**Distance Learning Development Guide**

K-12 Technologies

Customer Education

*Document version 2.5*

*DL Template version 2.5*

*mLG Widgets version 1.5*

**Table of Contents**

File Naming Conventions and Folder Structure 4

File Naming Convention 4

Content Structures 4

Graphics Requests 5

Audio Requests 5

Maintenance and Archiving 6

Introduction to Developer Software Packages and Add-ons 7

TextMate Snippets 7

Using Snippets 7

Tab-trigger Snippets 8

Hot-key Snippets 8

Miro Video Encoder 9

Automated Video Encoder Workflow 9

Templates and Settings 13

Recording Settings 13

Safari Text Settings 13

Publish Settings in Captivate 13

Compression 14

DOM Busters 14

Hotspot Bookmarklet 15

DL Template Version Bookmarklet 16

Introduction to Development Patterns 18

Demonstration Patterns 18

Classes 22

Get Started 22

Lay the Groundwork 22

Request Graphics 22

Remove Non-Class Index Pages 22

Prep the TOC page 22

Add / Remove Pages in course/ Folder 22

Prepare Class Pages Head and Top Nav Information 23

Add Content to Pages 23

Add Components to Each Page 23

Add Textual Content 25

Add Images 26

Capture Images 26

Add Images to Image Folder 26

Add Images to Each Page 26

Add Videos 26

Add Video Captions 27

Interactive Elements 30

Add Quiz Questions 30

Edit Matching Question HTML 31

Add Quiz Content: Hotspot questions 31

Add Try it! 31

Some Notes about Try It’s: 35

Add Character Dialogue 36

MIMS Development 39

MIM No-Video 39

Request Graphics 39

Rename Index Page 39

Remove Non-MIM HTML Pages from TOC and Course Folder 39

Prepare MIM Pages’ Head and Top Nav Information 40

Add Textual Content 41

Index Page 41

Working With Images 43

Add Resources (optional) 45

Remove Unused Folders 45

Modify help\_mim.html 46

Review and Test MIM 46

Submit MIM for Review 46

MIM Video 47

Add Closed Captions 48

mLearning Guide Development 50

Developing a Widget 50

Widget-Specific Instructions 51

Reviews 58

Appendix 60

List of CSS Classes 60

Lists 62

Links 62

Developer Checklist 63

Demos 63

Interactive Elements 63

## File Naming Conventions and Folder Structure

### File Naming Convention

The most important rule for effective file naming is consistency. Note: There are different conventions for Mastery in Minutes modules and full-length courses.

Filenames for courses should be kept as short as possible, while maintaining clarity and usability.

Use underscores, not spaces, in filenames. Some browsers and operating systems have difficulty recognizing file names with spaces.

Also, avoid using special characters in filenames. For example, avoid using: &, \ / : \* ? < > | % # $. Special characters are often reserved for use by the operating system or used in variable names.

**DL Naming Convention Breakdown**  
  
Each course is assigned a special name (course ID) that describes the content of the course or MIM. This name is an alphanumeric unique identifier.

Filenames for courses and MIMS consist of three parts: product, topic, and number.

**Course ID Example:** SN\_LP (Schoolnet Lesson Planner) or PS\_SOY2 (PowerSchool Start of Year Part 2)

**MIMS ID Example:** s01\_creating\_student\_group) Schoolnet – Creating a student Group) or m102\_edit\_data\_validation (Editing a Data Validation Rule in PowerSchool)

|  |  |
| --- | --- |
| Part | Description |
| Product | This includes an acronym for the relevant product.  *(All course ID’s begin with SN for Schoolnet or PS for PowerSchool)*  All *PowerSchool* MIMS ID’s start with ‘m’, and all Schoolnet MIMS ID’s start with ‘s’. The next number in the respective product MIM series follows the ID letter. |
| Topic | This includes the topic acronym for classes and an abbreviated title for MIMS. Developers are encouraged to keep the names as short as possible. |
| Version Number | A version number should be used at the end of the *archive* name to indicate the when a published file has changed, but the software version has not. |

### Content Structures

Using a uniform folder structure, we can easily identify all the components of a working course or MIM.

* The [COURSE\_TITLE] directory contains all files related to the course or MIM.
* The Resources directory organizes assets at the course level, such as transcripts, Keynote, original recordings, course notes, and peer review sheets.
* The [COURSE\_ID] directory contains the course assets at every stage of the development lifecycle, from early development, to peer review, to publication.
* Images and media elements are contained within corresponding section-numbered folders.

Example:

[COURSE\_TITLE] *(SN\_LP\_14.2\_v1)(PS\_DYSSE\_7\_v1)*

* \_RESOURCES *(inside: Word transcript doc with course description, objective, audience, etc. Downloadable resources such as QRC's. Current review sheet.)*
* \_SOURCE\_FILES
  + Captivate
  + Flash
  + Snagit Proj
  + Photoshop raw
  + Etc.
* [COURSE\_ID] *(SN\_LP)*
  + Course
    - SN\_LP\_01\_01.html
    - SN\_LP\_01\_02.html
    - SN\_LP\_01\_03.html
    - SN\_LP\_02\_01.html
    - SN\_LP\_02\_02.html
    - SN\_LP\_02\_03.html
    - SN\_LP\_03\_01.html
  + Img (image files [raw files should have the SAME name])
    - * 01\_01 (corresponds with html file name)
        + files within can be named to your preference but MUST have a unique name (ex. “carousel” can be added)
        + can put sub-folders, if desired
      * 02\_01
        + 02\_01\_01.png
        + 02\_01\_02.png
        + 02\_01\_03.png
      * 02\_02
        + 02\_02\_01a.png (carousel image series)
        + 02\_02\_02b.png
        + 02\_02\_03c.png
      * 03\_01
        + 03\_01\_login\_01.png
        + 03\_01\_02.png
        + 03\_01\_03.png

### Graphics Requests

Format:

* + Class request example: SN\_LP\_01\_01\_graphics
* MIMS request example: g35\_sp\_programs\_a
  + (Product, Course ID, Chapter, Page, Unique letter/text identifier )
* In Notes section for graphic requests, put the full title in the first request for that class or MIM.
* If you have several graphics in a single carousel or video, list them as **one** request, but make notes to the artist in the Notes section.
* Make sure you add the paragraph HTML tag(s) to break up text in the Notes area, as it is in the transcript.

### Audio Requests

Format:

* + SN\_LP\_01\_01\_audio
  + (Product, Course ID, Chapter, Page, Unique letter/text identifier)
* Pull out audio portions of the transcript into a separate document and attach it to the request

### Maintenance and Archiving

During development, a course or MIM has numerous files added and modified. This includes images, Captivate files, Flash files, HTML files, audio files, etc. Once a course or MIM has passed all review cycles and development is complete, the developer prepares the content to be published on PowerSource.   
  
Create an archived copy of each course, as well as an archive of all source files. Once you have finished your course, archive your files for upload to the shared server. To create a course or MIM archive, simply compress the entire course/MIM at the [COURSE\_ID] level.   
  
When we perform maintenance on a course or MIM, these archives can be retrieved and the developer will have all the necessary files to complete maintenance with minimal effort.

**Going Forward**Use this Universal Naming Convention (UNC) on all classes going forward from April 2013, and all classes currently in development. Any published class will remain as-is, until it comes up for maintenance. At the time of redevelopment, all file names will be edited to match the new UNC.

## Introduction to Developer Software Packages and Add-ons

To develop distance learning courses and Mastery in Minutes modules, developers use several software applications.

Adobe eLearning Suite is used to create Mastery in Minutes modules and the video portions in full-length classes. This package consists of several software applications. The applications used are Adobe Captivate 5, Photoshop CS5, Audition, and Flash CS5.

* **Captivate 5.5** is used for all development, as well as for packaging and publishing the modules.
* **Photoshop CS5** is integrated with Captivate 5, and is used for patching and editing backgrounds, screenshots, and images.
* **Flash CS5** is used for animations and special effects.
* **Audition** is used to edit audio within a module and is integrated into Captivate 5.

All audio files are recorded using Apple Garage Band, and split using Soundbooth, Fission, Audition, or Audacity (free download).

The class shell is developed in HTML5, Javascript, and CSS. The template package is used to create the course elements, and to house the final class content.

* **TextMate** is the text editor used to manipulate class and MIM code. TextMate uses code snippets and customized code to ensure that all of our courses are consistent across all developers.

### TextMate Snippets

If your local machine has not been configured to support TextMate snippets, you will need to download the latest course template from SharePoint and follow the instructions provided in the README.rft file in course\_template/dev/textmate/dl-jade/.

If you have already configured your machine for TextMate snippets and currently use them within the TextMate app, please update your snippets each time a new version of the template is released. Instructions for updating your TextMate snippets are provided below.

1. Download the new course template from SharePoint
2. Locate the dl-jade folder located in course\_template/dev/texmate/
3. Replace the dl-jade folder currently on your local machine with the dl-jade folder in the new template
4. Open the dl-jade folder you have updated in your local files in the Finder
5. Navigate to the textmate-bundle/ folder
6. Double click **DL Developer.tmbundle**
7. Click **Update**

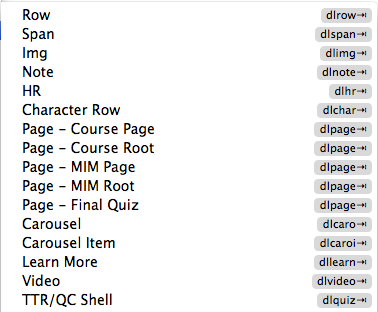
Once you have installed/updated your TextMate snippets they will be available for use within your TextMate app.

### Using Snippets

To begin using your snippets, open the new course\_template you downloaded from SharePoint in TextMate (if you have TextMate in your Mac dock, just drag and drop the class project folder on the TextMate icon). There are two types of snippets: tab trigger snippets and hot key snippets. Let’s first take a look at tab trigger snippets.

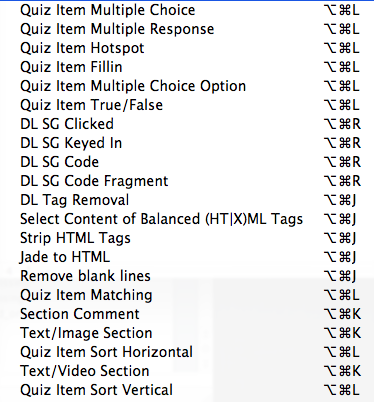
### Tab-trigger Snippets

In the image below, to the right of each type of code, you will see the tab trigger text and characters representing the tab key on your keyboard. So, that’s all you need to do to inject each type of code you find in this image. Type the text into an HTML page in TextMate and press **Tab**. When the code is injected into the page, you will see a portion of the code highlighted. When you change the value for that text, press **Tab**. You will be guided through a number of variables in the code that you may need to customize for your course.



### Hot-key Snippets

The other type of snippet is executed by using hot keys on your keyboard. If you want to add the base set of code for each of the items you see in the image below, you would use the hotkey listed out to the right of each item. You may notice that many of these items have the same pattern of hotkeys; that is ok. When you press those hotkeys, you will get a menu that allows you to choose which specific code you want to add.



When the code is injected into the page, you will see a portion of the code highlighted. When you change the value for that text, press **Tab**. You will be guided through a number of variables in the code that you may need to customize for the specific item.

### Miro Video Encoder

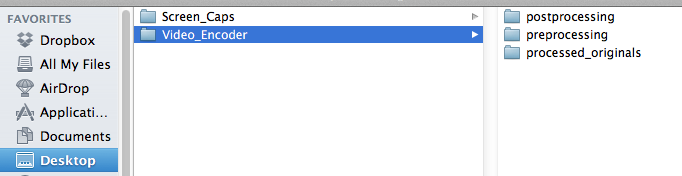
Miro Video Converter is used for compressing published videos before inserting them into your class. You can get the app in the DL Dropbox folder. Please remember to copy the file and paste it to your local machine. If you pull the file out of DL Dropbox to your local machine it will remove it from DL Dropbox.

### Automated Video Encoder Workflow

Since all of our products are now offered in an HTML5 framework we have to encode our videos in multiple formats to accommodate various browsers. The two formats we encode our videos to are .mp4 with h.264 codec and .ogv with ogg theora codec. To make this encoding process as painless as possible you can set up an automated video process on your local machine. Follow the steps below to set up the video encoding workflow.

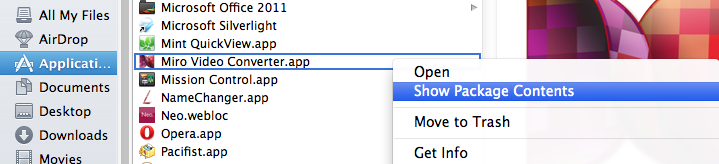
*Create Folders where the Videos will be Encoded*

1. Create a folder named Video\_Encoder on your Desktop. Then add three folders inside named: preprocessing, postprocessing, and processed\_originals.

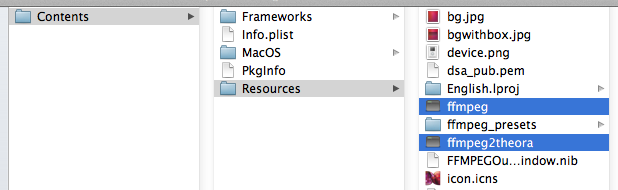


*Copy compiled binary ffmpeg and ffmpeg2theora to usr/local/bin*

1. Open the Terminal
2. In your Applications folder, right-click the Miro Video Encoder app and select **Show Package Contents**



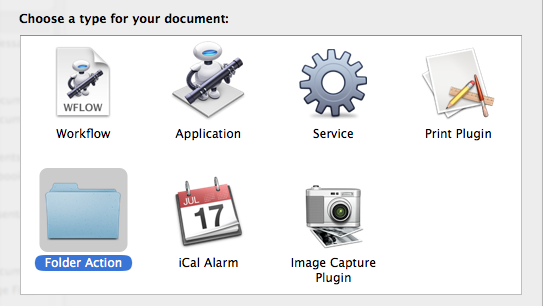
1. Navigate to Contents and then Resources. Inside you should see files named ffmpeg and ffmpeg2theora



1. Go back to the Terminal and enter **sudo cp –i**
2. Drag the ffmpeg binary file to the Terminal. It will probably look like this: **$ sudo cp -i /Applications/Miro\ Video\ Converter.app/Contents/Resources/ffmpeg**
3. Then, create a space at the end of the location and enter **/usr/local/bin**
4. It should look like this now: **$ sudo cp -i /Applications/Miro\ Video\ Converter.app/Contents/Resources/ffmpeg /usr/local/bin**
5. Press **Return**
6. Go back to the Terminal and enter **sudo cp –i**
7. Drag the ffmpeg2theora binary file to the Terminal. It will probably look like this:  **$ sudo cp -i /Applications/Miro\ Video\ Converter.app/Contents/Resources/ffmpeg2theora**
8. Then, create a space at the end of the location and enter **/usr/local/bin**
9. It should look like this now: **$ sudo cp -i /Applications/Miro\ Video\ Converter.app/Contents/Resources/ffmpeg2theora /usr/local/bin**
10. Press **Return**

*Create Video\_Encoder.workflow in Automator*

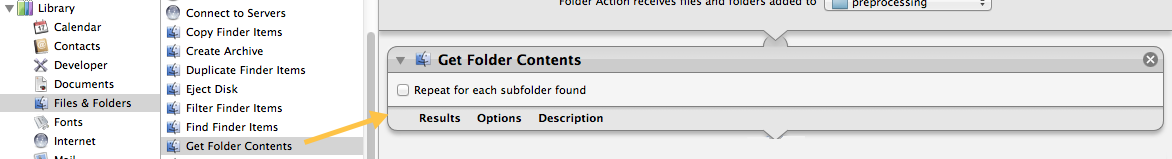
1. Open Automator and create a new Folder Action



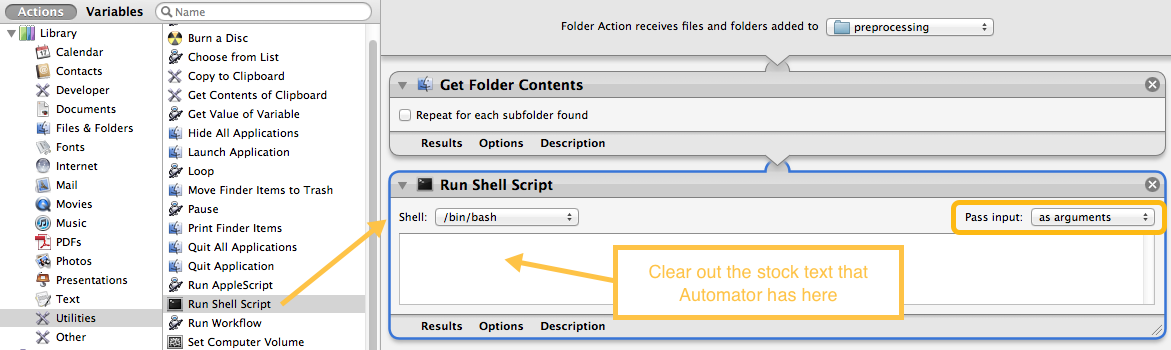
2. Click **Choose folder**, click **Other…**, and select the preprocessing folder located in the Video\_Encoder folder on your Desktop.



3. Drag the the action Get Folder Contents from Library > Files & Folders to the workflow.

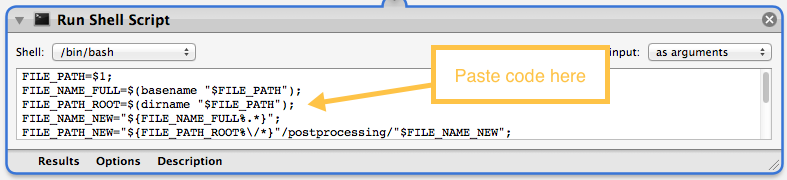


4. Drag the action Run Shell Script from Library > Utilities to the workflow. Change Pass input: value to "as arguments."



5. Paste this code into the Run Shell Script action

FILE\_PATH=$1;  
FILE\_NAME\_FULL=$(basename "$FILE\_PATH");  
FILE\_PATH\_ROOT=$(dirname "$FILE\_PATH");  
FILE\_NAME\_NEW="${FILE\_NAME\_FULL%.\*}";  
FILE\_PATH\_NEW="${FILE\_PATH\_ROOT%\/\*}"/postprocessing/"$FILE\_NAME\_NEW";  
/usr/local/bin/ffmpeg2theora --output="$FILE\_PATH\_NEW.ogv" "$FILE\_PATH";  
/usr/local/bin/ffmpeg -i "$FILE\_PATH" -acodec aac -strict experimental -ab 160000 -vcodec libx264 -threads 0 -f mp4 "$FILE\_PATH\_NEW.mp4"  
mv "$FILE\_PATH" "${FILE\_PATH\_ROOT%\/\*}"/processed\_originals/"$FILE\_NAME\_FULL";  
osascript -e 'tell app "System Events" to display dialog "Encoding Finished"';



6. Save the workflow as Video\_Encoder.workflow

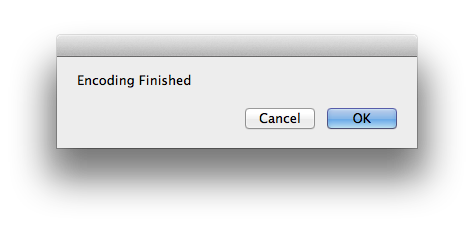
*How to use*

1. Grab all your mp4s you outputted from Captivate and drop them in the preprocessing folder

2. Go ahead and work on something else as your videos encode. You will see the gear in your Mac menu showing the folder action is running.



3. When you see the message "Encoding Finished" pop up, you can look in your post processing folder and see mp4 and ogg versions of all your videos. You will find the original mp4 from Captivate in your processed\_originals folder.



## Templates and Settings

All of the templates used for distance learning development are located on SharePoint at <http://teams.inside.pearson.com/func/PSSDL/default.aspx>. *Always use the latest templates to keep settings consistent and to ensure functionality.*

### Recording Settings

1. Always use the Safari browser for recording/screen captures. In most cases, we can avoid capturing the browser window so we can just focus on the product. Even in cases where the text says something like,“Use your browser’s print function,” we can just mention the action without pointing to File > print. However, there may be reason for us to take snaps using the full browser window. As long as we are consistent throughout the screenshots, we will be fine. If the MIM or class calls for the browser menus, then use it throughout.
2. When recording in Captivate, turn off Full Motion Recording, unless you need it to simulate a drag and drop scenario. Turn off the Mouse Wheel Actions setting in Recording Preferences.
3. Also, turn off Keyboard Tap Sounds in Recording Preferences.
4. In Recording Modes Preferences, turn off all Captions settings, as well as the Mouse setting for Highlight boxes.

NOTE: These settings should be turned off in the Captivate template already.

### Safari Text Settings

1. Open Safari.
2. Click the gear icon on the right.
3. Choose "Preferences".
4. Choose the "Appearance" tab.
5. Select the drop-down menu for "Font Smoothing".
6. Change the option from "Windows Standard" to "Medium - best for Flat Panel".

### Publish Settings in Captivate

1. Resolution: make sure all slides are set to High resolution. Captivate uses low res by default.
2. Audio quality: MP3, 96 kbps, 44.100 KHz
3. SWF quality: bmp High quality (24 bit)
4. Choose Media from the left toolbar. Select type as MP4 Video.
5. Where it says Select Preset, leave the default as Custom. Publish.

**Image and Recording Resolutions**

**Software Demo Video:** 850x520

**Intro video:** 700x520

**Artist Carousel:** 700x520

**Developer Carousel:** 700x520

**Try it!:** 850x520

**Hotspots:**850x520 or 700x520 (whatever is a best fit - developer choice) see screenshot for style guide

**Still Images from Artists:**700x520

**Still Image Screenshots of Application:**850x520, unless smaller size works better for project

**Learn more:** 850x520 or 700x520  (whatever is a best fit - developer choice)

**View images in final quizzes:** 850x520 or 700x520  (whatever is a best fit - developer choice)

### Compression

1. Use Miro Video converter to compress your videos before inserting them into your class files. Open Miro and drop mp4 in drop zone. Open the Pick a Device or Video Format menu and choose MP4 under Other Devices and Formats.  You will compress your original .mp4 twice: once in .mp4, and again as .ogv. Specific instructions in the Development section of this guide.

### DOM Busters

If a page has incorrect information, or information you do not wish to have in your demo recordings, use a DOM buster! This DOM buster works just like a bookmark. When you click it, whatever web page you are on becomes editable. It only edits your screen, not the actual site. To install:

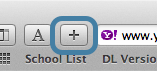
1. Create a new bookmark.
2. Name the bookmark “EDIT PAGE”.
3. Go into bookmarks and edit the new bookmark.
4. Delete the address and replace it with the following javascript:  
     
   javascript:document.body.contentEditable = 'true'; document.designMode='on'; void 0
5. Make sure when you copy that text, that it is one line of javascript. You may have to paste it into Notepad first.
6. Save the bookmark.
7. Create a second bookmark and name it "EDIT OFF"
8. Go into bookmarks and edit the new bookmark by replacing the address with the following one line of javascript (paste into Notepad first if needed to make sure it is one line)  
     
   javascript:document.body.contentEditable = 'false';document.designMode='off'; void 0
9. Save the bookmark
10. Go to a webpage.
11. Click the EDIT PAGE bookmark. Now edit that page like a Word doc!  
      
    When you use the DOM Buster and start to edit the page, the alignment of the page may shift. Some text items will change slightly based on what fonts you have installed on your system. Some page layout items may shift as well. This is due to the fact that editable items render differently. Do not take screenshots of the editable version of the page because of the shift.  
      
    Also, once you go into edit mode, the links on the page become un-clickable. Use your EDIT OFF bookmark to make the page links clickable again.
12. When you're done making edits, select your EDIT OFF bookmark  
      
    Use the "EDIT OFF" feature after you have made your changes, so the page renders properly. This new bookmark should be used after you are done editing a page with the "EDIT PAGE" bookmark and before you take your screenshot. Once you click the "EDIT OFF" bookmark, it will stop the edit process and use the fonts and layout that were intended for the page.

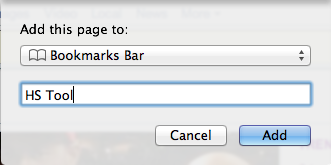
### Hotspot Bookmarklet

The hotspot bookmarklet is a new tool provided in the DL Template 2.0 package. You can find the javascript file located in course\_template/dev/.

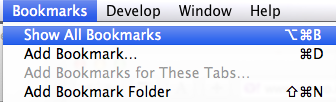
##### Installing the Hotspot Bookmark

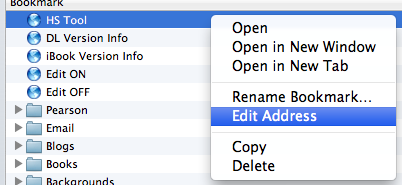
To install this bookmark in your Safari browser follow the steps below.

1. Click the add bookmark button in Safari   
   
2. Add the page to **Bookmarks Bar**, rename the bookmark **HS Tool**, and click **Add**.



1. In the Bookmarks menu, select **Show All Bookmarks**



1. Control-click the bookmark address and select **Edit Address**
2. Open hs\_config\_bmrklet.min.js file in TextMate, copy the code, and paste it into this area



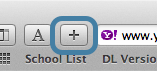
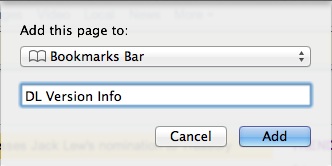
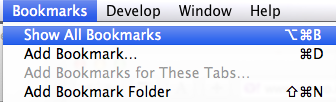
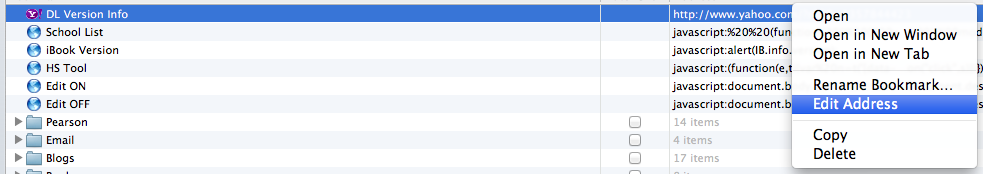
##### Using the Hotspot Bookmark

The hotspot bookmark is a tool that allows you to choose the location and size of a hotspot contained within a hotspot quiz question.

1. Open quiz page that contains the hotspot question in TextMate and Safari
2. Make sure the hotspot question is in view in Safari and click the hotspot bookmark you added to your browser.
3. Move and adjust the size of your hotspot to fit your question
4. When you have the hotspot sized and positioned the way you like, click inside the hotspot
5. Copy the code and replace the coordinates in your code with the ones provided in this code

### DL Template Version Bookmarklet

The DL Template Version bookmarklet gives developers the ability to quickly check the version number of the template their class is using. To install the bookmark, follow the instructions below.

1. Click the add bookmark button in Safari  
   
2. Add the page to **Bookmarks Bar**, rename the bookmark DL Version Info, and click **Add**
3. In the Bookmarks menu select **Show All Bookmarks**
4. Control-click the bookmark address and select **Edit Address**
5. Enter javascript:alert(DL.info.version);  
   
6. Now, when you have a course open in your browser, click the bookmark to see what version it is.

## Introduction to Development Patterns

Development Patterns are the common principles and descriptions the Distance Learning development team uses to build and communicate Distance Learning offerings. Each pattern provides a problem summary, solutions, and a rationale to explain why this pattern is being used. The solutions will also describe conditions of use.

### Demonstration Patterns

**Focusing Attention Using Action**

*Problem Summary*

The learner is listening to the narration in the tutorial, but not certain where to look on the screen. The learner should be paying attention to where the action is occurring within the application.

*Solution*

* Perform actions as most users would in the context of the application.
* Fill out forms, making sure to select the field so Safari will "highlight" it.
* Make the motion slow and methodical.
* Constant motion is not necessary. Pause to let the learner focus on the action occurring in the application.

*Rationale*

The brain is looking for visual changes. Motion causes the learner to focus on the action element rather than other areas of the screen.

**Focusing Attention Using Motion**

*Problem Summary*

The learner is listening to the narration in the tutorial, but not certain where to look on the screen. No data is being entered, and no action is required within the application.

*Solution*

* Move the cursor alongside the text of form elements. If it is text or an error message, move the cursor underneath the text.
* Make the motion slow and methodical.
* Constant motion is not necessary. Pause to let the learner focus on the text near the cursor.
* Using the mouse/cursor movement is the preferred method to point out items on the screen to the user.

*Rationale*

The brain is looking for visual changes. Motion causes the learner to focus on the thing in motion rather than other areas of the screen.

**Click Sounds**

*Problem Summary*

The screen or something on the screen changes abruptly, but it was not obvious what was clicked on in order to cause the action.

*Solution*

* Modify the mouse click to include the visual blue click. DO NOT use click sounds for the mouse in a demonstration.
* Click sounds should only be used for a Try it! interaction when clicking a button/link, opening a menu, or selecting a menu option.
* No typing sounds should be used when entering text.

*Rationale*

When no visual cue is present, the brain cannot process the sequence of action that led to the result.

**Zoom**

*Problem Summary*

When discussing an area of the application in the narration, the learner is not sure what to look at.

*Solution*

* Use the zoom feature in Captivate to highlight the area.
* Make sure to procure an image with better resolution to insert into the zoom area dialog.
* Single lines of text or small portions of the application shouldn't be zoomed.

*Rationale*

While not completely natural, zooming allows the learner to focus on an area of the application and view it in its natural, albeit larger state.

**Modal Windows**

*Problem Summary*

The voiceover narration must explain some elements of the application at length, but there is not always time to cut away from the application, nor is it best to remove the discussion from the context.

*Solution*

* Using the approved graphics and settings, use a modal window to highlight information.
* Insert the illustrations or text in the modal window.
* Center the modal window both horizontally and vertically.

*Rationale*

Modal windows should be used when extra text or visuals must be displayed and the learner must stay connected to a process on a single screen.

**Circles/Boxes**

*Problem Summary*

The learner's attention needs to be on a large, but specific area of the application as the narrator speaks.

*Solution*

Using the approved graphics, lay a box over the top of an area.

* Using the mouse/cursor movement is the preferred method to point out items on the screen to the user.
* You should never have more than one highlight box on the screen at a time.

You may also use a reverse highlight box to call attention to an area. **NOTE: Use these sparingly!** Reverse Highlight box settings:

* Reversed full
* Frame: black
* Border: 1 width
* Fill: 60% transparency

*Rationale*

More natural methods are preferred, but when referring to a larger area and time is not available to zoom in and out, the circle draws attention to the appropriate area of the screen.

**Trekkie Arrows**

*Problem Summary*

The learner's attention needs to be on a specific, small object in the application as the narrator speaks. No action is being performed.

*Solution*

* When possible, use the mouse cursor/arrow within the software to focus the learner’s attention on an area. The learner’s eye will follow the cursor movement naturally.
* If you can’t use the mouse for callouts, use the approved cursor style arrows, and place an arrow pointing to the item in question.
* By default, unless there are compelling reasons otherwise, use the Trekkie style arrow.
* In most cases, there should be no more than one arrow on the screen at a time. Exceptions would be when two elements refer to the same process or function.

*Rationale*

The learner needs to connect the audio with specific items on the screen so they can correctly associate the narration with what is in the application.

**Tooltips**

When developing for Schoolnet/Inform, the developer should record the tooltips that appear when the mouse hovers over a button/icon.

When developing for PowerSchool 7.0, the developer should NOT record the tooltips or blue highlights that appear when the mouse hovers over a menu item.

**Summary Slides (MIMS only)**

*Problem Summary*

The learner has just been through a step-by-step tutorial and needs a brief review of the instruction, but the length of the process does not make it possible to quickly review the steps.

*Solution*

* Following the content provided by curriculum, list the key points on the last slide.
* Provide a visual of the application or other graphic as deemed necessary.

*Rationale*

When the length of the process makes it difficult to review the key steps, the key points to remember reinforce what is most important for the learner to take away from the tutorial.

Use sentence case when creating the text for your summary slides. For example (slide heading): "To enroll a new student:" NOT "To Enroll a New Student:" Use this same rule for the heading in each summary box. Only use a capital letter for titles. When a specific button, link, or menu is being referenced, follow the application for word case. Make sure it matches the transcript.

**Summary boxes**

When a word is written as an action, such as “Click Submit”, the button or link text should be **Bold**.

## Classes

### Get Started

Download the transcript document for the class you will be developing, and download a new version of the template from SharePoint to construct the new class within.

### Lay the Groundwork

### Request Graphics

Review the Class transcript. Create graphics requests for all green text areas within the transcript. **Note:** Graphics should be separate from carousels, separated with a divider line. **The Get Started video found on the Introduction page should be a narrated video.** If your class wasn’t written this way, check with your writer to rework the text so it will work in the new narrated video format.

### Remove Non-Class Index Pages

Open the class folder and remove the “mim\_vid\_index.html” and “mim\_novid\_index.html” files. The only index file you should have in the project should be the “index\_lms.html” page. **Unlike the MIM index pages, you will leave this index page named the way it is.**  The “lms” addition in the name tells the LMS that this project has SCORM content included.

### Prep the TOC page

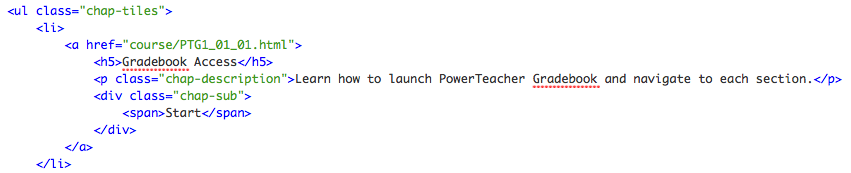
1. Use a text editor to open the toc.html file found in the **course/** folder of your class.
2. Remove all <li> items except the pages with the “ID” prefix (i.e. ID\_01\_02.html, ID\_Get\_Started.html, ID\_Final\_Quiz.html).
3. Replace the “ID” part of each link with the ID associated with the class you are developing (You can use your text editor’s search and replace function for this action, just be sure to make the search and replace case sensitive).
4. Add and remove <li> items to build out your toc based on the class outline provided in the transcript file (blue text). Chapters will be added as nav-headers and subtopics of a chapter will be added as html pages. Please refer to the naming convention document on SharePoint to align you file names to the standards put forth. The link for subtopic <li> items will provide the html file name as the “href=” and the subtopic name will be added as the link’s text.
5. When you have finished building out your toc, it’s time to create html pages for each toc subtopic.
6. To break out the TOC into the two-column format, add an id of “toc-col-break” to the divider right above the section you want to move into a second column. It should look like this: <li id="toc-col-break" class="divider"></li>

### Add / Remove Pages in course/ Folder

To add new class pages, within the course folder, copy the template.html page and paste it in the course/ folder. You will see a new file named “template copy.html.” You will rename this file the name of the page you want to add to your course. Refer to your completed toc.html file to see the list of pages you need to create. ID\_Get\_Started.html, ID\_Conclusion.html, ID\_Final\_Quiz.html, ID\_Thank\_You.html are added for you.

### Prepare Introduction Page

The index page of classes is called the Introduction. On the introduction page you will find a get started video and a vertical stack of chapter tiles. You can wait to source the get started video when you receive the animation from the artist. However, you can add chapter titles, descriptions, and links at this time.



As you see in the code above, within the .chap-tiles <ul> you have <li> elements that contain each chapter title, link to the first page within the chapter, and chapter description. You will need to add or remove <li> items until you get a number that matches the number of chapters in your class. Next, add the chapter title, link, and description to each <li> element. Note, you do not have to precede chapter titles with a number. The <li> items will automatically be numbered by the javascript. Once you finish adding this content, test your links to make sure they go to the first page of each chapter.

### Prepare Class Pages Head and Top Nav Information

Once you have all of your pages added to the course/ folder, use your text editor’s search and replace feature to modify the <title>, meta, and .brand link information for all pages in the class. Be sure to retain all files prepended with “help\_” (i.e. help\_course.html) to ensure the help overlay functions correctly.

Note: If you have a number contained in the text you present on a course or MIM page, mobile devices will sometimes take that number and treat it as a phone number changing its color and making it a link. If you would like to keep that number from being rendered as a telephone number on mobile devices, add the following meta information to the head of the HTML page.

<meta name = "format-detection" content = "telephone=no" />

### Add Content to Pages

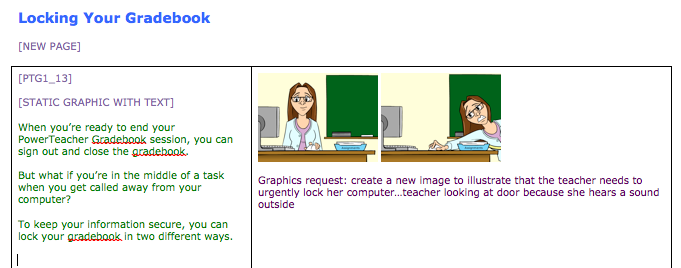
At this point, you should have your graphic requests submitted, your class should have all of its pages in the **course/** folder, your toc should be ready to go, and your <title>, meta, and .brand link updated. Now it’s time to add components to each subtopic page.

### Add Components to Each Page

As you begin to build content, be sure to build the content contained within the first chapter in its entirety before moving on to build content in other chapters, since a prototype review needs to be conducted prior to pre-audio (prototype review covers first chapter of course). However, if you are building the first chapter of the third chapter of a class, the process should look similar.

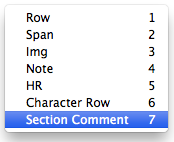
At this point in time you have created the shell of the course and added pages and navigation. Now it’s time to start building the inner framework. Each page can have a number of components on it. Some pages have text and a still image. Some pages have text and a still image accompanied by a carousel and a Try it!. You get the point. Your next step is to begin using TextMate snippets to add the frame of each component to the page.

Let’s take a look at an example.

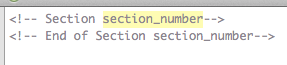


In the example above, the Locking the Gradebook page has a static graphic with text in the first section of the page.

To begin adding content to the Locking Your Gradebook page in TextMate, you first need to add a section comment to the page. Sections of a page are defined as content that fills a span12 in the bootstrap framework (i.e. span8 and span4). Putting components inside of section comments will help you pin point items editors comment on in reviews. To add a section comment using a snippet, position your cursor in the code where you want to add the comment and and press **option + command + K**. In the menu, select **Section Comment**.

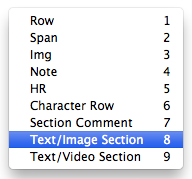


This will inject an opening and closing section comment. Type in the section number and it will change the value automatically on the closing comment.

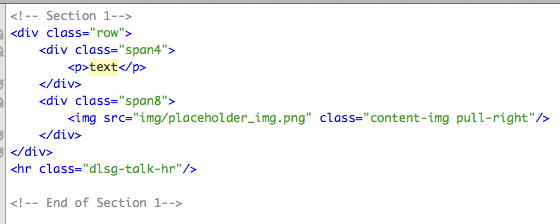




Now you are ready to add your section 1 components. To add some text with a still graphic you begin by using a snippet that will inject the framework of the code for you. Position your cursor inside the opening section comment and press **option + command + K**. In the menu, select **Text/Image Section**.



Now you will see the code added to your page. Once again, you will add the appropriate content to the highlighted areas and tab though until you finish.



If you don’t like to use snippets, you can also add in a section from the demo content pages you renamed x1.html, x2.html..etc. For example, I know that on x3.html there are a few sections on the page that contain one still image and some text. Copy one of these sections on the x3.html page and paste it into the Standards Review and Structure page. However, it is recommended to go the TextMate snippet route (much faster).

Now the Standards Review and Structure page has a placeholder section that you can add content into later. You would perform this same action for each page. If another page needs a carousel, you can copy the code for the placeholder carousel from the X\_Wide\_Carousel.html page and paste it into your subtopic page, or use the carousel snippet. You get the point. Once you have added all of the placeholder components for each subtopic page in your course you are ready to add textual content. **Please note:** the guideline for using a carousel: if the process is fewer than 10 steps and the concepts are straightforward. If there are more steps and the concepts require more explanation, then we go with a video.

### Add Textual Content

Since the transcript provides you all the text you need up front, adding textual content can be one of the first steps you take to begin the process of building out course content. If you are using snippets to build your content you can easily add textual content as you go. If you cut and pasted sections from other pages, you can swap out the placeholder text with the text from your class transcript. In your text editor, add text into each page component that has on screen text (general text areas, carousels, Learn Mores, quizzes, notes, etc.). If you come to a component that has a special creation process (i.e. Try it!, Explore it!, Dialogue, Videos, etc.) be sure to follow the steps outlined in this document for adding text to those components. Note that all headings within the carousel should be blue, and use the <h4> level heading.

### Add Images

### Capture Images

Perform screenshots using Captivate or Snag it! (standard size is 700x520) then resize images that need to be resized to 700x520 (this might be true for some Schoolnet screen caps - just make sure the larger size you are capturing maintains the 700x520 aspect ratio), and add annotations using Snagit!. The standardized highlighting tools will be an import file for Snagit! These will be posted on the Wiki. If you wish to use highlight boxes in a Captivate video file, download the Captivate highlight template.

### Add Images to Image Folder

Place finished images in **course/img/**. You can organize your images into folders within the course/img/ folder if you would like. Add graphic images from artists in this location as well. You can view a diagram of how to name image folders and image files on page 5 of this guide.

All your images (screen captures and graphic images) should be located in **course/img/**. You can also use a placeholder graphic until you get your images from your artist.

### Add Images to Each Page

Go through each page of the course and replace placeholder images with images you have added to your img folder. You replace images by changing the image name in the img src attribute. The path, up to the img folder is already provided in the img src attribute, so all you have to add is the folder and image name (if you have the image in a folder inside of the **course/img/** folder) or just the image name (if you placed the image directly in the **course/img/** folder).

*Image added directly to the* ***course/img/*** *folder*

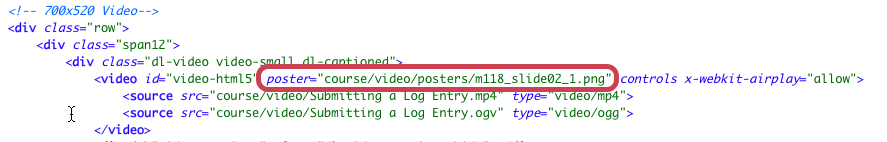
Image_49

*Image added to a folder inside the* ***course/img/*** *folder*

Image_12

### Add Videos

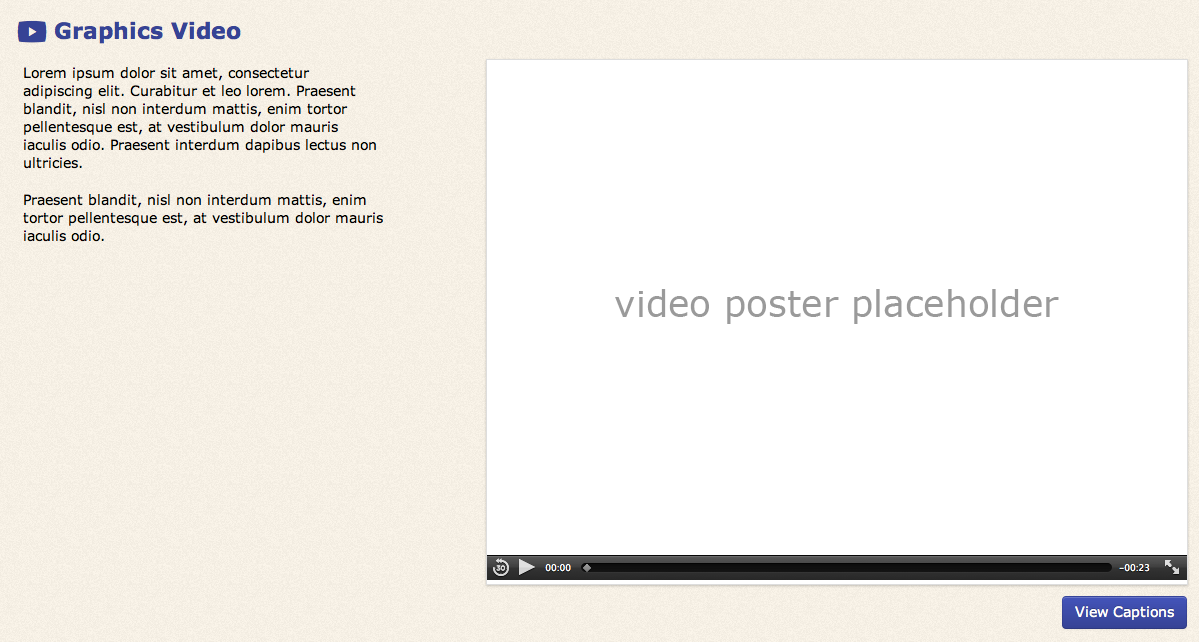
1. If you are creating a video that contains graphics and animations only, do so using the latest Captivate 700x520 template. If you are creating a software demonstration video, create the video with the latest Captivate 850x520 template.
2. Publish the video in Captivate as an .mp4 instead of in Flash. (File>Publish>Media>MP4 Video (\*.mp4). It doesn’t matter if you externalize the skin; the .mp4 export will create its own player.
3. Review your file once it’s published! If you have a blank screen where the Flash animation should be, set the start time of your Flash animation to .01 seconds in the timeline. Re-publish and review again.
4. If you have setup the automated video encoding workflow on your machine, drop the .mp4 video into the preprocessing folder. When the workflow finishes running you will find your new .mp4 and .ogv video files in the postprocessing folder. If you do not have the workflow setup on your machine, take the .mp4 video file and encode it in Miro.
   1. Open Miro and drop mp4 in drop zone. Open the Pick a Device or Video Format menu and choose **MP4** under Other Devices and Formats. The output will look like it has two mp4 extensions video\_name.mp4.mp4, just remove one of them so it looks like this: video\_name.mp4.
5. Use Miro once again to encode the .mp4 you created with Captivate to **Theora** format. Once converted, remove .theora from the file name so it reads video\_name.ogv. This allows us to be mobile-friendly, Firefox-ready, and HTML5 ready. <http://www.mirovideoconverter.com/>
6. Insert the video into your class template.
7. Add both the .mp4 and .ogv versions (converted from Miro) of the video to course/video/ and source your video files in the Main Content area of the Index.html file. Change the source video link to your video’s file name. Change both the .mp4 and .ogv source names.
8. Copy the first screen shot or graphic from the MIM to the course/video/posters folder and source it to the video poster (.png).

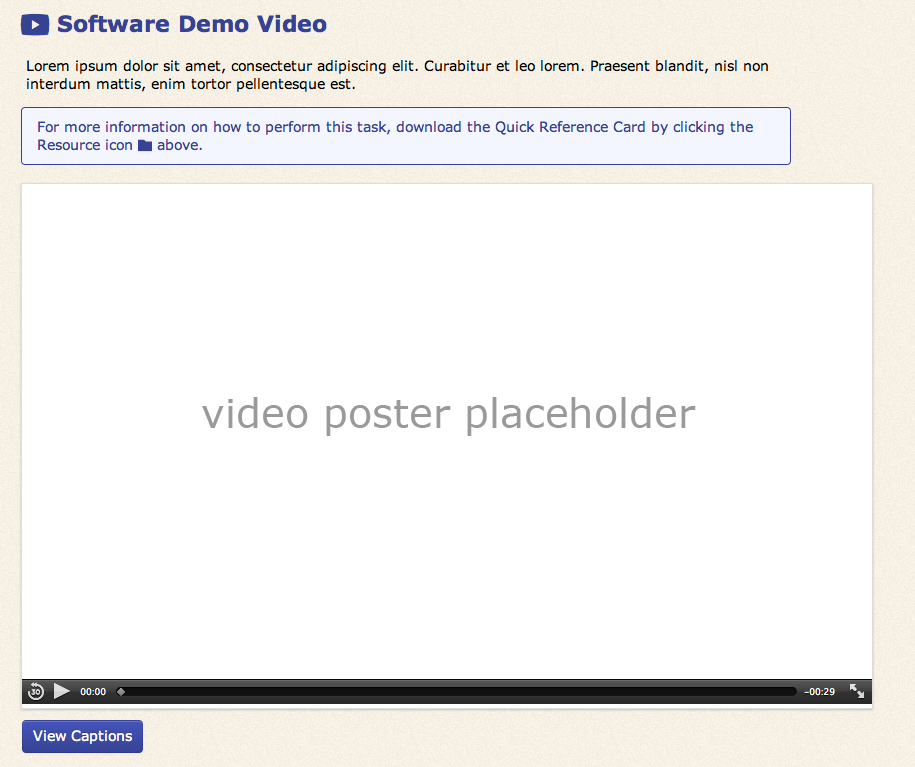


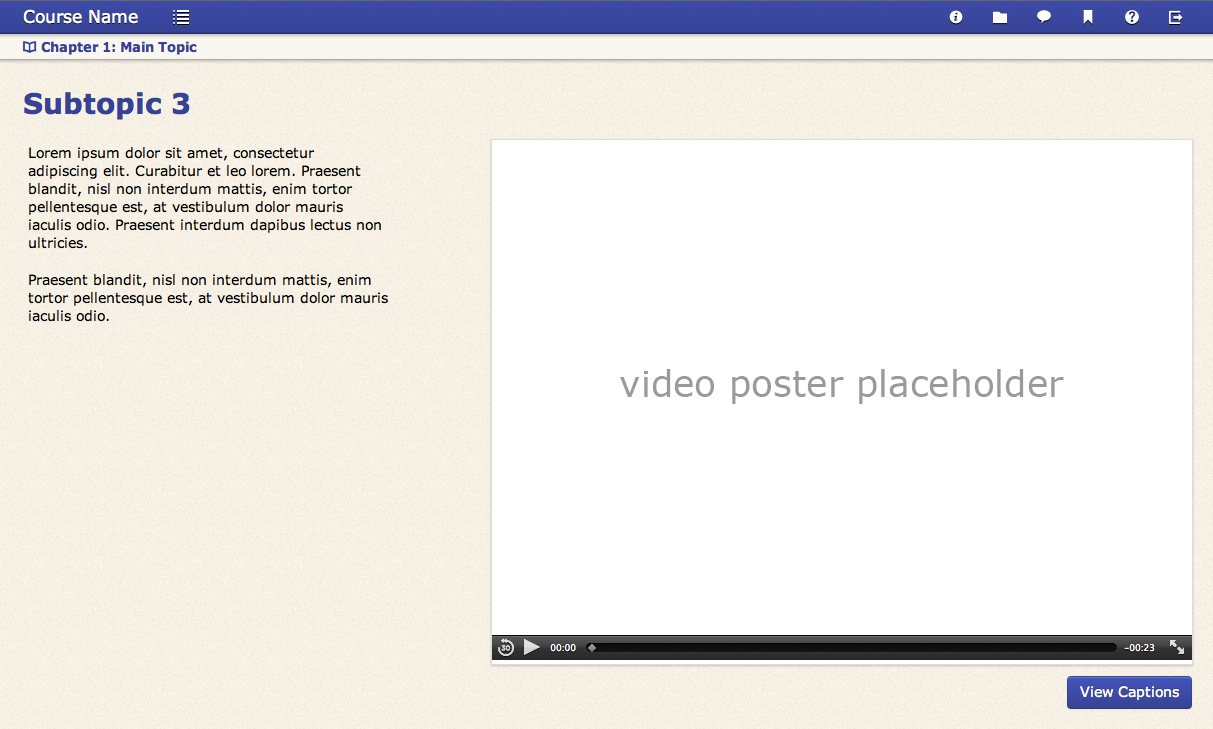
### Add Video Captions

See instructions for adding captions to videos in the MIM Video section of this document.

**Some Notes about video:**

* Make sure any lead-in images and/or text are separate and above the video. Do not use a video lead-in graphic as the video poster. Always use a screen cap of what the video contains as it’s poster image. In the case of graphic-only videos, introductory text will appear to the left of the video. Use hard returns to balance out text on the left with the video on the right. For example, if you only have a few sentences of text on the left, and the video seems overwhelming on the right, break up the text with some carriage returns to make it look more balanced.
* Headers: All Videos that follow another section (text, images, carousel, etc.) need a header. Use a div line between intro/lead-in images/text and video.  
  



* If a video is the first or only thing on a page, or doesn’t have lead in text, don’t use a video header. Just stick with the standard page header.  
  
* In the case of a graphics-only video (700x520), the video needs to be on the right, and text on left. Any accompanying text for the graphics-only video (that is not the narration text) should be to the left of the video. Software demonstrations (850x520) will have any accompanying text located above the video.
* The blue box for software demonstration videos will contain a boilerplate reference to the QRC. **We are no longer using the yellow “note” box in any of our products.**

**Code for the QRC note for videos:**

<div class="alert alert-info">For more information on how to perform this task, download the Quick Reference Card by clicking the Resource icon <i class="icomoon-folder"></i> above.</div>

**Code for a normal blue note:**

<div class="alert alert-info"><strong>Note:</strong> This is where you place the text for your note.</div>

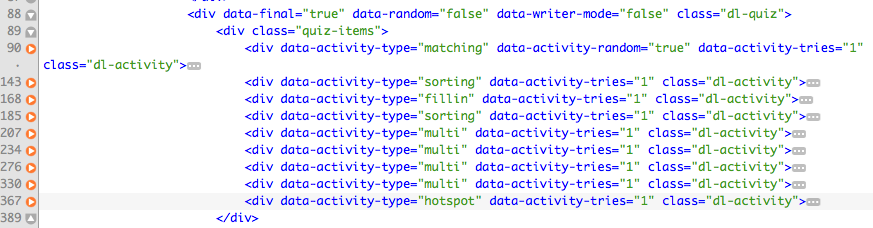
**Red box code (warning message):**

<div class="alert alert-warning"><strong>Warning:</strong> This is where you place text for a warning message. Use these sparingly.</div>

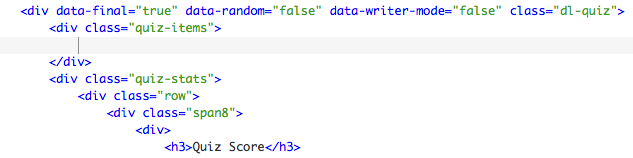
## Interactive Elements

### Add Quiz Questions

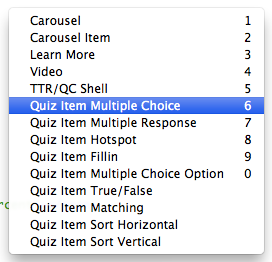
The best way to begin building a Final Quiz is to use the already provided ID\_Final\_Quiz.html page in the template. Simply change the ID prefix to your course ID. When you open the final quiz page in TextMate, you first want to get rid of all the sample questions. You will find all of the questions contained within the .quiz-items div. If you use the collapse arrows to the left to collapse each question, you will see 9 questions in this area.



Go ahead and remove them some you just have an empty .quiz-items div.



Insert your questions using TextMate snippets. Position your curson inside the quiz-items div and press option+command+L. Then select the type of question you want to add from the menu.

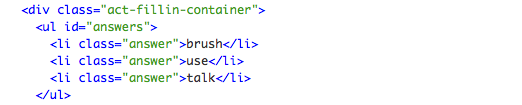


When the the question is added to the page, manipulate the code to fit your specific final quiz question.

### Edit Drag-and-Drop Fill-in Question HTML

The first step when editing a drag-and-drop fill-in question is to add the question statement. After you have added the question statement, it is a best practice to add each answer option provided in the transcript to the HTML.

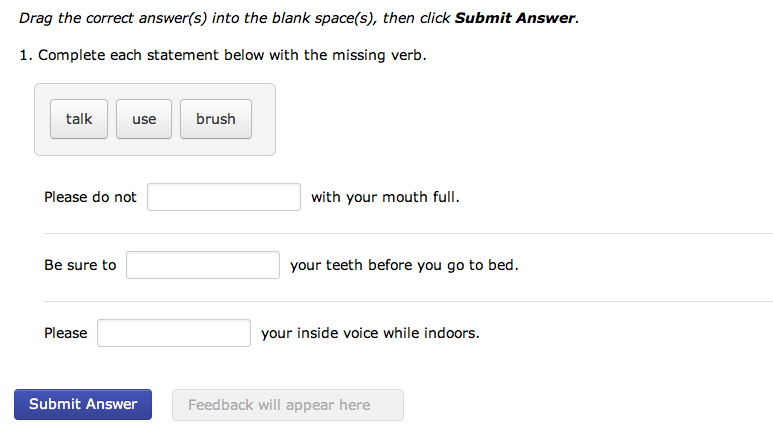
* Add each answer option as a separate <li> element in the #answers <ul>. Add to or subtract from the three default <li> elements to fit the number of answers provided in your question.



* The drag-and-drop fill-in can be presented in two different layouts: inline and block. The default layout of the fill-in will be the inline format. This format provides inputs within a line of text. The block format will present the text first, then provide the input below. In order to apply the inline format, a developer will not need to do anything to the fill-in added via a TextMate snippet or copied and pasted from the template. Both of these versions include the necessary class of “inline-fillins” on the .fillin-container div. In order to change the format to a block layout, remove the .inline-fillins class.

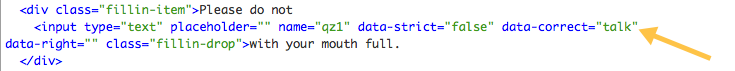
Macintosh HD:Users:umillwi:Desktop:Screen_Caps:Screen Shot 2013-08-29 at 9.43.42 AM.png

* After you choose the layout for your fill-in, begin adding the text and input(s) for each fill-in item.
  + Adding fillin-item text and inputs using the inline layout with separate lines.
    - If you would like to provide separate lines of text with inline fill-ins, you will need to add each line of text and accompanying input to a separate .fillin-item element. The text will go on either side of the input element. Below, you will see a live view of this and a code view.

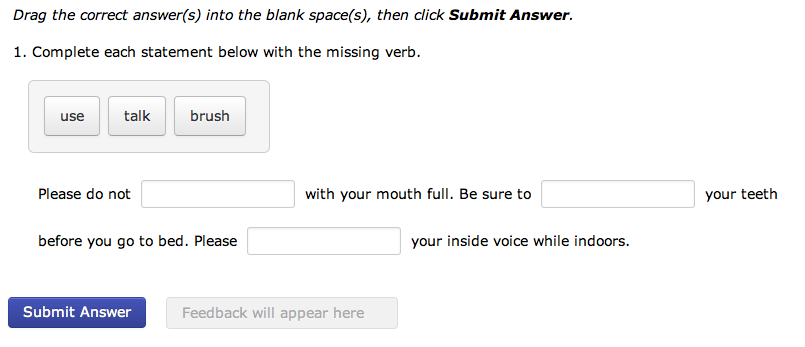


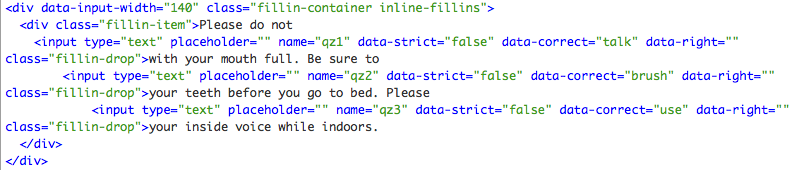


* + - After you have added the text and input element to the .fillin-item div, add the correct answer value of the fill-in to the data-correct attribute.

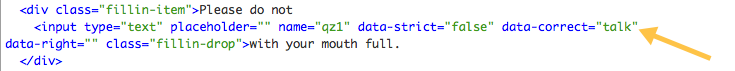


* + Adding fillin-item text and inputs using the inline layout with paragraph format.
    - If you would like to provide inputs within a paragraph, add your paragraph text to one .fillin item div and then add your inputs into the text where they need to go. Below, you will see a live view of this and a code view.





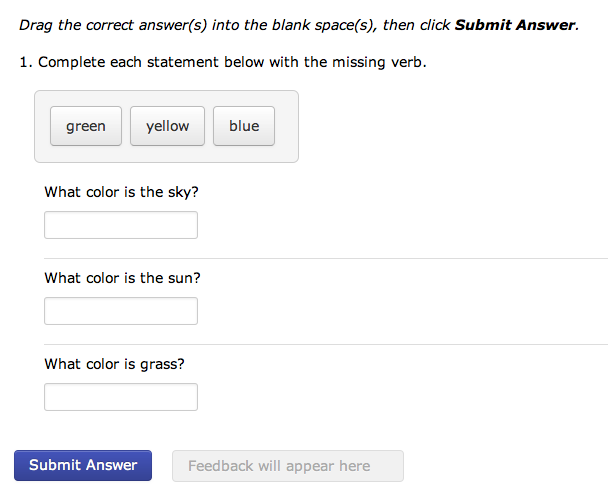
* + - After you have added the text and input elements to the .fillin-item div, add the correct answer value to each input’s data-correct attribute.

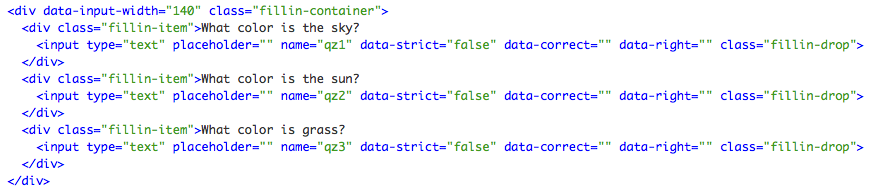


* + Adding .fillin-item text and inputs using block format.
    - If you would like to provide fill-in items with the text located above the fill-in input, first remove the .inline-fillins class from the .fillin-container div.

Macintosh HD:Users:umillwi:Desktop:Screen_Caps:Screen Shot 2013-08-29 at 11.07.49 AM.png

* + - Next, add each line of text and accompanying input to a separate .fillin-item element. The text will go before the input element. Below, you will see a live view of this and a code view.





* + - After you have added the text and input element(s) to the .fillin-item div(s), add the correct answer value to each input’s data-correct attribute.

Macintosh HD:Users:umillwi:Desktop:Screen_Caps:Screen Shot 2013-08-29 at 11.16.24 AM.png

* It is important that your fill-in inputs are all the same size in order that size does not give away the answer. It is also important that the size of inputs be large enough to include the largest answer plus the red “x” or checkmark feedback provided when answer is submitted. To adjust the size of inputs, modify the data-input-width attribute value. The value represents the width of inputs in pixels.

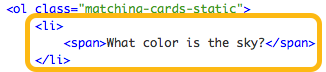
Macintosh HD:Users:umillwi:Desktop:Screen_Caps:Screen Shot 2013-08-29 at 11.27.58 AM.png

* Finally, make sure that each fillin-item input has a name that aligns it to a given fill-in but remains unique at the same time. For example, fi1-1, fi1-2, fi1-3 are examples of names that show that each input are a part of the first fill-in, but are unique in their numbering with the appended -1, -2, -3.

### Edit Matching Question HTML

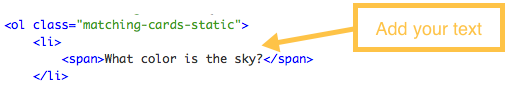
After you add your quiz statement like you do for other questions, on matching questions, you will need to pair two tiles. One tile will be a static tile and the other will be a random tile. The instructions explain below.

* Add to or subtract from matching-cards-static and matching-cards-random <li> items until you have the right number for your matching activity





* Add text for your static (left column) and random (right column) <li> items (note: the first static <li> item's match will be the first random <li> item, the second.. the second, etc)



### Add Quiz Content: Hotspot questions

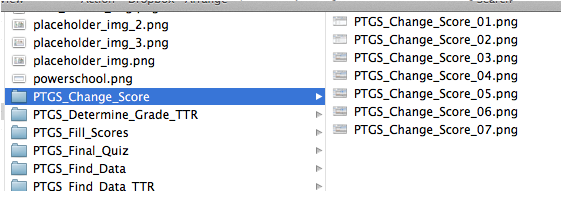
Once you have sourced the image/screen shot you would like to be visible in a hot spot question, the next step will be to set the hot spot size and location. Follow the steps provided below to accomplish this task.

1. Open quiz page that contains the hotspot question in TextMate and Safari
2. Make sure the hotspot question is in view in Safari and click the hotspot bookmark you added to your browser.
3. Move and adjust the size of your hotspot to fit your question
4. When you have the hotspot sized and positioned the way you like, click inside the hotspot
5. Copy the code inside the box
6. In the HTML of the quiz page, paste the code you have copied as the style attribute in the first div tag for the div with a class of “quiz-hot-drop”
7. To change the size of the screenshot/image used for a Hotspot question, change the data-orig-width and data-orig-height attributes.

### Macintosh HD:Users:lasumd:Desktop:Screen Shot 2013-10-22 at 3.19.29 PM.png

### Add Try it!

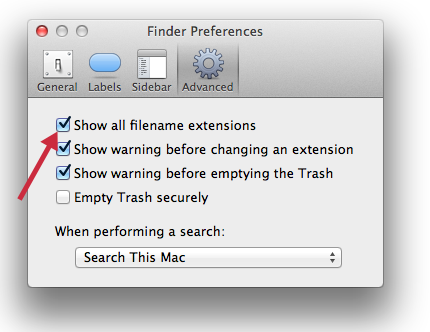
1. Capture screen caps for Try it! (be sure to capture every change on the screen including highlighted text boxes/form fields, etc). Save these images to their own unique folder in your **course/img** folder.



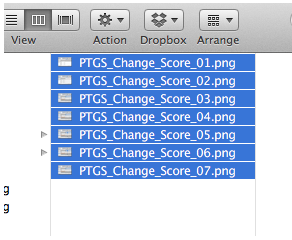
1. Next, copy this folder over to **dev/tryit\_tool/img/** folder.
2. In Safari, open the tryitmaker.html file located in **dev/tryit\_tool/**.
3. Open the Main Settings and add the title of the Try it! including the prefix "Try it!". (i.e. Try it! Title)
4. If an objective is provided in the transcript, add the text to the Objective field.
5. Add a unique id that will be associated only with this specific Try it!. (the id is for HTML/Javascript purposes - an id might look like this: "example-id" )
6. Select the product.
7. Choose whether the Try it! will be a "guided" Try it! or an "non-guided" Try it!. (Guided will include more detailed instructions and be used for first time practice opportunities, while Non-guided try its do less hand holding and are used if the task has already been practiced once before.)
8. If you are creating a "non-guided" Try it! and have instructions, pertinent to a task, that need to be visible (since the instructions accordion will be closed by default), add that information to the Instructions area. (i.e. user name and password for logging in, a certain store code, etc.)
9. Before performing the next step, open your Finder Preferences in the Finder toolbar.



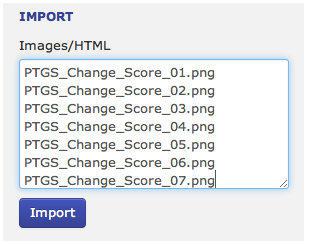
1. Then select Show all filename extensions.



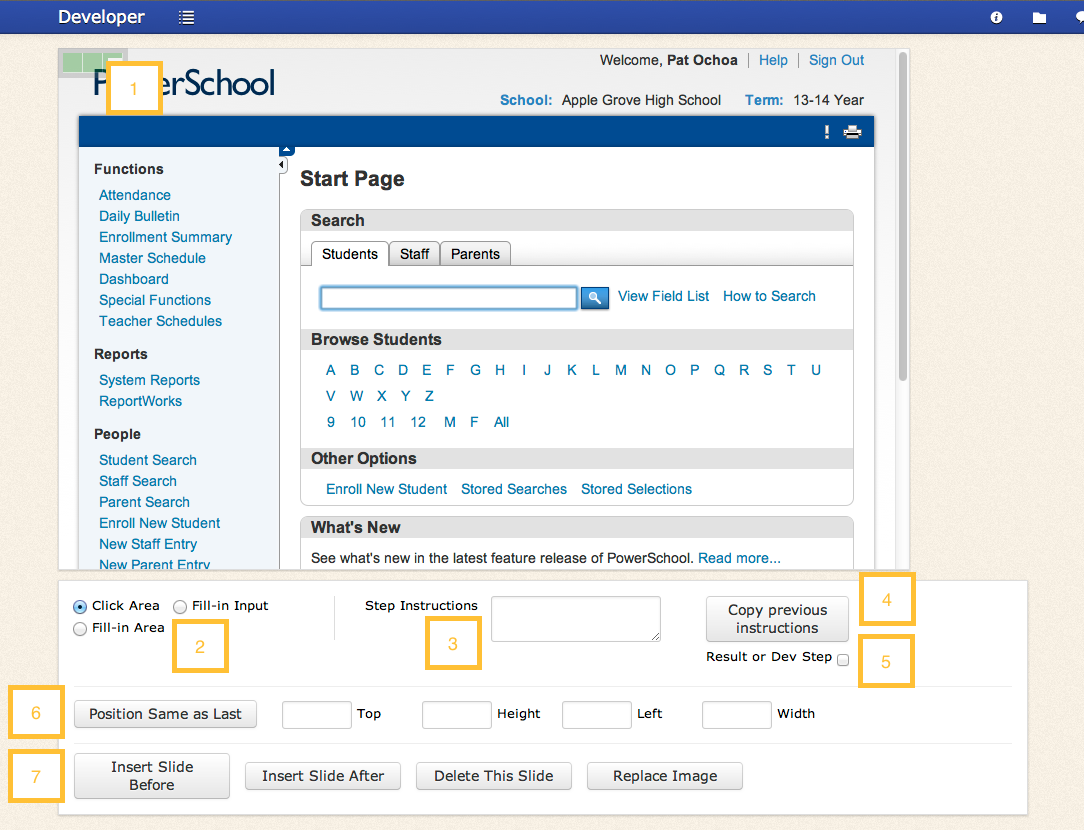
1. Next, add the name of the folder you added to the dev / tryit-tool / img folder. Make sure to add a "/" after the folder name. (i.e. img/folder\_name/ )
2. Now you are ready to add images to the Import section. Open the dev / tryit\_tool / img folder, select and copy all the images in the folder.



1. Paste the copied images into the Import section of the try it tool. Click **Import.**



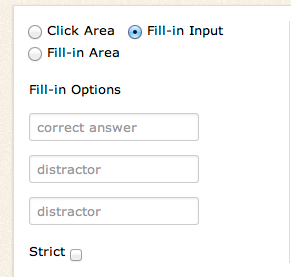
1. Now your images will be visible below, and slide-specific controls will be available to modify image actions. Resize your Safari browser to its full size so your images are full size and in the correct position.



1. Action zone - Move and size this zone over click and fillin regions.

2. Select which type of action is desired for the image. (Use Fillin Input for form fields and Fillin area for larger text entry boxes.)

Note: If you choose a Fill-in option, you will be presented with three input options. If the transcript only provides one option for the fill-in step, just add that option’s text into the first input. If you are provided multiple options in the transcript, add the correct option to the first input and the distractor options in the second and third inputs.



3. Add step instructions here.

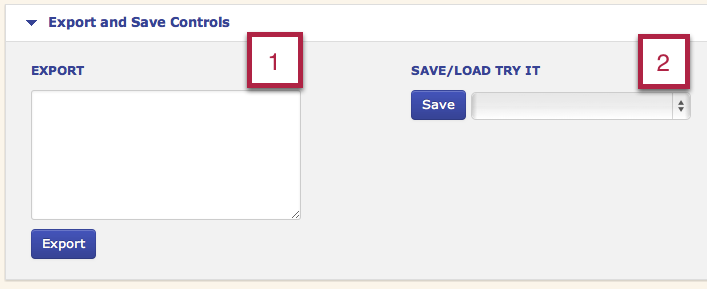
4. Select **Copy previous instructions** to copy instructions from previous step.

5. Select **Result or Dev Step** for the final slide that provides a completion message or for steps not directly related to the app (i.e. scrolling - scrolling is not necessary to accomplish the task for everyone).

6. Click **Position Same as Last** to move the "action zone" to the same location as on the previous slide, or manually position the box using the fields.

7. Slide controls - use these to insert new slides deleted slides, or replace images.

When you have finished building your Try it!, open the Export/Save accordion and click **Export** your Try it! HTML. Click **Save** to save the project to your browser's cache (Warning: if you clean your browser's cache, you will lose this content)

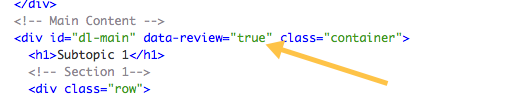


1. Click **Export** to export Try it! HTML. Copy and paste HTML into your course.

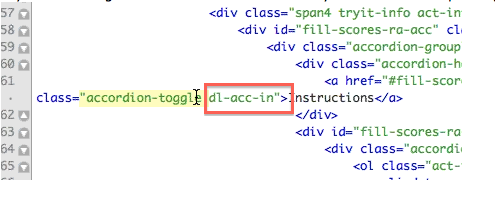
2. Click **Save** to save your project to the browser's cache. Open drop down menu to access previously saved projects.

### Some Notes about Try It’s:

* If you are in a review and would like to enable the Try it! review interface so you can non-sequentially navigate Try it! steps, set the data-review attribute value to “true” on the #dl-main div. When you are done reviewing the Try it!, set the value back to false.



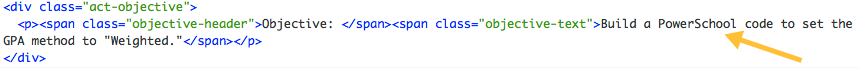
* If the Objective text is the same text as the header above the Try it, contact the writer to modify the objective so it has more detail.
* A Try it should always have a header (H2 header with the Try-it tag so you get the bull’s eye icon)
* Always select GUIDED in the Try it tool when developing a Try it
* When you take screen shots for the Try it’s, make sure your browser is zoomed out (Command 0) so you can line up the coordinates precisely for the “correct” call out box.
* If you are having issues with the arrows in the Try it accordions not pointing the correct way, add the class “dl-acc-in” to the instruction’s accordion toggle.



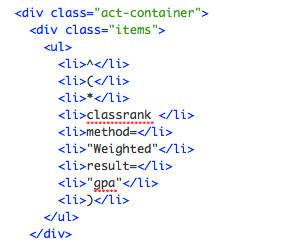
### Add Code Builder

The quickest way to add a Code Builder activity to an HTML page is to use a TextMate snippet. You can use the hotkey combination of option+command+L in TextMate to bring up a menu where you can choose the Code Builder component. If you do not use TextMate snippets, copy the existing Code Builder HTML from the X\_Code\_Builder.html page in the template and paste it into your course page.

1. Once the Code Builder activity is added to the page, begin by adding the objective text to the HTML.



1. Next, scroll down to the .items div and add each part of your code as a separate <li> element. Be sure to put the parts of code in the correct sequence from top to bottom.



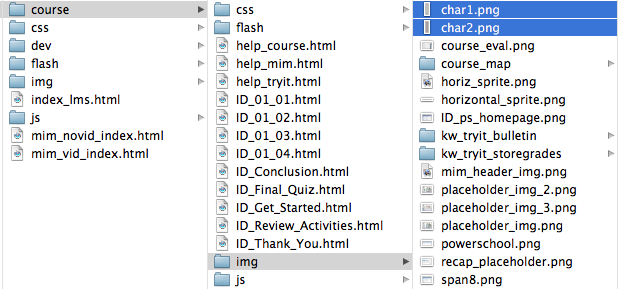
1. Save your Code Builder and give it a try to test the functionality.

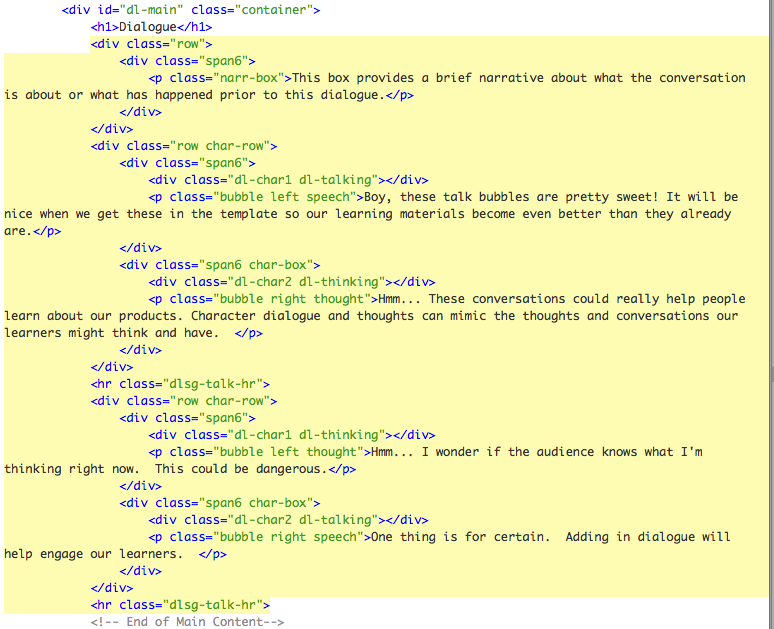
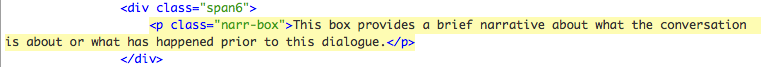
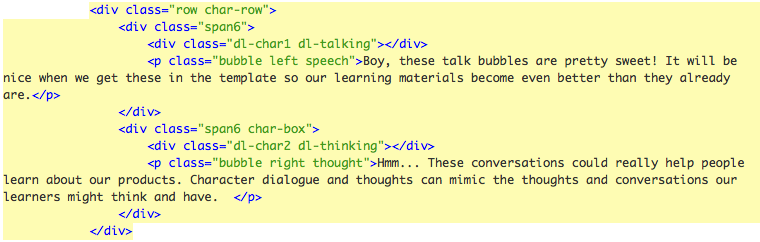
### Add Character Dialogue

Submit graphics request for character sprite(s). In the notes section of the graphics provide the character name, gender and short description, which way the character will face on the page, and the needed expressions. (Look at X\_Character\_Expression.html to determine what expressions you need to request.)

*Note for Artists: Odd numbered character sprites will have characters look to the left (your right) and even numbered character sprites will have characters look to the right (your left).*

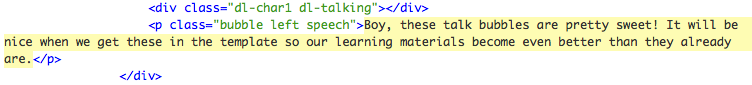
1. When you receive the character sprite images back from the artist, place the images in the course/img folder.



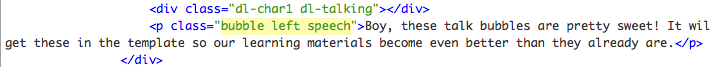
1. Copy dialogue code from X\_Dialogue.html page and add it to course page. 
2. Begin by editing the narr-box text. This text will provide lead in to the dialogue (scenario text, if you will). 
3. Now it's time to add characters and dialogue. Each row ("char-row") will have two characters contained in span6 divs. If you only have one character speaking, use one span6 div in the row. 

To change the character head and expression, modify the dl character and dl expression classes. You can see in the example above the first character head is being sourced from the char\_1.png because the class dl-char1 is used. The expression is talking represented by the class "dl-talking." The second character's head is sourced from the char\_2,png because the class is "dl-char2" and the expression is thinking represented by the class "dl-thinking." Modify these classes to complement the content and discussion taking place in your course. To view a list of the character expressions, see the X\_Character\_Expressions.html page.

1. Next, you will need to edit the speech/thought bubbles for each character. Paste in the text from the transcript that goes with each character’s speech or thought bubble.



1. After you add in the text, you will need to provide classes to style the speech or thought bubble. The first class you will add to the <p> tag is "bubble". That class creates the bubble graphic. The next class you will need to add determines the side of the bubble where the speech or thought lead in bubbles/triangles are located. If the character is on the left side of the bubble, add the class left to the <p> tag (right for the right). Finally, the last class determines if the bubble is a speech or though bubble. Add "speech" class for speech bubble and "thought" for thought bubble.



The example above is a speech bubble with a character head located to the left.

1. Finally, add or remove extra character rows to fit the content of your course (just like adding and removing items for carousels).

## MIMS Development

# Get Started

1. Download the MIM zip file, the keynote, and the transcript file from the appropriate folder on Sharepoint. Unzip the MIM zip file and rename the folder to include your MIM ID and the version number of the PS or SN software (i.e. M09\_7.0.2)
2. MIM no-Video / MIM Video? If the MIM you are developing will not be a video, you will use the “mim\_novid\_index.html” file located in the course\_template/ folder. If you have created a MIM video and are wanting to add the video to the new template, you will use the “mim\_vid\_index.html” file.

### MIM No-Video

### Request Graphics

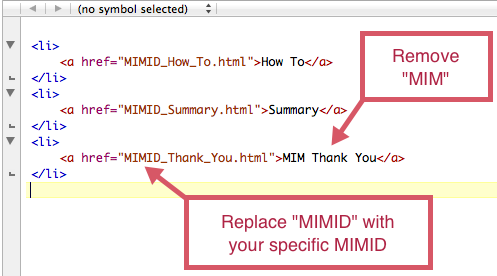
Review the MIM Keynote and transcript. Create graphics request for MIM.

### Rename Index Page

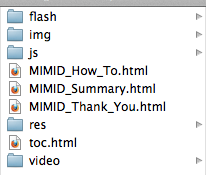
Open the MIM zip file and locate the “mim\_novid\_index.html” file. Rename the file to index.html and delete the other two remaining index files.

### Remove Non-MIM HTML Pages from TOC and Course Folder

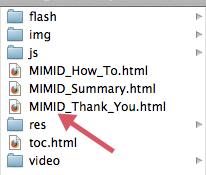
Use a text editor to open the toc.html file found in the course/ folder of your MIM. Remove all <li> items except the MIM Pages. Remove the header and divider leaving only the three MIM pages in the TOC. Replace the “MIMID” part of each link with the ID associated with the MIM you are developing. Remove “MIM” from the “MIMID\_Thank\_You.html” link text (Should read “Thank You”).



Now that you have prepared the TOC to include only MIM pages, it’s time to remove the non-MIM pages from the **course/** folder. Remove all pages from the course folder except “How\_To.html”, “Summary.html”, and “MIM\_Thank\_You.html”, “toc.html”, and “help\_mim.html” (leave resources.html if your MIM will have resources attached).



Then replace the “MIMID” part of each link with the ID associated with the MIM you are developing.



### Prepare MIM Pages’ Head and Top Nav Information

Open each of your MIM pages (index.html, MIMID\_How\_To.html, MIMID\_Summary.html, and MIMID\_Thank\_You.html) in a text editor. Begin by changing the text between these tags: <title></title>, <meta></meta>, and brand information to include the MIM title, author, and MIM description. Note: Keep the prefix “Schoolnet for PowerSchool:” in the MIM title text. Hide the Resources link in the top nav by adding the class "hidden-mim" to the containing <li> element.



Change .brand href to “index.html” here: <a class=”brand” href=”index.html”> MIM Name</a>. These links are in the Top Nav Bar areas of each page. Note: Pages contained in the course/ folder will have an href=”../index.html” but the index.html page will have an href=”index.html”.



### 

### Modify help-mim.html file

Open the help-mim.html file found in the **course/modals/help** folder. Remove the .hide class from zone1 and zone3 and add a .hide class to zone4.

### Add Textual Content

### Index Page

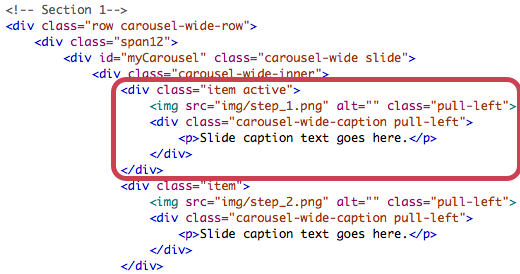
Open index.html file in text editor. Find the Main Content section of the HTML file. Copy and paste “story” text from transcript.html file into the second .span4 div in Section 1.



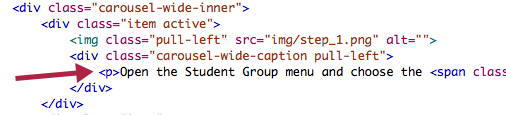
Depending upon how many graphics slides come back for your MIM, you may need to break this text up later. A second Section has been added to get you started when you do. However, once you have pasted your text in, you can move onto the other MIM pages for now.

#### “How To” Page

Open the “MIMID\_How\_To.html” file in your text editor. This page will contain software demonstration content. A carousel will be your tool of choice for displaying software demo slides. A wide carousel has been added for you in Section 1. Copy one of the divs with a class=”item” and paste it over-and-over until you have a div with a class=”item” for each slide you want to go in the carousel.

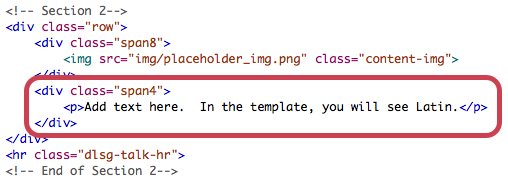


Begin with the first div with a class=”item”. Copy each slide’s caption text from the transcript and paste it into the areas that says <p>Slide caption text goes here.</p>.



If your MIM contains a graphic slide in the middle of the software demonstration, only include the first half of the software demonstration in this carousel, and proceed to steps 8 and 9 to add graphic slide and other software demonstration slides. However, if your MIM does not have a graphic slide in the middle of the software demonstration, add all demonstration slide to this carousel and delete Section 2 and Section 3 from the page.

* (optional) If your MIM has a graphic/story slide placed in the middle of the software demonstration, add the text for this slide in the second span4 div in Section 2 of the “MIMID\_How\_To.html” page.



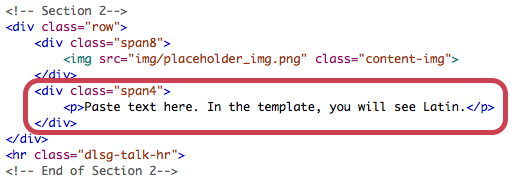
* (optional) If your MIM’s software demonstration is separated into two parts by a graphic slide, add remaining software demonstration slides and captions to wide carousel found in Section 3 following the instructions found in step 7.

#### Summary Page

Open MIMID\_Summary.html file in text editor. Copy the summary slide text (numbered steps) from the transcript and paste it into the span4 div in Section 1.



If your MIM has a closing graphic slide with text, copy the text for this slide from the transcript and paste it in the span4 div in Section 2.



If your MIM does not have a final graphic slide with text, delete Section 2.

#### Thank You Page

You will not need to add any text to this page.

### Working With Images

#### Capture Images

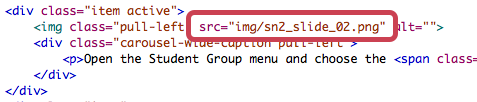
Perform screenshots, resize images that need to be resized to 700x520, and add annotations using Snagit! The standardized highlighting tools will be an import file for Snagit! These will be posted on the Wiki. Place finished images in mim\_name/course/img/. Name screenshots according to their corresponding slide number (i.e. youmimid\_slide\_02.png). Add graphic images from artists in this location as well, naming them accordingly, too.

#### Add Images to Image Folder

Add all your images (screen captures and graphic images) to **course/img/**. See page 5 of this guide for a diagram of how to name image folders and files.

#### Add Images to Each MIM Page

Go through each MIM page and replace placeholder images with images you have added to your img folder. You replace images by changing the image name in the img src attribute. The path is already provided in the img src attribute, all you have to do is replace the file name.

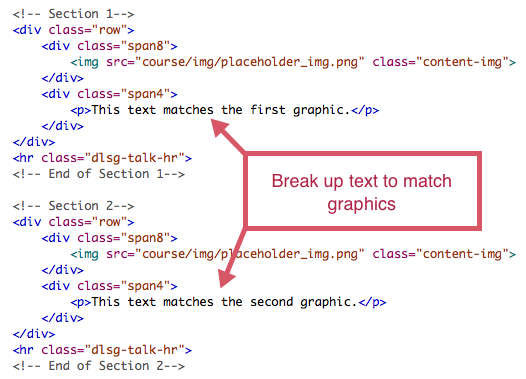


##### Index Page

When you begin to add images to the “index.html” page, you might notice that you have three or four graphic images but only two sections on the page. To add sections, just copy one of the existing sections and paste it below. Remember to change the comment for the new section to say Section 3 or 4 depending on where that section is placed in the series of sections.



After you have created a section for each graphic image. Break up the text that you pasted into Section 1, according to the artist’s instructions, and distribute the text between each content row matching it to its corresponding graphic.



##### How To Page

On the “How To” page you have one or two carousels and maybe a section that contains a graphic image and a span4 for text. To add images to each carousel slide, refer to the instructions contained in step 15. If you find that you have more than one graphic slide in between carousels, and you need to add sections of content to account for the multiple images, follow the same process outlined in step 16 for adding sections and distributing text.

##### Summary Page

There are two types of images you can add to the Summary page: summary slide and “ending” story graphics. Construct the summary slide image using the summary slide Captivate template that you can find in the DL Developer Wiki on Sharepoint. When your summary slide is finished in Captivate, screen capture the image and crop it down to 700x520. Add the summary slide image to **course/img/**.

Once you have your summary slide and graphic images added to the image folder, source the new images in the HTML on the Summary page. Use the same process outlined in step 15. If you need to add additional sections of content to account for multiple “ending” graphics, follow the process outlined in step 16.

##### Thank You Page

The “Thank You” image has already been added to the template, so there is no need to make any changes to this page.

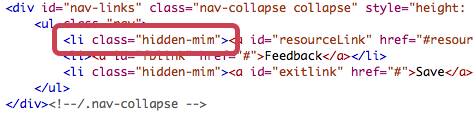
### Add Resources (optional)

If you are adding additional resources to your MIM, add them to **course/res/**.

After the resources have been added to the res folder, open the “resources.html” file and edit the resource links' href attribute to point to the files located in your res folder.

#### \*\*Note\*\*

If your MIM does not have any resources, you will need to hide the Resources link so it does not appear in the top navigation. Open each MIM html page file in your text editor and go to the top navigation block of the code. Toward the bottom you will see the Resources nav link. Add the class “hidden-mim” to the Resources <li> item on each MIM page.



### Remove Unused Folders

Finally, make one last pass of the course folder and remove the unused “video” and “flash” folders. If you have not added any resources, you can remove the “res” folder and “resources.html” page, as well.

### Modify help\_mim.html

Open help\_mim.html page (located in **course** folder) in your text editor and remove the “hide” class on <div class=”zone1”> and <div class=”zone3”>. Next, add the .hide class to <div class=”zone4”>. The opening div tag of zone4 should look like this: <div class=”zone4 hide”>. This will show the help overlay items needed for this type of mim and hide the mim video specific help zone.

### Review and Test MIM

Open the index.html file in your browser and look over the MIM one last time to check for mistakes. Be sure to clean out the cache of your browser before testing. Click every button and link on every page to make sure they function properly. Read over text to catch any mistakes you might have made. And test carousels to see if they move through each slide properly.

### Submit MIM for Review

When the MIM is ready for review, compress the project folder into a zip file and post to SharePoint. Notify reviewers and leads, and change task on Sharepoint.

## MIM Video

**Create Video**

Create 700x520 MIM using latest Captivate template. **Do not add captions to MIM in Captivate.** This will be done later. **When MIM has passed final review, you are ready to add it to the new template.**

**Publish**

Publish MIM in Captivate as an .mp4 instead of in Flash. (File>Publish>Media>MP4 Video (\*.mp4). It doesn’t matter if you externalize the skin; the .mp4 export will create its own player. **Make sure you remove the title slide before you publish.** You’ll add the MIMS title in the new template. Review your file once it’s published! If you have a blank screen where the Flash animation should be, set the start time of your Flash animation to .01 seconds in the timeline. Re-publish and review again.

**Compression**

Take .mp4 video file and encode it in Miro.

Open Miro and drop mp4 in drop zone. Open the Pick a Device or Video Format menu and choose **MP4** under Other Devices and Formats. The output will look like it has two mp4 extensions video\_name.mp4.mp4, just remove one of them so it looks like this: video\_name.mp4.

Use Miro once again to encode the .mp4 you created with Captivate to **Theora** format. Once converted, remove .theora from the file name so it reads video\_name.ogv. This allows us to be mobile-friendly, Firefox-ready, and HTML5 ready. <http://www.mirovideoconverter.com/>

**Create MIM**

* Download the latest version of the course template. Rename the main folder the MIM ID.
* Rename “mim\_vid\_index.html” to “index.html”. Remove “index\_lms.html” and “mim\_novid.html” files.
* Add both the .mp4 and .ogv versions (converted from Miro) of the MIM to course/video/ and source your video files in the Main Content area of the Index.html file. Change the source video link to your video’s file name. Change both the .mp4 and .ogv source names. (Choose the correct content area in the index.html file corresponding to the size of your MIM in the <!--777x577 video --> or <!-- 700x500 video--> sections.
* Remove the code for the video size that you do not need. (<!--777x577 video --> or <!-- 700x500 video--> )
* Attach the QRC, if you have one. If not, hide the Resource link by adding the class "hidden-mim" class to your tag. Change from <li> to <li class="hidden-mim"> in line 52.
* Hide the TOC by adding the class hidden-mim the opening <li> tag under <!--Title of course is here-->. Change it from <li class="dropdown"> to <li class="dropdown hidden-mim">in line 42.
* Remove all unused folders and pages from the **course** folder (leave course/js folder, /video, and “help\_mim.html” file). It’s ok to leave your unused folders there if you aren’t comfortable deleting them. The files are relatively small (much smaller than Articulate/Captivate).
* You will not have to modify the help\_mim.html file for mim videos since the help file is defaulted to accommodate this type of mim.
* Copy your converted video files (course/video) into the video folder within the course structure in the Easy Caption Tools folder. (dev/easycaptions\_tool/course/video) There will be two copies of your files (one copy in the course structure folder, and one copy in your Easy Captions video folder).
* Edit the easycaptions.html file (in the easycaptions\_tool folder) in TextMate to src the correct video file names. (Lines 92 and 93). Save and close.
* Launch the Easy Captions Tool (easycaptions.html) file in your **browser**. Use the Easy Caption tool to create captions for the video and output the caption code. (See Closed Captions Section below.)
* Change <title>, <meta>, and .brand information to include the MIM title, author, and MIM description. Hide your resource links.
* If everything looks good, open the "index.html" file in a browser, watch the video with the caption box open and review.
* If the MIM is free of errors, compress the project folder into a zip file and post to SharePoint. You can delete the Easy Captions Tool folder when your MIM is finalized.
* Notify Team Lead for posting; the final file with CC will be reviewed at this time.

### Add Closed Captions

**Loading Text**

* To start, click the Load Text button and either paste your prepared text in the text area, or type in the text you would like to sync with the video. The tool strips the formatting from your Word doc, but keeps the hard returns, so you should be able to cut and paste right from your Word transcript.
* Break up lines of text as needed. The ideal character limit is 90 characters, so longer sentences may have to be broken up further.
* Click the Import button to load text in the caption table.
* Continue to Break up lines of text as needed. (cut>paste) Note: An empty line with no caption and/or a data start value of 0 will not affect the captions, but it does create a new paragraph element in the transcript. You can remove the blank lines by clicking on the ”-” for those rows to eliminate additional code work later.

**Set Times to Display Captions**

* Start playing the video.
* As it plays, click the cell in the Data Start column when the video reaches the time you want the caption to start displaying. Normally, you won't need the Data End column. It is only required if there is a gap between one caption and the next where you want no caption displaying.
* If you make a mistake, you can scrub the video back and click the same cell again to set the time.

**Add and Remove Lines**

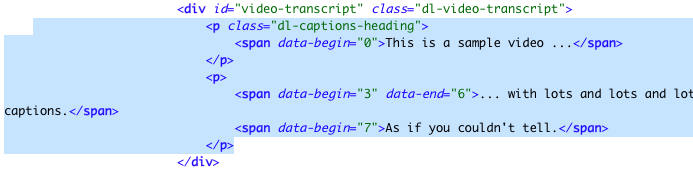
* Click the cell in the Del column to remove the caption on that row of the table.
* If you want to add a row, click the cell in the Add column and a row will be inserted directly below the row you clicked.

**Edit Text**

* To edit the text of the captions, simply click on the cell in the Text column. You can add or delete text in the cell.

**Export HTML**

* When you are finished editing the values for your captions, click the Export HTML button. The completed HTML needed will appear in the text area below the button.
* Select all the HTML and copy it.
* In a text editor, open the "index.html" file.
* Replace the example code (highlighted in blue below) with your captions. This is located in the <--video--> area, below your video src code.



* Click return between each of the </span><span> tags in the HTML so make your transcript easy to read. You’ll need to add in tags for bold text. You can always edit your transcript and timing in the HTML code if you have changes later. See the **Appendix** for a code list.

## mLearning Guide Development

mLearning Guides are developed using Apple’s iBooks Author, and are viewed by our customers on an iPad. The developer will add interactive Widgets to the content using Keynote or HTML5 and Javascript.

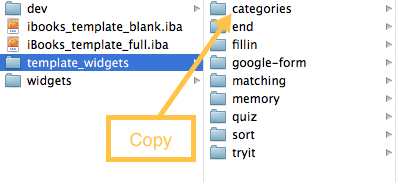
The class writer will have all of the content for the guide inserted into an iBooks Author file. This will have been reviewed for content and style and finalized before it gets passed to the developer for widgets.

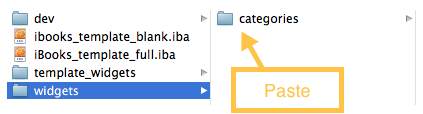
### Developing a Widget

Download a copy of the mLearning Guide template from Sharepoint. Work with the writer if there is already an mLearning Guide containing content for you.

**To create a new widget:**

1. Copy the template widget you want to duplicate, and paste it into widgets folder.

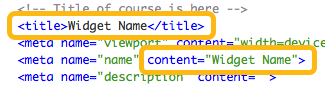




1. Rename main folder of copied widget the name of your new widget.



1. Open index.html file in TextMate
2. Replace "Widget Name" text in <title> and meta name with the name of your widget.



1. Give widget a unique id (id is assigned to h1 element).



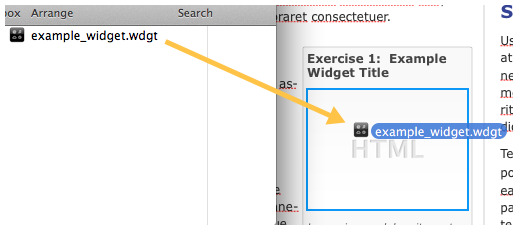
1. Add widget title to the h1 element.



1. Perform steps necessary for the specific widget you are working on (you can find these steps below in Widget-Specific Instructions).
2. If the widget performs properly, add .wdgt extension to the main widget folder.



1. Open iBooks Author project file and drop widget into the widget frame you have already added into the handout

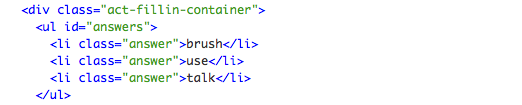


### Widget-Specific Instructions

### Edit Drag-and-Drop Fill-in Question HTML

The first step when editing a drag-and-drop fill-in question is to add the question statement. After you have added the question statement, it is a best practice to add each answer option provided in the transcript to the HTML.

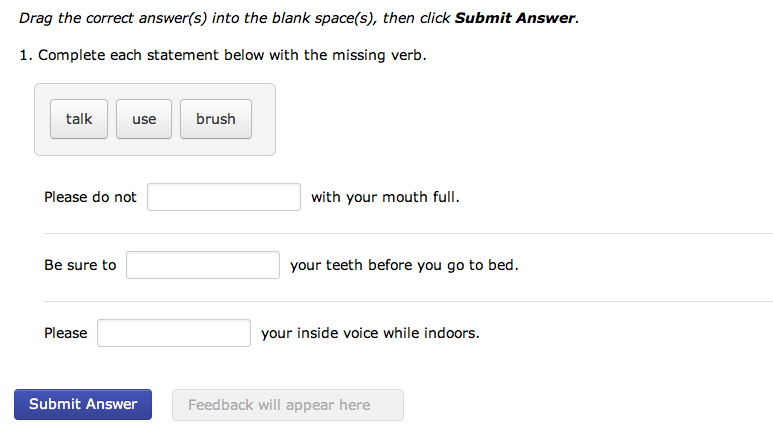
* Add each answer option as a separate <li> element in the #answers <ul>. Add to or subtract from the three default <li> elements to fit the number of answers provided in your question.



* The drag-and-drop fill-in can be presented in two different layouts: inline and block. The default layout of the fill-in will be the inline format. This format provides inputs within a line of text. The block format will present the text first, then provide the input below. In order to apply the inline format, a developer will not need to do anything to the fill-in added via a TextMate snippet or copied and pasted from the template. Both of these versions include the necessary class of “inline-fillins” on the .fillin-container div. In order to change the format to a block layout, remove the .inline-fillins class.

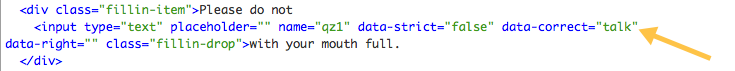
Macintosh HD:Users:umillwi:Desktop:Screen_Caps:Screen Shot 2013-08-29 at 9.43.42 AM.png

* After you choose the layout for your fill-in, begin adding the text and input(s) for each fill-in item.
  + Adding fillin-item text and inputs using the inline layout with separate lines.
    - If you would like to provide separate lines of text with inline fill-ins, you will need to add each line of text and accompanying input to a separate .fillin-item element. The text will go on either side of the input element. Below, you will see a live view of this and a code view.

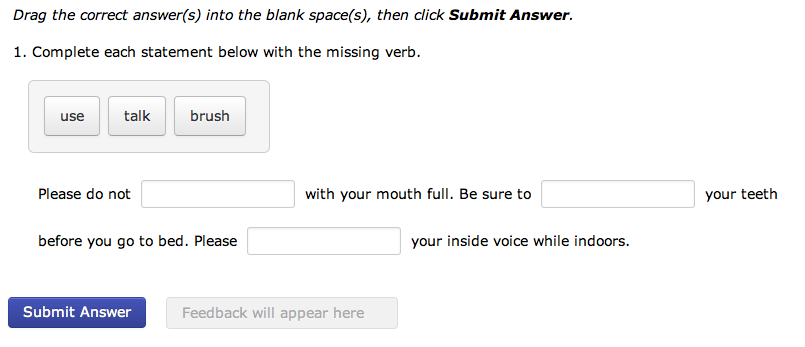


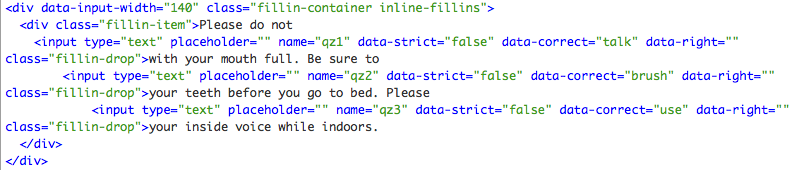


* + - After you have added the text and input element to the .fillin-item div, add the correct answer value of the fill-in to the data-correct attribute.

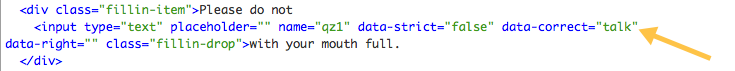


* + Adding fillin-item text and inputs using the inline layout with paragraph format.
    - If you would like to provide inputs within a paragraph, add your paragraph text to one .fillin item div and then add your inputs into the text where they need to go. Below, you will see a live view of this and a code view.





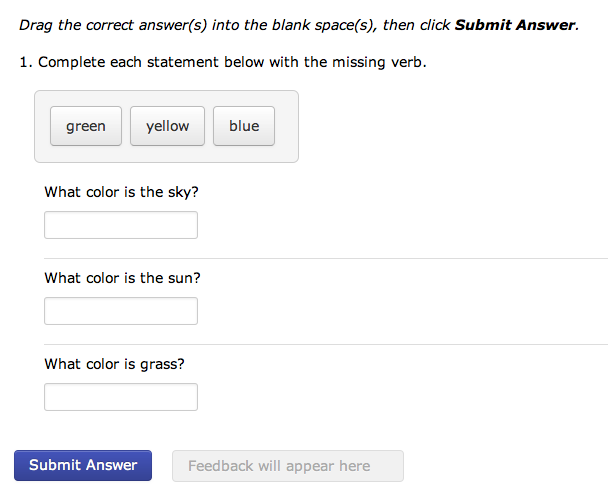
* + - After you have added the text and input elements to the .fillin-item div, add the correct answer value to each input’s data-correct attribute.

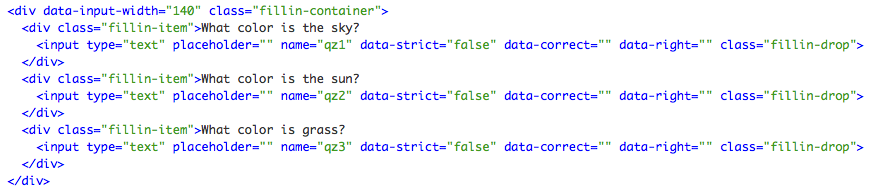


* + Adding .fillin-item text and inputs using block format.
    - If you would like to provide fill-in items with the text located above the fill-in input, first remove the .inline-fillins class from the .fillin-container div.

Macintosh HD:Users:umillwi:Desktop:Screen_Caps:Screen Shot 2013-08-29 at 11.07.49 AM.png

* + - Next, add each line of text and accompanying input to a separate .fillin-item element. The text will go before the input element. Below, you will see a live view of this and a code view.





* + - After you have added the text and input element(s) to the .fillin-item div(s), add the correct answer value to each input’s data-correct attribute.

Macintosh HD:Users:umillwi:Desktop:Screen_Caps:Screen Shot 2013-08-29 at 11.16.24 AM.png

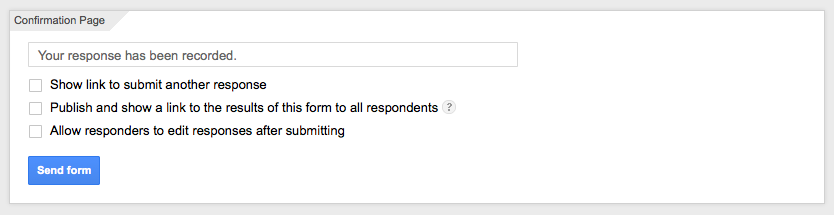
* It is important that your fill-in inputs are all the same size in order that size does not give away the answer. It is also important that the size of inputs be large enough to include the largest answer plus the red “x” or checkmark feedback provided when answer is submitted. To adjust the size of inputs, modify the data-input-width attribute value. The value represents the width of inputs in pixels.

Macintosh HD:Users:umillwi:Desktop:Screen_Caps:Screen Shot 2013-08-29 at 11.27.58 AM.png

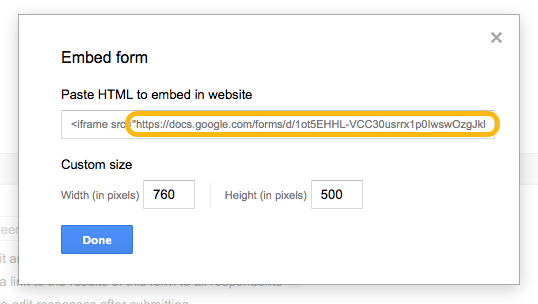
* Finally, make sure that each fillin-item input has a name that aligns it to a given fill-in but remains unique at the same time. For example, fi1-1, fi1-2, fi1-3 are examples of names that show that each input are a part of the first fill-in, but are unique in their numbering with the appended -1, -2, -3.
* Once all answers and fillin items have been added and fillin inputs completed, save the html file in TextMate and test in browser

**Google Form (Poll)**

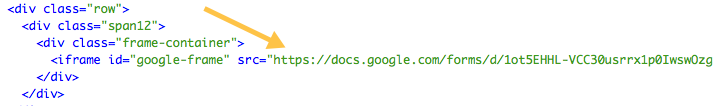
* Create the Google form in the designated Google Drive account for iBooks projects *[pearsonk12techforms@gmail.com/f0rms0719]*
* Google form instructional text (Added to Form Description field in Google Form): "Answer the questions below and tap Submit.
* When editing the form in Google Drive, make sure each checkbox in the Confirmation Page section is empty.



* Click File > Embed, and copy the embed src (this will just be the https://... part of that embed code)



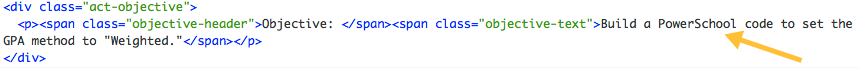
* Paste the URL into widget index.html page in the <iframe> src attribute



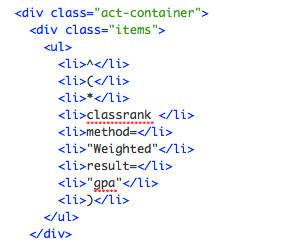
* Save the file in TextMate and open a browser to test

**Code Builder**

1. Begin by adding the objective text to the HTML.



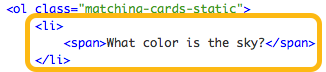
1. Next, scroll down to the .items div and add each part of your code as a separate <li> element. Be sure to put the parts of code in the correct sequence from top to bottom.



* Save your Code Builder and test widget in a web browser

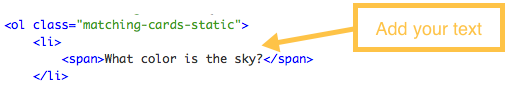
**Matching**

* Add to or subtract from matching-cards-static and matching-cards-random <li> items until you have the right number for your matching activity





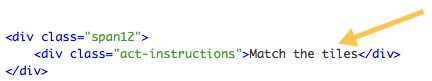
* Add text for your static (left column) and random (right column) <li> items (note: the first static <li> item's match will be the first random <li> item, the second.. the second, etc)



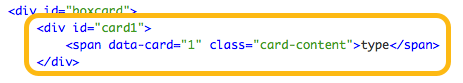
* Once you have added all of your matching items, test widget in a web browser

**Memory**

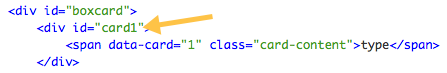
* Modify instructions to fit your specific memory exercise



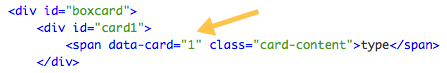
* Add to or subtract from cards to fit the number needed for your memory exercise (maximum of 10)

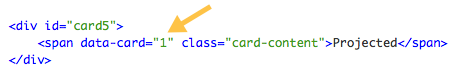


* Make sure each card has a unique id (i.e. card1, card2, card3)



* To signify that two cards match, give them the same data-card number (i.e. #card1 and #card5 match so you would give both a data-card value of 1)





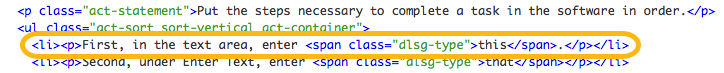
* Now that you have added the cards and text, modify the width of your cards to be optimized for the amount of text you have included on your cards. You can set the width of the memory cards by modifying the data-card-width attribute value. This value represents the width of each card in pixels.

Macintosh HD:Users:umillwi:Desktop:Screen_Caps:Screen Shot 2013-04-26 at 12.10.39 AM.png

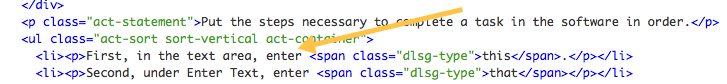
* Once you have added all of your memory items and adjusted the width of your cards, test widget in a web browser

**Sort**

* Add to or subtract from number of sorting <li> items so they match the number you need for your widget



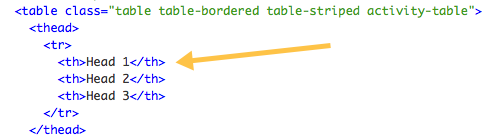
* Add text for each <li> item (be sure to add text in a sequential manner - the order of your <li> items is considered to be the correct sequence by the script)



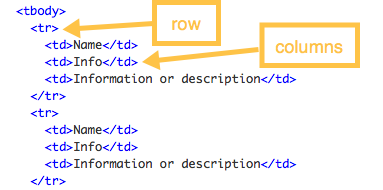
* Once you have added all of your <li> items, test widget in a web browser

**Table**

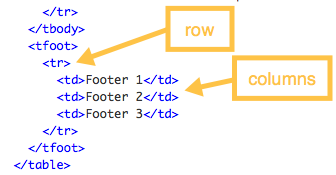
* Add header content in the <thead> element. The number of <th> elements you have will determine the number of columns in your table header.



* Add table body content in the <tbody> <tr> and <td> elements. Each <tr> element in the table body will create a row and each <td> element inside of a <tr> element will create a column within the row.



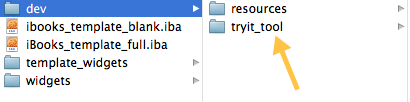
* (optional) If your table needs a footer, you can add footer content in the <tfoot> element. Once again, rows and columns are created with <tr> and <td> elements. If your table does not need a footer, just remove the <tfoot> element from the HTML.



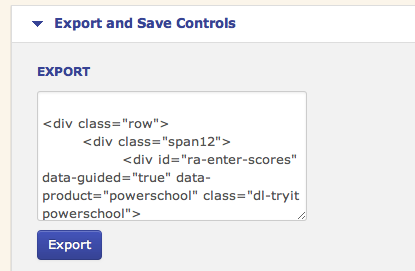
* Once you have constructed your table, test widget in a web browser

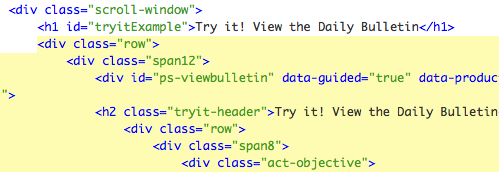
**Tryit**

* Take screen captures and construct Try it! using development tool found in dev/tryit\_tool location

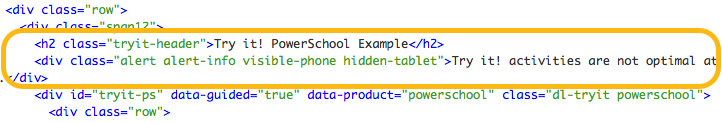


* When you finish constructing the try it in the dev tool, export the code from tool and paste into widget

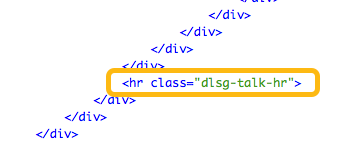




* Remove h2 and .alert div at the beginning of the Try it! code, and remove the <hr> element that is located at the end of the Try it! code.



* Remove the <hr class=”dlsg-talk-hr”> from the end of the Try it! code.



* Test widget in a web browser

**Quick Check**

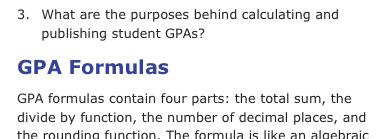
* Remove unwanted quick check questions (.dl-activity elements) from the templates quick check widget and add in the questions you need (you can use TextMate snippets to add new questions)
* Quick checks should include 1-2 questions
* Once you have added all of your questions, test widget in a web browser

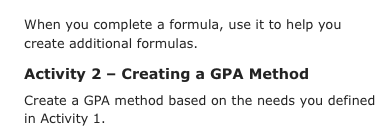
**Quiz**

* Remove unwanted quiz questions (.dl-activity elements) from the templates quiz widget and add in the questions you need for your quiz (you can use TextMate snippets to add new test questions except for sorting questions - you will need to copy and paste existing sorting question in template for those items)
* All Quizzes should contain 3-5 questions only.
* Once you have added all of your questions, test widget in a web browser

**Style Considerations:**

The same styles that are used in courses and MIMS apple to mLearning Guides, such as font styles. In addition, use the styles below for developing widgets.  
  
Remove any hard returns. Sometimes writers put two hard carriage returns in the printed versions of the material. Double-check for these using Find and Replace. Remove any spaces that come before headers in your project. See the examples below:





**Remove hyphenation from main body text, widget titles, and captions**

Highlight all of the text in the mLearning Guide and open the Inspector

In the Text tab under More, check Remove hyphenation from paragraphs

Highlight each widget title and caption and do the same

**Remember which widgets are figures vs. exercises**

Figures are widgets that provide information (such as a Keynote, Gallery, or Movie). Exercises are any widgets that solicit action on behalf of the user (such as Google-form, Quiz, Additional Practice, Matching, Memory, Sorting, Fill-in, Category Game).

**Widget titles and captions for Widgets**

Exercise X: activity name (i.e. Exercise 2: Matching or Exercise 1: Poll))

Captions: (Every widget should have one)

Sample captions: “Sort the GPA code.” “Match the codes with their definition.” (start with the verb, give a little info in the caption)

Inside widget- blue title header: Noun to describe content of activity. (i.e. GPA Codes)

**Figures**

Figure X: Title of chart/image (i.e. Figure 1: Security Groups)

Captions:

Gallery table widget: Tap to view table.

Gallery image widget: Tap to view image.

Keynote widgets: Pinch to zoom in and out of slide(s).

Movie widgets: Pinch to zoom in and out of the video.

Interactive Image widgets: Tap a label to zoom in. Tap the image to zoom out.  
  
**Image Note:**  The optimal resolution for images/pictures used in widgets is 1024x768.

## Reviews

When a MIM, class, or mLG is ready for a review cycle, change the task within Sharepoint, then email the Review Team.

Published class files should be uploaded to the Shared Server at smb://usrancasrvfs002.peroot.com/PSSPEDL. The review sheet and the transcript (if there were changes) should be uploaded to our Sharepoint site at <http://teams.inside.pearson.com/func/PSSDL/default.aspx> in the correct class/MIM folder.

ALL MIM and mLG files should be uploaded to our Sharepoint site. Remember to upload your published files, your review sheet, and the transcript file (if there were changes).

## Appendix

### List of CSS Classes

When it comes to tagging text within the template, in order to apply a style to the text, we have a couple directions you might take.

First, add template specific spans with unique classes. We do this for three types of situations: anything that is clicked, anything that is typed, and any code fragment. You use the span tags below for each of those scenarios.

**dlsg-click**

In instructions, identifies an element the user is going to click or select. For example, Click <span class="dlsg-click">Next</span> would look like this: Click **Next**

**dlsg-type**

Use this when a character or the user is typing something using the Courier New font. For example,

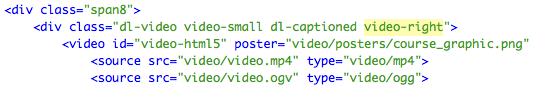
<span class="dlsg-type">Type, type, change, change</span> would look like this: Type, type, change, change.

**dlsg-fragment**

Use this for code fragments contained in a body of text. This will *italicize* the text. For example, Click <span class="dlsg-fragment">for (int i=0; i < 10; i++)</span> would look like this: *for (int i=0; i < 10; i++)*

**video-right**

When you have the video positioned in a span8 on the right side of the page, add this class to the video div contained in span8.



**dlsg-talk-hr-hide**

When you want the separation a header row provides between subtopics but do not want the blue line visible, just add"-hide" to the end of the header row class "dlsg-talk-hr".

Image_13

**video-header, video-header-h3**

Add these classes to your video h2 or h3 element to add the video icon to the header. The "video-header" class will be added to h2 header and the "video-header-h3" class will be added to an h3 element.

Image_42

*or*

Image_40

**tryit-header, tryit-header-h3**

Add these classes to your Try it! h2 or h3 element to add the try it icon to the header. The "tryit-header" class will be added to h2 header and the "tryit-header-h3" class will be added to an h3 element.

Image_10

*or*

Image_36

**exploreit-header, exploreit-header-h3**

Add these classes to your Explore it! h2 or h3 element to add the explore it icon to the header. The "exploreit-header" class will be added to h2 header and the "exploreit-header-h3" class will be added to an h3 element.

Image_16

*or*

Image_11

**carousel-header, carousel-header-h3**

Add these classes to your carousel h2 or h3 element to add the carousel icon to the header. The "carousel-header" class will be added to h2 header and the "carousel-header-h3" class will be added to an h3 element.

Image_50

*or*

Image_47

**Carousel header within carousel (on each slide in place of Steps)**

<h4 class=”carousel-header-h4”>slide header</h4>

**pull-right**

Add this to anything that you want to float right. ( i.e. Anytime you put and image on the right side of the page in a span8, add this class to the img to float it to the right side of the span8.)

Image_54

**hide**

Add this class to any element you want to remove visibly from the page, but want to remain in the backend HTML.

Image_48

### Lists

**Ordered and Unordered Lists**

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag. The list items are marked with bullets.

<ul>

<li>Coffee</li>

<li>Milk</li>

</ul>

Looks like this:

* Coffee
* Milk

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

The list items are marked with numbers.

<ol>

<li>Coffee</li>

<li>Milk</li>

</ol>

Looks like this:

1. Coffee
2. Milk

### Links

You can embed links within your class to reference other materials on a particular subject. Add a link like by using the href tag. For example, using this tag in a sentence creates a link to another DL course:

<a href="<https://powersource.pearsonschoolsystems.com/training/dl/main.action?course.id=34>"> Make sure to close your href tag with </a>.

Looks like this on the page:

You can learn more about these tasks by viewing the Distance Learning class PowerSchool Administration and Management, or by accessing the reference documents attached to this class.

## Developer Checklist

### Demos

1. Sign-in name on the right is consistent throughout. Use Pat Ochoa. Using existing name for course updates, unless most screenshots need to be redone. Majority rules.
2. Image resolution is consistent throughout course/video. You may use a half-size image for sections without much text. The width needs to be 700, but the height can vary.
3. Screen position remains the same throughout demo.
4. No abrupt scrolling/movement. Only use Full Motion Video recording for scrolling when absolutely necessary.
5. No typing sounds when entering text.
6. No clicking sounds when moving the mouse.
7. Steps follow script correctly.
8. Limited horizontal scrolling. Use if necessary on certain screens. Let application dictate.
9. No graphic artifacts such as blurred edges, pixelation, distortion, or remnants of merged graphics.
10. Preview full Captivate file in browser.
11. **Modal Windows:** Use only the windows in the approved library. Be consistent throughout course.
12. **Highlights:** Use only the blue/yellow highlights and arrows. The package can be downloaded from the developer wiki.
13. Whenever there are change logs or change histories, be sure that they make sense—does the date the historical grade was changed fit with when the original grade was recorded?
14. Closed Captions for videos should match the audio. Be sure format text with bold or the Courier New font, as listed in the transcript.

### Interactive Elements

1. Sign-in name on the right is consistent throughout. Use Pat Ochoa. Using existing name for course updates, unless most screenshots need to be redone. Majority rules.
2. With text entry boxes:

* Cursor is not present in text fields
* No blue "active" fields for text/user entry; use on demos only

1. Continuity—is the data filled in from the last field?
2. USE clicking sounds when clicking a button, opening a menu, selecting a menu option.
3. Make sure text entry box is aligned so you don’t create screen hitches.
4. Double-check click box/menu selection area size.
5. Menu item to click, select, or press is bold in user directions. The template will automatically add “Then press Tab/Enter” for you. Do not add this text to the instructions.
6. Limited horizontal scrolling. Use if necessary on certain screens. Let application dictate.
7. Time to Review and Quick Check sections should not have any additional text in the title.
8. The font should be **Bold Courier New** when the learner is to type/enter text, unless otherwise noted on the transcript.
9. Preview/test published file in browser.