



Getting Started with Studio

Create an Account	5
Create a New Course.....	6
Import a Course.....	6
Add Collaborators.....	7
Add Course Information	8
Add Schedule Information	8
Add Enrollment Information	8
Add a Course Overview.....	8
Add a Course Catalog Video	9
Add Weekly Time Requirements Information.....	11
Add a Syllabus, Schedule, or Course Handouts.....	11
Upload a File to the Assets Tab.....	11
Add Items to the Handouts Sidebar	12
Establish a Grading Rubric	13
Set Grade Brackets	13
Set a Grace Period.....	13
Create Assignment Types	14
Set Assignment Types in the Course.....	15
Troubleshooting.....	16
Build a Learning Sequence in Studio	16
Create a Discussion	19
Create an HTML Component.....	20
Create a Basic HTML Component	20
Link to Course Units.....	23
Import from LaTeX	25
Create a Problem.....	26
Create a Video.....	28
Set Content Release Dates and Visibility	30
Section	30
Subsection	31
Unit.....	31
View Course Content	31
Invite Students to Register	32

APPENDIX A: Template for Course Overview box	33
APPENDIX B: Sample Student Login Guide	34
APPENDIX C: Problem Types	36
Option Response.....	36
Sample Problem	36
Problem Code.....	36
Template.....	37
XML Attribute Information	37
Multiple Choice.....	38
Sample Problem	38
Problem Code.....	39
Template.....	39
XML Attribute Information	40
Checkbox	41
Sample Problem	41
Problem Code.....	41
Template.....	42
String Response.....	42
Sample Problem	42
Problem Code.....	43
Template.....	43
XML Attribute Information	43
Numerical Response	44
Sample Problem	44
Problem Code.....	45
Templates.....	46
XML Attribute Information	47
Formula Response	49
Sample Problem	49
Problem Code.....	50
Template.....	51
XML Attribute Information	51
Image Response	53
Sample Problem	53
Problem Code.....	54

Template.....	54
XML Attribute Information	54
Custom Response.....	55
Sample Problem	55
Problem Code.....	56
Templates.....	57
Chemical Equation Response	58
Sample Problem	58
Problem Code.....	59
Schematic Response.....	60
Sample Problem	60
Problem Code.....	61

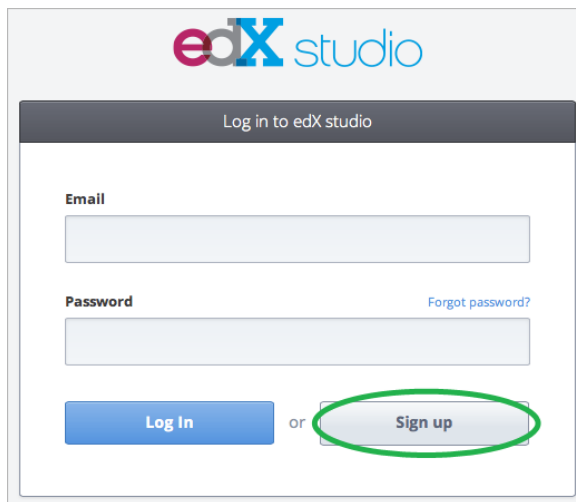
Getting Started with Studio

To begin using Studio, create an account with a unique user ID and password.

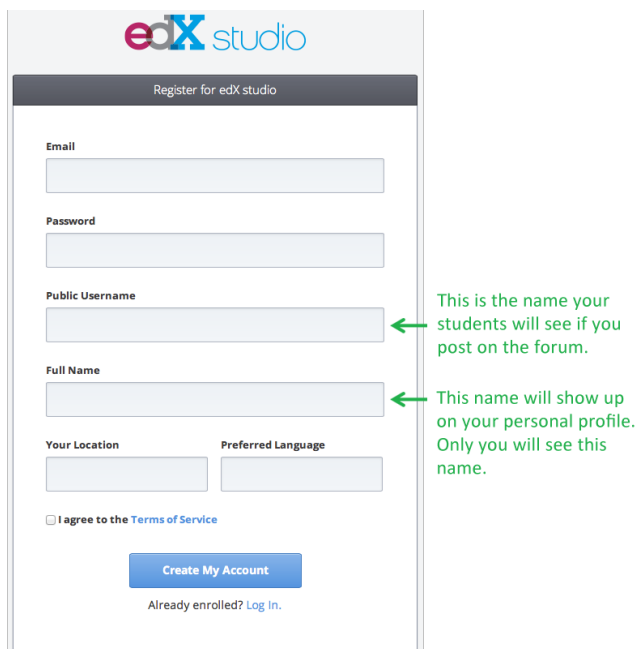
Create an Account

To create an account in Studio:

1. Go to <http://studio.edge.edX.org>. The following page opens.



2. Click **Sign up**. The following screen appears.



This is the name your students will see if you post on the forum.

This name will show up on your personal profile. Only you will see this name.

3. In the **Email** box, type your e-mail address.
4. In the **Password** box, type a unique password.

5. In the **Public Username** field, type the name you want students to see when you post on the user forum. (Most websites call this the "user name".)
6. In the **Full Name** box, type your full name. This name only appears on your user dashboard. Your students will not see this name.
7. Click **Create My Account**. After you click this button, you will receive an activation e-mail message. To finish creating your account, click the link that you receive in the activation e-mail.

Create a New Course

Note Enter your new course information carefully. This information becomes part of the URL for your course. The URL is very difficult to change after the course is created.

1. Sign in to your account. The **My Courses** page opens.
2. Click **New Course**. A screen opens.
3. 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.
4. For **Organization**, enter the name of your university. Do not include whitespace or special characters.
5. 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".

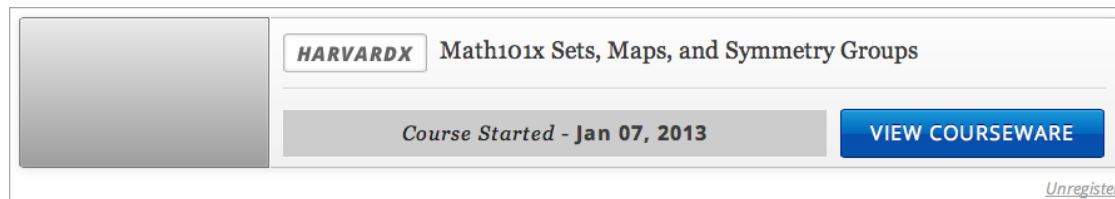
6. Click **Save**.

When you click **Save**, your new course opens to the **Courseware** tab. To view your course on your dashboard, click the Home icon in the top left corner of the screen.

Example: If you enter the following:

- **Course Name:** Sets, Maps and Symmetry Groups
- **Organization:** HarvardX
- **Course Number:** Math101x

The new course appears as follows on the dashboard.



Note that although the start date defaults to the current date, your course will not be advertised, so it will not be visible to the general public.

Import a Course

Note This feature should be used with caution! Importing a new course will delete all course content currently associated with your course and replace it with the contents of the uploaded file. **Importing a course cannot be undone.**

If you have already started using edXML to write a course, you can import your content into the edX studio system.

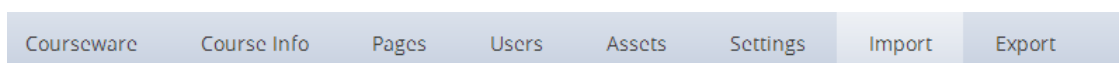
File uploads must be as a GZipped-compressed TAR file containing, at a minimum, a course.xml file within a course data directory. The tar.gz file must be named the same as the course data directory for the import process to succeed.

If your course has any problems with auto-generated url_name nodes, re-importing your course could cause the loss of student data associated with those problems.

If your course uses legacy layout structures, it might not be editable using the edX studio interface, although it will probably display in edX edge correctly. If you would like to ensure that your course is completely editable, ensure that all of your material is embedded in a chapter->sequential->vertical.

To import a course:

1. Click the **Import** tab.



2. Under **Course to Import**, click **Choose File**.
3. Locate the file that you want, and then click **Open**.

Add Collaborators

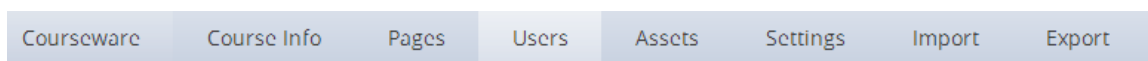
We have support for rudimentary collaborative editing of a course. Users must have registered at edge.edx.org or studio.edge.edx.org, and must have activated their account via the email link. If a user is not found, you will be notified.

Before you add a new user, consider the following.

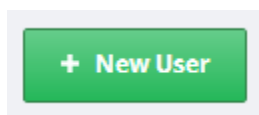
- Invited users have full permissions to edit your course, including deleting content created by anyone else.
- Invited users cannot currently grant new permissions on the course.
- Editing conflicts are currently not managed. Thus, the state of the course might change between refreshes of the page.

To give another user permission to edit your course:

1. Click the **Users** tab.



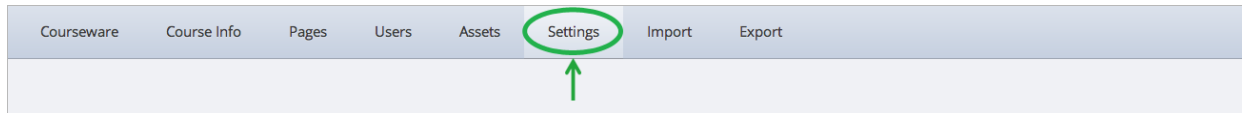
2. Click **New User**.



3. In the **email** box, type the email address of the user, and then click **Add User**.

Add Course Information

To add scheduling information, a description, and other information for your course, click the **Settings** tab at the top of the page.



Add Schedule Information

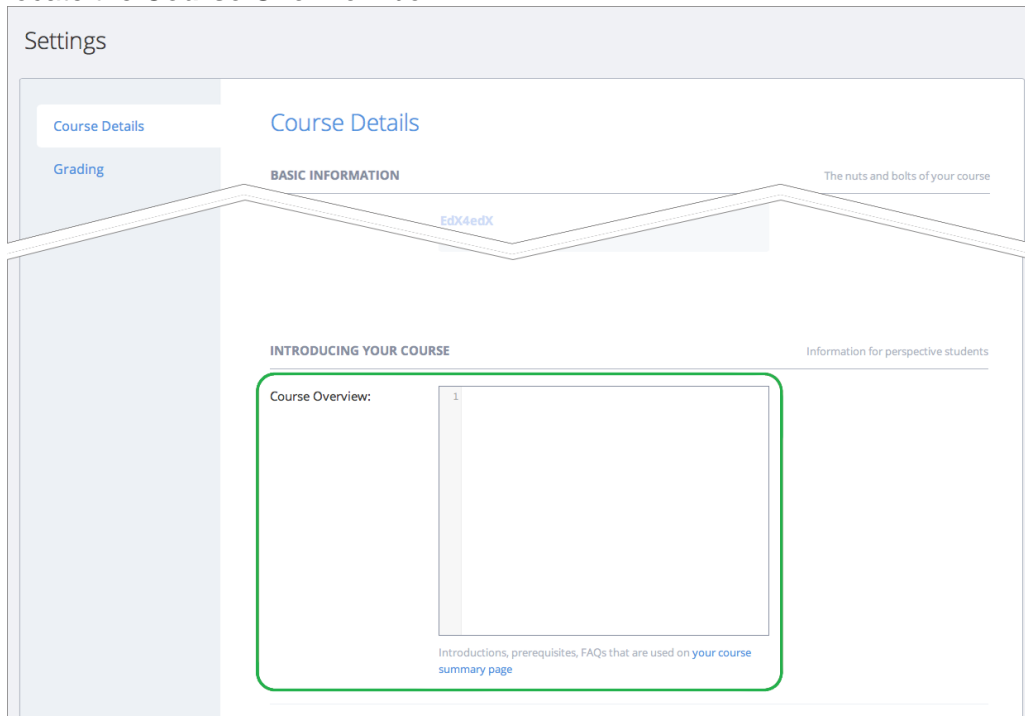
1. On the **Settings** tab, click **Course Details** in the left pane.
2. Scroll down to the **Course Schedule** section.
3. In the **Course Dates** section, enter the date you want your course to start in the **Start Date** box, and then enter the time you want your course to start in the **Start Time** box.
4. In the **Course Dates** section, enter the date you want your course to end in the **End Date** box, and then enter the time you want your course to end in the **End Time** box.

Add Enrollment Information

1. In the left pane, click **Course Details**.
2. Scroll down to the **Course Schedule** section.
3. In the **Enrollment Dates** section, enter the date you want enrollment for your course to start in the **Start Date** box, and then enter the time you want enrollment for your course to start in the **Start Time** box.
4. In the **Enrollment Dates** section, enter the date you want enrollment for your course to end in the **End Date** box, and then enter the time you want enrollment for your course to end in the **End Time** box.

Add a Course Overview

1. On the **Settings** tab, scroll down to the **Introducing Your Course** section, and then locate the **Course Overview** box.

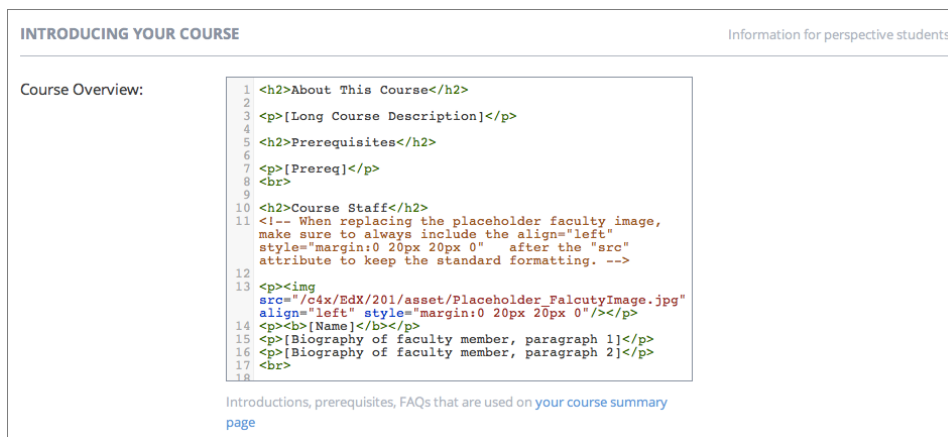


The content for this box must be formatted in XML. For a template that you can use that includes placeholders, see **Appendix A**.

2. In the **Course Overview** box, paste the XML that you just created.

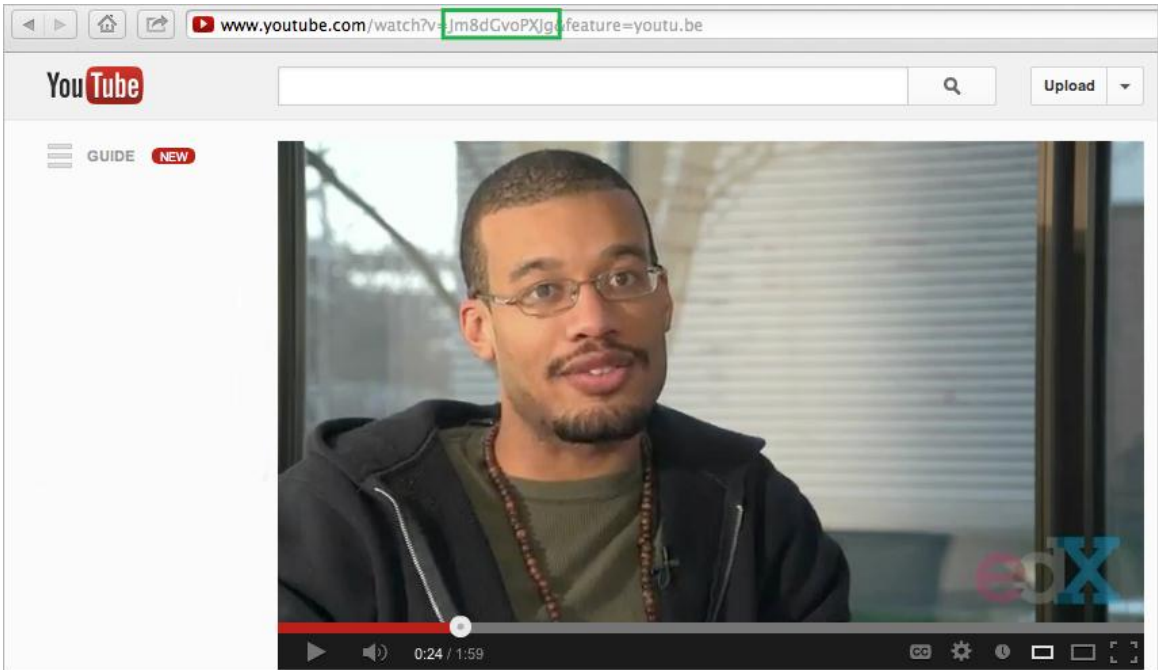
Note There is no save button. EdX Studio automatically saves your changes.

3. The following is example content for the **Course Overview** box.

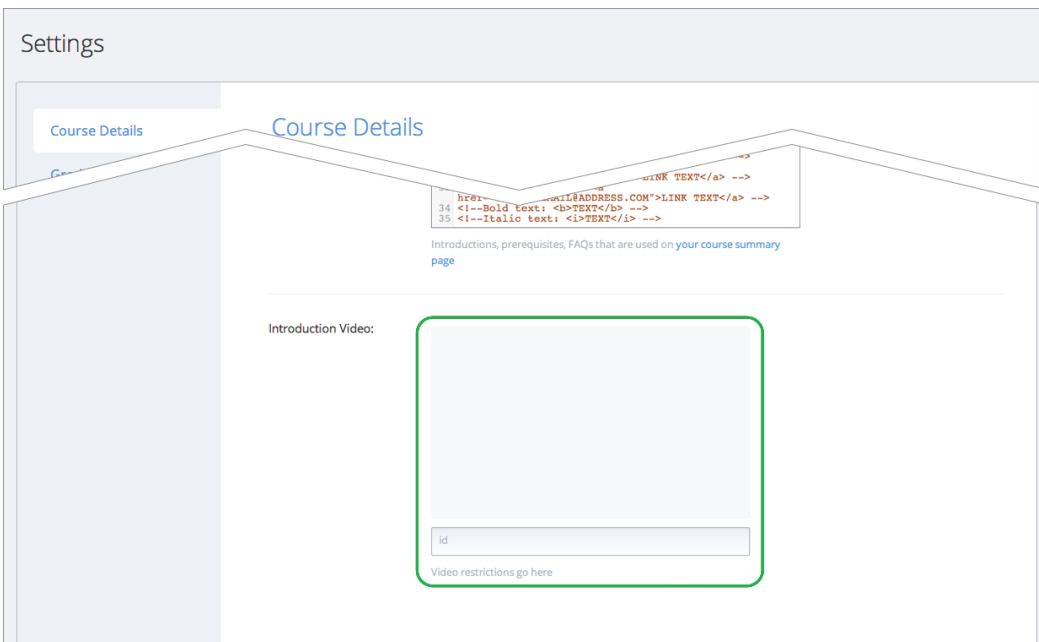


Add a Course Catalog Video

1. Upload the video that you want 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. On the **Settings** tab, scroll down to the **Introducing Your Course** section, and then locate the **Introduction Video** field. If you have not already added a video, you see a blank field above an **id** box.



3. In the **id** box, enter your video code. When you add the code, the video automatically appears in the field above the **id** box.

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

For example, your course introduction video appears as follows.



Add Weekly Time Requirements Information

1. On the **Settings** tab, scroll down to the **Requirements** section.
2. In the **Hours of Effort per Week** box, enter the number of hours you expect students to work on this course each week.

Add a Syllabus, Schedule, or Course Handouts

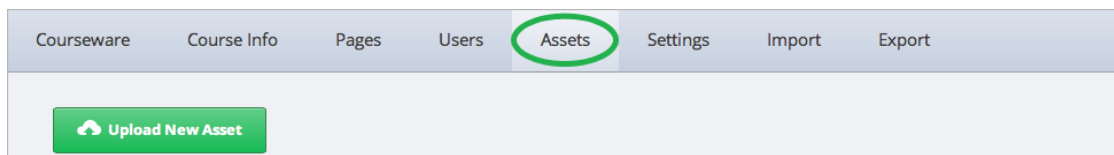
Your syllabus, schedule, and course handouts are available from the **Handouts** sidebar on the **Course Info** page. To add these items, you must create a file and upload the file to the **Assets** tab. You can then link to the file from a course page or display the file as an image.

Note You cannot upload videos to the **Assets** tab.

Upload a File to the Assets Tab


Warning: Uploading a file with the same name as a file that is already uploaded will cause the original file to be overwritten. There is not currently a warning when you try to upload a file that has the same name.

1. At the top of the page, click the **Assets** tab, and then click **Upload New Asset**.



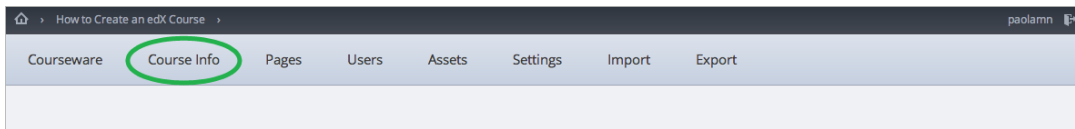
2. In the **Upload New File** dialog box, click **Choose File**.
3. In the **Open** dialog box, locate the file that you want, and then click **Open**.
4. To add another file, click **Load Another File**. To close the dialog box, click the **x** in the top right corner. The file appears on the **Assets** tab.

- Determine the URL where your newly uploaded file is located. To do this, close the upload window by clicking the **X** in the top right corner. You will see a list of your assets. In the **URL** column on the right, you can see the URL you will need to use to display an image or create a hyperlink.

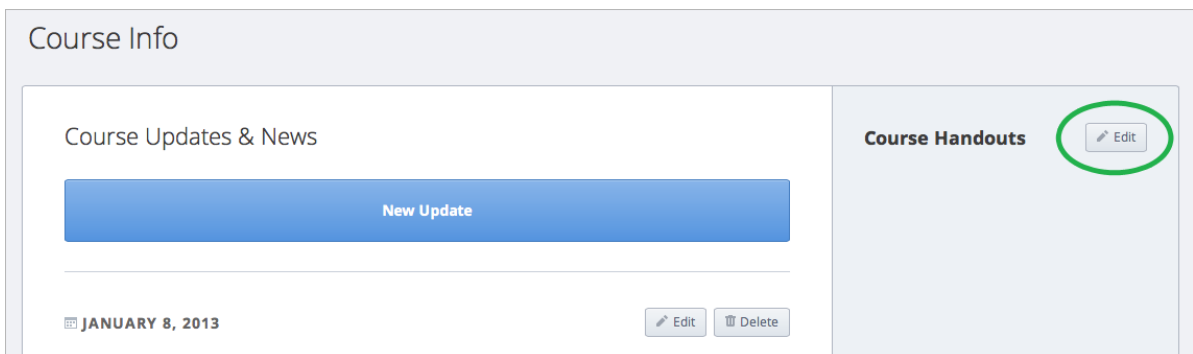
Name	Date Added	URL
 Chrysanthemum.jpg	07 February, 2013 at 06:16 PM	/c4x/MITx/Tere101/asset/Chrysanthemum.jpg

Add Items to the Handouts Sidebar

- Click the **Course Info** tab.



- Locate the **Course Handouts** sidebar on the right, and then click **Edit** next to **Course Handouts**.



- In the XML box that appears, create links to documents by entering the following syntax, where **URL OF FILE** is the URL that you noted in step 5 of **Upload a File to Your Assets Tab** and **LINK TEXT** is the text that you want to appear in the **Handouts** sidebar.

```
<p><a href="[URL OF FILE]">[LINK TEXT]</a></p>
```

To create a link to an image that you've uploaded, use the following syntax, where **URL OF FILE** is the URL that you noted in step 5 of **Upload a File to Your Assets Tab**.

```
<p></p>
```

For example, to create a link to the **HTML template for the "About" page** document whose URL is `/c4x/edX/edX101/asset/AboutPage_Template.txt`, use the following code.

```
<p><a href="/c4x/edX/edX101/asset/AboutPage_Template.txt">HTML Template for the "About" page</a></p>
```

For example, to create a link to the **CourseImage.jpg** file whose URL is `/c4x/edX/edX101/asset/CourseImage.jpg`, use the following code.

```
<p></p>
```

When you use this code, the following image appears.



Establish a Grading Rubric

Course grading can be split into three steps.

1. Set grade brackets.
2. Describe assignment types.
3. Setting Assignments in the courseware

Set Grade Brackets

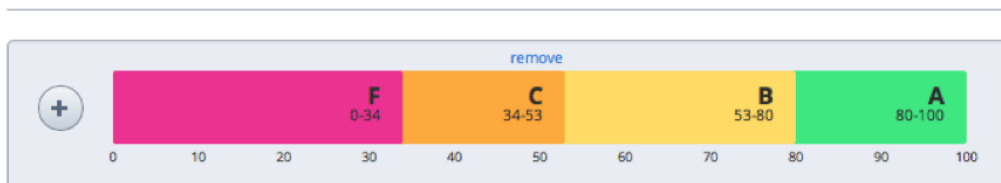
To set the thresholds for course grades:

1. Click the **Settings** tab.
2. In the left pane, click **Grading**.
3. Under **Overall Grade Range**, click and drag the dividing line between grade divisions to move each threshold up or down.

Note The default grade divisions are **Pass** and **Fail**. To add more grade divisions (such as A, B, C, or D), click the plus sign (+).

To remove a grade division, hover the mouse over the grade division, and then click the **Remove** link that appears above the grade division.

OVERALL GRADE RANGE



Set a Grace Period

1. Click the **Settings** tab.
2. In the left pane, click **Grading**.
3. Under **General Grading**, enter a value in the **Grace Period on Deadline** box.

Create Assignment Types

By default, Studio includes four assignment types for your course when you create the course: homework, lab, midterm exam, and final exam. You decide the weight of the student's total grade for each assignment type.

The example below has two types of assignments. The overall course grade is broken down as 40% Homework and 60% Final Exam. There are eight Homework assignments, and the grader will omit the lowest-scored assignment from the final grade. Thus, the seven remaining Homeworks are each worth $40/7 = 5.8\%$ of the final grade.

ASSIGNMENT TYPES

Assignment Type Name:	<input type="text" value="Homework"/>	<small>e.g. Homework, Labs, Midterm Exams, Final Exam</small>
Abbreviation:	<input type="text" value="HW"/>	<small>e.g. HW, Midterm, Final</small>
Weight of Total Grade:	<input type="text" value="40"/>	<small>e.g. 25%</small>
Total Number:	<input type="text" value="8"/>	<small>total exercises assigned</small>
Number of Droppable:	<input type="text" value="1"/>	<small>total exercises that won't be graded</small>
<input type="button" value="Delete"/>		

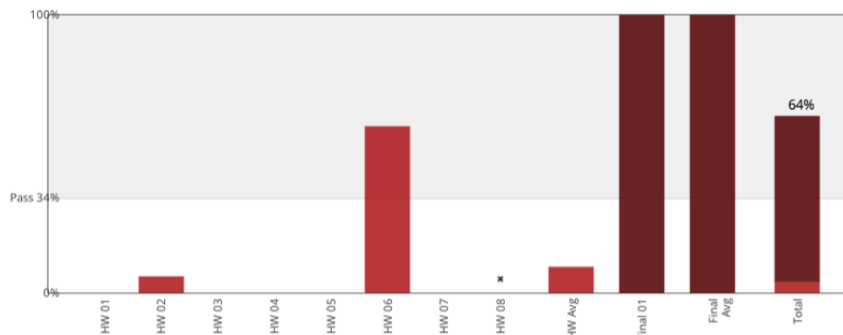
Assignment Type Name:	<input type="text" value="Final Exam"/>	<small>e.g. Homework, Labs, Midterm Exams, Final Exam</small>
Abbreviation:	<input type="text" value="Final"/>	<small>e.g. HW, Midterm, Final</small>
Weight of Total Grade:	<input type="text" value="60"/>	<small>e.g. 25%</small>
Total Number:	<input type="text" value="1"/>	<small>total exercises assigned</small>
Number of Droppable:	<input type="text" value="0"/>	<small>total exercises that won't be graded</small>
<input type="button" value="Delete"/>		

To create an assignment type, enter values in each of the following boxes.

Assignment Type Name: This is a general category of assessment (homework, exam, exercise). All assignments within an assignment type are given equal weight. This name will be visible to students.

Abbreviation: This is the short name that will appear next to an assignment on every student's **Progress** tab (see below).

Course Progress



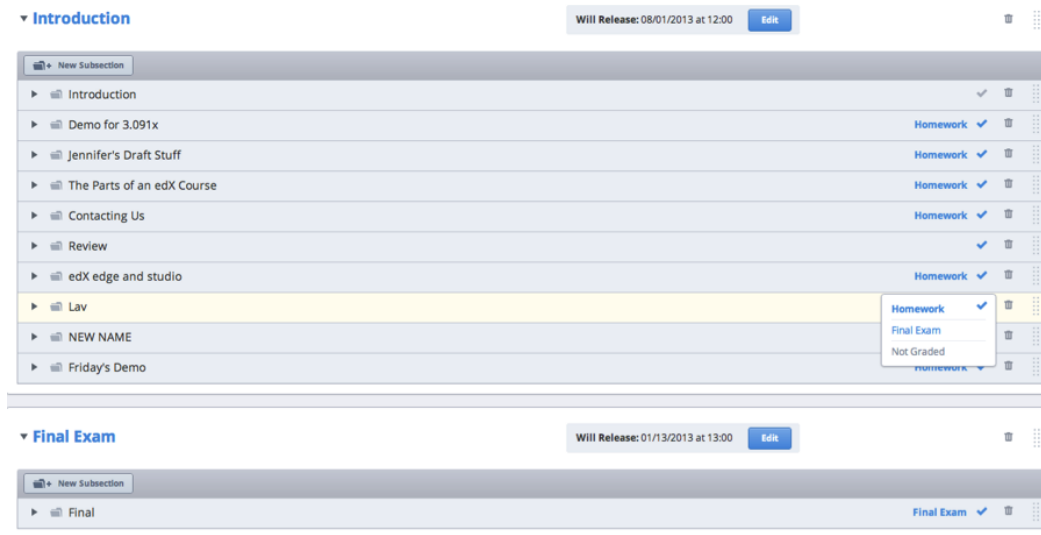
Weight of Total Grade: The assignments of a particular type together account for the percent value set in **Weight of Total Grade**.

Number of assignments and dropped assignments: The **Total Number** should equal the number of an assignment type you plan to present in your course. You can also allow students to drop a number of assignments, set in **Number of Droppable**. The grader will omit the lowest-scored assignments first.

Set Assignment Types in the Course

To set the assignment type for a subsection:

1. In Studio, click the **Courseware** tab.
2. Locate the subsection that you want.
3. On the right side of the screen, click the blue check mark for the subsection, and then select the assignment type.



Note If you change an assignment type name in the **Grading** page, be sure the assignment type names on the **Courseware** tab still match.

Troubleshooting

- In the **Weight of Total Grade** field, omit the % sign.
- Be sure that your **Weight of Total Grade** fields add up to at least 100.

Build a Learning Sequence in Studio

To build a learning sequence in Studio, create a new section, subsection, and unit. You then create components for the unit.

By default, all units are private. Students cannot see a unit until you explicitly release the subsection that contains the unit and change the unit's visibility to **Public**. For more information, see [Set Content Release Dates and Visibility](#).

To see the way unpublished material looks from the student perspective, click **Preview** in the right panel under **Unit Settings**.

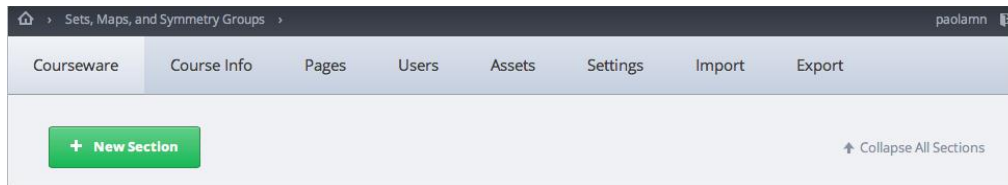
If you are working on a new draft of an already published unit, you can view the existing published version by clicking **Preview the Published Version** in the banner at the top of the screen, or you can view a preview of the new content by clicking **Preview** in the right panel under **Unit Settings**.

CAUTION!

The alpha version of Studio does not have versioning or browser refresh. 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, stale browser sessions can overwrite more recent content, so refresh your browser every time you work with a private unit or edit a draft of a public unit.

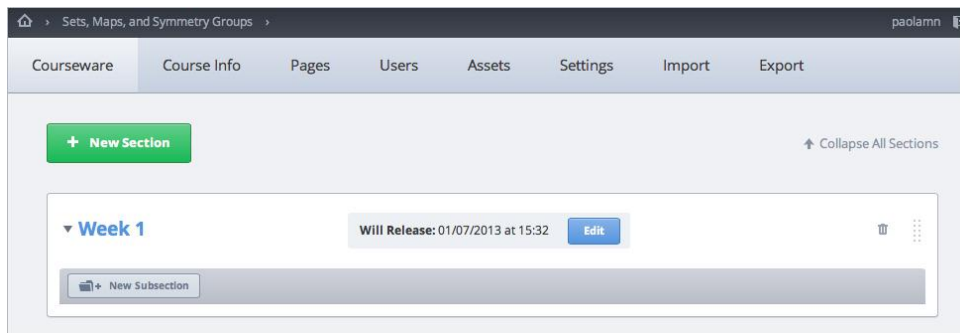
1. Sign in to Studio, and then click the course that you want.

2. Click the **Courseware** tab. You see the following screen.

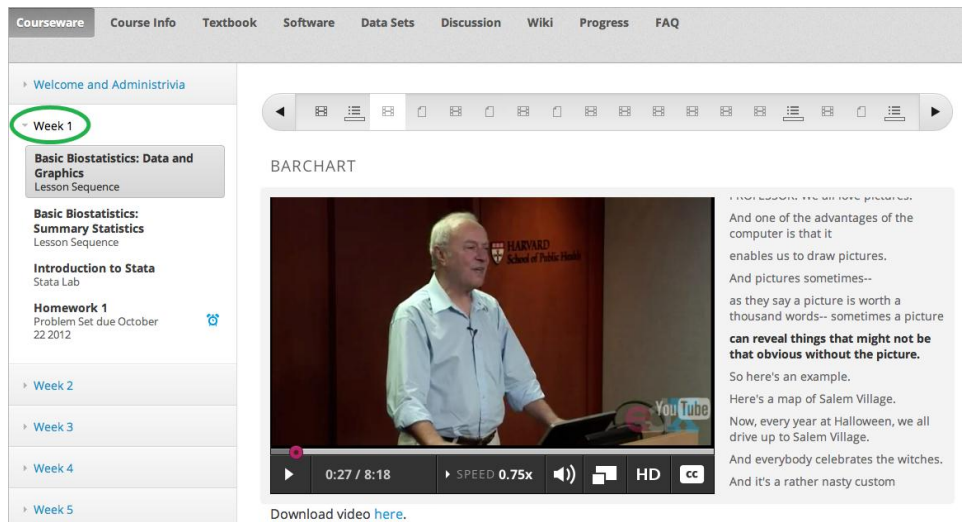


3. Click **New Section**, type a section name in the **New Section Name** box, and then click **Save**. *Note In most courses, the name of the first section is "Week 1".*

The name you enter will appear in the left pane.



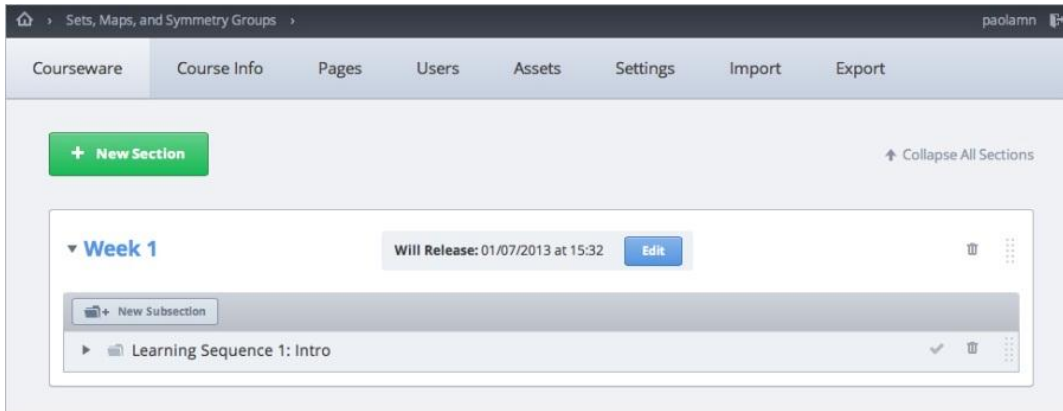
The name you enter also appears in the courseware accordion, as follows.



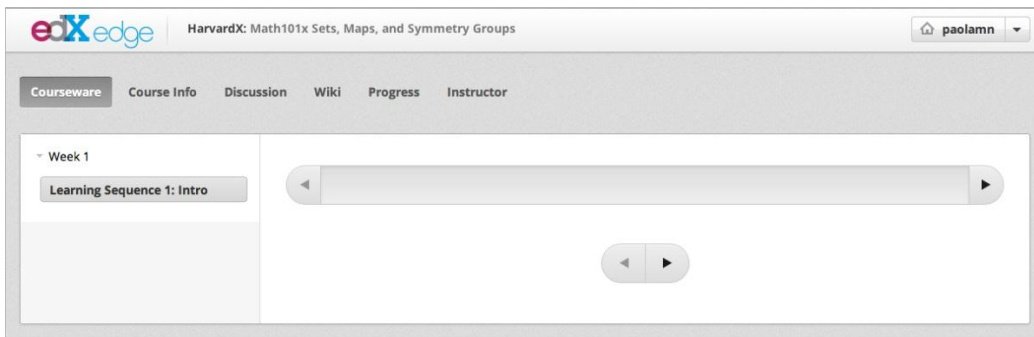
4. To create a new learning sequence or assessment in your section, click **New Subsection**.

5. In the **New Subsection** box, enter the name for this subsection, and then click **Save**.

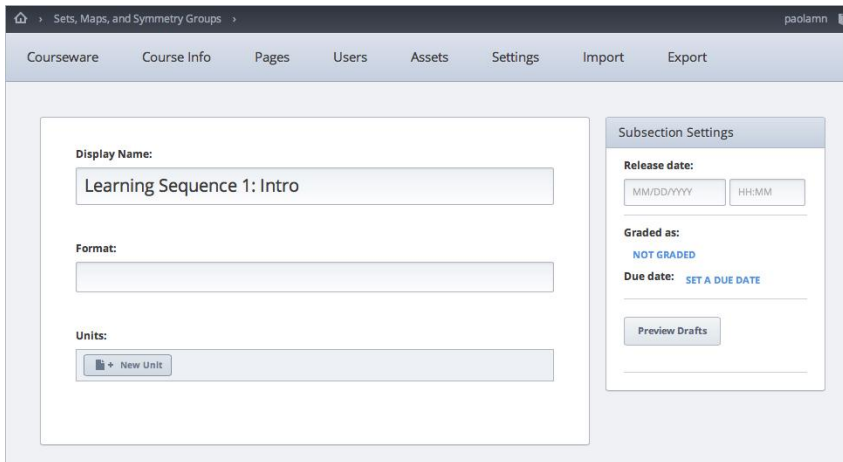
For example, if you enter **Week 1** as the section title and **Learning Sequence 1: Intro** as the subsection title, you see the following.



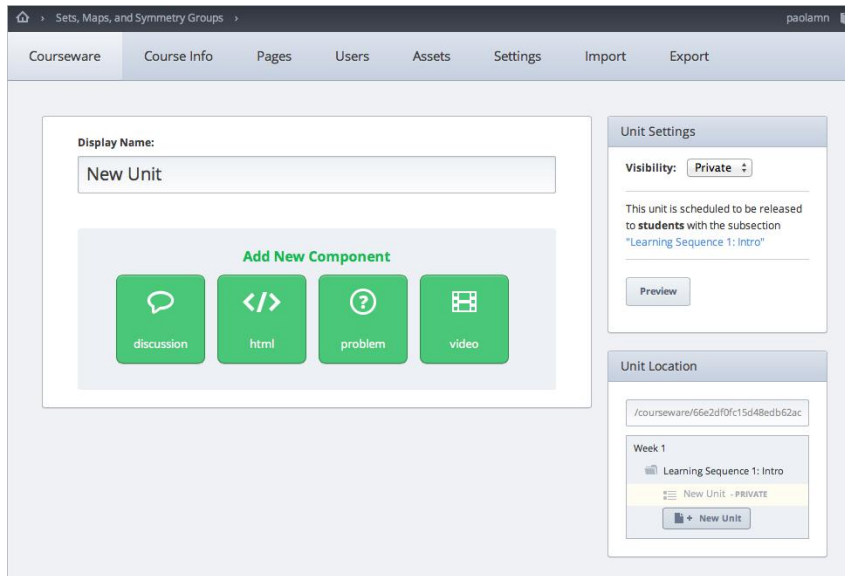
If you view your course as a student would see it, you see the following.



6. Click the new subsection that you just created. In this example, you would click **Learning Sequence 1: Intro**. You see the following screen.



Click New Unit. The following screen appears.



7. In the **Display Name** box, type the name of the unit.

Each unit has one or more components.

To create a discussion space where you or your students can post questions or participate in a discussion, click **Discussion**.

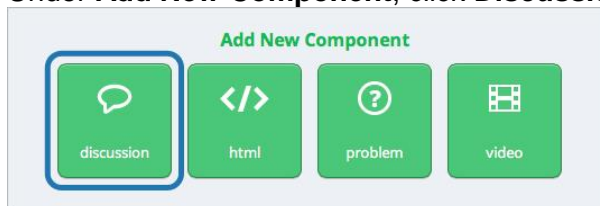
To create a component where you can add text, images, or other content, click **html**.

To create a problem for your students to solve, click **Problem**.

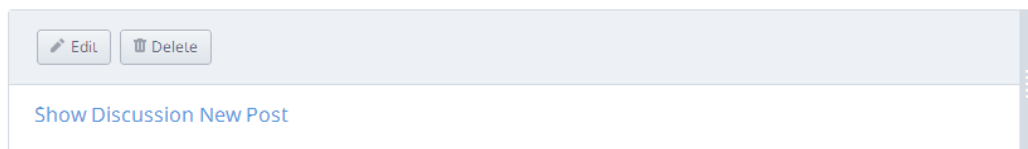
To add a video, click **Video**.

Create a Discussion

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



2. Click **Discussion Tag**. The following screen appears.



3. To start a new discussion, click **Edit**. The following screen appears.

for: Topic Level Student Visible Label

discussion_category: Week 1

display_name: Discussion Tag

id: 6002x_group_discussion_by_this

```

1 <discussion for="Topic-Level Student-Visible Label" id="6002x_group_discussion_by_this"
2 discussion_category="Week 1" />

```

Save Cancel

Create an HTML Component



The HTML component is the most basic component type. These components are the building blocks of text-based courses. They are used to add information such as text, lists, links, and images to units. For example, you can use these components between Problem components to add explanatory text. You can also use HTML components to import LaTeX code into your course.

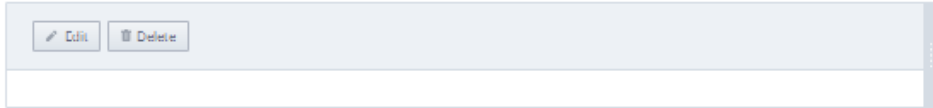
The HTML component editor has two views: Visual view and HTML view. Visual view offers you a WYSIWYG editor for editing a pre-formatted version of the text. HTML view gives you a text editor in which you can edit HTML code directly.

Note Studio processes the HTML code entered when saving it and before rendering it. Make sure that the component you created looks the way you expect live if you go back and forth between Visual and HTML view.

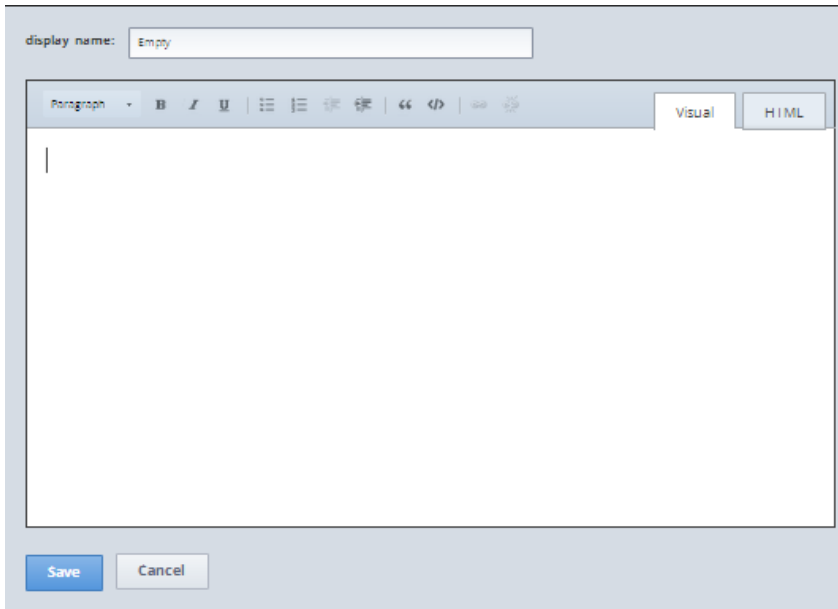
Create a Basic HTML Component

To create a basic, blank HTML component:

1. Under **Add New Component**, click **html**, and then click **Empty**. The following blank component appears.



2. In the blank component, click **Edit**. The HTML editor opens.



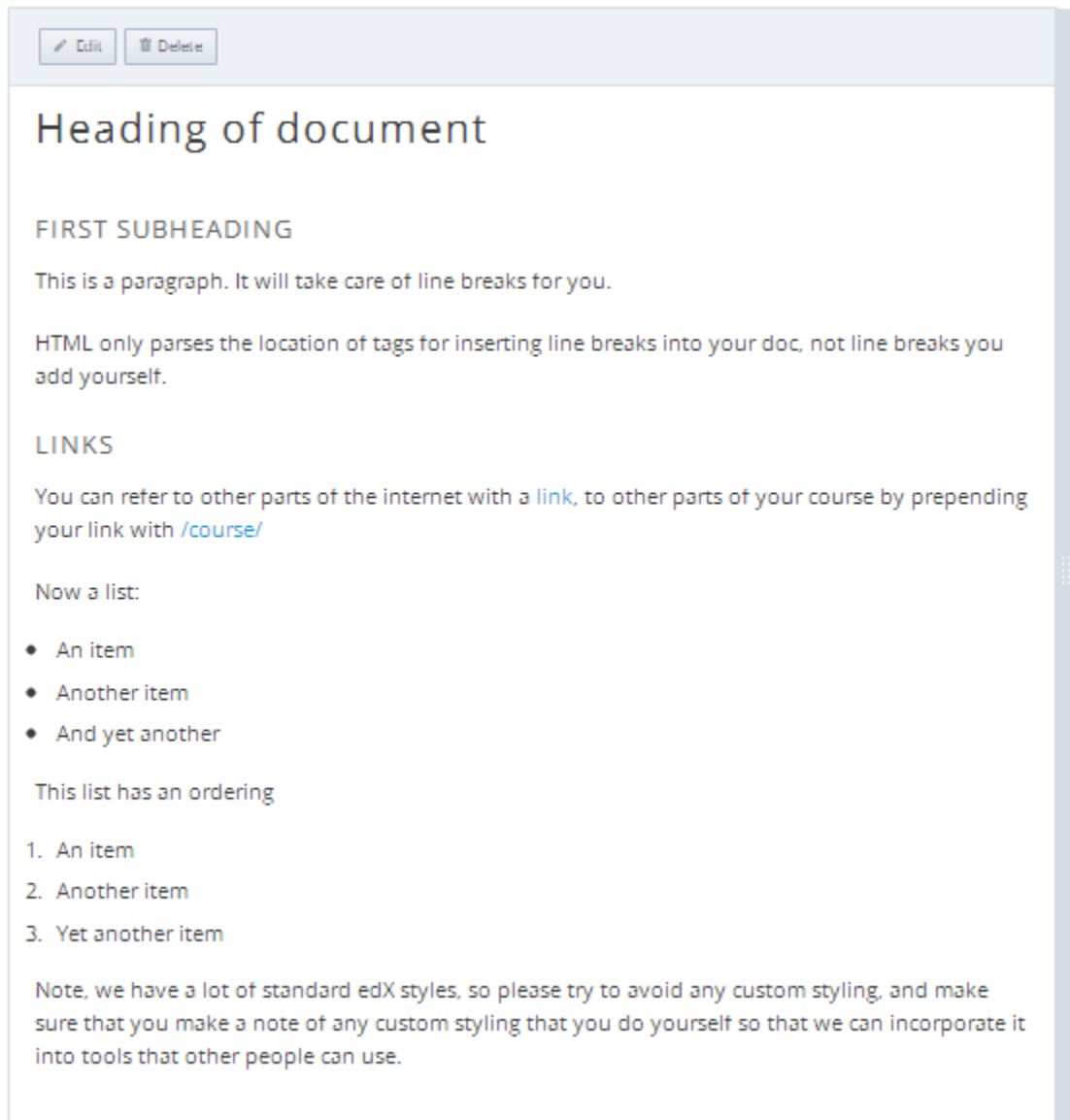
3. Enter the information that you want, and then click **Save**.

Note If you want to enter links to other pages or to images or to edit the HTML directly, switch to the **HTML** tab.

To create a basic HTML component that includes a template you can use:

1. Under **Add New Component**, click **html**, and then click **Announcement**.

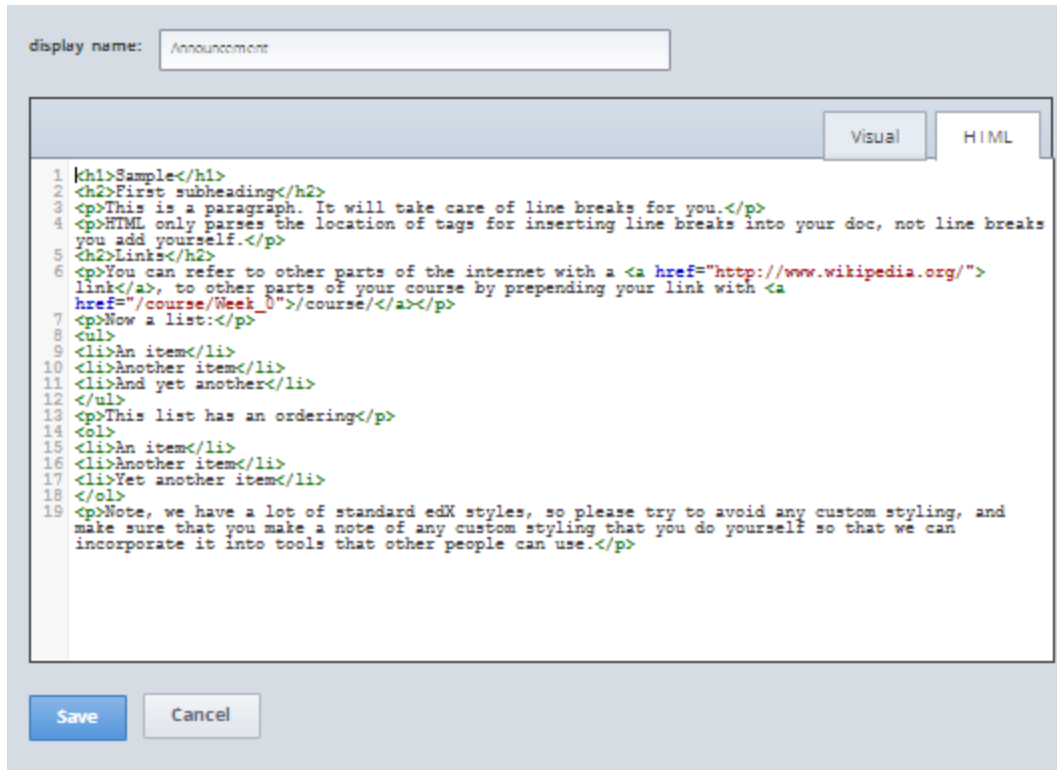
The following screen opens.



2. Click **Edit**.

The text editor opens in Visual view. Replace the template text with your announcement text.

Note If you want to enter links to other pages or to images or to edit the HTML directly, switch to the **HTML** tab.



3. Click **Save**.

Link to Course Units

To direct the student to a specific place in your own course, you must add an HTML link to that unit. To do this:

1. Determine the **relative directory** of your course.
 - a. On the **Settings** tab, click one of the three **your course URL** links that are located next to **Course Name**, **Organization**, and **Course Number**.

Course Details

BASIC INFORMATION		The nuts and bolts of your course
Course Name:	Test	This is used in your course URL, and cannot be changed
Organization:	MITx	This is used in your course URL, and cannot be changed
Course Number:	Test101	This is used in your course URL, and cannot be changed

The registration page for your course opens.

- b. In the address bar at the top of the page, locate the URL.
- c. Copy the part of the URL after “.org” and before “about”, including the forward slashes. The syntax is the following.

```
/courses/[organization]/[course_number]/[course_name]/
```

For example, for edX101: How to Create an edX Course from edX, the complete URL is the following.

```
https://edge.edx.org/courses/edX/edX101/How_to_create_an_edX_course/about
```

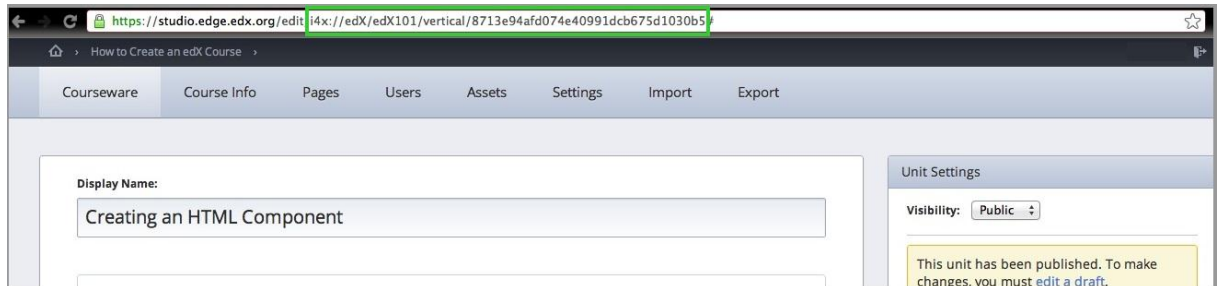
The relative directory is the following.

```
/courses/edX/edX101/How_to_create_an_edX_course/
```

2. Determine the **location ID** of the target unit. Studio generates the location ID for each unit when you create the unit. The location ID uses the following syntax.

```
i4x://<organization>/<course_number>/vertical/<url_name_of_unit>
```

To find the location ID, open the page of the unit you are trying to link to in Studio and look at the URL in the browser’s address bar. The location ID is the text in the URL after **edit**, as in the following example.



3. Open the unit that you want to link from.
4. Under **Add New Component**, click **html**, and then click **Empty**. A new, blank component appears.



5. Click **Edit**.
6. In the HTML editor that opens, click the **HTML** tab.
7. Next to the number 1, type the following. Replace **relative course directory** and **location id of unit**, and **link text** with your information.

```
<a href = "[[relative course directory]]/jump_to/[location id of unit]">[link text]</a>
```

For example, a link to the “Creating an HTML Component” unit in edX101 resembles the following.

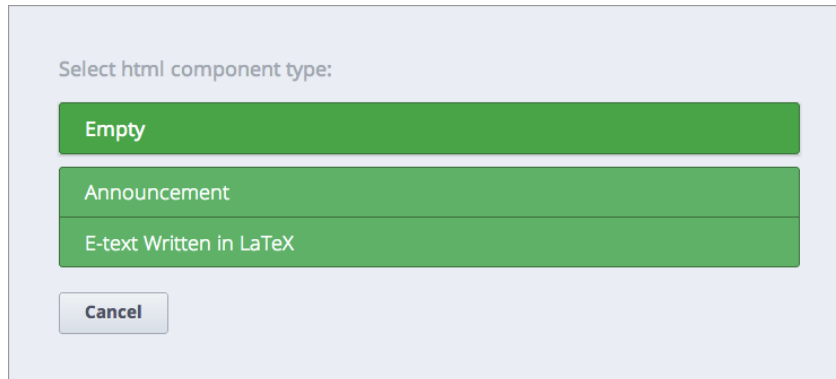
```
<a href = "/courses/edX/edX101/How_to_Create_an_edX_Course/jump_to/i4x://edX/edX101/vertical/8713e94afd074e40991dcb675d1030b5">Creating an HTML Component</a>
```


Import from LaTeX

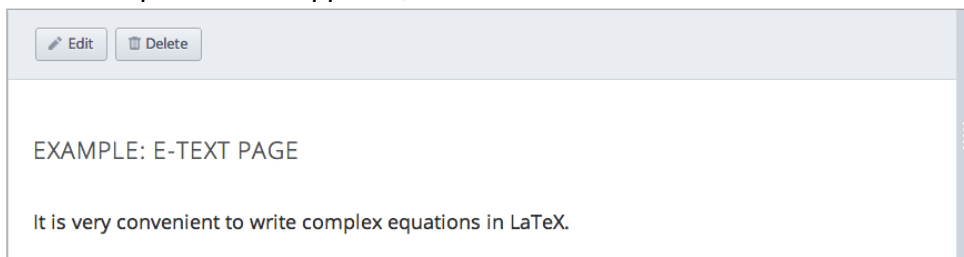
You can create an HTML component from imported LaTeX code.

Note This feature is currently under development.

1. Under **Add New Component**, click **html**, and then click **E-text Written in LaTeX**.



2. In the component that appears, click **Edit**.



3. The component editor opens. In the top left corner of the editor, click the yellow **Edit High Level Source** text.



4. In the **High Level Source Editing** screen that opens, replace the sample code with your LaTeX code.

```
HIGH LEVEL SOURCE EDITING
1 \subsection{Example of E-text in LaTeX}
2
3 It is very convenient to write complex equations in LaTeX.
4
5 \begin{equation}
6   x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
7 \end{equation}
8
9 Seize the moment.
10

Save & Compile to edX XML Save Refresh
```

5. Click **Save and compile to edX XML** to convert the LaTeX code into edX XML code.
Note Studio uses a third-party LaTeX processor to convert LaTeX code to XML. The LaTeX processor must be up and running.
6. Click **Save**. Verify that your newly created component looks the way you want it to.

Create a Problem

To add interactive, automatically graded exercises to your course content, use the Problem component. This component allows you to include an explanation that the student can see when the student clicks **Show Answer**.

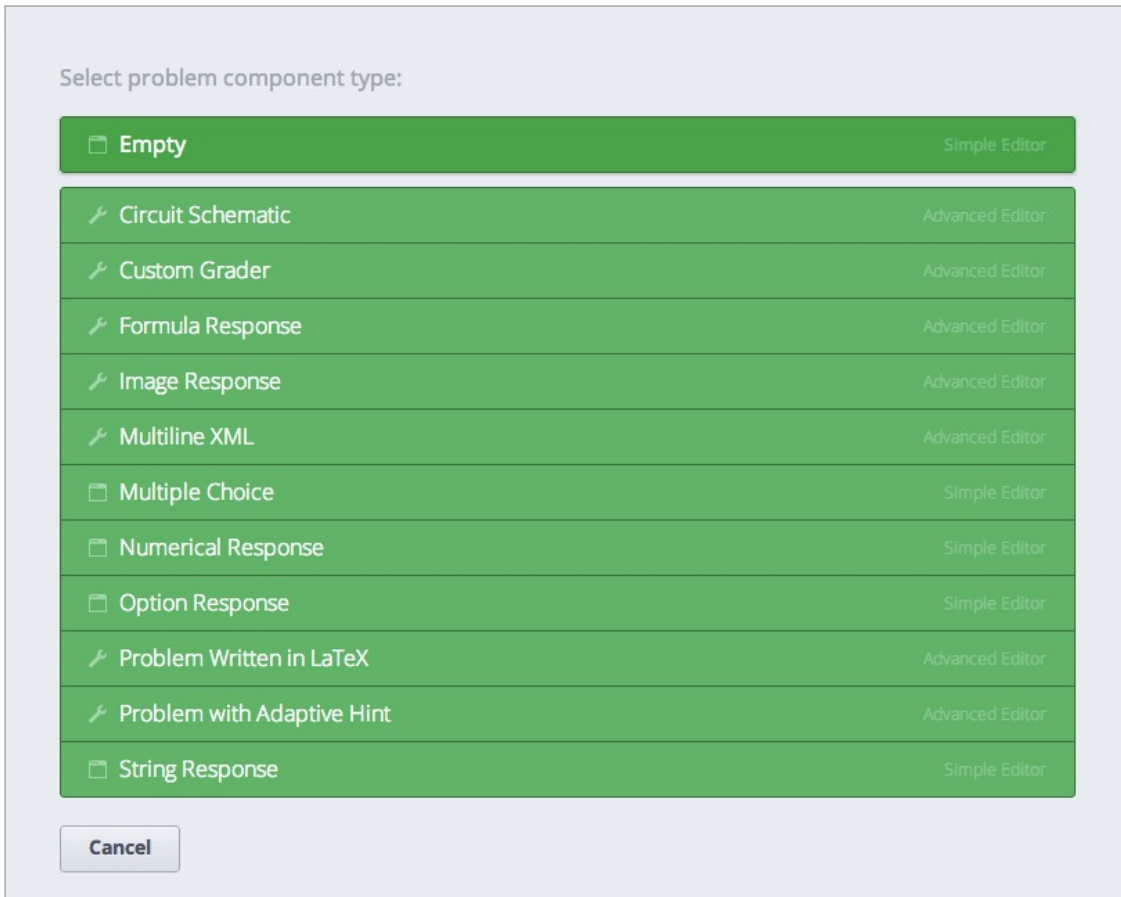
Studio offers several templates that you can use. Alternatively, you can create your own problem type in XML.

For detailed information about different problem types, see [Problem Types](#).

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



The **Select Problem Component Type** screen appears.

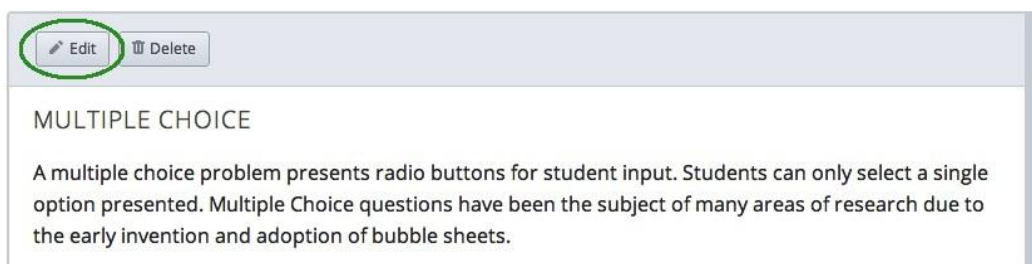


- In the **Select Problem Component Type** screen, click the problem type that you want.

Note To create your own problem in XML, click **Empty** to open a blank XML editor.

A new problem component with sample template text appears.

For example, if you click **Multiple Choice**, the following problem component appears.



- Click **Edit**. This opens the Simple Editor for the problem component. The following example shows this view for a multiple choice problem.

rerandomize:

showanswer:

display_name:

H1 [icons] ABC 123 [checkbox] [lightbulb] Advanced Editor ?

A multiple choice problem presents radio buttons for student input. Students can only select a single option presented. Multiple Choice questions have been the subject of many areas of research due to the early invention and adoption of bubble sheets.

One of the main elements that goes into a good multiple choice question is the existence of good distractors. That is, each of the alternate responses presented to the student should be the result of a plausible mistake that a student might make.

What Apple device competed with the portable CD player?

The iPad
 Napster
 The iPod
 The vegetable peeler
 Android
 The Beatles

[explanation] The release of the iPod allowed consumers to carry their entire music library with them in a format that did not rely on fragile and energy-intensive spinning disks.
[explanation]

4. In the **display_name** box, type the text that you want the student to see when the student hovers the mouse over the icon in the bar at the top of the page. This text also appears as a header for the problem.
5. Modify the problem text, and then click **Save** to save and check your work. Make sure to publish the draft you are working on to view the problem live.

Advanced Editor

Studio offers two interfaces for editing problem components.

- 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. The following is a multiple choice problem in the Advanced Editor.

Some of the simpler problem templates, including multiple choice, open in the Simple Editor and allow you to switch to the Advanced Editor. The more complicated problem types, such as Circuit Response, open in the Advanced Editor.

Note You can switch at any time from the Simple Editor to the Advanced Editor by clicking **Advanced Editor** in the Simple Editor interface. However, there is no way to go back from the Advanced Editor to the Simple Editor without making a new component.

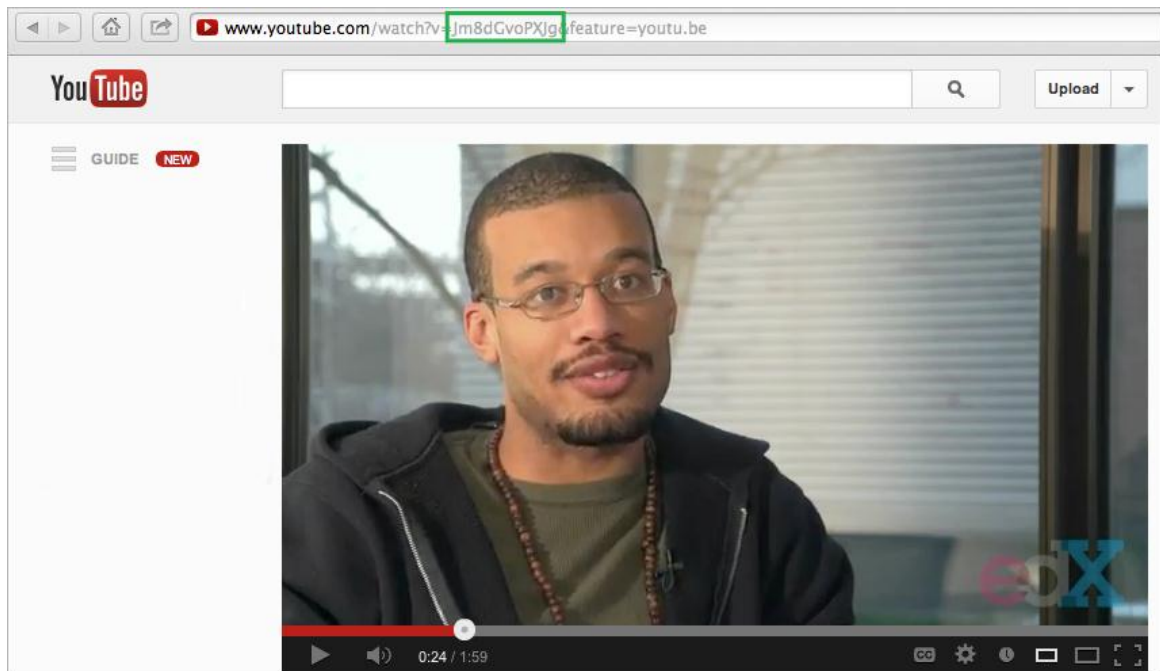
Create a Video

To add a video to the unit, you must upload your video to YouTube, and then create a video component. [You can also add a transcript to your video.](#)

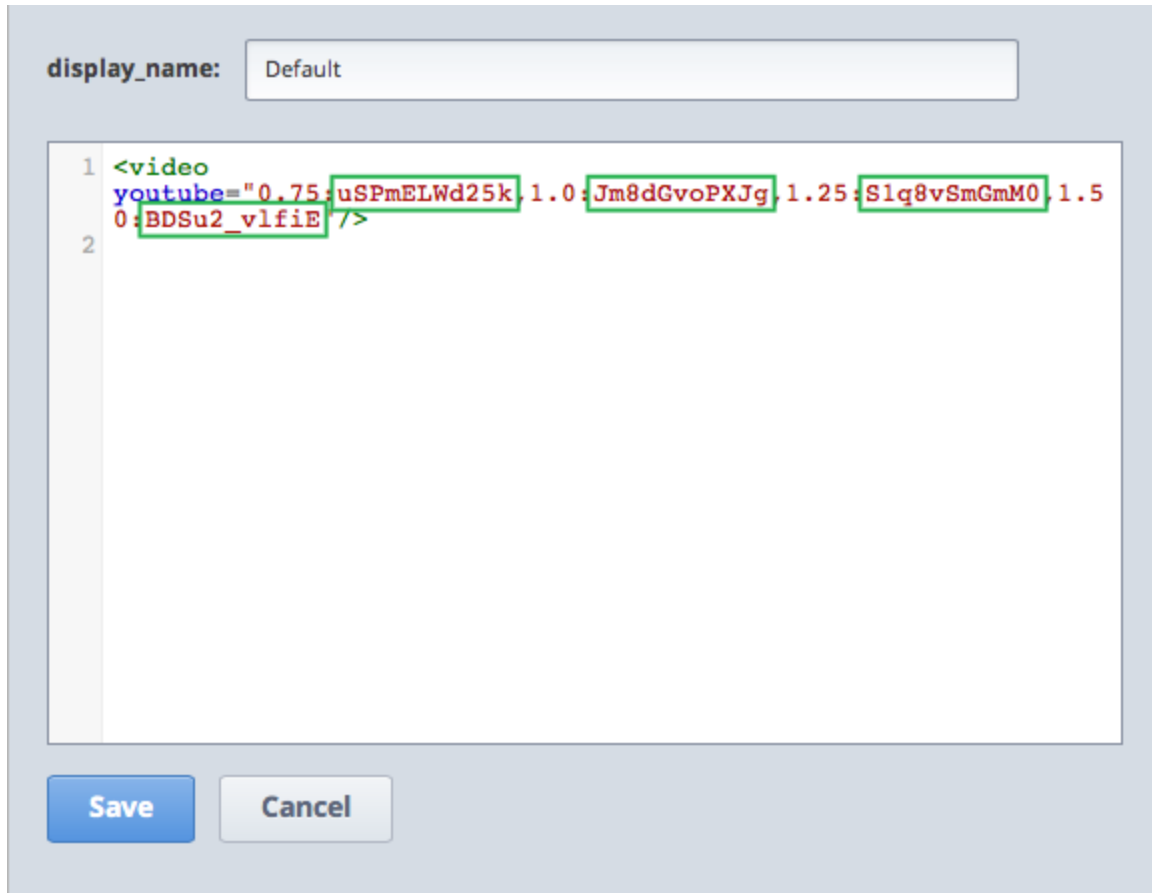
You can include videos that run at 0.75 speed, 1.25 speed, and 1.50 speed as well as at normal speed. To do this, you must upload each of these videos to YouTube separately.

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

1. Upload the video that you want 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. In Studio, go to the unit that you want.
3. Under **Add New Component**, click **Video**.
4. In the screen that appears, click **default**.
5. When the new video component appears, click **Edit**. A video component opens, and a sample video begins playing automatically.
6. In the **display_name** box, type the text that you want the student to see when the student hovers the mouse over the icon in the bar at the top of the page. This text also appears as a header for the video.
7. Change the codes in the green boxes to the YouTube codes that you noted in step 1. The first code (immediately to the right of "0.75:") corresponds to the video at 0.75 speed, the next corresponds to 1.0 speed, etc.



8. Click **Save**.

Note All videos embedded using the edX player begin playing automatically. There is currently no way to turn off the autoplay feature.

To add a transcript for your video:

1. Save your srt.sjson file as **subs_YOUTUBEID.srt.sjson**, where **YOUTUBEID** is the YouTube ID of your video.
2. Upload the **subs_YOUTUBEID.srt.sjson** file to your **Assets** tab.
3. Create a link to this file by following the steps in the [Add Items to the Handouts Sidebar](#) section.

Set Content Release Dates and Visibility

The release schedule for course material is determined by setting release dates for sections and subsections.

Section

To set the release date for a section:

1. Click **Courseware**.
2. Find the section you are looking for in the course outline.

3. Under **Will Release** date, click **Edit**, and then change the date.



Subsection

To set the release date for a subsection:

1. Click to open the subsection.
2. Locate the **Subsection Settings** box in the top right corner.
3. Under **Release Date**, change the date.

Unit

Individual units inherit the release date of the subsection they are in, but have an additional **Visibility** setting that can be set to **Public** or **Private**. Private units are never visible to students.

Note You can modify private units directly. To modify a unit that is marked as Public, you must create a draft.

To change the **Visibility** setting for a private unit:

1. Click to open the unit.
2. Locate the **Unit Settings** box in the top right corner.
3. Under **Unit Settings**, select a value for **Visibility**.

To change the **Visibility** setting for a public unit:

1. Click to open the unit.
2. Locate the **Unit Settings** box in the top right corner.
3. Under **Unit Settings**, click **edit a draft**.

You can verify that the draft looks the way you want it to before replacing the published version by clicking **Preview**.

View Course Content

There are four ways of viewing course content.

1. **In Studio**. You can see all material except published versions of Public units that are marked as having a Draft -- for these, you see the Draft instead.
2. **In Preview mode**. You can see the same material as in Studio.
3. **On Edge, as an instructor**. You can see only the units that have been marked Public (regardless of Release Date). Only the published versions appears on Edge; never the Drafts.

4. **On Edge, as a student.** Students can see the published version of material that is marked Public, but only after its Release Date has passed. You can check this view by setting up a student account for your course.

Invite Students to Register

To invite students to register for your course on Edge through the course registration page, direct students to the registration page, and provide instructions for completing the registration process.

1. Determine the link to your class registration page on Edge. To do this:
 - a. Click the **Settings** tab of your course in Studio, and then locate the **Course Details** section.
 - b. Under Basic Information, click one of the **your course URL** links. These are located under Course Name, Organization, and Course Number.
 - c. Copy the course URL from the address bar of your browser. The URL resembles the following.

```
https://edge.edx.org/courses/[PATH_TO_YOUR_CLASS]/about
```
2. View the sample student login guide in **Appendix B**. Make a copy of this guide and open it for editing in Google Docs, or download it for editing in Microsoft Word or another compatible text editor.

Note *The guide is taken from the Fall 2012 Python Programming course run by Bunker Hill Community College.*

3. Customize the guide template to your class by replacing all the highlighted text and images with content specific to your class. This includes:
 - Course name
 - Course number
 - Registration URL
 - Registration page screenshots
4. After you have customized this document, follow its instructions to make sure that you can create a student login and register for your class.
5. (Optional) Convert your file to a PDF file for easier dissemination.
6. Put together an invitation email or information packet to give out to your students. Include the registration URL and the student login guide you have created.

APPENDIX A: Template for Course Overview box

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>
<p></p><p></p>
<section class="prerequisites">
<h2>Prerequisites</h2>
<p>Add information about class prerequisites here.</p>
</section>
<p></p><p></p>
<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
youâ€™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> -->
```

APPENDIX B: Sample Student Login Guide

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.

Go to

[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 EDGE]

The above URL should bring you to the course registration page:

[REPLACE WITH SCREENSHOT OF YOUR REGISTRATION PAGE]

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.)

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].

APPENDIX C: Problem Types

Option Response

The Option Response input type allows the student to choose from a collection of answer options, presented as a drop-down list.

Option Response is structurally similar to Multiple Choice. Some conceptual differences between the two include the following.

- The Multiple Choice radio button format makes it easier for students to read very long response options.
- The Option Response drop-down input format makes it more likely for students to think of an answer and then search for it, rather than relying purely on recognition to answer the question.

The Multiple Choice format is more explicit and visual. This makes it a more appropriate choice for presenting tricky or complicated answer options which are intended to get the student to pause and think.

Sample Problem

Option Response is most similar to _____.

EXPLANATION

Like Option Response, Multiple Choice also allows students to select from a variety of pre-written responses.

Check

Hide Answer

Problem Code

```
<problem>
  <p>Option Response is most similar to _____.</p>
  <optionresponse>
    <optioninput
      options="('Multiple Choice','String Response',
               'Numerical Response','External Response',
               'Image Response')"
      correct="Multiple Choice"/>
  </optionresponse>
```

```

<solution>
  <div class="detailed-solution">
    <p>Explanation</p>
    <p>Like Option Response, Multiple Choice also allows students to select from a
variety of pre-written responses.</p>
  </div>
</solution>
</problem>

```

Template

```

<problem>

  <optionresponse>
    <optioninput
      options="'A', 'B'"
      correct="A"/>
  </optionresponse>

  <solution>
    <div class="detailed-solution">

      </div>
  </solution>
</problem>

```

XML Attribute Information

<optionresponse>

Attribute	Description	Notes
Children	Description	Notes
All standard HTML tags	Display text relevant to the problem.	
optioninput	A method to take input from students. See description below.	

<optioninput>

Attribute	Description	Notes
options	A list of options that students choose from.	<p>The attribute must be defined with double quotes, and the values in the list with single quotes.</p> <p>There must not be a space between the separating commas and the single-quote start of the option.</p> <p>The answers as displayed to students cannot contain</p>

		any quotes
correct	The option from the list of options that the grader will take as correct.	In order for students to get credit, this option must be typed exactly the same as the definition in 'options'
Children	Description	Notes

Multiple Choice

The Multiple Choice input type allows the student to select at most one choice from a collection of answer choices, presented as a list of radio buttons.

A Multiple Choice problem can have more than one correct answer, depending on how many choices are marked as correct in the underlying XML. If all choices are marked as incorrect, there is no correct response.

Multiple Choice is structurally similar to Option Response. Some conceptual differences between the two include the following.

- The Multiple Choice radio button format makes it easier for students to read very long response options.
- The Option Response drop-down input format makes it more likely for students to think of an answer and then search for it, rather than relying purely on recognition to answer the question.
- The Multiple Choice format is more explicit and visual. This makes it a more appropriate choice for presenting tricky or complicated answer options which are intended to get the student to pause and think.

Sample Problem

How many correct responses can a Multiple Choice question have?

- Only one
- Only zero or one
- Zero or more
- Only one or more
- Nobody knows
- Somebody might know :)

EXPLANATION

It depends on how many choices are marked as correct in the underlying XML.

Note that if all choices are marked as incorrect, there is no correct response.

Check

Hide Answer

Problem Code

```
<problem>
  <p><b>Example Problem</b></p>
  <p>How many correct responses can a Multiple Choice question have?</p>
  <multiplechoiceresponse>
    <choicegroup type="MultipleChoice">
      <choice correct="false" name="one">Only one</choice>
      <choice correct="false" name="zeroone">Only zero or one</choice>
      <choice correct="true" name="zeromore">Zero or more</choice>
      <choice correct="false" name="onemore">Only one or more</choice>
      <choice correct="false" name="noone">Nobody knows</choice>
      <choice correct="true" name="someone">Somebody might know :)</choice>
    </choicegroup>
  </multiplechoiceresponse>
  <solution>
    <div class="detailed-solution">
      <p>Explanation</p>
      <p>It depends on how many choices are marked as correct in the
        underlying XML.</p>
      <p>Note that if all choices are marked as incorrect, there is no
        correct response.</p>
    </div>
  </solution>
</problem>
```

Template

```

<problem>

  <multiplechoiceresponse>
    <choicegroup type="MultipleChoice">
      <choice correct="false" name="a">A</choice>
      <choice correct="true" name="b">B</choice>
    </choicegroup>
  </multiplechoiceresponse>

  <solution>
    <div class="detailed-solution">

      </div>
    </solution>
  </problem>

```

XML Attribute Information

<multiplechoiceresponse>

Attribute	Description	Notes
Children	Description	Notes
All standard HTML tags	to display text relevant to the problem	
choicegroup	a method to take input from students. See description below	

<choicegroup>

Attribute	Description	Notes
type	needs to have the value "MultipleChoice"	
Children	Description	Notes
choice	a description of an option to be displayed to students. Details below	

<choice>

Attribute	Description	Notes
name	(optional) a unique name that the backend will use to refer to this	

	option	
correct	"true" if this option is the correct answer, "false" if not	Only one choice in the choicegroup can have correct="true" in order for it to be possible to get the correct answer

Checkbox

The Checkbox input type allows the student to select zero or more choices from a collection of answer choices, presented as a list of checkboxes.

Remark: Questions with one Checkbox input type have exactly one correct response. All the choices marked as correct="true" have to be selected for the submitted answer (i.e. the response) to be considered correct.

In particular, the response of no boxes checked off could be the single correct response, and a Checkbox question, unlike a Multiple Choice question, cannot have zero correct responses.

Sample Problem

How many correct responses can a Checkbox question have?

Zero
 One
 Two or more
 Nobody knows
 Somebody might know :)

Problem Code

```

<problem>
<startouttext/>
  <p>How many correct responses can a Checkbox question have?</p>

<choiceresponse>
<checkboxgroup>
<choice correct="false"><text>Zero</text></choice>
<choice correct="true"><text>One</text></choice>
<choice correct="false"><text>Two or more</text></choice>

```

```
<choice correct="false"><text>Nobody knows</text></choice>
<choice correct="true"><text>Somebody might know :)</text></choice>
</checkboxgroup>
</choiceresponse>
</problem>
```

Template

```
<problem>

<choiceresponse>
<checkboxgroup>
<choice correct="false"><text>Zero</text></choice>
<choice correct="true"><text>One</text></choice>
</checkboxgroup>
</choiceresponse>
</problem>
```

String Response

The String Response input type provides an input box in which the student can enter a line of text, which is then checked against a specified expected answer.

A String Response input does not provide any answer suggestions, so it can be a good way to get the students to engage with the material more deeply in a sequence and look up, figure out, or remember the correct answer themselves.

Note that a student's answer in a String Response is marked as correct if it matches every character of the expected answer. This can be a problem with international spelling, dates, or anything where the format of the answer is not clear.

Sample Problem

What is the name of this problem? (What response type is this?)

● **Answer:** String Response

EXPLANATION

The name of this problem is "String Response," written without the punctuation marks.

Capitalization does not matter, so "string response" and "STRING RESPONSE" would also be correct answers.

Putting quotation marks around the answer, however, would be marked as incorrect.

Check

Hide Answer

Problem Code

```
<problem>
  <p><b>Example Problem</b></p>
  <p>What is the name of this unit? (What response type is this?)</p>
  <stringresponse answer="String Response" type="ci">
    <textline size="20"/>
  </stringresponse>
  <solution>
    <div class="detailed-solution">
      <p>Explanation</p>
      <p>The name of this unit is "String Response," written without the
punctuation.</p>
      <p>Arbitrary capitalization is accepted.</p>
    </div>
  </solution>
</problem>
```

Template

```
<problem>
  <stringresponse answer="REPLACE_THIS" type="ci">
    <textline size="20"/>
  </stringresponse>
  <solution>
    <div class="detailed-solution">
      </div>
    </solution>
</problem>
```

XML Attribute Information

<stringresponse>

Attribute	Description	Notes
type	(optional) "ci" if the student response should be graded case-insensitively. The default is to take case into consideration when grading.	
answer	The string that students need to enter in order to get credit.	
Children	Description	Notes
textline	used to accept student input. See description below	

<textline>

Attribute	Description	Notes
-----------	-------------	-------

math	(optional) If this attribute has any value at all, a math preview will display beneath the textbox showing well-formatted math corresponding to student input	
size	(optional) defines the size in character widths of the input box as it is displayed to students.	
hidden	(optional) if true, the textbox will be hidden from students.	
Children	Description	Notes

Numerical Response

The Numerical Response input type accepts a line of text input from the student and evaluates the input for correctness based on its numerical value. The input is allowed to be a number or a mathematical expression in a fixed syntax.

The answer is correct if it is within a specified numerical tolerance of the expected answer.

The expected answer can be specified explicitly or precomputed by a Python script.

Sample Problem

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.

Problem Code

```
<problem>
  <p><b>Example Problem</b></p>

  <p>What base is the decimal numeral system in?
    <numericalresponse answer="10">
      <textline />
    </numericalresponse>
  </p>

  <p>What is the value of the standard gravity constant <i>g</i>, measured in
  m/s<sup>2</sup>? Give your answer to at least two decimal places.
  <numericalresponse answer="9.80665">
    <responseparam type="tolerance" default="0.01" />
    <textline />
  </numericalresponse>
</problem>
```

```

    </numericalresponse>
</p>

<!-- Use python script spacing. The following should not be indented! -->
<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.
    <numericalresponse answer="$computed_response">
        <responseparam type="tolerance" default="0.0001" />
        <textline math="1" />
    </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/s<sup>2</sup>.
        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
        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
        <code>sqrt(pi^2+e^2)</code> into the editor.
        Other answers like <code>sqrt((pi-0)^2+(0-e)^2)</code> also work.
    </p>
    </div>
</solution>
</problem>

```

Templates

Exact values

```

<problem>

    <numericalresponse answer="10">
        <textline />
    </numericalresponse>

    <solution>
    <div class="detailed-solution">

        </div>
    </solution>
</problem>

```

Answers with decimal precision

```

<problem>

    <numericalresponse answer="9.80665">
        <responseparam type="tolerance" default="0.01" />
        <textline />
    </numericalresponse>

    <solution>

```

```

    <div class="detailed-solution">

    </div>
</solution>
</problem>

```

Answers with percentage precision

```

<problem>

    <numericalresponse answer="100">
        <responseparam type="tolerance" default="10%" />
        <textline />
    </numericalresponse>

    <solution>
    <div class="detailed-solution">

    </div>
</solution>
</problem>

```

Answers with a live math interpretation popup display

```

<problem>

    <numericalresponse answer="3.14159">
        <responseparam type="tolerance" default="0.00001" />
        <textline math="1" />
    </numericalresponse>

    <solution>
    <div class="detailed-solution">

    </div>
</solution>
</problem>

```

Answers with scripts

```

<problem>

<!-- Use python script spacing. The following should not be indented! -->
<script type="loncapa/python">
computed_response = math.sqrt(math.fsum([math.pow(math.pi,2), math.pow(math.e,2)]))
</script>

    <numericalresponse answer="$computed_response">
        <responseparam type="tolerance" default="0.0001" />
        <textline math="1" />
    </numericalresponse>

    <solution>
    <div class="detailed-solution">

    </div>
</solution>
</problem>

```

XML Attribute Information

<script>

| Attribute | Description | Notes |
|-----------|---|---|
| type | "loncapa/python" - a script written in python | A problem will behave the same way 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.

Like all python, indentation matters, even though it is embedded in XML. |
| Children | Description | Notes |
| | | |

<numericalresponse>

| Attribute | Description | Notes |
|---------------|--|--|
| answer | A value to which student input must be equivalent. Note that this expression can be expressed in terms of a variable that is computed in a script provided in the problem by preceding the appropriate variable name with a dollar sign. | Note that any numeric expression provided by the student will be automatically simplified on the grader's backend. |
| Children | Description | Notes |
| responseparam | used to specify a tolerance on the accepted values of a number. See description below | |
| textline | a format to take input from students. See description below | |

<responseparam>

| Attribute | Description | Notes |
|-----------|--|-------|
| type | "tolerance" - defines a tolerance for a number | |
| default | either a number or a percentage, defining how different from the provided answer a student answer can be while still getting full credit | |
| Children | Description | Notes |

| | | |
|--|--|--|
| | | |
|--|--|--|

<textline>

| Attribute | Description | Notes |
|-----------|---|-------|
| math | (optional) If this attribute has any value at all, a math preview will display beneath the textbox showing well-formatted math corresponding to student input | |
| size | (optional) defines the size in character widths of the input box as it is displayed to students. | |
| hidden | (optional) if true, the textbox will be hidden from students. | |
| Children | Description | Notes |
| | | |

Formula Response

The Formula Response input type accepts a line of text representing a mathematical expression from the student and evaluates the input for equivalence to a mathematical expression provided by the grader. Correctness is based on numerical sampling of the symbolic expressions.

The answer is correct if both the student-provided response and the grader's mathematical expression are equivalent to specified numerical tolerance, over a specified range of values for each variable.

This kind of response type can handle symbolic expressions. However, it places an extra burden on the problem author to specify the allowed variables in the expression and the numerical ranges over which the variables must be sampled in order to test for correctness.

Sample Problem

Write an expression for the product of R_1 , R_2 , and the inverse of R_3 .

● Answer: $(R_1 * R_2) / R_3$

Let c denote the speed of light. What is the relativistic energy E of an object of mass m ?

$E =$

● Answer: $m * c^2$

Let x be a variable, and let n be an arbitrary constant. What is the derivative of x^n ?

● Answer: $n * x^{(n-1)}$

EXPLANATION

Use standard arithmetic operation symbols and indicate multiplication explicitly.

Use the symbol \wedge to raise to a power.

Use parentheses to specify order of operations.

Problem Code

```
<problem>
  <p><b>Example Problem</b></p>
  <p>This is a short introduction to the Formula Response editor.</p>

  <p>Write an expression for the product of  $R_1$ ,  $R_2$ , and the inverse of  $R_3$ .</p>
  <formularesponse type="ci" samples="R_1,R_2,R_3@1,2,3:3,4,5#10" answer="$VoVi"
    <responseparam type="tolerance" default="0.00001"/>
    <textline size="40" math="1" />
  </formularesponse>

  <p>Let  $c$  denote the speed of light. What is the relativistic energy  $E$ 
of an object of mass  $m$ ?</p>
<script type="loncapa/python">
VoVi = "(R_1*R_2)/R_3"
</script>
```

```

<formularesponse type="cs" samples="m,c@1,2:3,4#10" answer="m*c^2">
  <responseparam type="tolerance" default="0.00001"/>
  <text><i>E</i> =</text> <textline size="40" math="1" />
</formularesponse>

  <p>Let <i>x</i> be a variable, and let <i>n</i> be an arbitrary constant. What is
the derivative of <i>x<sup>n</sup></i>?</p>
<script type="loncapa/python">
derivative = "n*x^(n-1)"
</script>
<formularesponse type="ci" samples="x,n@1,2:3,4#10" answer="$derivative">
  <responseparam type="tolerance" default="0.00001"/>
  <textline size="40" math="1" />
</formularesponse>

  <solution>
  <div class="detailed-solution">
    <p>Explanation</p>
    <p>Use standard arithmetic operation symbols and indicate multiplication
explicitly.</p>
    <p>Use the symbol <tt>^</tt> to raise to a power.</p>
    <p>Use parentheses to specify order of operations.</p>
  </div>
  </solution>
</problem>

```

Template

```

<problem>

<script type="loncapa/python">
answer_value = "n*x^(n-1)"
</script>
<formularesponse type="ci" samples="x,n@1,2:3,4#10" answer="$answer_value">
  <responseparam type="tolerance" default="0.00001"/>
  <textline size="40" math="1" />
</formularesponse>

  <solution>
  <div class="detailed-solution">

  </div>
  </solution>
</problem>

```

XML Attribute Information

<script>

| Attribute | Description | Notes |
|-----------|---|---|
| type | "loncapa/python" - a script written in python | <p>A problem will behave the same way 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.</p> <p>Like all python, indentation matters, even though it is embedded in</p> |

| | | |
|-----------------|--------------------|--------------|
| | | XML. |
| Children | Description | Notes |
| | | |

<formularesponse>

| Attribute | Description | Notes |
|------------------|--|--|
| type | "cs" - Case Sensitive, the default
"ci" - Case Insensitive, capitalization does not matter in variable names | |
| answer | An expression to which the student provided response must be equivalent. Note that this expression can be expressed in terms of a variable that is computed in a script provided with the problem by preceding the appropriate variable name with a dollar sign. | Multiplication must be explicitly stated with an asterisk. Students also must enter their input with multiplication as an asterisk. The phrase "mc^2" will be interpreted as a single value represented by the variable named "mc" raised to the second power. |
| samples | This attribute is one of the more complicated attributes in the EDXML tag system. It contains four related comma-delineated lists, separated by special delineators. The lists are in the format

<variables>@<lower_bounds>:<upper_bound>#<num_samples>

variables - a set of variables that are allowed as student input

lower_bounds - for every variable defined in variables, a lower bound on the numerical tests to use for that variable

upper_bounds - for every variable defined in variables, an upper bound on the numerical tests to use for that variable

num_samples - how many times to test the expression | |
| Children | Description | Notes |

| | | |
|---------------|--|--|
| responseparam | used to define an upper bound on the variance of the numerical methods used to approximate a test for equality. See description below. | |
| textbox | used to take student responses. See description below. | |

<responseparam>

| Attribute | Description | Notes |
|-----------|---|---|
| default | 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 sampling, 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 | Description | Notes |
| | | |

Image Response

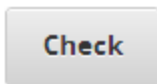
The Image Response input type presents an image and accepts clicks on the image as an answer.

Images have to be uploaded to the courseware Assets directory. Response clicks are marked as correct if they are within a certain specified sub rectangle of the image canvas.

Note *The Mozilla Firefox browser is currently not supported for this problem type.*

Sample Problem

You are given three shapes. Click on the triangle.



Problem Code

```
<problem>
  <p><b>Example Problem</b></p>
<startouttext/>
  <p>You are given three shapes. Click on the triangle.</p>
<endouttext/>
  <imageresponse>
    <imageinput src="/c4x/edX/edX101/asset/threeshapes.png" width="220" height="150"
rectangle="(80,40)-(130,90)" />
  </imageresponse>
</problem>
```

Template

```
<problem>
  <imageresponse>
    <imageinput src="Path_to_Image_File.png" width="220" height="150"
rectangle="(80,40)-(130,90)" />
  </imageresponse>
</problem>
```

XML Attribute Information

<imageresponse>

Attribute	Description	Notes
Children	Description	Notes

imageinput	used to get input based off of an image. See description below.	
------------	---	--

<imageinput>

Attribute	Description	Notes
src	the url from which the image should be taken	
height	The height that the image should be displayed with	
width	the width that the image should be displayed with	
rectangle	<p>an attribute with four embedded values in the format</p> <p>(<start_width>,<start_height>)-(<end_width>,<end_height>)</p> <p>All coordinates start with (0,0) at the top left corner and increase in value towards 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 in order to get credit for the problem.</p>	
Children	Description	Notes

Custom Response

A Custom Response input type accepts one or more lines of text input from the student and evaluate the inputs for correctness using an embedded Python script.

Sample Problem

This question consists of two parts.

First, enter two integers which sum to 10.

Now enter two (finite) decimals which sum to 20.

EXPLANATION

For the first part, any two numbers of the form n and $10-n$, where n is any integer, will work. One possible answer would be the pair 0 and 10.

For the second part, any pair x and $20-x$ will work, where x is any real number with a finite decimal representation. Both inputs have to be entered either in standard decimal notation or in scientific exponential notation. One possible answer would be the pair 0.5 and 19.5. Another way to write this would be $5e-1$ and $1.95e1$.

Check

Hide Answer

Problem Code

```
<problem>
  <p><b>Example Problem</b></p>
<script type="loncapa/python">

def test_add_to_ten(expect,ans):
    try:
        a1=int(ans[0])
        a2=int(ans[1])
    except ValueError:
        a1=0
        a2=0
    return (a1+a2)==10

def test_add(expect,ans):
    try:
        a1=float(ans[0])
        a2=float(ans[1])
    except ValueError:
        a1=0
```



```

    a2=0
    return (a1+a2)== float(expect)
</script>

<p>This question consists of two parts. </p>
<p>First, enter two integers which sum to 10. </p>
<customresponse cfn="test_add_to_ten">
    <textline size="40" /><br/>
    <textline size="40" />
</customresponse>

<p>Now enter two (finite) decimals which sum to 20.</p>
<customresponse cfn="test_add" expect="20">
    <textline size="40" /><br/>
    <textline size="40" />
</customresponse>

<solution>
    <div class="detailed-solution">
        <p>Explanation</p>
        <p>For the first part, any two numbers of the form  $n$ 
            and  $10-n$ , where  $n$  is any integer, will work.
            One possible answer would be the pair 0 and 10.
        </p>
        <p>For the second part, any pair  $x$  and  $20-x$  will work, where
             $x$  is any real number with a finite decimal representation. Both inputs have to
            be entered either in standard decimal notation or in scientific exponential notation.
            One possible answer would be the pair 0.5 and 19.5. Another way to write this would be
             $5e-1$  and  $1.95e1$ .
        </p>
    </div>
</solution>
</problem>

```

Templates

With displayed suggested correct answers

```

<problem>

<script type="loncapa/python">
def test_add(expect,ans):
    a1=float(ans[0])
    a2=float(ans[1])
    return (a1+a2)== float(expect)
</script>

<p>Enter two real numbers which sum to 20: </p>
<customresponse cfn="test_add" expect="20">
    <textline size="40" correct_answer="11"/><br/>
    <textline size="40" correct_answer="9"/>
</customresponse>

    <solution>
        <div class="detailed-solution">
            </div>
        </solution>
</problem>

```

With no suggested correct answers

```
<problem>

<script type="loncapa/python">
def test_add(expect,ans):
    a1=float(ans[0])
    a2=float(ans[1])
    return (a1+a2)== float(expect)
</script>

<p>Enter two real numbers which sum to 20: </p>
<customresponse cfn="test_add" expect="20">
    <textline size="40" /><br/>
    <textline size="40" />
</customresponse>

    <solution>
        <div class="detailed-solution">
            </div>
        </solution>
</problem>
```

Chemical Equation Response

The Chemical Equation Response input type is a special type of Custom Response that allows the student to enter chemical equations as answers.

Sample Problem

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



Some tips:

- Only real element symbols are permitted.
- Subscripts are entered with plain text.
- Superscripts are indicated with a caret (^).
- The reaction arrow (\rightarrow) is indicated with "->".

So, you can enter "H2SO4 -> H^+ + HSO4^-".

Problem Code

```
<problem>
  <p><b>Example Problem</b></p>
  <startouttext/>
  <p>Some problems may ask for a particular chemical equation.
  Practice by writing out the following reaction in the box below.</p>
  <center>\( \text{H}_2\text{SO}_4 \rightarrow \text{H}^+ +
  \text{HSO}_4^- \)</center>
  <br/>
  <customresponse>
    <chemicalequationinput size="50"/>
    <answer type="loncapa/python">

if chemcalc.chemical_equations_equal(submission[0], 'H2SO4 -> H^+ +
HSO4^-'):
    correct = ['correct']
else:
```

```
correct = ['incorrect']
```

```
</answer>
```

```
</customresponse>
```

```
<p> Some tips:<ul><li>Only real element symbols are permitted.</li><li>Subscripts are entered with plain text.</li><li>Superscripts are indicated with a caret (^).</li><li>The reaction arrow (\(\longrightarrow\)) is indicated with "->".</li></ul>
```

```
So, you can enter "H2SO4 -> H^+ + HSO4^-".</p>
```

```
<endouttext/>
```

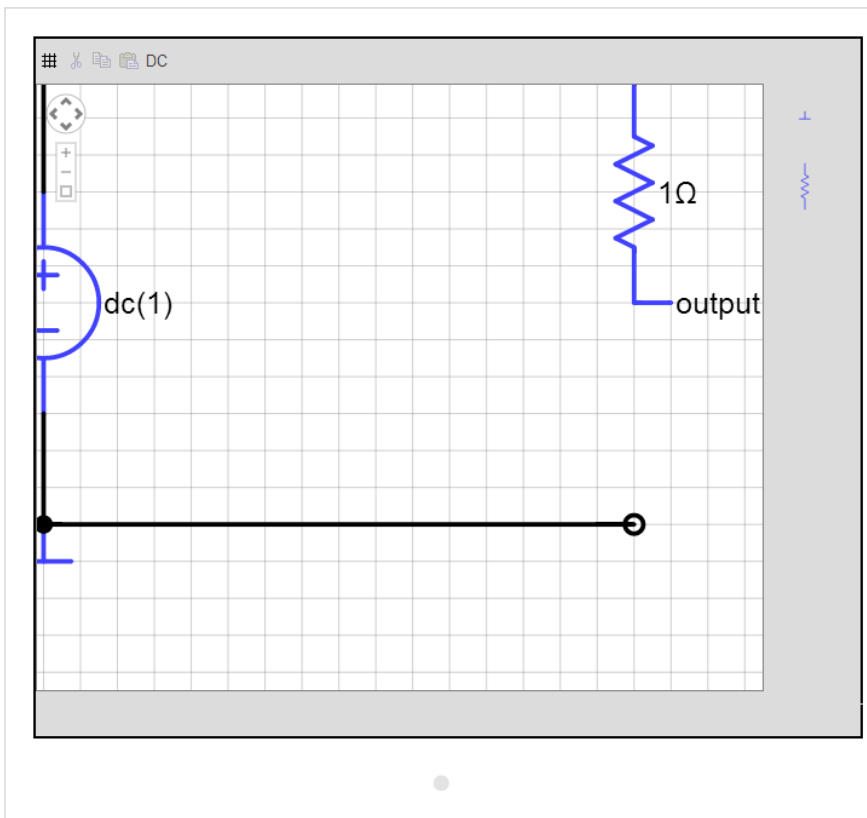
```
</problem>
```

Schematic Response

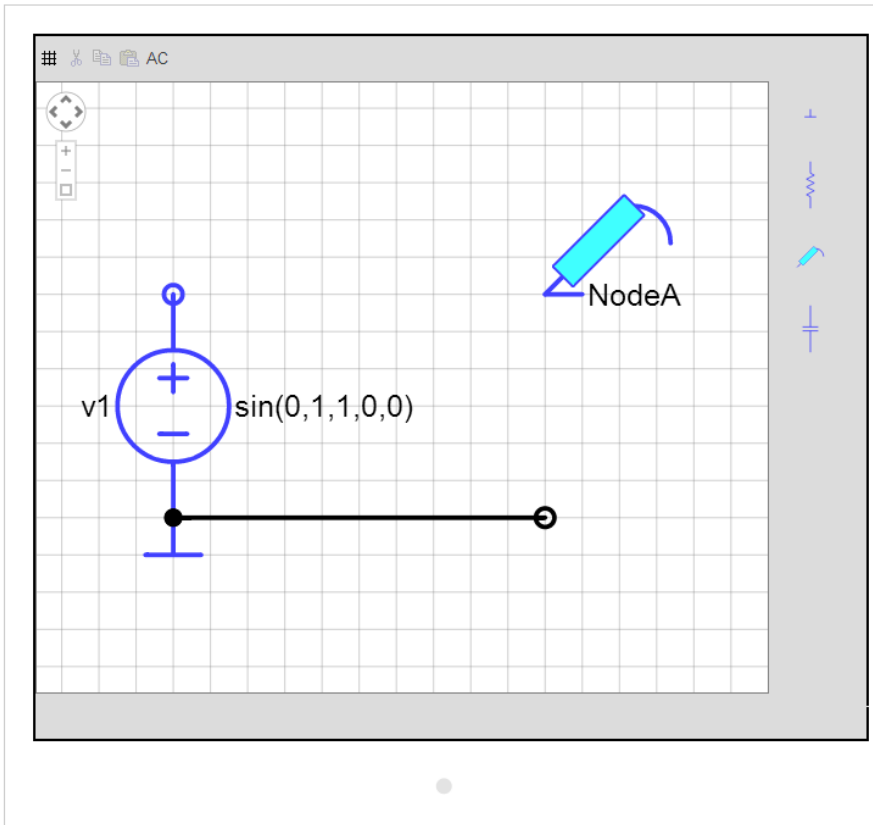
The Schematic Response input type provides an interactive grid on which the student can construct a schematic answer, such as a circuit.

Sample Problem

Make a voltage divider that splits the provided voltage evenly.



Make a high pass filter.



EXPLANATION

A voltage divider that evenly divides the input voltage can be formed with two identically valued resistors, with the sampled voltage taken in between the two.



A simple high-pass filter without any further constraints can be formed by simply putting a resistor in series with a capacitor. The actual values of the components do not really matter in order to meet the constraints of the problem.



Check **Hide Answer**

Problem Code

<problem>

Make a voltage divider that splits the provided voltage evenly.

```
<schematicresponse>
<center>
<schematic height="500" width="600" parts="g,r" analyses="dc"
initial_value=[[{"v",[168,144,0],{"value":"dc(1)","_json_":0},["1","0"]},{"r",[296,120,0],{"r":"1","_json_":1},["1","output"]},{"L",[296,168,3],{"label":"output","_json_":2},["output"]},{"w",[296,216,168,216]},{"w",[168,216,168,192]},{"w",[168,144,168,120]},{"w",[168,120,296,120]},{"g",[168,216,0],{"_json_":7},["0"]},{"view",-67.49999999999994,-78.49999999999994,1.6000000000000003,"50","10","1G",null,"100","1","1000"}]]
/>
</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},["1","0"]},{"w",[160,200,240,200]},{"g",[160,200,0],{"_json_":2},["0"]},{"L",[240,152,3],{"label":"NodeA","_json_":3},["NodeA"]},{"s",[240,152,0],{"color":"cyan","offset":"0","_json_":4},["NodeA"]},{"view",64.55878906250004,54.114697265625054,2.5000000000000004,"50","10","1G",null,"100","1","1000"}]]/>
</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, with the sampled voltage taken in between the two.</p>
<p></p>
<p>A simple high-pass filter without any further constraints can be formed by simply putting a resistor in series with a capacitor. The actual values of the components do not really matter in order to meet the constraints of the problem.</p>

```

```
<p></p>
</div>
</solution>
</problem>
```