

Warning: In peer assessments, the **due date** that you set for the subsection that contains the ORA problem is the date by which students must not only submit their own responses, but finish grading the required number of peer responses.

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 **Peer Grading** page opens.

The screenshot shows a web-based peer grading interface. On the left, there's a sidebar with a tree view showing 'Week 1' expanded, with 'Peer Grading Interface' selected. A sub-menu under it shows 'Peer Grading Interface' again. The main content area has a title 'Peer Grading'. Below it is a section titled 'INSTRUCTIONS' with the text 'Here are a list of problems that need to be peer graded for this course.' A yellow bar at the bottom displays the message 'Nothing to grade!'. Navigation arrows are visible at the top and bottom of the main content area.

When students submit responses for peer assessments in your course, the names of the problems appear in this interface.

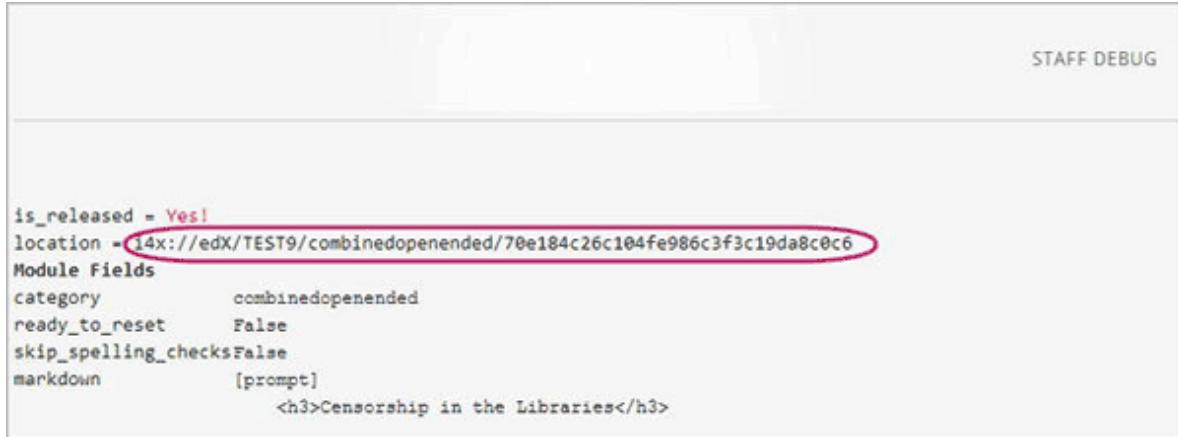
Add the Peer Grading Interface to an Individual Problem

When you add a peer grading interface for an individual problem, you must 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. 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 page 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**.



```
STAFF DEBUG

is_released = Yes!
location = i4x://edX/TEST9/combinedopenended/70e184c26c104fe986c3f3c19da8c0c6
Module Fields
category      combinedopenended
ready_to_reset  False
skip_spelling_checks False
markdown       [prompt]
                <h3>Censorship in the Libraries</h3>
```

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**.

The screenshot shows the "Peer Grading Interface" component editor. At the top, there are "Editor" and "Settings" tabs. Below the tabs, the configuration interface is displayed:

- Link to Problem Location:** A text input field containing "4x://MITx/Test" is highlighted with a red rounded rectangle.
- Problem Weight:** A numeric input field set to "1" with up and down arrows for adjustment.
- Show Single Problem:** A dropdown menu set to "True" is highlighted with a red rounded rectangle. To its right is a circular icon with a question mark.

Below the configuration area are two buttons: "Save" (blue) and "Cancel".

Help text for each setting is provided to the right of the input fields:

- "Link to Problem Location": "The location of the problem being graded. Only used when "Show Single Problem" is True."
- "Problem Weight": "Defines the number of points each problem is worth. If the value is not set, each problem is worth one point."
- "Show Single Problem": "When True, only the single problem specified by "Link to Problem Location" is shown. When False, a panel is displayed with all problems available for peer grading."

- Click **Save** to close the component editor.

Step 9. Test the Problem

Test your problem by adding and grading a response.

- In Studio, open the unit that contains your ORA problem.
- 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.
- In the LMS, locate your ORA question, and then type your response in the Response field under the question.

COMBINED OPEN ENDED RESPONSE

Open Response Assessments: Self AI Peer

[Hide Question](#)

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 if I have the right to remove that book from the shelf -- that work I abhor -- 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.

Response

[Save](#) [Submit](#)

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.

Staff Warning: Please note that if you submit a duplicate of text that has already been submitted for grading, it will not show up in the staff grading view. It will be given the same grade that the original received automatically, and will be returned within 30 minutes if the original is already graded, or when the original is graded if not.

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.

When you test your problem, you may want to submit test responses that contain little text, random characters, or other content that doesn't resemble the responses that you expect from your students. Open response assessments include a quality filter that prevents instructors and other students from seeing these "low-quality" responses. This quality filter is enabled by default. If you want to see all of your test responses, including the "low-quality" responses, disable the quality filter.

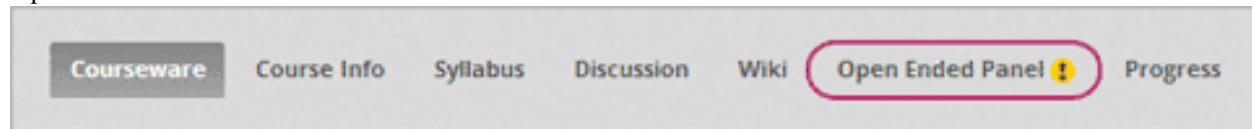
To disable the quality filter, open the problem component, click the **Settings** tab, and then set the **Disable Quality Filter** setting to **True**.

6.22.8 Grade an Open Response Assessment Problem

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.

The Staff Grading 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.

Open Ended Console

INSTRUCTIONS

Here are items that could potentially need your attention.

PEER GRADING

View all problems that require peer assessment in this particular course.

STAFF GRADING

New submissions to grade

View ungraded submissions submitted by students for the open ended problems in the course.

PROBLEMS YOU HAVE SUBMITTED

New grades have been returned

View open ended problems that you have previously submitted for grading.

FLAGGED SUBMISSIONS

Submissions have been flagged for review

View submissions that have been flagged by students as inappropriate.

When the **Staff Grading** page opens, information about your open response assessment appears in several columns.

Staff grading

INSTRUCTIONS

This is the list of problems that currently need to be graded in order to train AI grading and create calibration essays for peer grading. Each problem needs to be treated separately, and we have indicated the number of student submissions that need to be graded. You can grade more than the minimum required number of submissions—this will improve the accuracy of AI grading, though with diminishing returns. You can see the current accuracy of AI grading in the problem view.

PROBLEM LIST

Problem Name	Graded	Available to Grade	Required	Progress
Combined Open Ended Response	51	994	49	<div style="width: 10%;"><div style="width: 100px; height: 10px; background-color: #ccc;"></div></div>
Peer-Graded Essay	20	0	0	<div style="width: 10%;"><div style="width: 100px; height: 10px; background-color: #ccc;"></div></div>
Humanities Question -- Machine Assessed	119	0	0	<div style="width: 10%;"><div style="width: 100px; height: 10px; background-color: #ccc;"></div></div>

Problem Name	The name of the problem. Click the name of the problem to open it. Problems in your course do 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. 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.
Available to grade	The total number of ungraded student submissions.
Required	The number of responses remaining to be graded to train the algorithm for AI or to calibrate the responses for peer grading. 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.
Progress	A visual indication of your progress through the grading process.

Grade Responses

Warning: In peer assessments, the **due date** that you set for the subsection that contains the ORA problem is the date by which students must not only submit their own responses, but finish grading the required number of peer responses.

1. Go to the **Staff Grading** page.
2. Under **Problem Name**, click the name of the problem that you want.

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.

Staff grading

[« Back to problem list](#)

COMBINED OPEN ENDED RESPONSE

1 available | 0 graded | 100 more needed to start MI.
Machine learning error information: No model exists yet for this problem.

Please note that when you see a submission here, it has been temporarily removed from the grading pool. The submission will return to the grading pool after 30 minutes without any grade being submitted. Hitting the back button will result in a 30 minute wait to be able to grade this submission again.

Prompt ([Show](#))

Student Response

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 is because younger children should not be reading, listening, or watching material that is for adults or teenagers. They may like these type of things can be monitored is 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 has 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. Even though they are almost adults. Some parents may agree with some of the material. So, if their parents does 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.

Rubric

Select the criteria you feel best represents this submission in each category.

Writing Applications

- 0 points : The essay loses focus, has little information or supporting details, and the organization makes it difficult to follow.
- 1 points : The essay presents a mostly unified theme, includes sufficient information to convey the theme, and is generally organized well.

Language Conventions

- 0 points : The essay demonstrates a reasonable command of proper spelling and grammar.
- 1 points : The essay demonstrates superior command of proper spelling and grammar.

Written Feedback

Feedback for student (optional)

Flag as inappropriate content for later review

[Skip](#)

3. In the rubric below the response, select the option that best describes the response.

4. 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, if 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.

Written Feedback

Feedback for student (optional)

Flag as inappropriate content for later review

[Submit](#)

[Skip](#)

5. 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.

The screenshot shows the 'Staff grading' interface. At the top, there's a yellow bar with the text 'Grades saved. Fetching the next submission to grade.' Below it, a grey box displays 'COMBINED OPEN ENDED RESPONSE' with a note about machine learning error information and submission status. A 'Prompt (Show)' button is present. A large text area for 'Student Response' contains a paragraph about libraries and censorship. Under 'Rubric', there are sections for 'Writing Applications' and 'Language Conventions', each with two radio button options. A 'Written Feedback' section includes a text input field for 'Feedback for student (optional)' and a 'Flag as inappropriate content for later review' link. A 'Skip' button is also visible.

After you've graded all responses for this problem, **No more submissions to grade** appears on the page.

The screenshot shows the 'Staff grading' interface again, but this time the yellow bar at the top says 'No more submissions to grade.' A grey button labeled 'Re-check for submissions' is visible below the bar.

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.

Note: After you've graded enough responses for AI assessments to start, the number of responses in the **Available to grade** column decreases rapidly as the algorithm grades responses and returns them to your students in just a few seconds. No student data is lost.

Note: When a response opens for you to grade, it leaves the current “grading pool” that other instructors or students are grading from, which prevents other instructors or students from grading the response while you are working on it. If you do not submit a score for this response within 30 minutes, the response returns to the grading pool (so that it again becomes available for others to grade), even if you still have the response open on your screen.

If the response returns to the grading pool (because the 30 minutes have passed), but the response is still open on your screen, you can still submit feedback for that response. If another instructor or student grades the response after it returns to the grading pool but before you submit your feedback, the response receives two grades.

If you click your browser’s **Back** button to return to the problem list before you click **Submit** to submit your feedback for a response, the response stays outside the grading pool until 30 minutes have passed. When the response returns to the grading pool, you can grade it.

6.22.9 Access Scores and Feedback

Scoring

Scores for open response assessment problems that have more than one assessment type are not cumulative. That is, if a problem has a value of 10 points and it contains both a self assessment and an AI assessment, the total score is out of 10 points rather than 20.

The final score for an open response assessment is the score that the response receives in the last assessment that it undergoes. If a response doesn’t score high enough to move to another assessment, the final score is the score that the response receives in the last assessment that it qualifies for.

In peer assessment, the final score is the median of the scores that the response receives from each peer grader.

For example:

A 20-point problem includes self assessment, peer assessment, and AI assessment. To move from self assessment to peer assessment, the response must score 10 points. To move from peer assessment to AI assessment, the response must score 12 points.

- Student A gives himself 9 points in the self assessment. Because the minimum score to move from self assessment to peer assessment is 10, the response cannot continue to peer assessment. Student A’s final score is 9 out of 20.
- Student B gives herself 16 points in the self assessment. In the peer assessment, the response receives scores of 14, 11, and 10. Because the median is 11, the response cannot continue to AI assessment. Student B’s final score is 11 out of 20.
- Student C gives herself 17 points in the self assessment. In the peer assessment, the response receives scores of 10, 18, and 14. Because the median is 14, the response moves to AI assessment. AI assessment gives the response a score of 16. Student C’s final score is 16 out of 20.

Note that if the same 20-point problem changes the assessment order, the student’s final score may be different. For example, suppose the assessment order for the above example changes to self assessment, then AI assessment, and then peer assessment. To move from self assessment to AI assessment, the response must score 10 points. To move from AI assessment to peer assessment, the response must score 12 points.

- Student D gives himself 17 points in the self assessment. AI assessment gives the response a score of 16. In the peer assessment, the response receives scores of 10, 18, and 14; the median is 14. Student D’s final score is 14 out of 20.

Note: Because the assessment order can affect a student’s final score, we recommend that you specify peer assessments as the last assessments for ORA problems. If the last assessment is a self assessment, the student can award herself full points for a response, even if the response received low grades from the AI and peer assessments.

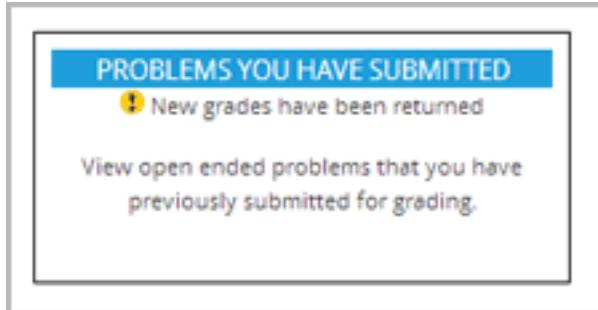
Accessing Scores

You access your scores for your responses to AI and peer assessment problems through the **Open Ended Console** page.

1. From any page in the LMS, 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

AI GRADED ESSAY

Open Response Assessments: AI

Show Question Response

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 is because younger children should not be reading, listening, or watching material that is for adults or teenagers. They way these type of things can be monitored is 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 want a parent has 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. Even though they are almost adults. Some parents not agree with some of the material. So, if their parents does not want them reading certain things their should be stickers on their cards. Even though there are plenty of opinions about what a persons child is reading, watching, or listening to I think there should be certain ways to watch what children are reading, watching or listening to.

New Submission

Submitted Rubric

Toggle Full Rubric Scored rubric from grader 1

Writing Applications

✓ 1 points : The essay presents a mostly unified theme, includes sufficient information to convey the theme, and is generally organized well.

Language Conventions

✓ 1 points : The essay demonstrates superior command of proper spelling and grammar.

Graded Peer Assessment

Open Response

Assessments: Peer

[Show Question](#)

[Response](#)

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 is because younger children should not be reading, listening, or watching material that is for adults or teenagers. They way these type of things can be monitored is 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 want a parent has 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. Even though they are almost adults. Some parents not agree with some of the material. So, if their parents does not want them reading certain things their should be stickers on their cards. Even though there are plenty of opinions about what a persons child is reading, watching, or listening to I think there should be certain ways to watch what children are reading, watching or listening to.

New Submission

Submitted Rubric

[Toggle Full Rubric](#)

[Scored rubric from grader 2](#)

Writing Applications

✓ 1 points : The essay presents a mostly unified theme, includes sufficient information to convey the theme, and is generally organized well.

Language Conventions

✓ 1 points : The essay demonstrates superior command of proper spelling and grammar.

You answer the question and gave support.

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.

Feedback not available yet

You need to peer grade 1 more submissions in order to see your feedback.

You have graded responses from 2 students, and 3 students have graded your submissions.

You have made 1 submissions.

6.23 Periodic Table Tool

You can create an interactive periodic table of the elements to help your students learn about various elements' properties. In the table below, detailed information about each element appears as the student moves the mouse over the element.

AN INTERACTIVE REFERENCE TABLE																																	
This interactive periodic table of the elements was first used in 3.091x Introduction to Solid State Chemistry.																																	
The table reveals detailed information about each element as you mouse over it.																																	
1 H Hydrogen 1.00794 g/mol	Atomic Weight 1.00794 g/mol	Atomic Volume 14.1 cm ³ /mol	Ionic Radius - pm	Density 0.089 g/cm ³	Melting Point -259.34 °C	Boiling Point -252.87 °C	Polarizability 0.667	Electronegativity 2.2	First Ionization Potential 13.598 eV	Crystal Structure Hex	Oxidation States 1	Electronic Configuration 1s ¹	Enthalpy of Fusion 0.0586 kJ/mol	Enthalpy of Vaporization 0.449 kJ/mol	Thermal Conductivity 0.001815 W·cm ⁻¹ ·K ⁻¹	Specific Heat Capacity 14.304 J g ⁻¹ ·K ⁻¹	Enthalpy of Atomization 217.57 kJ/mol	He 2 4															
Li 6.94	Be 9.01	B 10.81	C 12.01	N 14.01	O 16	F 19	Ne 20.18	Na 22.99	Mg 24.30	Al 26.98	Si 28.09	P 30.97	S 32.07	Cl 35.45	Ar 39.95	K 39.10	Ca 40.08	Sc 44.96	Ti 47.88	V 50.94	Cr 52	Mn 54.94	Fe 55.85	Co 58.93	Ni 58.69	Cu 63.55	Zn 65.39	Ga 69.72	Ge 72.61	As 74.92	Se 78.96	Br 79.90	Kr 83.80
Rb 85.47	Sr 87.62	Y 88.91	Zr 91.22	Nb 92.91	Mo 95.94	Tc 97.91	Ru 101.07	Rh 102.91	Pd 106.42	Ag 107.87	Cd 112.41	In 114.82	Sn 118.71	Sb 121.76	Te 127.60	I 126.90	Xe 131.29																
Cs 132.91	Ba 137.33	La 138.91	Hf 178.49	Ta 180.05	W 183.84	Re 186.21	Os 190.23	Ir 192.22	Pt 195.08	Au 196.97	Hg 200.59	Tl 204.38	Pb 207.20	Bi 208.98	Po 208.98	At 209.99	Rn 222.02																
Fr 223.02	Ra 226.03	Ac 227.03	Rf 261	Db 262	Sg 263	Bh 262	Hs 265	Mt 266	Uun 269	Uuu 272	Uub 277	Uut -	Uuq 285	Uup -	Uuh 289	Uus -	Uuo 293																
Ce 140.12	Pr 140.91	Nd 144.24	Pm 144.91	Sm 150.36	Eu 151.97	Gd 157.25	Tb 158.93	Dy 162.50	Ho 164.93	Er 167.26	Tm 168.93	Yb 173.04	Lu 174.97	Th 232.04	Pa 231.04	U 238.03	Np 237.05	Pu 244.06	Am 243.06	Cm 247.07	Bk 247.07	Cf 251.08	Es 252.08	Fm 257.10	Md 258.10	No 259.10	Lr 262.11						

6.23.1 Create the Periodic Table Tool

To create a periodic table, you need the following files:

- Periodic-Table.js
- Periodic-Table.css
- Periodic-Table-Colors.css
- PeriodicTableHTML.txt

To download all of these files in a .zip archive, click <http://files.edx.org/PeriodicTableFiles.zip>.

To create the periodic table, you need an HTML component.

1. Upload all of the files listed above *except PeriodicTable.txt* to the **Files & Uploads** page in your course.

2. In the unit where you want to create the problem, click **HTML** under **Add New Component**, and then click **HTML**.
3. In the component that appears, click **Edit**.
4. In the component editor, switch to the **HTML** tab.
5. Open the PeriodicTable.txt file in any text editor.
6. Copy all of the text in the PeriodicTable.txt file, and paste it into the HTML component editor. (Note that the PeriodicTableHTML.txt file contains over 6000 lines of code. Paste all of this code into the component editor.)
7. Click **Save**.

6.24 Poll Tool

You can run polls in your course so that your students can share opinions on different questions.

POLL QUESTION

All things being equal, should the government use public funds to invest in museums rather than sports arenas?

<input checked="" type="checkbox"/>	Yes	<div style="width: 63.6%; background-color: #555; height: 15px;"></div>	5780 (63.6%)
<input type="checkbox"/>	No	<div style="width: 36.4%; background-color: #ccc; height: 15px;"></div>	3308 (36.4%)

You responded "Yes" to the poll question. But what would you say to the following: What if public officials expect that more people will make use of and enjoy a sports arena? Should the government still use public funds to invest in museums instead?

Post your answer in the discussion section below, or view the discussion of those who responded "No" to the poll question [here](#).

Note: Creating a poll requires you to export your course, edit some of your course's XML files in a text editor, and then re-import your course. We recommend that you create a backup copy of your course before you create the poll. We also recommend that you only edit the files that will contain polls in the text editor if you're very familiar with editing XML.

6.24.1 Terminology

Sections, subsections, units, and components have different names in the **Course Outline** view and in the list of files that you'll see after you export your course and open the .xml files for editing. The following table lists the names of these elements in the **Course Outline** view and in a list of files.

Course Outline View	File List
Section	Chapter
Subsection	Sequential
Unit	Vertical
Component	Discussion, HTML, problem, or video

For example, when you want to find a specific section in your course, you'll look in the **Chapter** folder when you open the list of files that your course contains. To find a unit, you'll look in the **Vertical** folder.

6.24.2 Create a Poll

1. In the unit where you want to create the poll, create components that contain all the content that you want *except* for the poll. Make a note of the 32-digit unit ID that appears in the **Unit Identifier** field under **Unit Location**.
2. Export your course. For information about how to do this, see *Exporting and Importing a Course*. Save the .tar.gz file that contains your course in a memorable location so that you can find it easily.
3. Locate the .tar.gz file that contains your course, and then unpack the .tar.gz file so that you can see its contents in a list of folders and files.
 - To do this on a Windows computer, you'll need to download a third-party program. For more information, see [How to Unpack a tar File in Windows](#), [How to Extract a Gz File](#), [The gzip Home Page](#), or the Windows section of the [How to Open .tar.gz Files](#) page.
 - For information about how to do this on a Mac, see the Mac OS X section of the [How to Open .tar.gz Files](#) page.
4. In the list of folders and files, open the **Vertical** folder.

Note: If your unit is not published, open the **Drafts** folder, and then open the **Vertical** folder in the **Drafts** folder.

5. In the **Vertical** folder, locate the .xml file that has the same name as the unit ID that you noted in step 1, and then open the file in a text editor such as Sublime 2. For example, if the unit ID is e461de7fe2b84ebeabe1a97683360d31, you'll open the e461de7fe2b84ebeabe1a97683360d31.xml file.

The file contains a list of all the components in the unit, together with the URL names of the components. For example, the following file contains an HTML component followed by a Discussion component.

```
<vertical display_name="Test Unit">
  <html url_name="b59c54e2f6fc4cf69ba3a43c49097d0b"/>
  <discussion url_name="8320c3d511484f3b96bdedfd4a44ac8b"/>
</vertical>
```

6. Add the following poll code in the location where you want the poll. Change the text of the prompt to the text that you want.

```
<poll_question display_name="Poll Question">
  <p>Text of the prompt</p>
  <answer id="yes">Yes</answer>
  <answer id="no">No</answer>
</poll_question>
```

In the example above, if you wanted your poll to appear between the HTML component and the Discussion component in the unit, your code would resemble the following.

```
<vertical display_name="Test Unit">
  <html url_name="b59c54e2f6fc4cf69ba3a43c49097d0b"/>
  <poll_question display_name="Poll Question">
```

```

<p>Text of the prompt</p>
<answer id="yes">Yes</answer>
<answer id="no">No</answer>
</poll_question>
<discussion url_name="8320c3d511484f3b96bdedfd4a44ac8b"/>
</vertical>

```

7. After you add the poll code, save and close the .xml file.
8. Re-package your course as a .tar.gz file.
 - For information about how to do this on a Mac, see [How to Create a Tar GZip File from the Command Line](#).
 - For information about how to do this on a Windows computer, see [How to Make a .tar.gz on Windows](#).
9. In Studio, re-import your course. You can now review the poll question and answers that you added in Studio.

Note:

- Although polls render correctly in Studio, you cannot edit them in Studio. You will need to follow the export/import process outlined above to make any edits to your polls.
- A .csv file that contains student responses to the problem is not currently available for polls. However, you can obtain the aggregate data directly in the problem.

6.24.3 Format description

The main tag of Poll module input is:

```
<poll_question> ... </poll_question>
```

`poll_question` can include any number of the following tags: any xml and `answer` tag. All inner xml, except for `answer` tags, we call “question”.

poll_question tag

Xmodule for creating poll functionality - voting system. The following attributes can be specified for this tag:

`name` - Name of xmodule.
`[display_name| AUTOGENERATE]` - Display name of xmodule. When this attribute is not defined - display
`[reset | False]` - Can reset/revote many time (value = True/False)

answer tag

Define one of the possible answer for poll module. The following attributes can be specified for this tag:

`id` - unique identifier (using to identify the different answers)

Inner text - Display text for answer choice.

6.24.4 Example

Example of poll

```
<poll_question name="second_question" display_name="Second question">
    <h3>Age</h3>
    <p>How old are you?</p>
    <answer id="less18">&lt; 18</answer>
    <answer id="10_25">from 10 to 25</answer>
    <answer id="more25">&gt; 25</answer>
</poll_question>
```

Example of poll with unable reset functionality

```
<poll_question name="first_question_with_reset" display_name="First question with reset"
    reset="True">
    <h3>Your gender</h3>
    <p>You are man or woman?</p>
    <answer id="man">Man</answer>
    <answer id="woman">Woman</answer>
</poll_question>
```

6.25 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 problems can be text input problems.

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).

10

×

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?

Reset Show Answer(s)

6.25.1 Create a Problem with an Adaptive Hint

To create the above problem:

1. In the unit where you want to create the problem, 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 the code below.
5. Click **Save**.

```

<problem>
    <text>
        <script type="text/python" system_path="python_lib">
def test_str(expect, ans):
    print expect, ans
    ans = ans.strip("''")
    ans = ans.strip('\'')
    return expect == ans.lower()

def hint_fn(answer_ids, student_answers, new_cmap, old_cmap):
    aid = answer_ids[0]
    ans = str(student_answers[aid]).lower()
    print 'hint_fn called, ans=', ans
    hint = ''
    if '10' in ans:
        hint = 'If the ball costs 10 cents, and the bat costs one dollar more than the ball, how much'
    elif '.05' in ans:
        hint = 'Make sure to enter the number of cents as a whole number.'

    if hint:
        hint = "&lt;font color='blue'&gt;Hint: {0}&lt;/font&gt;".format(hint)
        new_cmap.set_hint_and_mode(aid,hint,'always')
    </script>
    <p>If a bat and a ball cost $1.10 together, and the bat costs $1.00 more than the ball, how much</p>
    <p>
        <customresponse cfn="test_str" expect="5">
            <textline correct_answer="5" label="How much does the ball cost?"/>
            <hintgroup hintfn="hint_fn"/>
        </customresponse>
    </p>
</text>
</problem>

```

6.25.2 Problem with Adaptive Hint XML

Template

```

<problem>
    <text>
        <script type="text/python" system_path="python_lib">
def test_str(expect, ans):
    print expect, ans
    ans = ans.strip("''")
    ans = ans.strip('\'')
    return expect == ans.lower()

def hint_fn(answer_ids, student_answers, new_cmap, old_cmap):
    aid = answer_ids[0]
    ans = str(student_answers[aid]).lower()
    print 'hint_fn called, ans=', ans
    hint = ''
    if 'INCORRECT ANSWER 1' in ans:
        hint = 'HINT FOR INCORRECT ANSWER 1'
    elif 'INCORRECT ANSWER 2' in ans:
        hint = 'HINT FOR INCORRECT ANSWER 2'
    </script>
    <p>
        <customresponse cfn="test_str" expect="5">
            <textline correct_answer="5" label="How much does the ball cost?"/>
            <hintgroup hintfn="hint_fn"/>
        </customresponse>
    </p>
</text>
</problem>

```

```
if hint:
    hint = "&lt;font color='blue'&gt;Hint: {0}&lt;/font&gt;".format(hint)
    new_cmap.set_hint_and_mode(aid,hint,'always')
</script>
<p>TEXT OF PROBLEM</p>
<p>
    <customresponse cfn="test_str" expect="ANSWER">
        <textline correct_answer="ANSWER" label="LABEL TEXT"/>
        <hintgroup hintfn="hint_fn"/>
    </customresponse>
</p>
</text>
</problem>
```

Tags

- <text>: Surrounds the script and text in the problem.
- <customresponse>: Indicates that this problem has a custom response.
- <textline>: Creates a response field in the LMS where the student enters a response.
- <hintgroup>: Specifies that the problem contains at least one hint.

Tag: <customresponse>

Attributes

(none)

Children

- <textline>
- <hintgroup>

Tag: <textline>

Attributes

Attribute	Description
label (required)	Contains the text of the problem.
size (optional)	Specifies the size, in characters, of the response field in the LMS.
hidden (optional)	If set to “true”, students cannot see the response field.
correct_answer (optional)	Lists the correct answer to the problem.

Children

(none)

Tag: <hintgroup>

Attributes

Attribute	Description
hintfn	Must be set to hint_fn (i.e., the tag must appear as <hintgroup hintfn="hint_fn"/>).

6.26 Problem Written in LaTeX

Warning: This problem type is still a prototype and may not be supported in the future. By default, the ability to create these problems is not enabled in Studio. You must change the advanced settings in your course before you can create problems with LaTeX. Use this problem type with caution.

If you have a problem that is already written in LaTeX, you can use this problem 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: 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. For more information about how to create mathematical expressions in Studio using MathJax, see *A Brief Introduction to MathJax in Studio*.

Estimate the energy savings (in J/y) if all the people (3×10^8) in the U. S. switched from U. S. code to low flow shower heads.

Energy saved =

✓

0.52

Answer: 0.52

6.26.1 Create a Problem Written in LaTeX

To create a problem written in LaTeX:

1. Enable the policy key in your course.
 - (a) In Studio, click **Settings**, and then click **Advanced Settings**.
 - (b) On the **Advanced Settings** page, scroll down to the **use_latex_compiler** policy key.
 - (c) In the **Policy Value** field next to the **use_latex_compiler** policy key, change **false** to **true**.
 - (d) At the bottom of the page, click **Save Changes**.
2. In the unit where you want to create the problem, click **Problem** under **Add New Component**, and then click the **Advanced** tab.
3. Click **Problem Written in LaTeX**.
4. In the component editor that appears, click **Edit**.
5. In the lower left corner of the component editor, click **Launch LaTeX Source Compiler**.
6. Replace the example code with your own code. You can also upload a Latex file into the editor from your computer by clicking **Upload** in the bottom right corner.
7. In the lower left corner of the LaTeX source compiler, click **Save & Compile to edX XML**.

6.27 Protex Protein Builder Tool

The Protex protein builder asks students to create specified protein shapes by stringing together amino acids. In the example below, the goal protein shape is a simple line.

DESIGNING PROTEINS IN TWO DIMENSIONS (1 point possible)

The protein builder allows you string together the building blocks of proteins, amino acids, and see how that string will form into a structure. You are presented with a goal protein shape, and your task is to try to re-create it. In the example below, the shape that you are asked to form is a simple line.

Amino Acids

Ale A	+ Arg R	- Alan N	- Asp D	Cys C
Gln Q	- Glu E	Gly G	+ His H	Ile I
Leu L	+ Lys K	Met M	Phe F	Pro P
Ser S	Thr T	Trp W	+ Tyr Y	Val V

Amino Acid Sequence

Folded Protein

Target Shape

Fold Disulfide Bonds OFF

Be sure to click "Fold" to fold your protein before you click "Check".

Check Save Show Answer(s) You have used 1 of 10 submissions

6.27.1 Create the Protein Builder Tool

To create the protein builder:

1. Under **Add New Component**, click **Problem**, and then click **Blank Advanced Problem**.
2. In the component that appears, click **Edit**.
3. In the component editor, paste the Problem component code from below.
4. Make any changes that you want, and then click **Save**.

6.27.2 Protein Builder Tool Code

```

<problem>
    <p>The protein builder allows you string together the building blocks of proteins, amino acids, a
    <p>Be sure to click "Fold" to fold your protein before you click "Check".</p>

<script type="loncapa/python">

def protex_grader(expect,ans):
    import json
    ans=json.loads(ans)
    if "ERROR" in ans["protex_answer"]:
        raise ValueError("Protex did not understand your answer. Try folding the protein.")
    return ans["protex_answer"]=="CORRECT"

</script>

<text>
    <customresponse cfn="protex_grader">
        <designprotein2dinput width="855" height="500" target_shape="W;W;W;W;W;W;W;W"/>
    </customresponse>
</text>
<solution>
    <p>
        Many protein sequences, such as the following example, fold to a straight line. You can play around
        with the sequence to see how it folds.
    </p>
    <ul>
        <li>
            Stick: RRRRRR
        </li>
    </ul>
</solution>
</problem>

```

In this code:

- **width** and **height** specify the dimensions of the application, in pixels.
- **target_shape** lists the amino acids that, combined in the order specified, create the shape you've asked students to create. The list can only include the following letters, which correspond to the one-letter code for each amino acid. (This list appears in the upper-left corner of the protein builder.)

A	R	N	D
C	Q	E	G
H	I	L	K
M	F	P	S
T	W	Y	V

6.28 Text Input Problem

In text input problems, 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 problems to allow for typographical errors.

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?

herd immunity



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).

Reset

Hide Answer(s)

6.28.1 Create a Text Input Problem

You can create text input problems in the Simple Editor or in the Advanced Editor.

Note: All problems must include labels for accessibility. The label generally includes the text of the main question in your problem. To add a label for a common problem, surround the text of the label with angle brackets pointed toward the text (>>label text<<).

Simple Editor

To create a text input problem in the Simple Editor, 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 problem.
5. Determine the text of the problem to use as a label, and then surround that text with two sets of angle brackets (>><<).
6. Select the text of the answer, and then click the text input button.

ABC

When you do this, an equal sign appears next to the answer.

7. In the component editor, select the text of the explanation, and then click the explanation button to add explanation tags around the text.



8. On the **Settings** tab, specify the settings that you want.

9. 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]

Advanced Editor

To create this problem in the Advanced Editor, click the **Advanced** tab in the Problem component editor, and then replace the existing code with the following code.

```
<problem>
<p>
    <em>This problem is adapted from an exercise that first appeared in MITx's 14.73x The Challenges o</em>
</p>
<p>What is the technical term that refers to the fact that, when enough people sleep under a bednet,</p>
<stringresponse answer=".+herd immunity.+" type="ci regexp">
    <additional_answer>community immunity</additional_answer>
    <additional_answer>population immunity</additional_answer>
    <textline size="20" label="What is the technical term that refers to the fact that, when enou</textline>
    <hintgroup>
        <stringhint answer="contact immunity" type="ci" name="contact_immunity_hint" />
        <hintpart on="contact_immunity_hint">
            <startouttext />
            In contact immunity, a vaccinated individual passes along his immunity to another per</startouttext />
        </hintpart >
        <stringhint answer="firewall immunity" type="ci" name="firewall_immunity_hint" />
        <hintpart on="firewall_immunity_hint">
            <startouttext />
            Although a firewall provides protection for a population, the term "firewall" is used</startouttext />
        </hintpart >
    </hintgroup>
</stringresponse>
<solution>
    <div class="detailed-solution">
        <p>Explanation</p>
        <p>The correct answer is <b>herd immunity</b>. As more and more people use bednets, the risk of m</p>
    </div>
</solution>
</problem>
```

6.28.2 Multiple Responses in Text Input Problems

You can specify more than one correct response for text input problems. For example, instead of requiring students to enter exactly “Dr. Martin Luther King, Junior,” you can allow answers of “Martin Luther King,” “Doctor Martin Luther King,” and other variations. To do this, you can use the Simple Editor or the Advanced Editor.

Simple Editor

To specify additional correct responses in the Simple Editor, include “or=” (without the quotation marks) before each additional correct response.

```
>>What African-American led the United States civil rights movement during the 1960s?<<
= Dr. Martin Luther King, Jr.
or= Dr. Martin Luther King, Junior
or= Martin Luther King, Jr.
or= Martin Luther King
```

Advanced Editor

To specify additional correct responses in the Advanced Editor, add an <additional_answer> for each correct response inside the opening and closing <stringresponse> tags.

```
<problem>

<p>What African-American led the United States civil rights movement during the 1960s?</p>

<stringresponse answer="Dr. Martin Luther King, Jr." type="ci" >
    <additional_answer>Dr. Martin Luther King, Junior</additional_answer>
    <additional_answer>Martin Luther King, Jr.</additional_answer>
    <additional_answer>Martin Luther King</additional_answer>
    <textline label="What African-American led the United States civil rights movement during the 1960s" />
</stringresponse>

</problem>
```

6.28.3 Case Sensitivity and Text Input Problems

By default, text input problems do not require a case sensitive response. You can change this and require a case sensitive answer.

To make a text input response case sensitive, you must use *The Advanced Editor*.

In the Advanced Editor, you see that the **type** attribute of the **stringresponse** element equals **ci**, for *case insensitive*. For example:

```
<stringresponse answer="Michigan" type="ci">
    <textline size="20"/>
</stringresponse>
```

To make the response case sensitive, change the value of the **type** attribute to **cs**.

```
<stringresponse answer="Michigan" type="cs">
    <textline size="20"/>
</stringresponse>
```

6.28.4 Response Field Length of Text Input Problems

By default, the response field for text input problems is 20 characters long.

You should preview the unit to ensure that the length of the response input field accommodates the correct answer, and provides extra space for possible incorrect answers.

If the default response field length is not sufficient, you can change it using *The Advanced Editor*.

In the advanced editor, in the XML block for the answer, you see that the **size** attribute of the **textline** element equals **20**:

```
<stringresponse answer="Democratic Republic of the Congo" type="ci">
  <textline size="20"/>
</stringresponse>
```

To change the response field length, change the value of the **size** attribute:

```
<stringresponse answer="Democratic Republic of the Congo" type="ci">
  <textline size="40"/>
</stringresponse>
```

6.28.5 Hints and Regular Expressions in Text Input Problems

You can provide hints that appear when students enter common incorrect answers in text input problems. You can also set a text input problem to allow a regular expression as an answer. To do this, you'll have to modify the problem's XML in the Advanced Editor.

The regular expression that the student enters must contain the part of the answer that the instructor specifies. For example, if an instructor has specified `<answer=".example answer.*" type="regexp">`, correct answers include `example answered`, two `example answers`, or even `==example answer==`, but not `examples` or `example anser`.

You can add `regexp` to the value of the `type` attribute, for example: `type="ci regexp"` or `type="regexp"` or `type="regexp cs"`. In this case, any answer or hint are treated as regular expressions.

6.28.6 Text Input Problem XML

Template

```
<problem>
  <p>Problem text</p>
  <stringresponse answer="**.Correct answer 1.**" type="ci regexp">
    <additional_answer>Correct answer 2</additional_answer>
    <additional_answer>Correct answer 3</additional_answer>
    <textline size="20" label="label text"/>
    <hintgroup>
      <stringhint answer="Incorrect answer A" type="ci" name="hintA" />
      <hintpart on="hintA">
        <startouttext />Text of hint for incorrect answer A<endouttext />
      </hintpart >
      <stringhint answer="Incorrect answer B" type="ci" name="hintB" />
      <hintpart on="hintB">
        <startouttext />Text of hint for incorrect answer B<endouttext />
      </hintpart >
      <stringhint answer="Incorrect answer C" type="ci" name="hintC" />
      <hintpart on="hintC">
```

```
<startouttext />Text of hint for incorrect answer C<endouttext />
  </hintpart>
</hintgroup>
</stringresponse>
<solution>
<div class="detailed-solution">
<p>Explanation or Solution Header</p>
<p>Explanation or solution text</p>
</div>
</solution>
</problem>
```

Tags

- `<stringresponse>`: Indicates that the problem is a text input problem.
- `<textline>`: Child of `<stringresponse>`. Creates a response field in the LMS where the student enters a response.
- `<additional_answer>` (optional): Specifies an additional correct answer for the problem. A problem can contain an unlimited number of additional answers.
- `<hintgroup>` (optional): Indicates that the instructor has provided hints for certain common incorrect answers.
- `<stringhint />` (optional): Child of `<hintgroup>`. Specifies the text of the incorrect answer to provide the hint for. Contains answer, type, name.
- `<hintpart>`: Contains the name from `<stringhint>`. Associates the incorrect answer with the hint text for that incorrect answer.
- `<startouttext />`: Indicates the beginning of the text of the hint.
- `<endouttext />`: Indicates the end of the text of the hint.

Tag: `<stringresponse>`

Indicates that the problem is a text input problem.

Attributes

Attribute	Description
answer (re- quired)	Specifies the correct answer. To designate the answer as a regular expression, add “regexp” to the type attribute. If you do not add “regexp” to the type attribute, the student’s answer must match the value in this attribute exactly.
type (op- tional)	Can specify whether the problem is case sensitive and allows regular expressions. If the <code><stringresponse></code> tag includes <code>type="ci"</code> , the problem is not case sensitive. If the tag includes <code>type="cs"</code> , the problem is case sensitive. If the tag includes <code>type="regexp"</code> , the problem allows regular expressions. A type attribute in a <code><stringresponse></code> tag can also combine these values. For example, <code><stringresponse type="regexp cs"></code> specifies that the problem allows regular expressions and is case sensitive.

Children

- `<textline />` (required)
- `<additional_answer>` (optional)
- `<hintgroup>` (optional)

Tag: <textline />

Creates a response field in the LMS where the student enters a response.

Attributes

Attribute	Description
label (required)	Contains the text of the problem.
size (optional)	Specifies the size, in characters, of the response field in the LMS.
hidden (optional)	If set to “true”, students cannot see the response field.
correct_answer (optional)	Lists the correct answer to the problem.

Children

(none)

Tag: <additional_answer>

Specifies an additional correct answer for the problem. A problem can contain an unlimited number of additional answers.

Attributes

(none)

Children

(none)

Tag: <hintgroup>

Indicates that the instructor has provided hints for certain common incorrect answers.

Attributes

(none)

Children

- <stringhint> (required)

Tag: <stringhint>

Specifies a common incorrect answer to the problem.

Attributes

Attribute	Description
answer (required)	The text of the incorrect answer.
name (required)	The name of the hint that you want to provide.
type	Specifies whether the text of the specified incorrect answer is case sensitive. Can be set to “cs” (case sensitive) or “ci” (case insensitive).

Children

- <hintpart> (required)

Tag: <hintpart>

Associates a common incorrect answer with the hint for that incorrect answer.

Attributes

At-tribute on	Description The name of the hint. This must be the same as the name attribute of the <code><stringhint></code> tag. (The <code><stringhint></code> tag provides the name of the hint and the incorrect answer to associate with the hint. The <code><hintpart></code> tag contains the name of the hint and the text of the hint.)
---------------	--

Children

- `<startouttext />` (required)
- `<endouttext />` (required)

Tags: `<startouttext />` and `<endouttext>`

Surround the text of the hint.

Attributes

(none)

Children

(none)

6.29 Word Cloud Tool

In a word cloud tool, 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. The graphic becomes larger as more students answer. Students can both see the way their peers have answered and contribute their thoughts to the group.

For example, the following word cloud was created from students' responses to a question in a HarvardX course.

This exercise first appeared in HarvardX's PH278x: Human Health and Global Environmental Change course, spring 2013.

6.29.1 Create a Word Cloud Tool

To create a word cloud tool:

1. Add the Word Cloud advanced component.
 - (a) On the **Settings** menu, click **Advanced Settings**.
 - (b) 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).
 - (c) Under **Policy Value**, place your cursor between the brackets, and then enter the following. Make sure to include the quotation marks.
"word_cloud"
 - (d) At the bottom of the page, click **Save Changes**.

The page refreshes automatically. At the top of the page, you see a notification that your changes have been saved.

- (e) Return to the unit where you want to add the specialized problem. The list of possible components now contains an Advanced component.
2. In the unit where you want to create the problem, click **Advanced** under **Add New Component**.
3. In the list of problem types, click **Word Cloud**.
4. In the component that appears, click **Edit**.
5. 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.
6. Click **Save**.

6.30 Write-Your-Own-Grader Problem

In custom Python-evaluated input (also called “write-your-own-grader” problems), the grader uses a Python script that you create and embed in the problem to evaluates a student’s response or provide hints. These problems can be any type. Numerical input and text input problems are the most popular write-your-own-grader problems.

CUSTOM PYTHON-EVALUATED INPUT (2/4 points)

This question has two parts.

Part 1: Enter two integers that sum to 10.

 Answer: 3

 Answer: 7

Part 2: Enter two integers that sum to 20.



EXPLANATION

For part 1, 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 part 2, 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.

Reset

Hide Answer(s)

You have used 3 of 5 submissions

Custom Python-evaluated input problems can include the following:

- *Chemical Equation Problem*
- *Custom JavaScript Problem*
- *Gene Explorer Tool*
- *Molecule Editor Tool*
- *Protex Protein Builder Tool*

<pre><script type="loncapa/python"> <customresponse cfn="test_add_to_ten"> <customresponse cfn="test_add" expect="20"> <textline size="10" correct_answer="3"/></pre>	Indicates that the problem contains a Python script.
	This tag includes the <code>size</code> , <code>correct_answer</code> , and <code>label</code> attributes. The <code>correct_answer</code> attribute is optional.

You can create one of these problems in *Answer Tag Format* or *Script Tag Format*.

6.30.1 Answer Tag Format

The answer tag format encloses the Python script in an `<answer>` tag:

```
<answer>
if answers[0] == expect:
    correct[0] = 'correct'
    overall_message = 'Good job!'
else:
    correct[0] = 'incorrect'
    messages[0] = 'This answer is incorrect'
    overall_message = 'Please try again'
</answer>
```

Important: Python honors indentation. Within the `<answer>` tag, you must begin your script with no indentation.

The Python script interacts with these variables in the global context:

- `answers`: An ordered list of answers the student provided. For example, if the student answered 6, `answers[0]` would equal 6.
- `expect`: The value of the `expect` attribute of `<customresponse>` (if provided).
- `correct`: An ordered list of strings indicating whether the student answered the question correctly. Valid values are "correct", "incorrect", and "unknown". You can set these values in the script.
- `messages`: An ordered list of messages that appear under each response field in the problem. You can use this to provide hints to users. For example, if you include `messages[0] = "The capital of California is Sacramento"`, that message appears under the first response field in the problem.
- `overall_message`: A message that appears beneath the entire problem. You can use this to provide a hint that applies to the entire problem rather than a particular response field.

Create a Custom Python-Evaluated Input Problem in Answer Tag Format

To create a custom Python-evaluated input problem using an `<answer>` tag:

1. In the unit where you want to create the problem, 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 the following code.
5. Click **Save**.

```

<problem>
    <p>What is the sum of 2 and 3?</p>

    <customresponse expect="5">
        <textline math="1" />
    </customresponse>

    <answer>
if answers[0] == expect:
    correct[0] = 'correct'
    overall_message = 'Good job!'
else:
    correct[0] = 'incorrect'
    messages[0] = 'This answer is incorrect'
    overall_message = 'Please try again'
    </answer>
</problem>

```

Important: Python honors indentation. Within the `<answer>` tag, you must begin your script with no indentation.

6.30.2 Script Tag Format

The script tag format encloses a Python script that contains a “check function” in a `<script>` tag, and adds the `cfn` attribute of the `<customresponse>` tag to reference that function:

```

<problem>

<script type="loncapa/python">

def test_add(expect, ans):
    try:
        a1=int(ans[0])
        a2=int(ans[1])
        return (a1+a2) == int(expect)
    except ValueError:
        return False

def test_add_to_ten(expect, ans):
    return test_add(10, ans)

</script>

<p>Enter two integers that sum to 10. </p>
<customresponse cfn="test_add_to_ten">
    <textline size="10"/><br/>
    <textline size="10"/>
</customresponse>

</problem>

```

Important: Python honors indentation. Within the `<script>` tag, the `def check_func(expect, ans):` line must have no indentation.

The `check` function accepts two arguments:

- `expect` is the value of the `expect` attribute of `<customresponse>` (if provided)

- `answer` is either:
 - The value of the answer the student provided, if the problem only has one response field.
 - An ordered list of answers the student provided, if the problem has multiple response fields.

The `check` function can return any of the following to indicate whether the student's answer is correct:

- `True`: Indicates that the student answered correctly for all response fields.
- `False`: Indicates that the student answered incorrectly. All response fields are marked as incorrect.
- A dictionary of the form: `{ 'ok': True, 'msg': 'Message' }` If the dictionary's value for `ok` is set to `True`, all response fields are marked correct; if it is set to `False`, all response fields are marked incorrect. The `msg` is displayed beneath all response fields, and it may contain XHTML markup.
- A dictionary of the form

```
{ 'overall_message': 'Overall message',
  'input_list': [
    { 'ok': True, 'msg': 'Feedback for input 1' },
    { 'ok': False, 'msg': 'Feedback for input 2' },
    ... ] }
```

The last form is useful for responses that contain multiple response fields. It allows you to provide feedback for each response field individually, as well as a message that applies to the entire response.

Example of a checking function:

```
def check_func(expect, answer_given):
    check1 = (int(answer_given[0]) == 1)
    check2 = (int(answer_given[1]) == 2)
    check3 = (int(answer_given[2]) == 3)
    return {'overall_message': 'Overall message',
            'input_list': [
                { 'ok': check1, 'msg': 'Feedback 1' },
                { 'ok': check2, 'msg': 'Feedback 2' },
                { 'ok': check3, 'msg': 'Feedback 3' } ] }
```

The function checks that the user entered 1 for the first input, 2 for the second input, and 3 for the third input. It provides feedback messages for each individual input, as well as a message displayed beneath the entire problem.

Create a Custom Python-Evaluated Input Problem in Script Tag Format

To create a custom Python-evaluated input problem using a `<script>` tag:

1. In the unit where you want to create the problem, 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 the following code.
5. Click **Save**.

Problem Code:

```
<problem>
<p>This question has two parts.</p>

<script type="loncapa/python">
```

```

def test_add(expect, ans):
    try:
        a1=int(ans[0])
        a2=int(ans[1])
        return (a1+a2) == int(expect)
    except ValueError:
        return False

def test_add_to_ten(expect, ans):
    return test_add(10, ans)

</script>

<p>Part 1: Enter two integers that sum to 10. </p>
<customresponse cfn="test_add_to_ten">
    <textline size="10" correct_answer="3" label="Integer #1"/><br/>
    <textline size="10" correct_answer="7" label="Integer #2"/>
</customresponse>

<p>Part 2: Enter two integers that sum to 20. </p>
<customresponse cfn="test_add" expect="20">
    <textline size="10" label="Integer #1"/><br/>
    <textline size="10" label="Integer #2"/>
</customresponse>

<solution>
    <div class="detailed-solution">
        <p>Explanation</p>
        <p>For part 1, any two numbers of the form <i>n</i> and <i>10-n</i>, where <i>n</i> is any integer between 0 and 9 inclusive. For part 2, any pair <i>x</i> and <i>20-x</i> will work, where <i>x</i> is any real number between 0 and 20 inclusive.</p>
    </div>
</solution>
</problem>

```

Templates

The following template includes answers that appear when the student clicks **Show Answer**.

```
<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>Problem text</p>
<customresponse cfn="test_add" expect="20">
    <textline size="10" correct_answer="11" label="Integer #1"/><br/>
    <textline size="10" correct_answer="9" label="Integer #2"/>
</customresponse>

<solution>
    <div class="detailed-solution">
        <p>Solution or Explanation Heading</p>
        <p>Solution or explanation text</p>
    </div>

```

```
</solution>
</problem>
```

The following template does not return answers when the student clicks **Show Answer**. If your problem doesn't include answers for the student to see, make sure to set **Show Answer** to **Never** in the problem component.

```
<problem>

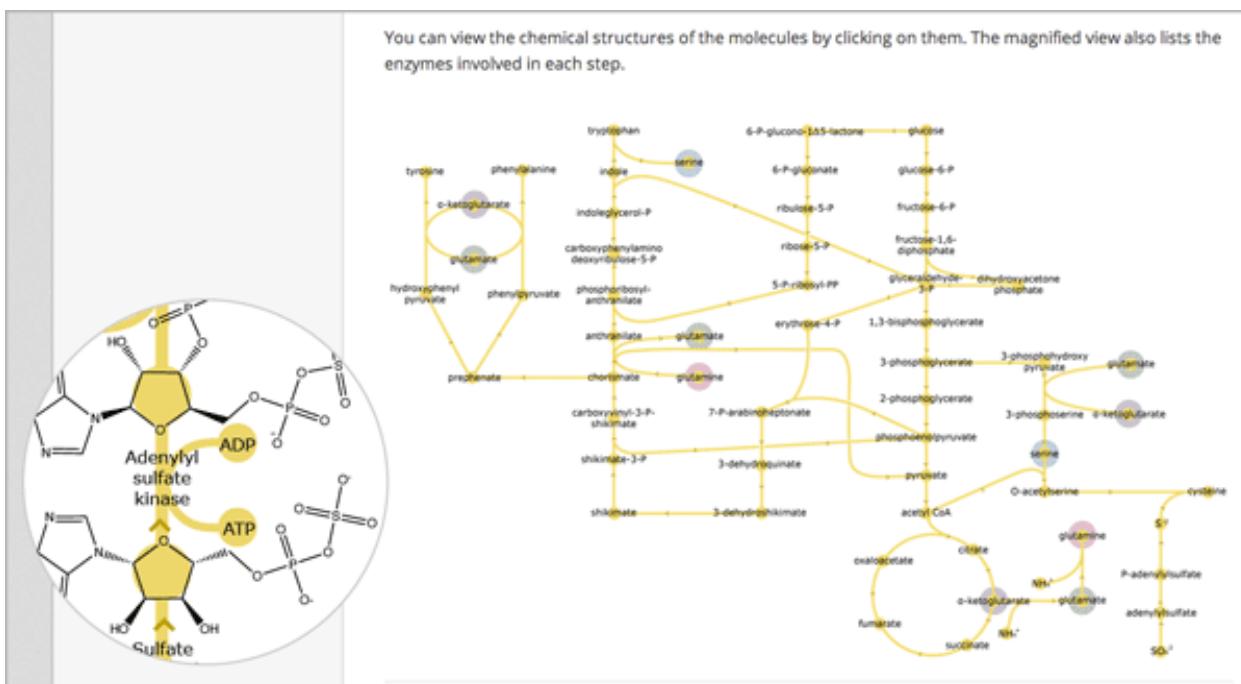
<script type="loncapa/python">
def test_add(expect,ans):
    a1=float(ans[0])
    a2=float(ans[1])
    return (a1+a2)== float(expect)
</script>

<p>Enter two real numbers that sum to 20: </p>
<customresponse cfn="test_add" expect="20">
    <textline size="10" label="Integer #1"/><br/>
    <textline size="10" label="Integer #2"/>
</customresponse>

<solution>
    <div class="detailed-solution">
        <p>Solution or Explanation Heading</p>
        <p>Solution or explanation text</p>
    </div>
</solution>
</problem>
```

6.31 Zooming Image Tool

You may want to present information to your students as an image. If your image is very large or very detailed, students may not be able to see all the information in the image. You can use the zooming image tool to enlarge areas of your image as the student moves the mouse over the image, as in the example below.



6.31.1 Components of a Zooming Image Tool

To create a zooming image tool, you need the following files.

- The image that you want students to see when they access the unit.
 - The image that appears in the magnified area when a student clicks the regular image. This image may be larger than the regular image.
 - The **jquery.loupeAndLightbox.js** JavaScript file. Every zooming image tool uses this JavaScript file, and you won't make any changes to it. [To download this file, right-click here](#), and then click **Save Link As** to save the file on your computer.

6.31.2 Create a Zooming Image Tool

1. Upload your regular-sized image file, your small image file, and the **jquery.loupeAndLightbox.js** file to the **Files & Uploads** page. For more information about how to do this, see *Adding Files to a Course*.
 2. Under **Add New Component**, click **html**, and then click **Zooming Image**.
 3. In the new component that appears, click **Edit**.
 4. In the component editor, replace the default problem text with your own text.
 5. Switch to the **HTML** tab.
 6. Replace the following placeholders with your own content.
 - Replace the following file name and path with the name and path of the image that you want to appear magnified when the user hovers over the regular image.
https://studio.edx.org/c4x/edX/DemoX/asset/pathways_detail_01.png
For example, your file name and path may be **/static/Image1.jpg**.

- Replace the following file name and path with the name and path of the image that you want to appear when the page opens.

https://studio.edx.org/c4x/edX/DemoX/asset/pathways_overview_01.png

For example, your file name and path may be [/static/Image2.jpg](#).

- Replace the following name and file path with the name and path of the JavaScript file for your course.

<https://studio.edx.org/c4x/edX/DemoX/asset/jquery.loupeAndLightbox.js>

Because you uploaded the `jquery.loupeAndLightbox.js` file to the **Files & Uploads** page, your file name and path is [/static/jquery.loupeAndLightbox.js](#).

- (Optional) If you want the magnified area to be larger or smaller, change the **width** and **height** values from 350 to larger or smaller numbers. For example, you can change both numbers to 450. Or, if you want a horizontal oval instead of a circle, you can change the **width** value to a number such as 500 and the **height** value to a number such as 150.

The HTML in your zooming image tool may resemble the following.

The screenshot shows the 'Editing: Zooming Image' interface in the edX Studio. At the top, there are 'Editor' and 'Settings' tabs, with 'Editor' selected. Below them are 'Visual' and 'HTML' tabs, with 'HTML' selected. The main area contains the following HTML code:

```
1 <h2>ZOOMING DIAGRAMS</h2>
2 <p>Some edX classes use extremely large, extremely detailed graphics. To make it
   easier to understand we can offer two versions of those graphics, with the zoomed
   section showing when you click on the main view.</p>
3 <p>The example below is from <a href="https://www.edx.org/course/mit/7-
   00x/introduction-biology-secret-life/1014" target="_blank">7.00x: Introduction to
   Biology</a> and shows a subset of the biochemical reactions that cells carry out.
   </p>
4 <p>You can view the chemical structures of the molecules by clicking on them. The
   magnified view also lists the enzymes involved in each step.</p>
5
6 <div class="zooming-image-place" style="position: relative;">
7   <a class="loupe" href="/static/Image1.jpg">
8     
9   </a>
10  <div class="script_placeholder" data-src="/static/jquery.loupeAndLightbox.js" />
11 </div>
12 <script type="text/javascript">// <![CDATA[
13 JavascriptLoader.executeModuleScripts($('.zooming-image-place').eq(0), function() {
14   $('.loupe').loupeAndLightbox({
15     width: 500,
16     height: 150,
17     lightbox: false
18   });
19 });
20 // ]]></script>
21 <div id="ap_listener_added"></div>
22
```

At the bottom of the editor are 'SAVE' and 'CANCEL' buttons.

7. Click **Save** to save the HTML component.

6.32 A Brief Introduction to MathJax in Studio

To write clear and professional-looking symbols and equations, we use a LaTeX-like language called **MathJax**. Your MathJax equations can appear with other text in the paragraph (inline equations) or on their own lines (display equations).

- For inline equations, you can do either of the following.
 - Surround your Mathjax expression with backslashes and **parentheses**.
`\(equation \)`
 - Surround your Mathjax expression with [mathjaxinline] tags. Note that these tags use square brackets ([]).
`[mathjaxinline] equation [/mathjaxinline]`
- For display equations, you can do either of the following.
 - Surround your Mathjax expression with backslashes and **brackets**.
`\[equation \]`
 - Surround your Mathjax expression with [mathjax] tags. Note that these tags use square brackets ([]).
`[mathjax] equation [/mathjax]`

You can use MathJax in HTML (text) components and in Problem components.

Note: Complete MathJax documentation (together with a testing tool) can be found at <http://www.onematicalcat.org/MathJaxDocumentation/TeXSyntax.htm>.

6.32.1 HTML (Text) Components

In the HTML component editor, you can use MathJax both in Visual view and in HTML view.

Editing: Text Editor Settings

Heading 3 B I π Σ \int $\frac{d}{dx}$ $\frac{\partial}{\partial x}$ $\frac{\partial^2}{\partial x^2}$ $\frac{\partial^3}{\partial x^3}$ $\frac{\partial^4}{\partial x^4}$ $\frac{\partial^5}{\partial x^5}$ $\frac{\partial^6}{\partial x^6}$ $\frac{\partial^7}{\partial x^7}$ $\frac{\partial^8}{\partial x^8}$ $\frac{\partial^9}{\partial x^9}$ $\frac{\partial^{10}}{\partial x^{10}}$

Visual HTML

Inline Example

You can add equations such as $\mathcal{B} = \nabla \times \mathcal{A}$ inline with the rest of your text.

Display Example

Even though the expression $\mathcal{B} = \frac{\mu_0 I}{4\pi} \int \frac{d\vec{l} \times \hat{r}}{r^2}$ appears inline in the editor, the brackets mean that it will appear on its own line on the unit page in Studio and in the LMS.

SAVE **CANCEL**

HTML component editor: Visual view

Editing: Text Editor Settings

Visual HTML

```

1 <h3>Inline Example</h3>
2 <p>You can add equations such as \(\mathcal{B} = \nabla \times \mathcal{A}\) inline with the rest of your text.</p>
3 <h3>Display Example</h3>
4 <p>>Even though the expression \(\mathcal{B} = \frac{\mu_0 I}{4\pi} \int \frac{d\vec{l} \times \hat{r}}{r^2}\) appears inline in the editor, the brackets mean that it will appear on its own line on the unit page in Studio and in the LMS.</p>

```

SAVE **CANCEL**

HTML component editor: HTML view

Edit **Delete**

Inline Example

You can add equations such as $\mathcal{B} = \nabla \times \mathcal{A}$ inline with the rest of your text.

Display Example

Even though the expression

$$\mathcal{B} = \frac{\mu_0 I}{4\pi} \int \frac{d\vec{l} \times \hat{r}}{r^2}$$

appears inline in the editor, the brackets mean that it will appear on its own line on the unit page in Studio and in the LMS.

Studio: Unit page

Inline Example

You can add equations such as $\mathcal{B} = \nabla \times \mathcal{A}$ inline with the rest of your text.

Display Example

Even though the expression

$$\mathcal{B} = \frac{\mu_0 I}{4\pi} \int \frac{d\vec{l} \times \hat{r}}{r^2}$$

appears inline in the editor, the brackets mean that it will appear on its own line on the unit page in Studio and in the LMS.

LMS

6.32.2 Problem Components

In the Problem component editor, you can use MathJax both in the Simple Editor and in the Advanced Editor.

In the example problem below, note that the Einstein equation in the explanation is enclosed in backslashes and parentheses, so it appears inline with the text. The Navier-Stokes equation is enclosed in backslashes and brackets, so it appears on its own line.

Editing: Multiple Choice Editor Settings

H1 ABC 123 Advanced Editor ?

This nonsensical question is for demonstration purposes only.

Which of the following equations is the most complex?
 $E = mc^2$
 $A = \pi r^2$
 $\rho(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}) = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}$
 $V = IR$

[explanation]
 Although a true understanding of Einstein's equation $E = mc^2$ relating mass and energy requires years of study, the Navier-Stokes equation $\rho(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}) = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}$ is computationally more complex (and frequently gives students nightmares).
[explanation]

SAVE **CANCEL**

Problem component editor: Simple Editor

Editing: Multiple Choice Editor Settings

```

1 <problem>
2 <p>This nonsensical question is for demonstration
purposes only.</p>
3
4 <p>Which of the following equations is the most
complex?</p>
5 <multiplechoiceresponse>
6   <choice correct="false">\( E = m c^2 \)</choice>
7   <choice correct="false">\( A = \pi r^2 \)</choice>
8   <choice correct="true">\( \rho(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}) = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f} ) \)</choice>
9   <choice correct="false">\( V = I R \)</choice>
10 </multiplechoiceresponse>
11 </p>
12 <h2>Explanation</h2>
13 <p>Although a true understanding of Einstein's
equation  $E = mc^2$  relating mass and energy
requires years of study, the Navier-Stokes equation  $\rho(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}) = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}$  is computationally more complex (and
frequently gives students nightmares).</p>
17
18 </problem>
```

SAVE **CANCEL**

Problem component editor: Advanced Editor

Edit **Delete**

MULTIPLE CHOICE (1 point possible)

This nonsensical question is for demonstration purposes only.

Which of the following equations is the most complex?

$E = mc^2$
 $A = \pi r^2$
 $\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}$
 $V = IR$

EXPLANATION

Although a true understanding of Einstein's equation $E = mc^2$ relating mass and energy requires years of study, the Navier-Stokes equation

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}$$

is computationally more complex (and frequently gives students nightmares).

Check **Save**

Studio: Unit page

MULTIPLE CHOICE (1 point possible)

This nonsensical question is for demonstration purposes only.

Which of the following equations is the most complex?

$E = mc^2$
 $A = \pi r^2$
 $\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}$
 $V = IR$

EXPLANATION

Although a true understanding of Einstein's equation $E = mc^2$ relating mass and energy requires years of study, the Navier-Stokes equation

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}\right) = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}$$

is computationally more complex (and frequently gives students nightmares).

Check **Save**

LMS

Releasing Your Course

7.1 Testing Your Course

The way your course looks in Studio is not the way students see and experience it. To test your course, while you develop course content and afterward, you must view and interact with your course from a student's point of view.

From within Studio, you can:

- *Preview Your Course*
- *Switch Between Studio and Your Live Course*

For information about setting up a beta test for your course, see *Beta Testing a Course*.

7.1.1 Preview Your Course

When you view your course through 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.

Using Preview mode is the only way to see content that is set to Private **as a student would see it**.

You can enter Preview mode in two ways.

- On any subsection page, click **Preview Drafts**.

You are editing a draft.

View the Live Version

Display Name: Working with Videos

Building a Computer Memory Element

BUILDING A COMPUTER MEMORY ELEMENT

Unit Settings

This is a draft of the published unit. To update the live version, you must [replace it with this draft](#).

This unit is scheduled to be released to **students** on **Jan 01, 1970 at 05:00 UTC** with the subsection [Lesson 1 - Getting Started](#)

Delete Draft **Preview**

- On any Unit page, click **Preview**.

The following example shows the **Preview** button for a unit that is set to Public.

Building and Running an edX Course, Release

The screenshot shows the edX Studio interface. On the left, there's a list of units under the heading 'Display Name: Working with Videos'. One unit is expanded, showing its title 'Building a Computer Memory Element' and a preview of its content 'BUILDING A COMPUTER MEMORY ELEMENT'. On the right, there's a 'Unit Settings' panel. It contains a message about the unit being a draft of a published unit, a scheduled release date of Jan 01, 1970, and a 'Delete Draft' and 'Preview' button. The 'Preview' button is highlighted with a red circle.

The following example shows the **Preview** button for a unit that is set to Private.

This screenshot is similar to the one above, showing the 'Working with Videos' unit in Studio. The unit is set to 'Private' and scheduled for release on Jan 01, 1970. The 'Preview' button in the Unit Settings panel is highlighted with a red circle.

7.1.2 Switch Between Studio and Your Live Course

While you work in Studio, you can switch to your live course to see how your course appears to students. Because you are logged in as a course team member, when you switch to your live course you see the **Staff View**.

The screenshot shows the edX Staff View interface. At the top, there's a header with the edX logo, the course name 'edX: Open_DemoX edX Demonstration Course', and a 'staff' dropdown. Below the header, there's a navigation bar with tabs for 'Courseware', 'Course Info', 'Discussion', 'Wiki', 'Progress', 'Instructor', and 'Staff view' (which is highlighted). On the left, there's a sidebar with sections for 'Introduction', 'Example Week 1: Getting Started', and 'Example Week 2: Get Interactive'. In the center, there's a slide with the title 'FIND YOUR STUDY BUDDY' and a button labeled 'VIEW UNIT IN STUDIO'.

In **Staff View**:

- You see all of the units that are set to **Public**, regardless of the release dates of the containing section or subsection.
- You do not see units that are set to **Private**. To see Private units, you must use Preview mode as described in *Preview Your Course*.
- You can access the Instructor Dashboard, which has features and reports that help you run your course.

When you view your live course in **Staff View**, you can execute tests to make sure your course works the way you intend. For example, the course team can work through all of the problems to verify that the correct answer gets a green check for correct, and that any answer other than the correct one gets a red X for incorrect.

Go to Your Live Course from Studio

To switch from Studio to your live course, click **View Live** on any of these pages:

- The Course Outline page.

The screenshot shows the 'Course Outline' page with two sections: 'Example Week 1: Getting Started' and 'Example Week 2: Get Interactive'. Each section contains subsections like 'Lesson 1 - Getting Started', 'Homework - Question Styles', 'Lesson 2 - Let's Get Interactive!', 'Homework - Labs and Demos', and 'Homework - Essays'. A sidebar on the right provides instructions on what can be done on the page, including creating new sections and subsections, setting release dates, and managing assignment types. The 'View Live' button at the top right of the page is highlighted with a red circle.

- Any Subsection page.

The screenshot shows a 'Subsection Settings' page for 'Lesson 2 - Let's Get Interactive!'. It includes fields for 'Release Day' (01/01/2014) and 'Release Time (UTC)' (05:00). The 'Graded as' field is set to 'NOT GRADED'. A 'SET A DUE DATE' link is present. At the bottom are 'Preview Drafts' and 'View Live' buttons, with 'View Live' circled in red. On the left, there's a list of units under 'Units'.

Subsection Settings	
Release Day	Release Time (UTC)
01/01/2014	05:00
Graded as: NOT GRADED	
SET A DUE DATE	
Preview Drafts	View Live

Units:

- Lesson 2 - Let's Get Interactive!
- An Interactive Reference Table
- Zooming Diagrams
- Electronic Sound Experiment

+ New Unit

- The Unit page, if the Unit is Public.

Building and Running an edX Course, Release

The screenshot shows the edX Studio interface. On the left, there's a video player window titled "BUILDING A COMPUTER MEMORY ELEMENT". The video frame displays a hand-drawn diagram of a memory cell with annotations like "Building a memory element ...", "Ⓐ First attempt", and a play button. To the right of the video player is a "Unit Settings" panel. At the top of this panel is a "Visibility" dropdown set to "Public". Below it, a message states: "This unit has been published. To make changes, you must [edit a draft](#)". Another message indicates: "This unit is scheduled to be released to students on Jan 01, 2013 at 05:00 UTC with the subsection [Lesson 1 - Getting Started](#)". A red circle highlights the "View Live" button. On the far right, there's a "Unit Location" sidebar with a tree view of courseware sections: "Example Week 1: Getting Started" (selected), "Lesson 1 - Getting Started" (selected), "Getting Started" (selected), "Working with Videos" (selected), "Videos on edX", "Video Demonstrations", "Video Presentation Styles", "Interactive Questions - PRIVATE", "Reading Assignments", and a "+ New Unit" link.

When you click **View Live**, the Courseware page or the corresponding subsection or unit page opens.

Go to Studio from Your Live Course

To switch from your live course to Studio, click **View {page} in Studio** on any of these pages:

- Any unit (shown in the *Switch Between Studio and Your Live Course* section above).
- The Instructor Dashboard.

The screenshot shows the top navigation bar of the Instructor Dashboard. It includes tabs for "Courseware", "Course Info", "Discussion", "Wiki", "Progress", and "Instructor" (which is currently selected). Below the tabs are two buttons: "BACK TO STANDARD DASHBOARD" and "VIEW COURSE IN STUDIO".

Instructor Dashboard

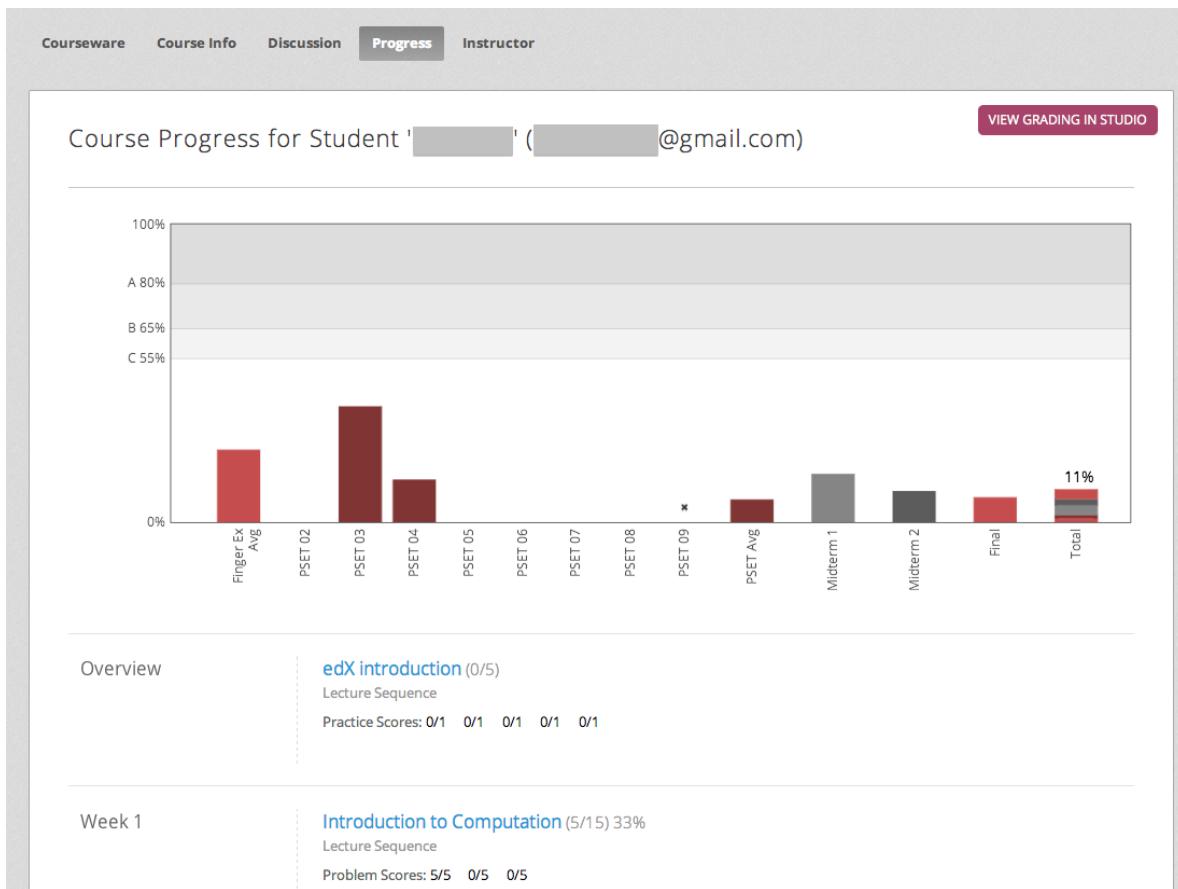
[COURSE INFO](#) [MEMBERSHIP](#) [STUDENT ADMIN](#) [DATA DOWNLOAD](#) [ANALYTICS](#)

ENROLLMENT INFORMATION

Total number of enrollees (instructors, staff members, and students)

For information about the tasks you can complete on the Instructor Dashboard, see *Running Your Course*.

- The **Course Progress** page for a specified student.



For information about checking a student's progress, see *Review How Grading Is Configured for Your Course*.

7.2 Publishing Your Course

When you have set up your course, established a grading system, and organized your course content, you are ready to publish your course and make it available to students.

Understanding the content your students can view, and knowing how to change what students can view, is complex. Read the following sections carefully:

- *Understanding Content Students Can View*
- *Release Dates*
- *Public and Private Units*
- *Modifying Public Units*

7.2.1 Understanding Content Students Can View

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.

- The **Release Date**. Sections and subsections have release dates. If the current date of a section or subsection is before the release date, the content of that course element is not yet published, and not visible to students.

For a student to view a subsection, both it and its containing section must be have a release date earlier than the current date. It is possible that a section is released, but a subsection within it is not released. In this case, students cannot view that subsection.

Course staff can see sections and subsections before the release date in the LMS.

- The unit must be **Public**. All units have a **Visibility** setting that is **Public or Private**. When you create a unit, it is **Private** by default.

Students cannot view a **Private** unit, even if the containing section and subsection are released.

Students cannot view a **Public** unit if the containing section and subsection are *not* released.

Course staff *cannot* see Private units in the LMS.

In summary, for students to see content, the unit must be **Public**, and the containing section and subsection must be released. If all these criteria are not met, students do not see that unit.

Continue reading this chapter for more details.

7.2.2 Release Dates

Release dates specify the dates when content is available to students. Release dates 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 in the LMS regardless of its release date.

Note: For courses created before January 14, 2014, when you create a new section or subsection, the default release date is January 1, 1970. This means that public units are immediately published, and you must change this date to the future to prevent students from viewing the content.

For courses created after January 14, 2014, the default release date is January 1, 2030. In this case, you must adjust the release date to the day you want students to be able to view the content.

Set the Release Date for a Section

You can set release date and time for each section. Before the release date and time, students are not able to view any content in that section.

To set a release date for a section:

1. In the **Will Release:** field next to the section title, click **Edit**.
2. Enter the release date in MM/DD/YYYY format, and the UTC time.
3. Click **Save**.

Set the Release Date for a Subsection

Subsections inherit the release date of the section they are in, but you can change this date so that different subsections are released at different times.

Note that if the release date for a subsection falls before the release date for the section that contains it, students will not be able to see the subsection until the release date for the *section* has passed. Section release dates override subsection release dates.

To set the release date for a subsection:

1. Open the subsection.
2. Locate the **Subsection Settings** box in the top right corner.
3. Enter the release date in MM/DD/YYYY format, and the UTC time.

Synch the Release Date for a Subsection

You or other course staff could inadvertently set the release date for a subsection earlier than the release date for the containing section. In this situation, the subsection is not released until the section is released.

To help you keep your course and release dates organized, Studio flags subsections with earlier release dates than their containing section. In this situation, when you open the subsection, in the Subsection Settings, you see the following message:

The date above differs from the release date of <Section Name> – <Section release date and time>. Sync to <Section Name>.

Click **Sync to <Section Name>** to have the subsection inherit the later section release date.

7.2.3 Public and Private Units

Units are released at the release date of the subsection they are in.

In addition, units have a **Visibility** setting that you can set to **Public** or **Private**.

When you create a unit, it is Private by default. A Private unit is never visible to students, even if it is contained by a subsection that has been released.

When you change the visibility setting of a unit from Private to Public, you publish the unit and its contents. You must set the Visibility to Public for students to be able to see the unit.

Course staff cannot see Private units in the LMS.

7.2.4 Modifying Public Units

To make revisions to a unit that has been published, you create and edit a draft of that unit.

Warning: There are additional implications to modifying the public unit that has graded problem components students may have already completed. See *Modifying a Released Problem* for more information.

To create a draft, go to the unit's page, and then click **edit a draft** in the right pane.

The screenshot shows the course editor interface for a unit named "Unit 1". The main content area displays the title "The Fruits of Learning" and a quote: "'Tis one of our goals here at edX To circumvent end-of-course headX. We hope that our students". The right sidebar, titled "Unit Settings", shows the visibility is set to "Public". A message box states: "This unit has been published. To make changes, you must edit a draft." Below it, a scheduled release message says: "This unit is scheduled to be released to students on Jan 01, 2010 at 12:00 UTC with the subsection "Subsection 1a"". A "View Live" button is also present.

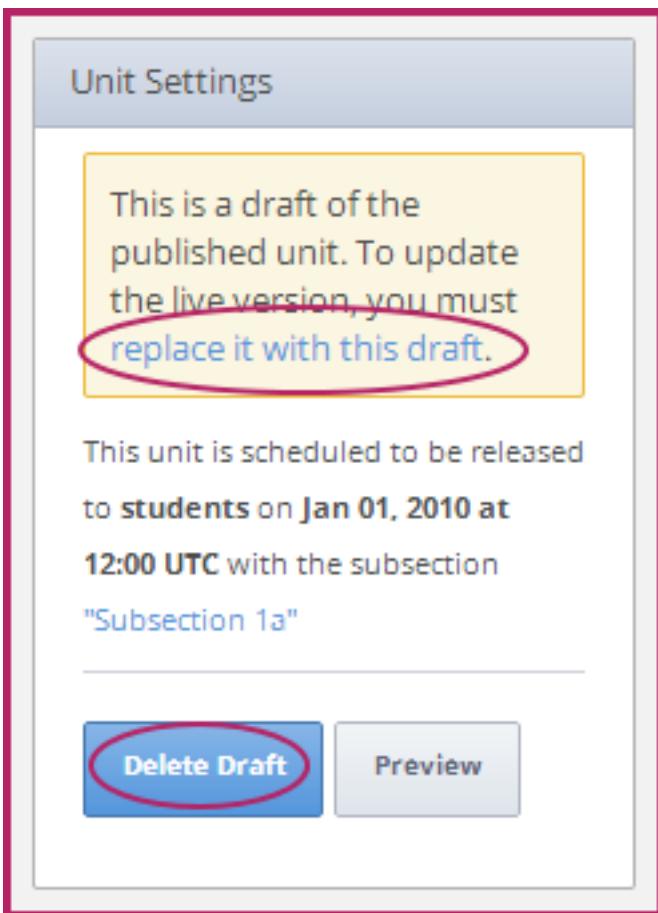
When you edit a draft of a unit, you can view the unit's contents in two ways.

- 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**.

The screenshot shows the course editor interface for a unit draft named "Working with Videos". The main content area displays the title "Building a Computer Memory Element" and a section titled "BUILDING A COMPUTER MEMORY ELEMENT". The right sidebar, titled "Unit Settings", shows a message: "This is a draft of the published unit. To update the live version, you must replace it with this draft." Below it, a scheduled release message says: "This unit is scheduled to be released to students on Jan 01, 1970 at 05:00 UTC with the subsection Lesson 1 - Getting Started". Buttons for "Delete Draft" and "Preview" are shown, with "Preview" being circled.

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**.



Warning: Historical versions of units are not stored by Studio. After you replace the live version with a new draft, you cannot revert the unit to the previous version.

7.3 Beta Testing a Course

When a course is being designed in Studio, the course team can choose **View Live** to assess their progress in designing the course, test their work, find gaps and errors, and mimic the student experience. To gather information about the overall experience of taking a course, you can also conduct a beta test of the course.

Beta testers have early access to the course. Beta testers are not members of the course team or staff: they don't have information about "how it's supposed to work". They use their own computers and internet connections to view videos, follow links, and complete problems. They interact with the course exactly as students will, but see the course materials before other enrolled students do.

- *The Beta Testing Process*
- *Qualities of Good Beta Testers*
- *What to Test*
- *How Beta Testers See Course Content*
- *Add Beta Testers*
- *Add_Testers_Bulk*

- *Issue Reporting During a Course*

7.3.1 The Beta Testing Process

During a beta test, you gather information about the experience of taking your course in an effort to improve its quality. The phases of the beta testing process, and some questions that you may find useful to consider as you prepare for each phase, follow.

1. **Planning:** What are your objectives for the test? What is the schedule for testing?
2. **Recruitment:** How many testers do you need? How much time do you expect testers to spend? When do you need them to complete their feedback?
3. **Provide access to your course:** Can testers access your entire course immediately, or are its sections and subsections available at different times? How will that affect the schedule?
4. **Collect feedback:** How do you want testers to provide feedback? Who checks for feedback, and how often?
5. **Evaluate feedback:** How is feedback evaluated? Who decides what changes to make as a result, and on what schedule?
6. **Conclusion:** How do you acknowledge the efforts of your testers? What do you share with them about the results of their efforts?

There is no one answer to any of these questions. They are included here as background on the role that beta testing can play in the preparation of your course.

7.3.2 Qualities of Good Beta Testers

A beta test is valuable in part because it is unscripted. Your beta testers are not following a predetermined series of steps, or using a particular computer environment, when they access your course. When you recruit beta testers, however, you may find these skills and characteristics to be helpful.

- Curiosity.
- Attention to detail for identifying problems and inconsistencies.
- Solid communication skills for reporting problems to course staff.

Your beta testers should also have varying levels of knowledge about the course content:

- To recognize when material is wrong or missing, at least one tester should know the content well.
- To identify material that is unclear, at least one tester should be less familiar with the content.

Depending on the objectives you set for your beta test, you may want to consider recruiting testers who use assistive technologies, who have different native languages, or who have varying levels of familiarity with computer software.

Using Course Staff as Beta Testers

Course staff can provide valuable feedback about your course. However, they are typically stakeholders in the success of your course and have a significant amount of knowledge about it. As a result, they can be too close to the course to interact with it in the same way as students will. They can also be either reluctant to provide feedback, or overly zealous.

If you do want a staff member to be a beta tester, a different, second email address must be used for this additional role. The privileges of the course staff role override those of a beta tester, so a second registration on the site, activation, and enrollment in the course are necessary using this second email address. The staff member must log in using the second email address (with the beta tester role) in order to experience the course as a student.

7.3.3 What to Test

Beta testers should interact with everything in the course.

- Click all links.
- Watch all videos.
- Download video transcripts and other files.
- Complete all problems.

As they work, beta testers log issues and questions for the course team.

Note: Beta testers can make discussion posts and wiki contributions. You may want to ask your testers not to make posts or contributions about unreleased courseware, and alert your discussion administration team to watch for posts from the beta testers.

7.3.4 How Beta Testers See Course Content

To beta test a course, you:

- Define a number of days before section and subsection release days for testing to begin.
- Identify beta testers.

Designated beta testers see course content before students can. Beta testers access courseware:

	Yes	No
Before the Course Enrollment Date	X	
Before the Course Start Date	X	
Before the section Release Day	X	
Before the subsection Release Day	X	
Before the unit is Public		X
Before a draft replaces a live unit		X

The course team can continue to add content in Studio after the beta test begins. When new content is ready for testing, be sure to change the **Visibility** setting for any units that are Private to Public. See *Units* or *Publishing Your Course*.

Define When the Beta Test Starts

To define the start of the beta test, you use Studio to specify a number of days before the **Release Day** of each section and subsection in your course.

1. In Studio, select **Settings > Advanced Settings**.
2. Scroll down to the `days_early_for_beta` **Policy Key**.
3. Set the **Policy Value** to a number of days before the **Release Day** for the section and subsection.
4. Click **Save Changes**.

Example

Beta tester access to courseware for a course with `days_early_for_beta = 20` and 2 sections:

Course Enrollment Date = 31 August	31 August; earlier if enrolled by course staff
Course Start Date = 15 September	26 August
section 1 Release Day = 15 September	26 August
section 2 Release Day = 22 September	2 September
subsection 1 Release Day = 22 September	2 September
subsection 2 Release Day = 24 September	4 September
subsection 2, unit 1 Visibility = Public	4 September
subsection 2, unit 2 Visibility = Public; draft in progress	4 September for Public version; No access to draft
subsection 2, unit 3 Visibility = Private	No access

In this example, the beta testers cannot access all of the courseware when the beta test starts on 26 August: they can access section 1 but not section 2. You may want to provide a schedule of section availability to your testers so that they can plan their time appropriately. Future release dates do not display to the beta testers in the courseware.

7.3.5 Add Beta Testers

Before you can add beta testers:

- You need the email address or username of each beta tester.
- Each beta tester must register and activate a user account for that email address/username.

When you add beta testers, note the following.

- If the beta test starts before the course **Enrollment Start Date**, testers cannot enroll themselves in your course. However, you can enroll the beta testers prior to the **Enrollment Start Date**.
- If you add the beta testers after the defined start of the beta test, and they are enrolled in the course, they see your course on their dashboards when they log in.
- If you add beta testers before the test starts, or if they are not enrolled, they do not see your course on their dashboards. You can enroll the beta testers in the course yourself, and you may want to send an email message to remind them when the test begins.

If you have a number of beta testers that you want to add, you can use the “batch add” option to add them all at once, rather than individually. With this feature, you have options to enroll the beta testers in the course (before or after the **Enrollment Start Date**) and send an email message to notify the beta testers that they have been added.

To add multiple beta testers:

1. View the live version of your course.
2. Click **Instructor**, and then click **Try New Beta Dashboard**.
3. Click **Membership**.
4. In the **Batch Beta Tester Addition** section of the page, enter one or more addresses or usernames separated by commas or line feeds (the Enter key). You can also copy data from a CSV file of email addresses and paste it here.
5. To enroll the beta testers in your course, leave **Auto Enroll** selected. If the beta test starts before the course **Enrollment Start Date**, testers cannot enroll themselves in your course.
6. To send an email message to the beta testers, leave **Notify users by email** selected. An example of the message that is sent to a beta tester who is not enrolled (or auto enrolled) in the course follows.

Dear twobeta

You have been invited to be a beta tester for [REDACTED] at edge.edx.org by a member of the course staff.

Visit [https://edge.edx.org/courses/\[REDACTED\]/about](https://edge.edx.org/courses/[REDACTED]/about) to join the course and begin the beta test.

6. Click **Add beta testers**.

To remove the Beta Tester role from one or more users, enter their email addresses in the **Batch Add Beta Testers** field and then click **Remove beta testers**.

Note: The **Auto Enroll** option has no effect when you click **Remove beta testers**. The user's role as a beta tester is removed; course enrollment is not affected.

To add a single beta tester:

1. View the live version of your course.
2. Click **Instructor** then **Try New Beta Dashboard**.
3. Click **Membership**.
4. In the **Administration List Management** section, use the drop-down list to select **Beta Testers**.
5. Under the list of users who currently have that role, enter an email address or username and click **Add Beta Tester**.

If the beta test starts before the **Enrollment Start Date** of your course, you can also enroll the beta tester. See *Enrollment*.

To remove the Beta Tester role from users individually, find the user in the list of beta testers, and then click **Revoke access** to the right of that user's email address.

7.3.6 Issue Reporting During a Course

Despite the efforts of the course team and the beta testers, additional problems, questions, and issues can occur while a course is running.

- Often, students report issues by asking questions in a discussion. Your discussion administration team can watch for posts that indicate problems.
- To let students know the best way to report problems, you can post information on the **Course Info** page or post it in a discussion.

7.4 Course Launching Activities

To launch a course, you prepare the course itself, the staff, and the students. This chapter includes a *Course Launch Checklist* to use as a model for your schedule of activities.

To help you communicate to the staff and to all course participants when you are launching the course, and also while the course runs, you can send email messages from the Instructor Dashboard. See *Bulk Email* and *Email Task History Report*.

7.4.1 Course Launch Checklist

As the start date for your course approaches, a checklist or timeline of activities can help you make sure that your course, and your students, are ready to begin. Suggestions for activities to complete before your course starts follow.

Verify Course Settings

- Check the course start date and time in Studio. See *Set Important Dates for Your Course*.

- Review the grading policy, and set a grace period for homework assignment due dates. See *Establishing a Grading Policy*.

Review First Week Content

- Verify that all units are present and published. See *Units*.
- Check all assignments for completeness and verify their due dates. See *Working with Problem Components*.
- Verify that videos, transcripts, and download links are in place and working.
- Review feedback from the course team and beta testers to be sure that the content has been thoroughly reviewed and tested.

Welcome Students

- Two months before the course start date, prepare and send a welcome email message to currently enrolled students. See *Send Email Messages to Course Participants*.
- Compose a welcome message and add it to the **Course Info** page. See *Add a Course Update*.
- Verify that a syllabus and other references are available on the **Course Handouts** page. See *Add Course Handouts*.
- One month before the course start date, prepare and send a welcome email message to currently enrolled students.
- One week before the course start date, prepare and send a welcome email message to currently enrolled students.
- Start an “Introduce Yourself” topic in a discussion thread. For a MOOC, you may want to manage the size of the thread by distributing student responses across multiple threads. For example, you can start different threads for introductions based on geographical location, such as “Introduce Yourself: Europe”, “Introduce Yourself: North America”, etc. See *Run a discussion*.
- On the course start date, prepare and send a launch email message to currently enrolled students.

Prepare Staff

- Define communication methods for all course contributors, including staff, instructors, and the discussion team. For example, set up a course-specific email address.
- Verify that all course contributors know how to record their work, report issues, and collaborate on tasks.
- Verify that the instructors and course staff selected for your course have the correct role assignments in the LMS. See *Staffing*.
- Verify that discussion admins, discussion moderators, and community TAs have registered and activated their user accounts, enrolled in the course, and been assigned their roles. See *Assign Discussion Administration Roles*.
- Define methods for managing discussions and guidance for discussion moderators, and distribute to the discussion team. See *Moderate Discussions* and *Guidance for Discussion Moderators*.

7.4.2 Bulk Email

With the bulk email feature, you can send email messages to course participants directly from the Instructor Dashboard. Messages can use HTML styling, and can include links to videos, social media pages for the course, and other material. All course contributors who are assigned the course staff or instructor role can use this feature to communicate with course participants before, during, and after the course run.

Note: The bulk email feature is currently in limited release, and is enabled for new courses only. A gradual rollout of this feature is planned for 2014.

Message Addressing

When you send an email message from the Instructor Dashboard, you choose its recipients by selecting one of these predefined groups:

- **Myself**, to test out a message before sending it to a larger group.
- **Staff and Instructors**, to contact other members of the administrative team.
- **All (students, staff and instructors)**, to communicate with currently enrolled students and the administrative team.

Note: Students can opt not to receive email messages through the **Email Settings** link for each course on their dashboards. Email messages are not sent to these students.

When you use the bulk email feature, consider that messages **cannot be cancelled** after they are sent. Before you send a message to all course participants, be sure to review each draft carefully, and send the message to yourself for thorough testing.

Send Email Messages to Course Participants

To send an email message to course participants:

1. View the live version of your course.
2. Click **Instructor > Try New Beta Dashboard**.
3. Click **Email**.
4. Select who you want to send the message to from the **Send to** dropdown list. You can select:
 - **Myself**
 - **Staff and Instructors**
 - **All (students, staff and instructors)**
5. Enter a **Subject** for the message. A subject is required.
6. Enter the the text for the message. Messages can use HTML styling, including text formatting and links. The email message editor offers the same formatting options as the HTML component editor in Studio. See *Working with HTML Components*.
7. Click **Send Email**. The status of the message displays in the **Pending Instructor Tasks** section of the page.

Message Queueing

When you send a message, it is queued for processing as a bulk email task. Multiple courses use the same queue to complete these tasks, so it can take some time for your message to be sent to all of its recipients. If your course is a MOOC, consider limiting the number of messages that you send to all course participants to no more than one per week.

On the **Email** page, the **Pending Instructor Tasks** section shows the status of queued messages.

PENDING INSTRUCTOR TASKS

Email actions run in the background. The status for any active tasks - including email tasks - appears in a table below.

Task Type	Task inputs	Task ID	Requester	Submitted	Duration (sec)	State	Task Status	Task Progress
bulk_course_email	{"email_id": 2, "to_option": "myself"}	e7321ef6-8be6-4f29-94a3-a5b5e7eb5f4c		2014-03-17T14:55:37+00:00	0	PROGRESS	Incomplete	Progress: emailed 0 of 0 so far (out of 1)

You can perform other tasks on the Instructor Dashboard or navigate to other pages while you wait for your message to be sent.

7.4.3 Email Task History Report

You can produce a report of all of the bulk email tasks sent for your course. For each message sent, the report includes the username of the requester, the date and time it was submitted, the duration and state of the entire task, the task status, and the task progress.

You can use this history to investigate questions relating to the bulk email message that have been sent, such as:

- How frequently students are sent course-related email messages.
- Whether a message was sent successfully.
- The number of people who were sent course-related messages over time.

Review the Email Task History

To produce the Email Task History report:

1. View the live version of your course.
2. Click **Instructor > Try New Beta Dashboard**.
3. Click **Email**.
4. In the **Email Task History** section of the page, click **Show Email Task History**. A report like the following example displays on the Instructor Dashboard.

EMAIL TASK HISTORY

To see the status for all bulk email tasks ever submitted for this course, click on this button:

Show Email Task History								
Task Type	Task inputs	Task ID	Requester	Submitted	Duration (sec)	State	Task Status	Task Progress
bulk_course_email	{"email_id": 2, "to_option": "myself"}	e7321ef6-8be6-4f29-94a3-a5b5e7eb5f4c		2014-03-17T14:55:37+00:00	6	SUCCESS	Complete	Message successfully emailed for 1 recipients

For messages with a State of “Success”, the Task Progress can show an informational message such as “Message successfully emailed for 13457 recipients (skipping 29) (out of 13486)”. To interpret this message, note that:

- The first number indicates the number of messages sent to the selected recipients.
- The second “skipping” number indicates the number of enrolled users who have opted out of receiving course email messages on their dashboards.

- The final “out of” number indicates the number of users in the set of selected recipients who were enrolled in the course when you sent the email message. For email messages addressed to a large number of users, the number of enrolled students can change while the message is queued for processing.

7.5 Staff Debug Info

The edX system keeps track of students’ progress through a course – recording when the student watches videos, responds to problems, and so on. If you are a staff member on a course, some of that data is visible to you for debugging purposes. Under every problem is a **Staff Debug Info** button: clicking this button opens a popup with metadata about the problem.

None of this information should be necessary for day-to-day usage of edX, but for the sake of clarity, some of these fields are documented here:

is_released Indicates whether the problem is visible to students.

location An internal unique identifier that corresponds to this problem. If you are having trouble with a problem, and need assistance from the edX support team, including this value will make it easier for them to track down the issue you’re having with the problem.

markdown The text of the problem, in Markdown format. This is often written using Studio.

display_name The name of the problem, as shown to the student.

max_attempts The maximum number of times that a student can attempt to answer the problem correctly.

attempts The number of times that the currently logged in student has attempted to answer the problem correctly, so far. Every time this student attempts to answer this question, this number will go up, until it reaches `max_attempts`.

7.6 Time Zones

Overview Released course materials become visible to all students at once, and assignments with a due date will be due for all students at once, at the time specified by the setting. However, a number of places on edX and Studio present a time setting without specifying a time zone. Unless specified otherwise, most dates and times in Studio and edX are in UTC, not in your local time zone! When you specify date and time settings that do not have a time zone label, you need to convert values to UTC. You should also ensure that students and instructors know how to interpret time settings for your course.

Details Time is stored in a specific time zone. However, this time zone may not be visible in the interface. Unlabeled time values are specified, stored, and viewed in UTC.

EdX and Studio handle time zones as follows.

- All times, labeled and unlabeled, are saved to the server in UTC (a.k.a. UTC or Z).
- Unlabeled times are displayed both in Studio and on edX/Edge in UTC.
- Times labeled with a particular time zone in Studio are specified in that time zone, and are converted to UTC. (This is rare.)

For settings in Studio that are labeled with a time zone, such as the course start and end dates, enter the setting in the time zone specified (usually your local time zone).

For time settings that are not labeled with a time zone, such as release dates and due dates for course content, convert from your local time zone, and set the dates and time settings in UTC. You can use an online time zone converter to convert from your local time zone.

Note When you use an online converter, enter both the day and the time to account for daylight saving time.

Example US Eastern Standard Time is “UTC-5”, so a New York winter due date of 5:00pm (17:00) should be entered as 10:00pm (or 22:00) in Studio. US Daylight Saving Time, however, is UTC-4, so a New York summer due date of 5:00pm would be entered as 9:00pm in Studio.

Most of these time settings are also not labeled in the Student view on edX/Edge. When you set due dates for an assignment, make sure to tell students how to interpret the due date. You can choose one of the following options.

- Notify students in advance that all times, unless otherwise labeled, are displayed in UTC, and point them to a time zone converter to convert to their local time zone.
- Allow students to assume that all due dates are specified in their local time zone, and specify an unadvertised grace period to invisibly extend all the due dates in your course. For example, some courses set a grace period of “1 day, 6 hours, and 1 minute” to accommodate differences in time zones and any potential system issues.

Note Setting a grace period is generally not recommended. It can lead to problems not closing “when they should”, and may be misleading to your students.

If you have further questions about specifying times and time zones, or are experiencing inconsistencies in due date or release date behavior, please contact us from the edX Studio Help page.

References

<http://help.edge.edx.org/discussions/questions/61-time-zones>

<http://help.edge.edx.org/discussions/questions/23-grace-periods>

Running Your Course

8.1 Staffing

You can designate a team of people to help you run your course. Two roles are available to give your administrative team members access to different options for working with students, grades, and other members of the staff.

- Course Staff
- Instructors

The administrative team that helps you run your course in the LMS can include some, all, or none of the people who help you set up the course in Studio. You assign these administrative roles in the LMS, separately from the roles you assign in Studio. For more information on setting up a team in Studio, see *Add Course Team Members*.

You can also designate teams of people to beta test your course and to moderate and manage its discussions by assigning other LMS roles. The beta testers and discussion administrators must be enrolled in your course, but they do not need to have the Course Staff or Instructors role or a Studio role. For more information, see *Beta Testing a Course* and *Assign Discussion Administration Roles*.

8.1.1 Administrative Team Roles

To provide access to features on the LMS Instructor Dashboard you assign the Course Staff role or the Instructors role.

Course Staff can:

- View the course before the Course Start Date.
- Enroll and unenroll students.
- Access student grades.
- Reset student attempts to answer a question correctly.
- See course HTML errors.
- Send email messages to course participants.

Instructors have access to all of the same options for running the course as the course staff. They can also:

- Add and remove Course Staff.
- Add and remove other Instructors.
- Add and remove Beta Testers.
- Add and remove Discussion Admins, Discussion Moderators, and Discussion Community TAs.

8.1.2 Assign Staff Roles

Before you can assign the Course Staff or Instructors role to a team member:

- You need the email address or username of each team member.
- Each team member must register a user account for that email address/username, activate the account, and enroll in your course.

To assign a staff role:

1. View the live version of your course.
2. Click **Instructor** then **Try New Beta Dashboard**.
3. Click **Membership**.
4. In the **Administration List Management** section, use the drop-down list to select **Course Staff or Instructors**.
5. Under the list of users who currently have that role, enter an email address or username and click **Add** for the role type.

To remove an assigned role, view the list of users and then click **Revoke access**.

8.2 Enrollment

Course authors and instructors can enroll students in a course, see how many people are enrolled, and, when necessary, unenroll students.

Students can enroll themselves in a course during its defined enrollment period. For a www.edx.org course, enrollment is publicly available to anyone who registers an edX account. For other courses, such as those on edge.edx.org, enrollment is limited to students who know the course URL and students you explicitly enroll.

- *Registration and Enrollment*
- *Enroll Students in a Course*
- *View an Enrollment Count*
- *Unenroll Students from a Course*

8.2.1 Registration and Enrollment

Before a student can enroll in a course, he or she must:

1. Register a user account, which includes supplying a valid email address, on www.edx.org, edge.edx.org, or your implementation of the edX platform. Each platform requires a separate user account.
2. Activate the registered account by following the emailed instructions.

As long as the course **Enrollment End Date** has not passed, students who have registered and activated user accounts can enroll themselves in www.edx.org courses, or can enroll in other courses if they know the URL. For a more detailed description of this process from a student's point of view, see *Sample Student Login Guide*.

Course authors and instructors, however, can enroll students in a course either before or after the students register their user accounts.

To work on a course, all course staff members must also have registered and activated user accounts and be enrolled in the course.

8.2.2 Enroll Students in a Course

You enroll students, and other course staff members, in your course by supplying their email addresses. After the **Enrollment End Date** for a course students can no longer enroll themselves; however, you can still explicitly enroll students.

When you enroll people in a course you have these options:

- **Auto Enroll.** When you choose this option, the people who you enroll do not need to complete an explicit course enrollment step. Of the list of email addresses that you supply, those that correspond to a registered user account are immediately enrolled in the course, and your course displays on the dashboards on log in. Email addresses on the list that do not match a registered user account are enrolled as soon as that account is registered and activated.

If you do not select **Auto Enroll**, the people who you enroll must also actively locate your course and enroll themselves in it. These students see the course on their dashboards after they have done so.

- **Notify students by email.** When you choose this option, an email message is automatically sent to each of the email addresses that you supply. The message includes the name of the course and, for students who are not already registered, a reminder to use that same email address to register.

An example of the email message that a student received when this option was selected during enrollment follows. In this example, the student already had a registered and activated edx.org account, and both **Auto Enroll** and **Notify students by email** were selected.

The screenshot shows an email inbox with one message. The subject line is "You have been enrolled in Introduction to Computer Science and Programming". The message is from "no-reply@registration.edx.org via amazonses.com" and was sent "to me" at "10:36 AM (4 minutes ago)". The message content is as follows:

Dear [redacted]

You have been enrolled in Introduction to Computer Science and Programming at courses.edx.org by a member of the course staff. The course should now appear on your courses.edx.org dashboard.

To start accessing course materials, please visit https://courses.edx.org/courses/MITx/6.00x/2013_Spring/

This email was automatically sent from courses.edx.org to [redacted]

To enroll students or staff members:

1. View the live version of your course.
2. Click **Instructor** then **Try New Beta Dashboard**.
3. Click **Membership**.
4. In the **Batch Enrollment** section of the page, enter an email address or multiple addresses separated by commas or line feeds.

You can copy and paste data from a CSV file of email addresses. However, note that this feature is better suited to courses with smaller enrollments, rather than courses with massive enrollments.

5. To streamline the course enrollment process, leave **Auto Enroll** selected.
6. To send students an email message, leave **Notify students by email** selected.
7. Click **Enroll**.

8.2.3 View an Enrollment Count

After you create a course, you can access the total number of people who are enrolled in it. When you view an enrollment count, note that:

- In addition to students, the enrollment count includes the course author, course team members, instructors, and course staff. (To work with a course in Studio or the LMS, you must be enrolled in that course.)
- Students can unenroll from courses, and course authors and instructors can unenroll students when necessary.

Note: The enrollment count displays the number of currently enrolled students and course team staff. It is not a historical count of everyone who has ever enrolled in the course.

To view the enrollment count for a course:

1. View the live version of your course.
2. Click **Instructor** then **Try New Beta Dashboard**.
3. Click **Course Info** if necessary.

The **Enrollment Information** section of the page that opens shows the total number of people who are currently enrolled.

You can also view or download a list of the people who are enrolled in the course. See *Student Data*.

8.2.4 Unenroll Students from a Course

You can remove students from a course by unenrolling them. To prevent students from re-enrolling, course enrollment must also be closed. You use Studio to set the **Enrollment End Date** for the course to a date in the past. See *Set Important Dates for Your Course*.

Note: Unenrollment does not delete data for a student. An unenrolled student's state remains in the database and is reinstated if the student does re-enroll.

To unenroll students, you supply the email addresses of enrolled students.

1. View the live version of your course.
2. Click **Membership**.
3. In the **Batch Enrollment** section of the page, enter an email address or multiple addresses separated by commas or line feeds.
4. To send students an email message, leave **Notify students by email** selected.

Note: The **Auto Enroll** option has no effect when you click **Unenroll**.

5. Click **Unenroll**. The course is no longer listed on the students' dashboards, and the students can no longer contribute to discussions or the wiki or access the courseware.

8.3 Discussions

Discussions, or discussion forums, foster interaction among your students and between students and course staff. You set up discussion topics and categories when you create your course, and then run and moderate discussions throughout the course to guide participation and develop course community.

Discussions are also excellent sources of feedback and ideas for the future.

For options you can use to run and moderate discussions, see the following topics:

- Set Up Discussions for Your Course
- Run a discussion
- Moderate Discussions
- Close Discussions to New Posts

8.3.1 Set Up Discussions for Your Course

Discussions in an edX course include both the specific topics that you add to course units as discussion components, and broader forums on course-wide areas of interest, such as Feedback, Troubleshooting, or Technical Help, that you can add as discussion categories.

Add Units With a Discussion Component

Typically, all units are added during the design and creation of your course in Studio. To add a component to a unit, follow the instructions in *Working with Discussion Components*.

This type of discussion is subject to the release date of the section that contains it. Students cannot contribute to these discussions until that date.

Create Discussion Categories

All courses include a page named **Discussion**. When you create a course, a discussion category named General is available for you to include by default. You can add more discussion categories to guide how students share and find information during your course. Categories might include Feedback, Troubleshooting, or Technical Help. Discussions in these categories can begin as soon as your course is available.

To create a discussion category:

1. Open your course in Studio.
2. Select **Settings > Advanced Settings**.
3. Scroll down to the **Policy Key** for **discussion_topics**. By default, its **Policy Value** is:

Policy Key:	Policy Value:
discussion_topics	<pre>{ "General": { "id": "i4x-test_doc-SB101-course-2014_Jan" } }</pre>

4. Copy the three lines provided for the General discussion category and paste them above the closing brace:

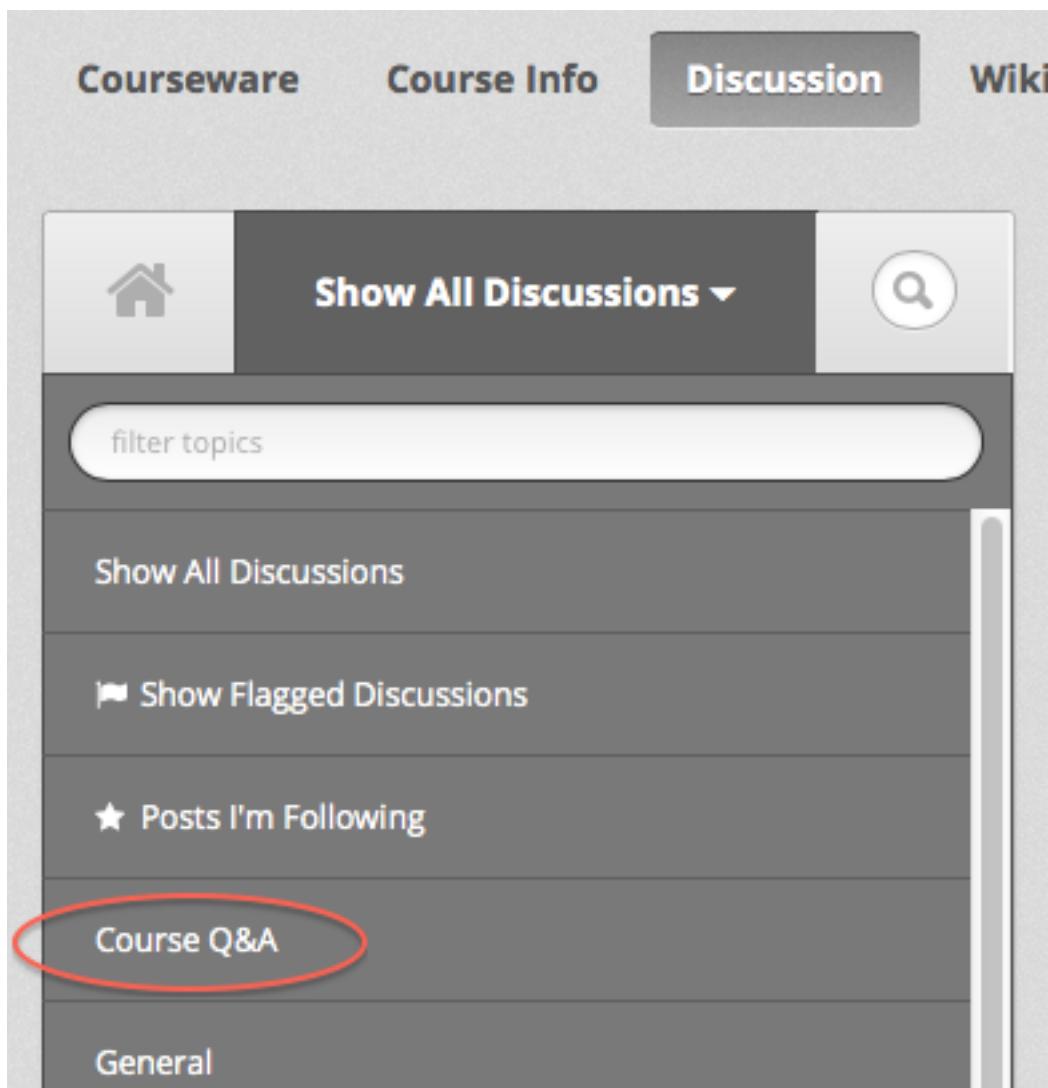
Policy Key:	Policy Value:
discussion_topics	<pre>{ "General": { "id": "i4x-test_doc-SB101-course-2014_Jan" } "General": { "id": "i4x-test_doc-SB101-course-2014_Jan" } }</pre>

5. Replace the second “General” with the quoted name of your new discussion category.
6. Change the second id value to a unique identifier. For example, append a reference to the category name.
7. Add a comma after the first closing brace.

Policy Key:	Policy Value:
discussion_topics	<pre>{ "General": { "id": "i4x-test_doc-SB101-course-2014_jan" }, "Course Q&A": { "id": "i4x-test_doc-SB101-course-2014_jan_faq" } }</pre>

8. Click **Save Changes**.

When students click the **Discussion** page for your course, the drop-down list includes this new category.



Assign Discussion Administration Roles

You can designate a team of people to help you run course discussions. Different options for working with discussion posts are available to discussion administrators with these roles:

- Discussion moderators can edit and delete posts, review posts flagged for misuse, close and reopen threads, pin posts and endorse responses, and, if the course is cohorted, see posts from all cohorts. Responses and comments made by moderators are marked as “Staff”.
- Discussion community TAs have the same options for working with discussions as moderators. Responses and comments made by community TAs are marked as “Community TA”.
- Discussion admins have the same options for working with discussions as moderators. Admins can also assign these discussion management roles to more people while your course is running, or remove a role from a user whenever necessary. Responses and comments made by admins are marked as “Staff”.

Note: Discussion responses and comments made by course staff and instructors are also marked as “Staff”.

Before you can assign roles to your discussion administrators, you need their email addresses or usernames.

- To get this information for a staff member, on the Instructor Dashboard click **Membership** and then select **Course Staff** from the drop-down list.
- To get this information for an enrolled student, on the Instructor Dashboard click **Data Download > Download profile information as a CSV**.

Tip: These instructions are for the new Instructor Dashboard: click **Try New Beta Dashboard**.

To assign a role:

1. View the live version of your course.
2. Click **Instructor** then **Try New Beta Dashboard**.
3. Click **Membership**.
4. In the Administration List Management section, use the drop-down list to select Discussion Admins, Discussion Moderators, or Discussion Community TAs.
5. Under the list of users who currently have that role, enter an email address or username and click **Add** for the role type.
6. To remove an assigned role, view the list of users and then click **Revoke access**.

8.3.2 Run a discussion

On an ongoing basis, discussion administrators run the course discussions by making contributions and guiding student posts into threads. Techniques that you can use throughout your course to make discussions successful follow.

Seed a Discussion

Before you contribute to a discussion, you can decide whether you want to be identified as a staff member or community TA, or to appear like other students’ work. Depending on the subject and your purpose, one or the other might be more appropriate to spark discussion and inform students.

You can also post anonymously. Regardless of your role, you can choose to make a post anonymous. However, you may want to discourage your students from posting anonymously, and therefore choose not to use this option yourself.

- To identify your posts with your role, log in with your discussion administrator email address and add the post or response. The responses and comments that you make include a colored banner with either “Staff” or “Community TA”.
- To post as a student, you must set up an alternate account with a different email address, go to the course URL and register, and then join the discussion. When you post as a student, your responses and comments do not have a banner or other distinguishing features: they appear the same as any other student post.

Note: Posts by discussion administrators do not include a colored “Staff” or “Community TA” banner. The colored banner appears only when discussion administrators respond to existing posts or make a comment.

Use Conventions in Discussion Subjects

To identify certain types of posts and make them easier for your students to find, you can define a set of standard tags to include at the beginning of the subject. Examples follow.

- Use “[OFFICIAL]” at the start of announcements about changes to the course.
- Post information about corrected errors with a subject that begins “[ERRATA]”.
- Direct students to use “[STAFF]” in the subject of each post that needs the attention of a course staff member.

Minimize Thread Proliferation

To encourage longer, threaded discussions rather than many similar, separate posts, discussion administrators can use these techniques. However, be aware that long threads (with more than a 200 responses and comments) can be difficult to read and slow to display, and can therefore result in an unsatisfactory experience in the discussion.

- Pin a post. Pinning a post makes it appear first in the discussion, so that it is more likely that students will see and respond to it. Otherwise, each post is listed in reverse chronological order or sorted as each student chooses. You can write your own post and then pin it, or pin a post by any author. Click **Pin Thread**.

Discussion guidelines

+ 0

A
less than a minute ago

Please follow these guidelines to make the discussions useful and build a positive community.

- Before asking a question, check to see if someone else has already done so.
- Write a concise, descriptive title for your post, then provide enough information for others to understand the question or problem.
- "Upvote" good posts: click on the green "plus" button so that good posts and/or responses can be found more easily.
- Include [STAFF] at the beginning of a post title to notify the course staff of your question or problem.
- Asking for or posting answers to assignments violates the honor code. These posts will be edited or deleted, and students who use the discussions in this way may be unenrolled from the course.

 Edit  Delete  Close

 Pin Thread  Report Misuse

- Endorse a response. Endorsing a response indicates that it provides value to the discussion, such as a correct answer to a question. Click the **check mark** that displays at upper right of the response.



My experience has been that reading the text first helps me understand the main point, while the videos present a broader range of diverse views to expand on it. If I only watch the videos, I tend to lose sight of the course-specific material and don't do as well on the quizzes.

Report Misuse

- Close a thread. You can respond to a redundant post or response by pasting in a link to the thread that you prefer students to contribute to, and then prevent further thread interaction by closing the entire post or a specific response. Click the **Close** button that displays below the post or response to close it.
- Provide post/response/comment guidelines. A set of *Guidance for Discussion Moderators* or a post in the General discussion can provide guidance about when to create a new thread, respond to an existing post, or comment on a response.

8.3.3 Moderate Discussions

Discussion administrators monitor discussions and keep them productive. They can also collect information, such as areas of particular confusion or interest, and relay it to the course staff.

Developing and sustaining a positive discussion culture requires that sufficient moderator time is dedicated to reviewing and responding to discussions. Keeping up-to-date with a large MOOC forum requires a commitment of 5 or more hours per week, and involves reading posts, replying to and editing posts, and communicating with the other discussion administrators and course staff.

For information on setting up moderators for your course, see *Assign Discussion Administration Roles*.

Provide Guidelines for Students

You can develop a set of best practices for discussion participation and make them available to students as a course handout file or on a defined page in your course. These guidelines can define your expectations and optionally introduce features of edX discussions.

Develop a Positive Discussion Culture

Monitors can cultivate qualities in their own discussion interactions to make their influence positive and their time productive.

- Encourage quality posts: thank students whose posts have a positive impact and who answer questions.
- Check links, images, and videos in addition to the text that students post. Edit offensive or inappropriate posts quickly, and explain why.
- Review posts with a large number of votes and recognize “star posters” publicly and regularly.
- Stay on topic yourself: before responding to a post, be sure to read it completely.
- Maintain a positive attitude. Acknowledge problems and errors without assigning blame.

- Provide timely responses. More time needs to be scheduled for answering discussion questions when deadlines for homework, quizzes, and other milestones approach.
- Discourage redundancy: before responding to a post search for similar posts. Make your response in the most pertinent or active thread, then use links to direct other posts to that thread.
- Publicize issues raised in the discussions: add questions and their answers to an FAQ discussion category, or announce them on the Course Info page.

For a template that you can use to develop guidelines for your course moderators, see *Guidance for Discussion Moderators*.

Edit Posts

Posts and responses can be edited by discussion moderators, community TAs, and admins. Posts that include spoilers or solutions, or that contain inappropriate or off-topic material, should be edited quickly to remove text, images, or links.

1. Log in to the course with your discussion administrator username.
2. Click the **Edit** button below the post or response.
3. Remove the problematic portion of the post, or replace it with standard text such as “[REMOVED BY MODERATOR]”.
4. Communicate the reason for your change. For example, “Posting a solution violates the honor code.”

Delete Posts

Posts and responses can be deleted by discussion moderators, community TAs, and admins. Posts that include spam or abusive language may need to be deleted, rather than edited.

1. Log in to the course with your discussion administrator username.
2. Click the **Delete** button below the post or response.
3. Click **OK** to confirm the deletion.

Important: If a post is threatening or indicates serious harmful intent, contact campus security at your institution. Report the incident before taking any other action.

Respond to Reports of Misuse

Students can use the **Report Misuse** flag to indicate posts that they find inappropriate. Moderators, community TAs, and admins can check for posts that have been flagged in this way and edit or delete them as needed.

1. View the live version of your course and click **Discussion** at the top of the page.
2. On the drop-down list of discussion topics click **Show Flagged Discussions**.
3. Review each post listed as a flagged discussion. Posts and responses show a flag and **Misuse Reported** in red font; comments show only a red flag.
4. Edit or delete the post. Alternatively, leave the post unchanged and click **Misuse Reported** or the flag to remove the notification.

Block Users

For a student who continues to misuse the course discussions, you can unenroll the student from the course. See *Unenroll Students from a Course*. If the enrollment period for the course is over, the student cannot re-enroll.

8.3.4 Close Discussions to New Posts

You can close the discussions for your course so that students cannot add posts. Course discussions can be closed temporarily, such as during an exam period, or permanently, such as when a course ends.

When you close the discussions for a course, all of the discussion components in course units, and all of the discussion categories on the **Discussion** page, are affected.

- Existing discussion contributions remain available for review.
- Students cannot add posts, respond to posts, or comment on responses. However, students can continue to vote on existing posts, report posts for misuse, and follow posts.
- Course Staff, Instructors, Discussion Admins, Discussion Moderators, and Discussion Community TAs are not affected when you close the discussions for a course. Users with these roles can continue to add to discussions.

Note: To assure that your students understand why they cannot add to discussions, you can add the dates that discussions are closed to the **Course Info** page and post them to a General discussion.

Start-End Date Format Specification

To close course discussions, you supply a start date and time and an end date and time in Studio. You enter the values in this format:

["YYYY-MM-DDTHH:MM", "YYYY-MM-DDTHH:MM"]

where:

- The dates and times that you enter are in the Universal Coordinated (UTC) time zone, not in your local time zone.
- You enter an actual letter **T** between the numeric date and time values.
- The first date and time indicate when you want course discussions to close.
- The second date and time indicate when you want course discussions to reopen. If you do not want the discussions to reopen, enter a date that is far in the future.
- Quotation marks enclose each date-time value.
- A comma and a space separate the start date-time from the end date-time.
- Square brackets enclose the start-end value pair.
- You can supply more than one complete start and end value pair. A comma and a space separate each pair.

For example, to close course discussions temporarily for a final exam period in July, and then permanently on 9 August 2014, you enter:

["2014-07-22T08:00", "2014-07-25T18:00"], ["2014-08-09T00:00", "2099-08-09T00:00"]

You enter these values between an additional pair of square brackets which are supplied for you in Studio.

Define When Discussions Are Closed to Posts

To define when discussions are closed to new posts, and when they reopen:

1. Open your course in Studio.
2. Select **Settings > Advanced Settings**.
3. Scroll down to the **Policy Key** for **discussion_blackouts**.
4. In the **Policy Value** field, place your cursor between the supplied square brackets. Use the required date format specification to enter the start and end dates for each time period during which you want discussions to be closed to new posts.

When you enter the dates and times from the example above, the **Policy Value** field looks like this:

Policy Key:	Policy Value:
discussion_blackouts	<pre>[["2014-07-22T08:00", "2014-07-25T18:00"], ["2014-08-09T00:00"]]</pre>

5. Click **Save Changes**.

Studio reformats your entry to add line feeds and indentation, like this:

Policy Key:	Policy Value:
discussion_blackouts	<pre>[["2014-07-22T08:00", "2014-07-25T18:00"], ["2014-08-09T00:00", "2099-08-09T00:00"]]</pre>

8.4 Guidance for Discussion Moderators

Discussion forums are hugely important tools in running a successful MOOC; they allow for substantive community development, in addition to being excellent sources of feedback and ideas for future iterations of the course.

Moderators are the key to effectively managing these online communities. Moderators keep the discussions productive and relay important information (errata, student confusion or interest with particular topics, etc.) to the rest of the course staff. Discussions can be moderated by any of a number of members of the course team, but dedicating enough time to moderation is the best way to cultivate a successful discussion culture.

Please feel free to use some or all of the information in this chapter to guide the contributions of your discussion moderators.

8.4.1 Responsibilities

- Answer basic questions posed by students, and direct students with questions to the right resources: syllabi, course documents, course updates, useful lecture segments, example problems, etc.

- Relay reports of errata, common misconceptions and questions, highly disruptive participants, bugs, and amusing or interesting posts to the most appropriate course staff.
- Enforce the Honor Code by editing or deleting problem answers, or requests for answers, promptly.
- Edit out offensive content from the discussion, and remind the originators of discussion etiquette and expectations.
- Communicate problems and successes to your fellow moderators.
- Identify students whose presence in the discussions has a strongly positive impact. These students can be promoted to community TA role and publicly acknowledged.
- Add helpful items to the FAQ or Course Info page.

8.4.2 Qualities of good discussion moderators

- Good or great content knowledge: stellar students from previous years often make good moderators.
- Solid communication skills: the ability to organize positive, consistent, and effective communication with students, other moderators, and course staff.
- Time: keeping up-to-date with the discussions for a large MOOC 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: this is the best predictor of moderator excellence!

8.4.3 Best practices for discussion moderation

- Always maintain a positive attitude. Keeping a positive attitude is crucial to encouraging participation in the discussion community.
- Encourage discussion between students. Actively thank students who answer the questions of others.
- Use the course's FAQ and Course Info pages as resources. Provide links to these pages in your responses when appropriate. Suggest that information be added to these resources when necessary to respond to a common question or fix confusion.
- Always make it obvious that you have read the student's question. When posting a response, make sure that you are on topic. Respond in the context of the thread.
- Develop a discussion persona. Try to answer your questions in your own slightly unique way.

8.4.4 Guidelines for specific types of posts

Certain types of posts require more attention from the moderators than others, or may need to be handled in a particular way.

Time sensitive problems

- Try to be present on discussion threads when assignment due dates are approaching or new content is being released. The discussions tend to be extra busy at these times.
- Please alert the course staff about problems that need to be dealt with quickly, such as problems with a graded assignment. Setting up a course email address that is checked frequently is a good way to manage these alerts.
- Content Questions

- Assist with content questions sensitively, but be careful not to post spoilers. Do not ask students to post their solutions!
- A good guiding question can be better than an answer.

Redundant posts

- When possible, help discourage redundancy by responding to such posts with links to an earlier or higher quality thread that asks the same question.
- When responding to a post, search for similar posts and respond to the most pertinent thread. Redirect the other posts to the thread with your response and then close the redundant threads.

Off topic, inappropriate, or offensive posts

- Don't simply delete them; instead, edit and explain why they were edited. Inappropriate posts include spoilers, solutions, and information on how to pirate educational materials.
- Check links that students post. If offensive sites and materials are found, they need to be edited quickly.

Bugs or errors

- Check if there is in fact an error.
- If not, suggest to the students that they check their work.
- If so, contact course staff, and notify the thread that the error has been reported.
- Use language that does not assign blame or discourage users from the platform.

You might say: "Thanks for letting us know about that issue. We are working with edX to get it fixed as quickly as possible."

Feature requests

- Keep an organized collection of feature requests cultivated by the course team. That list can subsequently be shared with an edX product manager, who will log those requests internally.
- Reply to the post to let the person know that their request was heard, without promising that the feature will be implemented.

You might say: "Thanks for your suggestion. I've logged it for review by edX staff, who will prioritize feature requests on the development roadmap."

- Search the discussions for other similar requests, and respond to and close those as well.

8.5 Course Data

After you create a course, you can access information about it from the Instructor Dashboard. You can find:

- Identifying information for the course.
- Whether the course has started or ended.
- The defined grade cutoff for passing or failing.

8.5.1 Review Course Data

To view course data:

1. View the live version of your course.
2. Click **Instructor** then **Try New Beta Dashboard**.
3. Click **Course Info** if necessary.

The **Basic Course Information** section of the page that opens lists information about the course.

BASIC COURSE INFORMATION

- Organization: **edX**
- Course Number: **edx204**
- Course Name: **2014_Q2**
- Course Display Name: **Introduction to Data Analytics**
- Has the course started? **No**
- Has the course ended? **No**
- Grade Cutoffs: **Pass: 0.5**

8.5.2 Sources in Studio of the Basic Course Information

The course data that displays on the Instructor Dashboard is defined in Studio, or derived from data that you define in Studio.

- **Organization:** Specified in Studio when you create the course. Becomes part of the course URL, and cannot be changed.
- **Course Number:** Specified in Studio when you create the course. Becomes part of the course URL, and cannot be changed.
- **Course Name:** Specified in Studio when you create the course. Becomes part of the course URL, and cannot be changed. In Studio, this field is labeled **Course Run**.
- **Course Display Name:** Specified in Studio when you create the course. In Studio, this field is labeled **Course Name**.

This name can be changed in Studio (not recommended if your course is live): select **Settings > Advanced Settings** and scroll down to the policy key labeled **display_name**. This setting changes the **Course Display Name** in the LMS only.

The illustration that follows shows the information collected by Studio for a new course side-by-side with the same information in the **Basic Course Information** section of the Instructor Dashboard.

Create a New Course (Studio)

Course Name *

The public display name for your course.

Organization *

The name of the organization sponsoring the course. Note: This is part of your course URL, so no spaces or special characters are allowed. This cannot be changed, but you can set a course run.

Course Number *

The unique number that identifies your course within your organization. Note: This is part of your course URL, so no spaces or special characters are allowed and it cannot be changed.

Course Run *

The term in which your course will run. Note: This is part of your course URL, so no spaces or special characters are allowed and it cannot be changed.

BASIC COURSE INFORMATION (LMS)

- Organization: **edX**
- Course Number: **edx204**
- Course Name: **2014_Q2**
- Course Display Name: **Introduction to Data Analytics**
- Has the course started? **No**
- Has the course ended? **No**
- Grade Cutoffs: **Pass: 0.5**

- Has the course started:** Derived from the **Course Start Date** and the current date. This date can be changed in Studio (not recommended if your course is live): select **Settings > Schedule & Details**.
- Has the course ended:** Derived from the **Course End Date** and the current date. This date can be changed in Studio (not recommended if your course is live): select **Settings > Schedule & Details**.
- Grade Cutoffs:** Specified in Studio when you define the cutoff for a failing grade. Students who earn exactly the cutoff value pass the course. Grading can be changed in Studio (not recommended if your course is live): select **Settings > Grading**.

8.6 Student Data

You can access data about the students who are enrolled in your course at any time after you create the course.

For information about the data you can access, see the following topics:

- Guidance for working with personal information*
- Access student data*
- Access anonymized student IDs*

You can also view charts of certain student demographics for graded problems. See *Grade and Answer Data*.

8.6.1 Guidance for working with personal information

The information that edX collects from site registrants includes personal information that can be used to identify, contact, and locate individuals. This information is available to course authors for the students who are enrolled in their courses.

Course authors should follow the policies established by their organizations and comply with the legal requirements of their locales to prevent public distribution or misuse of this information.

8.6.2 Access student data

You can download data about the students who are currently enrolled in your course in a CSV (comma-separated values) file. For courses that have fewer than 200 students enrolled, you can also view data for enrolled students on the Instructor Dashboard.

Student-reported data

When students register with edX, they select a public username and supply information about themselves. Most of this information is optional, so not all of the students who are enrolled in your course provide it.

Sign Up for edX

E-mail *

e.g. yourname@domain.com

Password *

.....

Public Username *

e.g. yourname (shown on forums)

Full Name *

e.g. Your Name (for certificates)

Ed. Completed **Gender** **Year of birth**

-- -- --

Mailing address

Goals in signing up for edX

I agree to the Terms of Service*

I agree to the Honor Code*

Students then register for as many individual courses as they choose, which enrolls them in each selected course.

You can access this self-reported information for all of the students who are enrolled in your course:

- username
- name
- email
- year_of_birth
- gender
- level_of_education
- mailing_address
- goals

The student data that is available to course staff always reflects the set of live, current enrollments. Students can register for your course throughout the defined enrollment period, and they can unregister from a course at any time, which unenrolls them. Students can also change their email addresses and full names at any time. As a result, you may want to download student data periodically to gain insights into how the student population changes over time.

Note: In the future, edX may also request that students select a language and location. This data is not collected at this time.

Download or view student data

You can download student data to learn about population demographics at a specific point in time, compare demographics at different points in time, and plot trends in the population over time.

In addition to the data for enrolled students, data for the course staff is included in the file or display.

Important: Do not navigate away from this page while you wait for the data to be prepared. The larger the enrollment for your course, the longer it takes to create and output the data.

For smaller courses, you also have the option to view student data on the Instructor Dashboard.

To download or view student data:

1. View the live version of your course.
2. Click **Instructor** then **Try New Beta Dashboard**.
3. Click **Data Download**.
4. To download data about enrolled students in a CSV file, click **Download profile information as a CSV**.

You are prompted to open or save the enrolled_profiles.csv file. All student-supplied data is included without truncation.

5. To display data about enrolled students, click **List enrolled students' profile information**. (This option is available only for courses with an enrollment of less than 200.)

A table of the student data displays, with one row for each enrolled student. Longer values, such as student goals, are truncated.

username	name	email	language	location	year_of_birth	gender	level_of_education	mailing_address	goals
[REDACTED]	[REDACTED]	[REDACTED]			[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	Learn more about moocs and EMPLOYEE
[REDACTED]	[REDACTED]	[REDACTED]			[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	

Note: The columns for language and location are included in this report for backward compatibility only. This data is no longer collected during student registration.

View demographic distributions

You can view a course-wide summary of certain demographic distributions for your currently enrolled students. The total count for each value reported for gender and educational attainment displays on the Instructor Dashboard. Because this data is optional, the totals for each of these self-reported values are likely to be lower than your course enrollment total. You can also view a chart with the ages of all currently enrolled students.

To display demographic data for your students:

1. View the live version of your course.
2. Click **Instructor** then **Try New Beta Dashboard**.
3. Click **Analytics**.
 - The Year of Birth section displays a chart of enrolled students plotted by year of birth.
 - The Gender Distribution and Level of Education sections show tables with counts of responses made by enrolled students.

LEVEL OF EDUCATION

Level of Education	Count
Elementary/primary school	465
None	749
Bachelor's degree	27722
Secondary/high school	22965
Master's or professional degree	14680
No Data	9557
Doctorate	769
Junior secondary/junior high/middle school	2603
Other	2501

GENDER DISTRIBUTION

Gender	Count
No Data	9078
Male	62529
Other	170
Female	13357

“No Data” is the sum of the students for whom no value exists for the demographic.

Data for individual students is not shown, and you cannot download data directly from this page. See *Download or view student data*.

8.6.3 Access anonymized student IDs

Some of the tools that are available for use with the edX platform, including external graders and surveys, work with anonymized student data. If it becomes necessary for you to deanonymize previously anonymized data, you can download a CSV file to use for that purpose.

To download a file of assigned user IDs and anonymized user IDs:

1. View the live version of your course.
2. Click **Instructor > Try New Beta Dashboard**.
3. Click **Data Download > Get Student Anonymized IDs CSV**.

You are prompted to open or save the (course-id)-anon-id.csv file for your course. This file contains the user ID that is assigned to each student at registration and its corresponding anonymized ID. Values are included for every student who ever enrolled for your course.

You can use the data in this file together with the data in the enrolled_profile.csv file of student data and in a *course_id_grade_report_date.csv* file for your course to research and deanonymize student data.

8.7 Grade and Answer Data

You can review information about how grading is configured for your course, and access student grades, at any time after you create the course. You can also make adjustments to how a problem is graded, for a single student or all students. For information about the grading data that you can access and the changes you can make, see the following topics:

- *Review How Grading Is Configured for Your Course*
- *Access Student Grades*
- *Adjust Grades*

To review student answers to the problems in your course, you can download data for course problems or review a graph for a selected problem. See *Student Answer Distribution*.

For information about how you establish a grading policy and work with the Problem components in your course, see *Establishing a Grading Policy* or *Working with Problem Components*.

Important: If you make changes to your grading policy after a course starts, students can see the effect of your changes on their **Progress** pages. Be sure to announce any changes on your Course Info page.

8.7.1 Review How Grading Is Configured for Your Course

You can review the assignment types that are graded and their respective weights on the Instructor Dashboard.

You establish a grading policy for your course when you create it in Studio. While the course is running, you can view an XML representation of the assignment types in your course and how they are weighted to determine students' grades.

1. View the live version of your course.
2. Click **Instructor > Try New Beta Dashboard**.
3. Click **Data Download > Grading Configuration**.

A list of the assignment types in your course displays. In this example, Homework is weighted as 0.3 (30%) of the grade.

8.7.2 Access Student Grades

You can generate and review your students' grades at any time during your course. You can generate grades for all currently enrolled students, or check the progress of a single student (who can be enrolled or unenrolled).

Generate Grades for Enrolled Students

When you initiate calculations to grade student work, a process starts on the edX servers. The complexity of your grading configuration and the number of students enrolled in your course affect how long this process takes. You can download the calculated grades in a CSV (comma-separated values) file when the grading process is complete. You cannot view student grades on the Instructor Dashboard.

To generate grades for the students who are currently enrolled in your course:

1. View the live version of your course.
2. Click **Instructor > Try New Beta Dashboard**.
3. Click **Data Download**.
4. To start the grading process, click **Generate Grade Report**.

A status message indicates that the grading process is in progress. This process can take some time to complete, but you can navigate away from this page and do other work while it runs.

5. To track the progress of the grading process, reload the page in your browser and scroll down to the **Pending Instructor Tasks** section.

Download Grades for Enrolled Students

After you request a grade report for your students, the result is a time-stamped CSV file that includes columns to identify each student: id, email, and username. It also includes a column for every assignment that is included in your grading configuration: each homework, lab, midterm, final, and any other assignment type you added to your course.

Important: Because this file contains confidential, personally identifiable data which may be subject to the Family Educational Rights and Privacy Act (FERPA), be sure to follow your institution's data stewardship policies when you open or save this file.

To download a file of student grades:

1. View the live version of your course.
2. Click **Instructor > Try New Beta Dashboard**.
3. Click **Data Download**.
4. To open or save a grade report file, click the `{course_id}_grade_report_{date}.csv` file name at the bottom of the page.

Note: To prevent the accidental distribution of student data, you can only download these files by clicking the links on this page. Do not copy these links for reuse elsewhere, as they expire within 5 minutes. The links on this page also expire if the page is open for more than 5 minutes: if necessary, refresh the page to generate new links.

Interpret the Grade Report

Grade reports provide a snapshot of cumulative course scores, by assignment, of every currently enrolled student.

You can open `{course_id}_grade_report_{date}.csv` files in a spreadsheet application to sort, graph, and compare data.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
id	email	username	grade	HW 01	HW 02	HW 03	HW 04	HW 05	HW 06	HW 07	HW Avg	Midterm	Final
0.43	1	1	1	1	0	0	0	0.666666667	0.75	0			
0.05	0	1	0	0	0	0	0	0.166666667	0	0			
0.38	1	1	1	0.5	0	0	0	0.583333333	0.666666667	0			
0.3	1	1	0	0	0	0	0	0.333333333	0.666666667	0			
0.13	0	0	0.5	0	0	0	0	0.083333333	0.333333333	0			

The CSV file contains one row of data for each student, and columns that provide the following information.

- Student identifiers, including an internal `id`, `email` address, and `username`.
- `grade`, with the total score a student has currently attained in the course. This value is expressed as a decimal: a student with a grade of 0.65 has earned 65% of the credit in the course, and a student with a grade of 1 has earned 100%.
- Each `{assignment type} {number}` defined in your grading configuration, with the score a student attained for that specific assignment. For example, column `HW 03` shows the scores for the third homework assignment.
- An `{assignment type} Avg` with each student's current average score for that assignment type: for example, `HW Avg`.

Note: The grade reports do not include information about individual questions within the assignments, or include student answer distributions.

Check the Progress of a Single Student

To check a single student's progress, you can locate the specific row on the grade report or review the student's **Progress** page. The **Progress** page includes a chart that plots the score the student has earned for every graded assignment and the total grade as of the current date. Below the chart, each assignment and the score attained are listed.

To review a student's **Progress** page, you supply an email address or username. You can check the progress for students who are either enrolled in, or who have unenrolled from, the course.

Students can view a similar chart and assignment list (of their own progress only) when they are logged in to the course.

To view the **Progress** page for a student:

1. View the live version of your course.
2. Click **Instructor > Try New Beta Dashboard**.
3. Click **Student Admin**.
4. In the Student-Specific Grade Inspection section, enter the student's email address or username.
5. Click **Student Progress Page**.

The **Progress** page for the student displays a chart with the grade for each homework, lab, midterm, final, and any other assignment types in your course, and the total grade earned for the course to date.

Courseware Course Info Discussion Progress Instructor

Course Progress for Student ' [REDACTED] ([REDACTED] @gmail.com)

VIEW GRADING IN STUDIO

A bar chart titled 'Course Progress' showing student scores across different assignments and exams. The y-axis represents the percentage of the total grade, ranging from 0% to 100% in increments of 15%. The x-axis lists assignments: Finger Ex Avg, PSET 02, PSET 03, PSET 04, PSET 05, PSET 06, PSET 07, PSET 08, PSET 09, PSET Avg, Midterm 1, Midterm 2, Final, and Total. The bars show the following approximate values:

Assignment	Score (%)
Finger Ex Avg	~75%
PSET 03	~85%
PSET 04	~65%
PSET 09	*
PSET Avg	~15%
Midterm 1	~45%
Midterm 2	~35%
Final	~25%
Total	11%

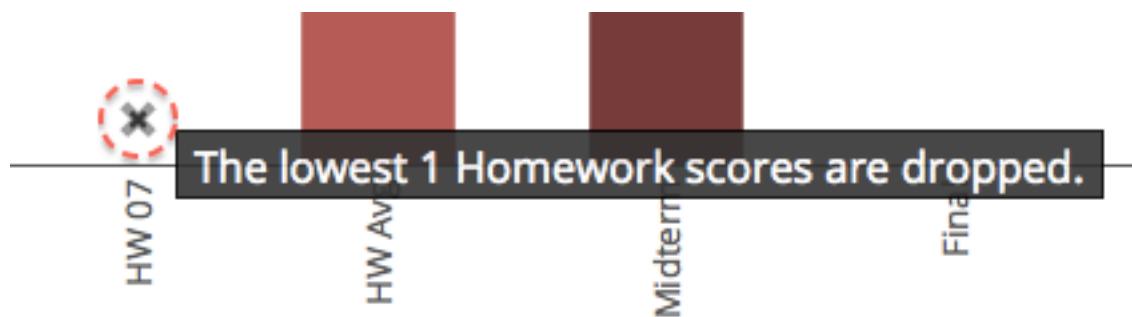
Overview

[edX introduction](#) (0/5)
Lecture Sequence
Practice Scores: 0/1 0/1 0/1 0/1

Week 1

[Introduction to Computation](#) (5/15) 33%
Lecture Sequence
Problem Scores: 5/5 0/5 0/5

To learn more about a particular assignment, move the cursor onto the value in the chart. A brief description displays.



Below the chart, subsections are listed on the left and the units that contain assignments are listed on the right. The student's individual problem scores display.

First stanza

Line 1 (1/1) 100%

Homework

Problem Scores: 1/1

Dusk (1/1) 100%

Homework

Problem Scores: 1/1

Parsing the distinctions (5/5) 100%

Homework

Problem Scores: 5/5

Second Verse (3/4) 75%

Midterm Exam

Problem Scores: 1/1 0/1 1/1 1/1

Interpret the Student Progress Page

The chart of a student's scores on the **Progress** page and the rows of data on the grade report present assignment scores in a similar order. However, the total, cumulative score earned for the course is placed in a different position on the **Progress** page.

In this example grade report, the indicated student has a current grade of 0.43 (43%).

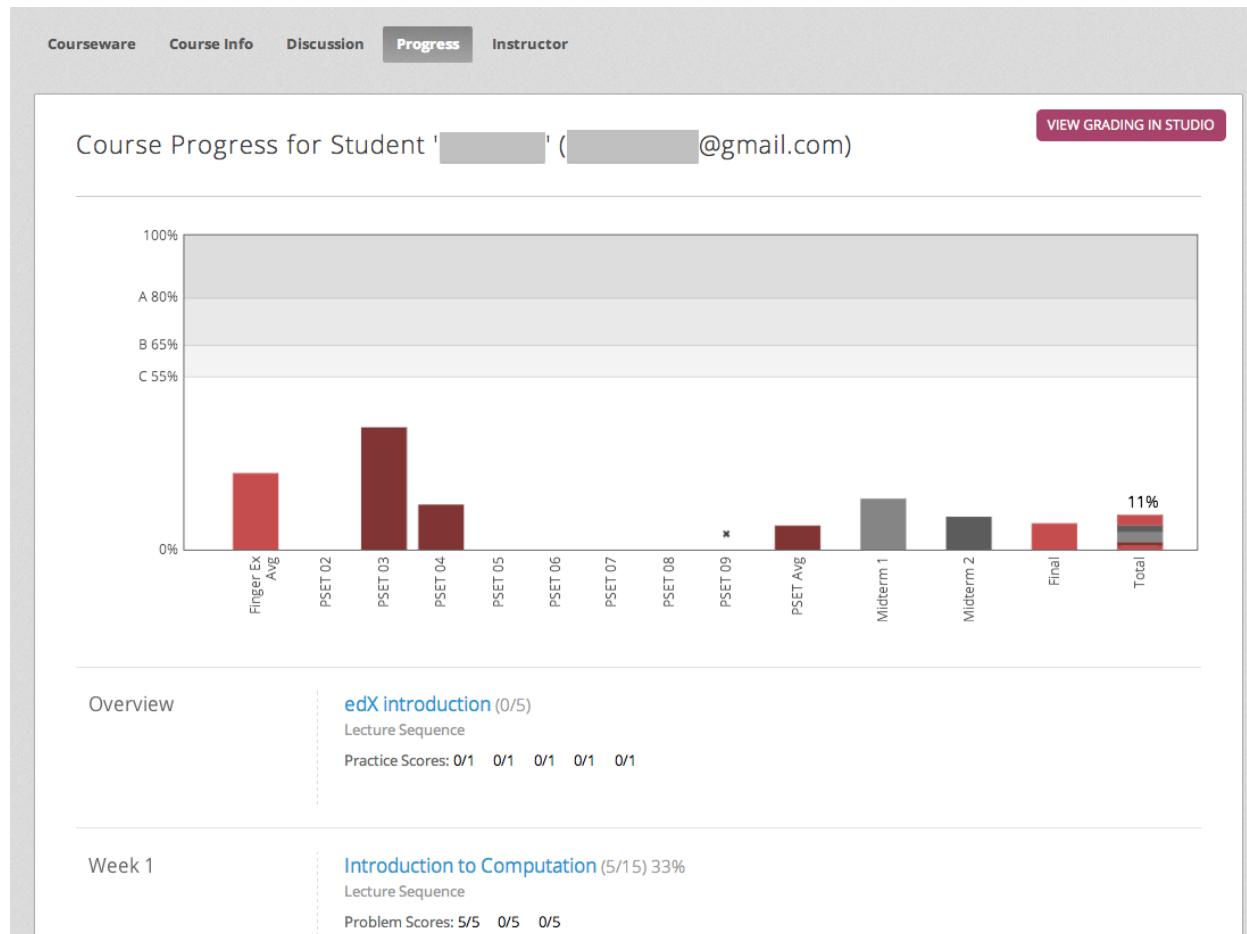
A	B	C	D	E	F	G	H	I	J	K	L	M	N
id	email	username	grade	HW 01	HW 02	HW 03	HW 04	HW 05	HW 06	HW 07	HW Avg	Midterm	Final
			0.43	1	1	1	1	0	0	0	0.6666666667	0.75	0
			0.05	0	1	0	0	0	0	0	0.1666666667	0	0
			0.38	1	1	1	0.5	0	0	0	0.5833333333	0.6666666667	0
			0.3	1	1	0	0	0	0	0	0.3333333333	0.6666666667	0
			0.13	0	0	0	0.5	0	0	0	0.0833333333	0.3333333333	0

- On each of the first four homework assignments the student scored 1 (100%), but currently has a 0 (0%) on each of the remaining three assignments.

Notice, however, that the student's current average score for homework assignments is listed as 0.6666666667 (67%): in this course, the homework assignment with the lowest score is dropped, so this average is over six assignments rather than all seven.

- The student has a score of 0.75 (75%) on the midterm, and a score of 0 (0%) on the final.

On the student's **Progress** page, you see the same information graphically represented; however, the student's “total” of 43% is on the far right.



The chart on the **Progress** page includes y-axis labels for the grade ranges defined for the course. In this example, Pass is set to 60%, so at the end of the course students with a grade of 0.60 or higher can receive certificates.

Check a Student's Answer Submission and Submission History

For a single student and problem, you can review the exact response submitted, the number of attempts made, and the date and time of the submission. You identify the student by supplying a username.

To review a response submitted by a student:

1. View the live version of your course.
2. Click **Courseware** and navigate to the component that contains the problem you want to review.
3. Display the problem and then click **Submission History** below the problem.
4. Enter the username for the student whose work you want to review, then click **View History** at the end of the page.

Information about the response or responses provided by the student displays.

To close the Submission History Viewer, click on the browser page outside of the viewer.

8.7.3 Adjust Grades

You can adjust grades for one student at a time, or for all of the enrolled students in the course. For example, your course beta testers can evaluate numerous different correct and incorrect responses to verify that your course is set up as you intend. Students can also report errors while a course is running.

When an error is discovered or corrected, or if you decide that you must modify a problem after students or beta testers have attempted to answer it, you can either:

- Rescore the submitted answers to reevaluate the work.
- Reset the number of attempts made to answer the question correctly so that students can try again.

To make these adjustments, you need to specify a problem by supplying the unique identifier from its URL.

Find the Unique Identifier for a Problem

When you create each of the problems for a course, edX assigns a unique identifier. To make grading adjustments for a problem, or to view data about it, you need to specify this identifier.

To find the unique identifier in the URL for a problem:

1. View the live version of your course.
2. Click **Courseware** and navigate to the component that contains the problem you want to review.
3. Display the problem, and click **Staff Debug Info**.

Information about the problem displays, including its location or URL. This URL ends with the type of module, which is typically “problem”, and the unique identifier.

```
is_released = Yes!
location = i4x://test_doc/SB101/problem/d46b0a33732c44e793be724653ea843f
```

4. To copy the identifier for the problem, select it, right click, and choose **Copy**.

Note: If the URL does not include “problem” before the identifier, you will need to specify that module identifier as well. Select and copy both the module identifier and the problem identifier.

To close the Staff Debug viewer, click on the browser page outside of the viewer.

Rescore Student Submissions

Each problem that you define for your course includes a correct answer, and may also include a tolerance or acceptable alternatives. If you decide to make a change to these values, you can rescore any responses that were already submitted. For a specified problem, you can rescore the work submitted by a single student, or rescore the submissions made by every enrolled student.

Note: You can only rescore problems that have a correct answer entered in edX Studio. Problems that are scored by an external grader cannot be rescored with this procedure.

To specify the problem you want to rescore, you need its unique identifier. See *Find the Unique Identifier for a Problem*.

To rescore a problem:

1. View the live version of your course.
2. Click **Instructor > Try New Beta Dashboard**.

3. Click **Student Admin**.
4. Rescore the problem, either for an individual student or for all students.

To rescore a problem for one student, you work in the **Student-Specific Grade Adjustment** section of the page. Enter the student's email address or username and the unique problem identifier, and then click **Rescore Student Submission**.

To rescore a problem for all enrolled students, you work in the **Course-Specific Grade Adjustment** section of the page. Enter the unique problem identifier, and then click **Rescore ALL students' problem submissions**.

5. When you see a dialog box that notifies you that the rescore process is in progress, click **OK**.

This process does not take long for a single student, but can take some time to complete for all enrolled students. The process runs in the background, so you can navigate away from this page and do other work while it runs.

6. To view the results of the rescore process, click either **Show Background Task History for Student** or **Show Background Task History for Problem**.

A table displays the status of the rescore process for each student or problem.

Reset Student Attempts

When you create a problem, you can limit the number of times that a student can try to answer that problem correctly. If unexpected issues occur for a problem, you can reset the value for one particular student's attempts back to zero so that the student can begin work over again. If the unexpected behavior affects all of the students in your course, you can reset the number of attempts for all students to zero.

For information about modifying a released problem, including other workarounds, see *Modifying a Released Problem*.

Note: To reset the number of attempts for a problem, you need its unique identifier. See *Find the Unique Identifier for a Problem*.

To reset student attempts for a problem:

1. View the live version of your course.
2. Click **Instructor > Try New Beta Dashboard**.
3. Click **Student Admin**.
4. To reset the number of attempts for one student, you work in the Student-Specific Grade Adjustment section of the page. Enter the student's email address or username and the unique problem identifier, then click **Reset Student Attempts**.
5. To reset the number of attempts for all enrolled students, you work in the Course-Specific Grade Adjustment section of the page. Enter the unique problem identifier then click **Reset ALL students' attempts**.
6. A dialog opens to indicate that the reset process is in progress. Click **OK**.

This process does not take long for a single student, but can take some time to complete for all enrolled students. The process runs in the background, so you can navigate away from this page and do other work while it runs.

7. To view the results of the reset process, click either **Show Background Task History for Student** or **Show Background Task History for Problem**.

A table displays the status of the reset process for each student or problem.

8.7.4 Student Answer Distribution

For certain types of problems in your course, you can download a CSV file with data about the distribution of student answers. Student answer distribution data is included in the file for problems of these types:

- Checkboxes (<choiceresponse>)
- Dropdown (<optionresponse>)
- Multiple choice (<multiplechoiceresponse>)
- Numerical input (<numericalresponse>)
- Text input (<stringresponse>)
- Math expression input (<formularesponse>)

The file includes a row for each problem-answer combination selected by your students. For example, for a problem that has a total of five possible answers the file includes up to five rows, one for each answer selected by at least one student. For problems that use rerandomization (the **Randomization** setting in Studio), there is one row for each problem-variant-answer combination selected by your students.

The CSV file contains the following columns:

Column	Description
ModuleID	The internal identifier for the Problem component.
PartID	For a Problem component that contains multiple problems, the internal identifier for each individual problem. For a Problem component that contains a single problem, the internal identifier of that problem.
Correct Answer Count	0 if this AnswerValue is incorrect, or 1 if this AnswerValue is correct.
ValueID	The number of times that students entered or selected this answer as their most recent submission for the problem or problem variant. For problems with the number of Attempts set to a value greater than 1, this means that each student contributes a maximum of 1 to this count, even if the same answer is provided in multiple attempts.
AnswerValue	The internal identifier of the answer choice for checkboxes and multiple choice problems. Blank for dropdown, numerical input, text input, and math expression input problems.
Variant	The text label of the answer choice for checkboxes, dropdown, and multiple choice problems. The value entered by the student for numerical input, text input, and math expression input problems. For problems that use the Randomization setting in Studio, contains the unique identifier for a variant of the problem. Blank for problems that do not use the Randomization setting, or that use the Never option for this setting.
Problem Display Name	The Display Name defined for the problem.
Question	The label for accessibility that appears above the answer choices or the text entry field for the problem. In Studio's Simple Editor, this text is surrounded by two pairs of angle brackets (>>Question<<). Blank for questions that do not have an accessibility label defined.

Entries are sorted by the value in each column, starting with the ModuleID on the left and continuing through the columns to the right.

Download the Student Answer Distribution Report

An automated process runs periodically on the edX servers to update the CSV file of student answer data. A link to the most recently updated version of the CSV file is available on the Instructor Dashboard.

To download the most recent file of student answer data:

1. View the live version of your course.
2. Click **Instructor > Try New Beta Dashboard**.
3. Click **Data Download**.

4. At the bottom of the page, click the `{course_id}_answer_distribution.csv` file name. You may have to scroll down to find this file.

Frequently Asked Questions about the Student Answer Distribution Report

Answers to questions about the student answer distribution report follow.

My course doesn't have a student answer distribution report. How can I generate it?

Student answer distribution reports are generated automatically, and refreshed several times each day. The `{course_id}_answer_distribution.csv` file displays after all of the `{course_id}_grade_report_{date}.csv` files. Be sure to scroll to the end of the list of available reports.

Why are some problems missing from this report? The ones that are missing do have the problem types listed under Student Answer Distribution.

This report includes only problems that at least one student has answered since early March 2014. For those problems, this report only includes activity that occurred after October 2013.

Why don't I see an AnswerValue for some of my problems?

For checkboxes and multiple choice problems, the answer choices actually selected by a student after early March 2014 display as described in the previous answer. Answer choices selected by at least one student after October 2013, but not selected since early March 2014, are included on the report but do not include an **AnswerValue**. The **ValueID** does display the internal identifiers, such as `choice_1` and `choice_2`, for those answers.

Why don't I see a Question for some of my problems?

The value in the **Question** column is the accessibility label for the problem. For more information about how to set up labels for problems, see *Creating Exercises and Tools*.

Also, for problems that use the **Randomization** setting in Studio, if a particular answer has not been selected since early March 2014, the **Question** is blank for that answer.

My students are saying that they answered a question, but it isn't showing up in the report. How can that be?

Only questions that have a **Maximum Attempts** setting of 1 or higher are included on the report.

I made a correction to a problem after it was released. How can I tell which students tried to answer it before I made the change?

Problem **Count** values reflect the entire problem history. If you change a problem after it is released, it may not be possible for you to determine which answers were given before and after you made the change.

Why is the same answer showing up in two different rows when I view the report in a spreadsheet?

Some spreadsheet applications can alter the data in the CSV report for display purposes. For example, for different student answers of "0.5" and ".5" Excel correctly includes the two different lines from the CSV, but displays the **AnswerValue** on both of them as "0.5". If you notice answers that appear to be the same on separate lines with separate counts, you can review the actual, unaltered data by opening the CSV file in a text editor.

Why are there strange characters in the report when I view it in a spreadsheet?

The CSV file is UTF-8 encoded, but not all spreadsheet applications interpret and render UTF-8 encoded characters correctly. For example, a student answer distribution report with answer values in French displays differently in Microsoft Excel for Mac than in OpenOffice Calc.

Answer Values in Microsoft Excel for Mac:

ValueID	AnswerValue
choice_0	J'ai particip__ la discussion.
choice_0	J'ai regard_ la vid_o: "Une focale sur la non-democratie"
choice_2	le politique
choice_0	le pouvoir
choice_1	l'_conomie

Answer Values in OpenOffice Calc:

ValueID	AnswerValue
choice_0	J'ai participé à la discussion.
choice_0	J'ai regardé la vidéo: "Une focale sur la non-democratie"
choice_2	le politique
choice_0	le pouvoir
choice_1	l'économie

If you notice characters that do not display as expected in a spreadsheet, try a different spreadsheet application such as LibreOffice or Apache OpenOffice to open the CSV file.

Interpret the Student Answer Distribution Report

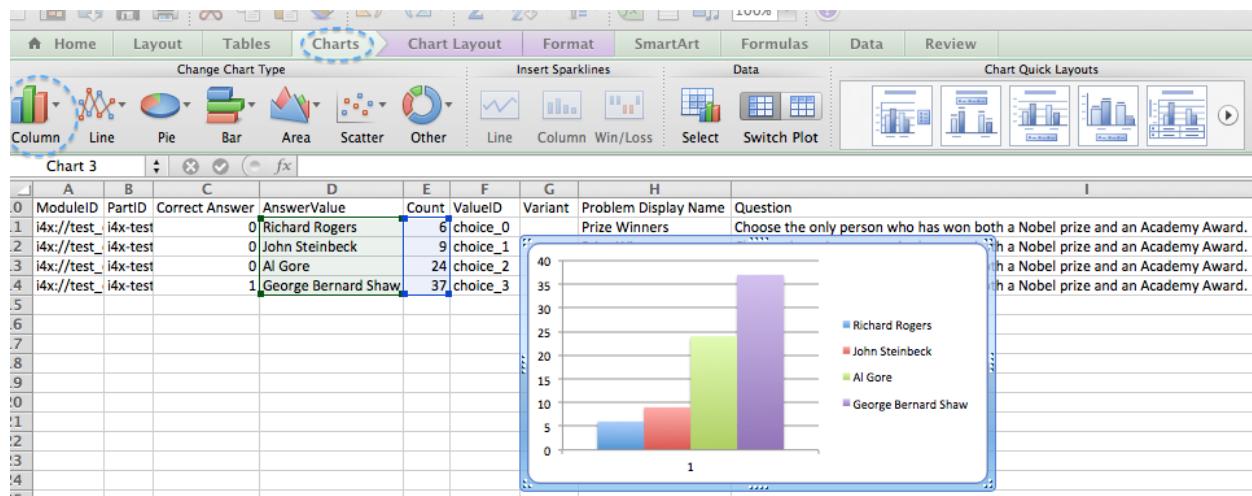
You can use the Student Answer Distribution report to review student responses to assignments, which can then help you evaluate the structure and completeness of your courseware and problem components.

As an example, you define a text input question in Studio to have a single correct answer, “Warfarin”. When you produce the Student Answer Distribution report, you verify that this answer was in fact marked correct: there is a 1 in the **Correct Answer** column for this **AnswerValue**.

A	B	C	D	E	F	G	H	I
Mod	Partl	Correct Answer	Count	Valu	AnswerValue	Vari	Problem Disp	Question
i4x://i4x-t		0	107		Coumadin		Other Danger Name	an anticoagulant that is now used to prevent thrombosis, but that originally was used to poison rodents.
i4x://i4x-t		0	55		Warfarin or Coumadin		Other Danger Name	an anticoagulant that is now used to prevent thrombosis, but that originally was used to poison rodents.
i4x://i4x-t		1	6454		Warfarin		Other Danger Name	an anticoagulant that is now used to prevent thrombosis, but that originally was used to poison rodents.
i4x://i4x-t		0	1		Cubicin		Other Danger Name	an anticoagulant that is now used to prevent thrombosis, but that originally was used to poison rodents.
i4x://i4x-t		0	3		Coufarin		Other Danger Name	an anticoagulant that is now used to prevent thrombosis, but that originally was used to poison rodents.

However, as you view the report you notice other student answers that you did not set up to be marked as correct in Studio, but that you might (or might not) also consider to be correct, such as “Warfarin or Coumadin”. The **Correct Answer** column shows that the other answers were marked incorrect (0), but for future iterations of your course you may want to revise the question or update the problem to evaluate additional variations of the answer as correct.

Many spreadsheet applications offer data visualization options, such as charts or graphs. Charts can help make your students’ common misconceptions easier to identify.



In this example, the Student Answer Distribution report is open in Microsoft Excel. To create a chart that shows how many of your students chose various answers to a multiple choice question, you move the **AnswerValue** and **Count** columns next to each other. After you click and drag to select the report cells that contain the data you want to chart, you click the Charts toolbar and then click the type of chart you want.

Note: Refer to the help for the spreadsheet application that you use for information on using these options. You may have to make changes to your spreadsheet, such as reordering columns. Save a copy of the file you originally downloaded as a backup before you begin.

You can adjust your course content based on common student mistakes. While most students in this example selected the correct answer, the number of incorrect answer(s) can guide future changes to the courseware.

View a Histogram of Scores for a Single Problem

You can view a chart of the score distribution for a specified problem.

Note: To view the score distribution for a problem, you need its unique identifier. You can display a histogram for problems that have the /problem/ prefix in the unique identifier. See *Find the Unique Identifier for a Problem*.

To display the distribution of scores for a problem:

1. View the live version of your course.
2. Click **Instructor > Try New Beta Dashboard**.
3. Click **Analytics**.
4. In the Score Distribution section, select a problem by using its unique identifier.

The **Analytics** page updates to display a histogram of scores for that problem.

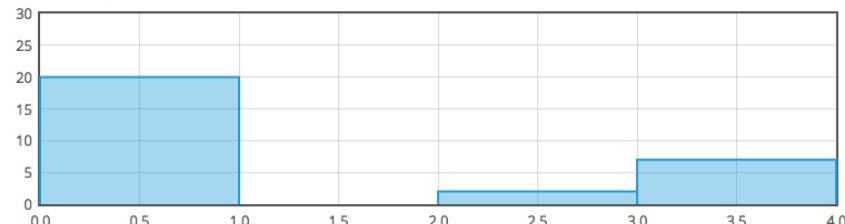
SCORE DISTRIBUTION

The chart below displays the score distribution for each standard problem in your class, specified by the problem's url name. Scores are shown without weighting applied, so if your problem contains 2 questions, it will display as having a total of 2 points.

Problem: 04ffaf1b4e804163a56216ba1ea34f75 ▾

Last Updated: 2014-03-06T00:01:10+00:00

57 students scored.



Interpret a Score Histogram

The histogram of scores for a selected problem provides the following information.

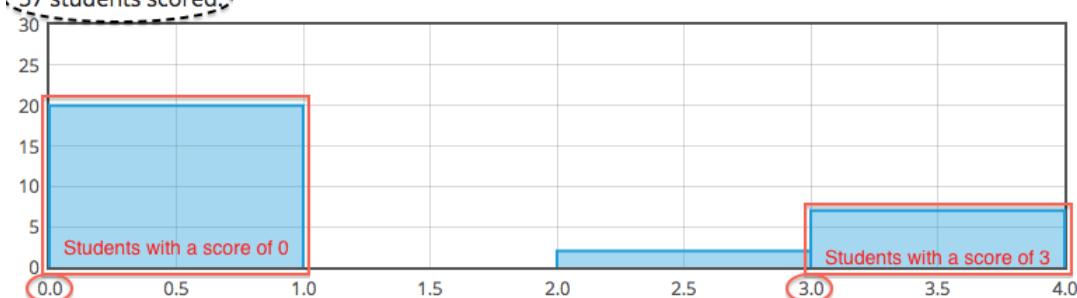
SCORE DISTRIBUTION

The chart below displays the score distribution for each standard problem in your class, specified by the pr without weighting applied, so if your problem contains 2 questions, it will display as having a total of 2 poin

Problem: 04ffaf1b4e804163a56216ba1ea34f75 ▾

Last Updated: 2014-03-06T00:01:10+00:00

57 students scored.



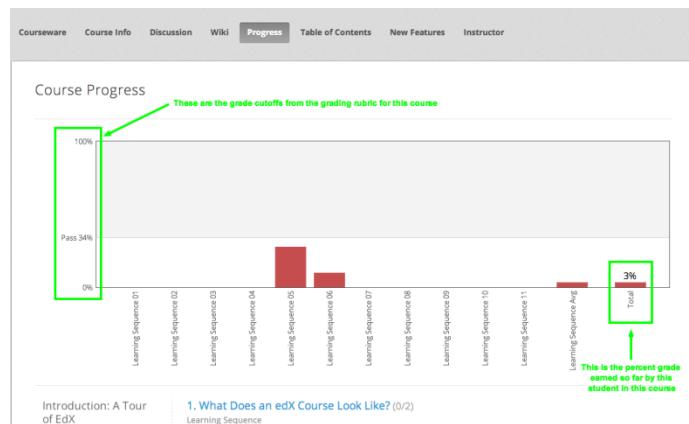
- The x-axis indicates the number of points that students can receive for the problem, unweighted. Although decimal values are marked on the x-axis, each score is a whole number. The whole number value at the left of a plotted bar is the score that applies to it.
- The y-axis indicates the number of students who have answered and checked the problem.
- Above the graph, a number of **students scored** displays. This number indicates the total number of database records that exist for the problem: it includes not only students who have answered and checked the problem to receive a score, but also students who have opened the problem but not yet completed it.
- An automated process runs approximately weekly on the edX servers to update score histograms. This process runs less frequently than the process that updates the student answer distribution report: at certain times during a course the number of students shown in a score histogram can be quite different than the **Count** for the same problem in a student answer distribution report.

8.8 Checking Student Progress and Issuing Certificates

The grading policy and stored problem scores are used to record progress through the course, determine final grades, and issue certificates at the end. This unit will give you some advance information about how the grading policy will be visible to the students during the run of the course and what you will need to do at the end of the course to give out grades.

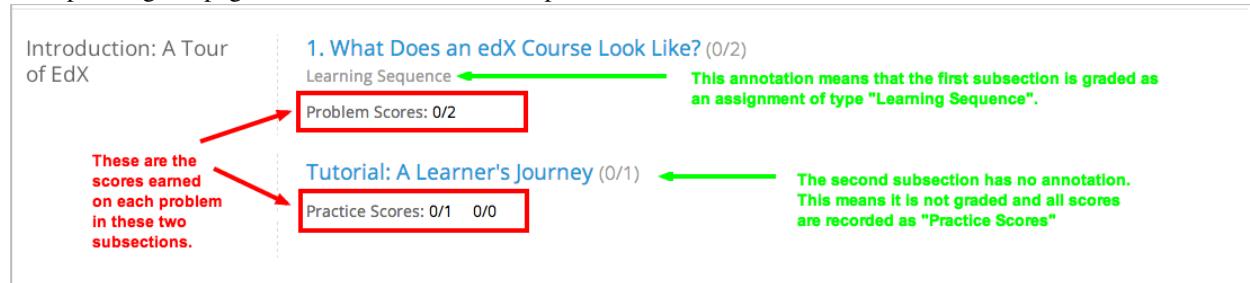
8.8.1 A Student's View

Students can check their progress by clicking on the **Progress** tab in the course. The student's progress through the graded part of the course is displayed at the top of this page, above the subsection scores, as a chart with entries for all the assignments, total percentage earned in the course so far and percent needed for each grade cutoff. Here is an example of a student's progress through edX101.



The student can see from this page that edX101 was graded as a Pass/Fail course with a cutoff of 34% and that the grading rubric contained one assignment type, called Learning Sequence, consisting of 11 assignments total. Furthermore, this particular student has only submitted correct responses to two assignments, and that her current total percent grade in the course is 6%. By hovering over each progress bar, the student can get further statistics of how much each assignment was counted as.

As was mentioned in the unit on Viewing Scores, further down on the Progress page is a list of all the subsections in the course, with the scores recorded for the student for all problems in the course. Here is a scrolled down view of the example Progress page for the student in the example above:



Again, note that point scores from graded sections are called “Problem Scores”, while point scores from ungraded sections are called “Practice Scores”.

8.8.2 Check Progress of Students as an Instructor

To check the progress of the student, go to the Instructor Dashboard of your course click the Grades page. The Instructor Dashboard for courses sometimes changes as more course-specific tools get added. Here is the current view of the top of the Grades page of the Instructor dashboard for edX101:

The screenshot shows the 'Instructor' tab selected in the top navigation bar. Below it, the 'Instructor Dashboard' header is visible. Underneath, there's a section titled '[GRADES | ADMIN | FORUM ADMIN | ENROLLMENT | DATADUMP | MANAGE GROUPS]'. This section contains several buttons: 'Gradebook', 'Grade summary', 'Dump list of enrolled students', 'Dump Grades for all students in this course' (disabled), 'Download CSV of all student grades for this course', 'Dump all RAW grades for all students in this course', 'Download CSV of all RAW grades', and 'Download CSV of answer distributions'. Below these buttons is a section titled 'STUDENT-SPECIFIC GRADE INSPECTION AND ADJUSTMENT'. It includes a text input field for 'edX email address or their username:' and a button 'Get link to student's progress page'. At the bottom of this section, it says 'and, if you want to reset the number of attempts for a problem, the urlname of that problem' followed by a text input field and a button 'Reset student's attempts'.

Here you see several options for viewing or downloading student grades, viewing individual progress through a course or resetting problem attempts.

Note: Only the top part of the Grades page is shown. The page continues with some more information about course statistics. The link to the student's progress page should give you a view of exactly what the student would see, including scores for graded and ungraded assignments.

Note: The stored scores visible to you on the Instructor tab and to the students from the Progress tab in the course are a snapshot of the current state of the problem score database. They may be slightly out of sync with actual problem scores. (Asynchronicities may happen if, for example, the weight of a live problem was changed during an assignment, and not everyone has yet resubmitted their answers for that problem.) Scores and grades are usually recomputed at the end of the semester before determining final grades and issuing Certificates.

8.8.3 Assign Final Grades and Issue Certificates

The final grades of a student in the course and the grading rubric you have set are used to determine whether the student has earned a Certificate of Mastery for the course. The process for issuing Certificates has to be started manually by you or by the edX support team at the end of the course.

Information for Your Students

9.1 Sample Student Login Guide

Create a login guide for your individual course.

STUDENT LOGIN GUIDE FOR [COURSE NUMBER]

How to Set Up Your edX Account and Register for Your Class

Welcome to [COURSE NAME] on edX! This guide will walk you through setting up your student account with edX, registering for [COURSE NAME], and accessing the course materials.

Step 1: Open the course registration page

Example

[https://edge.edx.org/courses/\[YOUR_CLASS_PATH\]/about](https://edge.edx.org/courses/[YOUR_CLASS_PATH]/about) [REPLACE WITH URL TO YOUR REGISTRATION PAGE ON edX]

https://edge.edx.org/courses/edX/PHOTO101/Digital_SLR_Photography_101/about

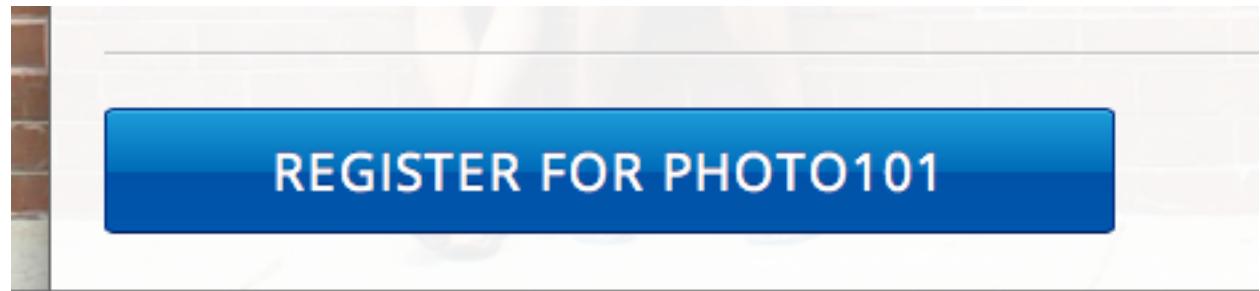
The above URL should bring you to the course registration page: [REPLACE WITH SCREENSHOT OF YOUR REGISTRATION PAGE]

Example

Building and Running an edX Course, Release

The screenshot shows the edX course page for PHOTO101: Digital SLR Photography 101. At the top, there's a navigation bar with links like "Online Python execution", "Welcome", "Redrock mic...cam Forum", "www.dtic....=ADA512533", "Planning", "Planning Yo...cumentation", "Instructor ...kboard Help", "Blackboard ...ning Center", "DCTV | Vide", "About PHOTO101", and "PENDIX C: Sample Student Login Guide — edX 0.1 documentation". Below the navigation is a search bar with "FIND COURSES" and the edX logo. A user profile for "Toddi" is visible. The main content area features a banner with three photos of people in a classroom setting. Below the banner, the course title "PHOTO101: Digital SLR Photography 101" is displayed with the edX logo. A large blue button labeled "REGISTER FOR PHOTO101" is prominent. To the right, there's a sidebar with social sharing icons (Twitter, Like, Email) and course details: "Course Number: PHOTO101", "Classes Start: May 06, 2013", and "Classes End: Jul 31, 2013".

Step 2: Request to register for the course Click on the blue Register for [NUMBER OF YOUR CLASS] button: [REPLACE WITH PART OF YOUR SCREENSHOT ABOVE]



A pop-up user authentication window will appear. It will ask you to log in or sign up for edX. (If you already have an account with edX, use it to log in. You should now be registered for the course. Skip to Step 5. Otherwise, go on with Step 3.)

Register for edX

https://edge.edx.org/register?course_id=edX/PHOTO101/Digital_SLR_Photography_101&enrollment_action=enroll

Free Webinar...ine Webinar Online Python...e execution Welcome — ...umentation Redrock mic...cam Forum www.dtic...=ADAS12533 Planning Planning Yo...cumentation Instructor ~...kboard Help

edXedge REGISTER NOW log in

WELCOME!
Register below to create your edX account

Please complete the following fields to register for an account.
Required fields are noted by **bold text and an asterisk (*)**.

E-mail *

Password *

Public Username *

Will be shown in any discussions or forums you participate in

Full Name *

Needed for any certificates you may earn (cannot be changed later)

Highest Level of Education Completed Gender Year of Birth

Mailing Address

Please share with us your reasons for registering with edX

I agree to the [Terms of Service](#) * I agree to the [Honor Code](#) *

create my edX account

[About](#) [Jobs](#) [Press](#) [FAQ](#) [Contact](#)

EdX is a non-profit created by founding partners Harvard and MIT whose mission is to bring the best of higher education to students of all ages anywhere in the world, wherever there is Internet access. EdX's free online MOOCs are interactive and subjects include computer science, public health, and artificial intelligence.

[View our FAQs for answers to commonly asked questions.](#) Once registered, most questions can be answered in the course specific discussion forums or through the FAQs.

[Terms of Service and Honor Code - Privacy Policy](#)

© 2013 edX, some rights reserved.

Step 3: Create your account with edX

Fill out the form in the pop-up window: (Some tips: Use an email address you check regularly. Also, the username you choose will also be your screen name in the course. This means the only name we will know you by is the username (not a first and last name). Please include your last name as part of your username so we can tell who you are on the forums.)

Step 4: Activate your edX account

Check your email. This is where your course activation link has been sent. You must click on this to finalize your access to the course.

Congratulations! You should now have an edX account and be registered for [ENTER YOUR COURSE NUMBER].

Proceed with Step 5 to access the class materials.

Step 5: Log in to edX and go the course website

Log in to your account on edX at <https://edge.edx.org/> by using the username and password you just created. You will be taken to the main landing page of your edX edge account, which should now include a listing for [ENTER YOUR

COURSE NUMBER]: [REPLACE WITH SCREENSHOT OF YOUR COURSE]

You can now click **View Courseware** to enter the main course website.

If you are experiencing problems with registration or with accessing the class website, please contact [SUPPORT CONTACT NAME] at [CLASS SUPPORT EMAIL].

9.2 Math Response Formatting for Students

In numerical input problems, the student's response may be more complicated than a simple number. Expressions like `sqrt(3)` and even `1+e^(sin(pi/2)+2*i)` are valid, and evaluate to 1.73 and $-0.13 + 2.47i$, respectively.

The parser renders text that students enter into “beautiful math” that appears below the problem’s response field:

$1+e^{(\sin(\pi/2)+2\cdot i)}$

$1 + e^{\sin\left(\frac{\pi}{2}\right) + 2 \cdot i}$

$\sqrt{(\pi - 0)^2 + (0 - 3)^2}$

$\sqrt{(\pi - 0)^2 + (0 - 3)^2}$

$\sqrt{\pi^2 + e^2}$

$\sqrt{\pi^2 + e^2}$

$\sin(\pi/5)$

$\sin\left(\frac{\pi}{5}\right)$

The form contains two examples of mathematical expressions:

- $n^*x^{(n-1)}$
- $n \cdot x^{n-1}$

Students can enter any of the following into the response field.

9.2.1 Numbers

- Integers: ‘2520’
- Fractions: $2/3$
- Normal floats: ‘3.14’
- With no integer part: ‘.98’
- Scientific notation: ‘1.2e-2’ (=0.012)
- More scientific notation: ‘-4.4e+5’ = ‘-4.4e5’ (=−440,000)
- SI suffixes: ‘2.25k’ (=2,250). The full list:

Suffix	Stands for	Example
%	percent	$0.01 = 1e-2$
k	kilo	$1000 = 1e3$
M	mega	$1e6$
G	giga	$1e9$
T	tera	$1e12$
c	centi	$0.01 = 1e-2$
m	milli	$0.001 = 1e-3$
u	micro	$1e-6$
n	nano	$1e-9$
p	pico	$1e-12$

The largest possible number handled currently is exactly the largest float possible (in the Python language). This number is $1.7977e+308$. Any expression containing larger values will not evaluate correctly, so it’s best to avoid this situation.

9.2.2 Default Constants

Simple and commonly used mathematical/scientific constants are included by default. These include:

- i and j as $\sqrt{-1}$
- e as Euler’s number (2.718...)
- g : gravity (9.80 m/s²)
- π
- k : the Boltzmann constant (~1.38e-23 in Joules/Kelvin)
- c : the speed of light in m/s (2.998e8)
- T : the positive difference between 0K and 0°C (285.15)

- q: the fundamental charge (~1.602e-19 Coulombs)

9.2.3 Greek Letters

The parser automatically converts the following Greek letter names into the corresponding Greek characters:

alpha	beta	gamma	delta
epsilon	varepsilon	zeta	eta
theta	vartheta	iota	kappa
lambda	mu	nu	xi
pi	rho	sigma	tau
upsilon	phi	varphi	chi
psi	omega		

Note: epsilon is the lunate version, whereas varepsilon looks like a backward 3.

9.2.4 Operators and Functions

- Use standard arithmetic operation symbols.
- Indicate multiplication explicitly by using an asterisk (*).
- Use a caret (^) to raise to a power.
- Use an underscore (_) to indicate a subscript.
- Use parentheses to specify the order of operations.

The normal operators apply (with normal order of operations): + - * / ^. Also provided is a special “parallel resistors” operator given by |||. For example, an input of 1 ||| 2 would represent the resistance of a pair of parallel resistors (of resistance 1 and 2 ohms), evaluating to 2/3 (ohms).

Currently, factorials written in the form ‘3!’ are invalid, but there is a workaround. Students can specify fact (3) or factorial (3) to access the factorial function.

The default included functions are the following:

- Trig functions: sin, cos, tan, sec, csc, cot
- Their inverses: arcsin, arccos, arctan, arcsec, arccsc, arccot
- Other common functions: sqrt, log10, log2, ln, exp, abs
- Factorial: fact (3) or factorial (3) are valid. However, you must take care to only input integers. For example, fact (1.5) would fail.
- Hyperbolic trig functions and their inverses: sinh, cosh, tanh, sech, csch, coth, arcsinh, arccosh, arctanh, arcsech, arccsch, arccoth

9.3 Open Response Assessments for Students

9.3.1 Introduction to Open Response Assessments

Note: Modify this section according to your course. For example, you can delete sentences such as “For more information, see *Peer Assessment*” and “For more information, see *Artificial Intelligence (AI) Assessment*” if your

ORA problem doesn't contain peer assessments or AI assessments and you want to delete these sections from this document.

Open response assessments allow you to submit a short written answer, an essay, or a file such as an image or computer code file.

When you come to an open response assessment problem, you see the name of the problem, the assessment types, the text of the question, the field where you'll enter your response, and the **Save** and **Submit** buttons.

Problem name
CENSORSHIP IN THE LIBRARIES

Open Response	Assessments:	Self	Peer	AI
---------------	--------------	------	------	----

Question text

[Hide Question](#)

"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 if I have the right to remove that book from the shelf -- that work I abhor -- 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

In the response field below, write a persuasive essay reflecting your views on censorship in libraries. Do you believe that certain materials, such as books, music, and movies, should be removed if they are judged offensive? Support your position with convincing arguments from your own experience, observations, and/or reading.

When you've entered your response, click **Submit**. (You can also click **Save** if you need to stop working on your response for a while and come back to it later.) After you click **Submit**, the rubric for this question will appear, and you'll give yourself a score based on the rubric.

After you finish your self assessment, the response will move on to AI and peer assessments. You can check for your score at any time on the **Open Ended Panel** tab. Allow a few days for the AI and peer responses to be completed.

Student's response

Response

No, I do not think that books, magizines, movies and music should be removed. However, I do think that the books, magizines, movies, and music should be monitred by age. The reason why is because younger children sould not be reading , listening, or watching material that is for adults or teenagers. They way these type of things can be monitred is by the child's library card. The card could have their name, age, grade, and date of birth. If there is a certain book, magizine, movie, and music that the child want a parent has to be there with them. The parent needs to show a form of identification so that the librian knows that the parent is of age. Teenagers should also be moniterd. Event though they are almost adults. Some parents not agree with some of the material. So, if their parents does not want

If an open response assessment asks you to submit a file, you'll also see a button that you'll click to upload your file.

The screenshot shows an edX problem page titled "L15 PROBLEM 6". At the top left is a "Open Response" button. To its right are "Assessments:" buttons for "Self" and "Peer". Below these is a "Hide Question" link. The main text area says: "In this problem you will be given the opportunity to get feedback on your graph from your classmates. In exchange for the feedback, you must give feedback to at least 3 other submissions." A large text input field is labeled "Response". At the bottom left are "Choose File" and "No file chosen" buttons, with "Choose File" circled in red. Below these are "Save" and "Submit" buttons.

The *assessment types* can include *self assessment*, *peer assessment*, and *artificial intelligence (AI) assessment*. The assessment types run in the order in which they appear in the problem.

- In a self assessment, you assess your response according a rubric that the instructor has created. For more information, see *Self Assessment*.
- In a peer assessment, you grade responses that your peers have submitted while several of your peers grade your response. For more information, see *Peer Assessment*.
- In an AI assessment, a computer algorithm grades your response. For more information, see *Artificial Intelligence (AI) Assessment*.

An open response assessment problem doesn't have to use all assessment types. For example, one problem may use self assessment and AI assessment, while another problem may use self assessment and peer assessment, and another problem may use only peer assessment.

You'll answer open response assessment problems in much the same way that you answer other problems. For more information about how to submit responses, see *Submit a Response*.

When you submit a response to an open response assessment, the next step depends on the type of assessment that the problem uses. For more information, see *Self Assessment*, *Peer Assessment*, and *Artificial Intelligence (AI) Assessment*.

After you submit your response, your score will be available shortly - sometimes within a few minutes. For information about how to access your score after your response has been graded, see *Access Scores and Feedback*.

If you want to experiment with open response assessments, you can try out the open assessment problems in the [edX Demo](#) course. To get started, go to the [Self-Assessed Essay](#) unit, and then enter a response in the **Response** field under the question. You can enter your own response, or you can use one of the sample responses in the [Sample Answers](#) unit.

9.3.2 Submit a Response

Submitting a response is slightly different if you’re submitting a written response or uploading a file.

1. Enter the response that you want to submit.
 - If you’re submitting a written response, type your response in the **Response** field.
 - If you’re uploading a file, click **Choose File** under the **Response** field. In the dialog box that opens, select the file that you want to upload, and then click **Open**.
2. 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**. An “Answer saved, but not yet submitted” message appears directly under the **Save** and **Submit** buttons.

After you submit your response, the assessment types start running in the order in which they appear in the problem. For more information, see *Self Assessment*, *Peer Assessment*, or *Artificial Intelligence (AI) Assessment*.

9.3.3 Self Assessment

Note: You can delete this section if your ORA problem doesn’t use self assessments.

In a self assessment, the rubric for the problem appears below your response immediately after you submit the response. You then assess your response based on the rubric.

Perform a Self Assessment

1. Submit a response to a self-assessed ORA problem.
2. When the rubric appears, compare your response with the rubric, and select the option that you think is appropriate for each category.

Rubric

Select the criteria you feel best represents this submission in each category.

Writing Applications

- 0 points : The essay loses focus, has little information or supporting details, and the organization makes it difficult to follow.
- 1 points : The essay presents a mostly unified theme, includes sufficient information to convey the theme, and is generally organized well.

Language Conventions

- 0 points : The essay demonstrates a reasonable command of proper spelling and grammar.
- 1 points : The essay demonstrates superior command of proper spelling and grammar.

Submit assessment

3. Click **Submit assessment**.

Your response appears, and you can see the scores that you gave yourself.

9.3.4 Peer Assessment

Note: You can delete this section if your ORA problem doesn't use peer assessments.

In a peer assessment, several students in the course grade your response while you grade other students' responses. You have to grade a number of your peers' responses before you receive your score. (After you grade the minimum number of responses required to receive your score, you can grade as many additional responses as you want.)

After you submit your response for grading, the following message appears under your response.

Your response has been submitted. Please check back later for your grade.

Warning: In peer assessments, the **due date** is the date by which you must not only submit your own response, but finish grading the required number of your peers' responses.

Peer Grading Interface

The area where you'll grade 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.

The screenshot shows the 'Peer Grading' interface. At the top, it says 'Peer Grading'. Below that is a section titled 'INSTRUCTIONS' with the text: 'Here are a list of problems that need to be peer graded for this course.' A table follows, listing two problems:

Problem Name	Due date	Graded	Available	Required	Progress
Peer-Graded Essay	No due date	3	960	0	<div style="width: 100%;"><div style="width: 100%;"> </div></div>
Milankovitch Cycle - Short Answer Response	No due date	3	825	3	<div style="width: 25%;"><div style="width: 100%;"> </div></div>

Perform a Peer Assessment

Warning: In peer assessments, the **due date** is the date by which you must not only submit your own response, but finish grading the required number of your peers' responses.

Performing a peer assessment has several steps. You can find detailed instructions for each step below.

1. *Step 1: Access responses from other students*, either in the body of the course or from the **Open Ended Console** page.
2. *Step 2: Learn to grade* (this process is called *calibration*).
3. *Step 3: Grade responses* from other students.

Step 1: Access responses from other students

Note: Modify the content in this section according to your course. For example, if your students can only grade by using the **Open Ended Console** page, change the introductory sentence below, and delete the second and third bullets.

Note 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 haven't submitted a response or no responses are available for grading, you see a yellow message in the interface.

Peer Grading

INSTRUCTIONS

Here are a list of problems that need to be peer graded for this course.

You currently do not have any peer grading to do. In order to have peer grading to do:

- You need to have submitted a response to a peer grading problem.
- The instructor needs to score the essays that are used to help you better understand the grading criteria.
- There must be submissions that are waiting for grading.

There are several ways to access other students' responses, depending on the way that the course is set up.

- Through the **Open Ended Console** page. This option is always available for every course. To access the **Open Ended Console** page, click the **Open Ended Panel** tab at the top of any page in the course. When you see the list of problems that have responses available to grade, click the name of the problem that you want to open it.

Peer Grading

INSTRUCTIONS

Here are a list of problems that need to be peer graded for this course.

Problem Name	Due date	Graded	Available	Required	Progress
Peer-Graded Essay	No due date	3	960	0	<div style="width: 100%;"><div style="width: 100%;"> </div></div>
Milankovitch Cycle - Short Answer Response	No due date	3	825	3	<div style="width: 65%;"><div style="width: 100%;"> </div></div>

- Through the courseware, in a specific unit. This option is only available if the instructor has included a peer grading interface for the problem in the body of the course. To access responses in the courseware, go to the unit that contains the open response assessment problem. Scroll down past the response that you submitted until you see the peer grading interface that appears below the problem.

PEER-GRADED ESSAY						
Open Response		Assessments: Peer				
Edit Question Hide Question						
<h3>Censorship in the Libraries</h3> <p>"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 if I have the right to remove that book from the shelf -- that work I abhor -- 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</p> <p>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.</p>						
Edit Response Hide Response						Response
<p>No, i do not think that books, magizines, movies and music should be removed. However, I do think that the books, magizines, movies, and music should be monitored by age. The reason why is because younger children should not be exposed to mature content. Listening, or watching material that is not appropriate for their age group can lead to negative outcomes. It's important to protect children from harmful content while still allowing them to explore different genres and styles of media.</p>						
Peer Grading						
INSTRUCTIONS						
Here are a list of problems that need to be peer graded for this course.						
Problem Name	Due date	Graded	Available	Required	Progress	
Peer-Graded Essay	No due date	6	773	0	<div style="width: 100%; background-color: #f0ad4e; height: 15px;"></div>	

- Through the courseware, in a separate section. This option may not be available for your course. If it is, you'll see the section for peer grading in the course accordion on the left side of your screen. For example, MIT's 6.00x: Introduction to Computer Science and Programming course has a separate section that holds all the course peer grading interfaces. To access peer grading for a problem, you click the problem name.

The screenshot shows the edX course interface. At the top, there is a navigation bar with tabs: Courseware (which is selected), Updates & News, Calendar, Wiki, Discussion, Progress, and Open Ended Panel. On the left, there is a sidebar with a tree view of course content: Overview, Week 1, Week 2, Week 3, Week 4, Week 5, Peer Grading Panel (which is expanded), Final Exam. Under Peer Grading Panel, there are five items: Peer Grading L12 Problem 5, Peer Grading L12 Problem 6, Peer Grading L12 Problem 7, Peer Grading L15 Problem 6, and Peer Grading L15 Problem 7. A yellow callout box highlights the Peer Grading L12 Problem 5 item. The main content area is titled "Peer Grading". It contains a section titled "INSTRUCTIONS" with the text: "Here are a list of problems that need to be peer graded for this course." Below this, a yellow box contains the text: "You currently do not have any peer grading to do. In order to have peer grading to do:" followed by a bulleted list: "• You need to have submitted a response to a peer grading problem.", "• The instructor needs to score the essays that are used to help you better understand the grading criteria.", and "• There must be submissions that are waiting for grading." Navigation arrows are located at the bottom of both the sidebar and the main content area.

Step 2: Learn to grade

Before you grade your peers' responses, you must learn to grade the same way that an instructor would. In this process, called *calibration*, you'll grade several responses that an instructor has already graded. If your grading is similar to the instructor's, you can begin grading other students' responses to the question.

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**.

The screenshot shows a feedback page titled "HOW DID I DO?". The text on the page reads: "The score you gave was: 1. The instructor score is: 1" and "Your score matches the instructor score!". Below this, it says "Instructor Scored Rubric: Writing Applications : 0 | Language Conventions : 1 |". At the bottom, there is a "Continue" button.

HOW DID I DO?

The score you gave was: 0. The instructor score is: 1

You may want to review the rubric again.

Instructor Scored Rubric:
Writing Applications : 0 | Language Conventions : 1 |

Continue

When you click **Continue**, the next student response appears for you to grade, and you see a yellow **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.

Step 3: Grade responses

When you grade a peer assessment response, you can not only select options in the rubric, 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 apply to the response, as you did in the calibration process.

If you have concerns about the response, you can select other options to flag the response for instructor review. You don't have to fill out the rubric before you select these options.

- If you aren't sure how to grade the response, select the **I am unsure about the scores I have given above** check box.
- If the response is offensive, or if you suspect that it contains plagiarized material, select the **This submission has explicit, offensive, or (I suspect) plagiarized content** check box.

2. Under **Written Feedback**, write a comment about the score that you gave the response.
3. Click **Submit**. You see a **Successfully saved your feedback** message at the top of the screen, and the next response opens.
4. 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 *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 that you want to continue grading.

Note: When a response opens for you to grade, it leaves the current “grading pool” that other instructors or students are grading from, which prevents other instructors or students from grading the response while you are working on it. If you do not submit a score for this response within 30 minutes, the response returns to the grading pool (so that it again becomes available for others to grade), even if you still have the response open on your screen.

If the response returns to the grading pool (because the 30 minutes have passed), but the response is still open on your screen, you can still submit feedback for that response. If another instructor or student grades the response after it returns to the grading pool but before you submit your feedback, the response receives two grades.

If you click your browser's **Back** button to return to the problem list before you click **Submit** to submit your feedback for a response, the response stays outside the grading pool until 30 minutes have passed. When the response returns to the grading pool, you can grade it.

9.3.5 Artificial Intelligence (AI) Assessment

Note: You can delete this section if your ORA problem doesn't use AI assessments.

In an AI assessment, an instructor grades a sample set of student responses to the open response assessment problem. A machine learning algorithm then creates a model based on the instructor's scores and grades the remaining students' responses.

After you submit your response to an AI assessment, the following message appears under your response.

Your response has been submitted. Please check back later for your grade.

Depending on the time that it takes for the instructor to grade a sample set of responses, you may receive your grade within minutes, or you may have to wait a few days. You won't receive a notification when your score is ready, so keep checking back.

For more information about accessing your scores, see *Access Scores and Feedback*.

9.3.6 Access Scores and Feedback

Note: Modify the text in this section to apply to your course.

For *self assessments*, the score that you give yourself appears as soon as you submit the score.

For *peer assessments* and *AI assessments*, you'll access your scores 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. The status for each problem is either **Waiting to be Graded** or **Finished**.
4. If **Finished** appears in the **Status** column for the problem you want, click the name of the problem 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.

AI GRADED ESSAY

Open Response

Assessments: AI

Show Question

Response

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 is because younger children should not be reading, listening, or watching material that is for adults or teenagers. They way these type of things can be monitored is 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 want a parent has 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. Even though they are almost adults. Some parents not agree with some of the material. So, if their parents does not want them reading certain things their should be stickers on their cards. Even though there are plenty of opinions about what a persons child is reading, watching, or listening to I think there should be certain ways to watch what children are reading, watching or listening to.

New Submission

Submitted Rubric

Toggle Full Rubric

Scored rubric from grader 1

Writing Applications

✓ 1 points : The essay presents a mostly unified theme, includes sufficient information to convey the theme, and is generally organized well.

Language Conventions

✓ 1 points : The essay demonstrates superior command of proper spelling and grammar.

For peer assessments, you can also see the written feedback that your response received from different graders.

Open Response

Assessments: Peer

Show Question

Response

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 is because younger children should not be reading, listening, or watching material that is for adults or teenagers. They way these type of things can be monitored is 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 want a parent has 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. Even though they are almost adults. Some parents not agree with some of the material. So, if their parents does not want them reading certain things their should be stickers on their cards. Even though there are plenty of opinions about what a persons child is reading, watching, or listening to I think there should be certain ways to watch what children are reading, watching or listening to.

New Submission

Submitted Rubric

Toggle Full Rubric

Scored rubric from grader 2

Writing Applications

✓ 1 points : The essay presents a mostly unified theme, includes sufficient information to convey the theme, and is generally organized well.

Language Conventions

✓ 1 points : The essay demonstrates superior command of proper spelling and grammar.

You answer the question and gave support.

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.

Feedback not available yet

You need to peer grade 1 more submissions in order to see your feedback.

You have graded responses from 2 students, and 3 students have graded your submissions.

You have made 1 submissions.

For more information about grading peer assessments, see *Peer Assessment*.

9.3.7 Resubmitting a Response

Note: You can delete this section if you don't allow students to submit multiple responses.

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.