Let's Design for Web!

Let's Design for Web!

RICHARD ADAMS

Introduction

This eBook contains the tutorials for the Ryerson online class, GCM 362, “Web and Cross-media.” The eBook was created from separate PDF files with tutorial instructions so that all tutorials would be in one place.

Thanks to all who contributed to the tutorials, including:

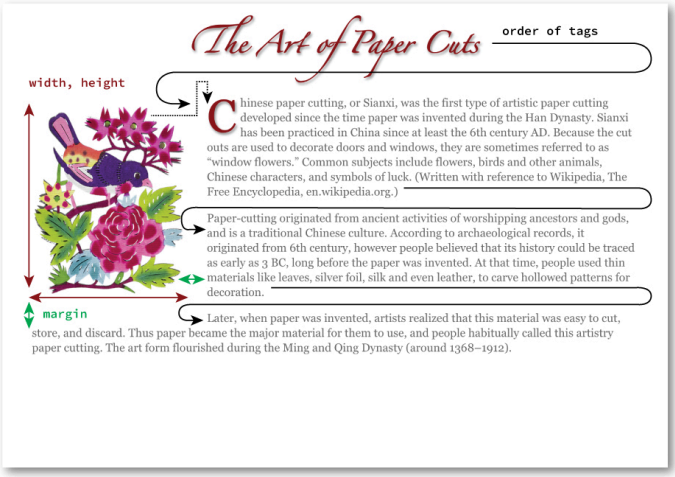
- Ahmed (Am) Sagarwala (GCM ’08), jQuery
- Mark Corrigan (GCM ’13), MailChimp
- Chris Ambedkar (GCM ’15, MDM ’16), user experience design
- Prof. Dr. Okke Schlüter, HdM-Stuttgart, DesignAgility
- Prof. Bettina Tabel, HdM-Stuttgart, Design Critique

Tutorial 1 • Getting Started

Page Layout for the Web

Web pages differ from print pages in that the designer never knows what computer, device, operating system, monitor, screen, browser, or window size the reader will be using. The page size is determined by the user's browser window, so you can only control:

- order of page elements (tags)
- tag width and height
- spacing around the tag (margin or padding)
- text wrap ("float")



Web page with positioning characteristics outlined.

Tags

HTML. Web pages are described in the hypertext markup language (HTML) and styled using cascading style sheets (CSS). Page elements like paragraphs, headings, and images are specified in tags, enclosed in less-than and greater-than symbols, such as opening and closing , for paragraph. The tags are similar to highlighting, by telling where the highlight color should start and where it should end.

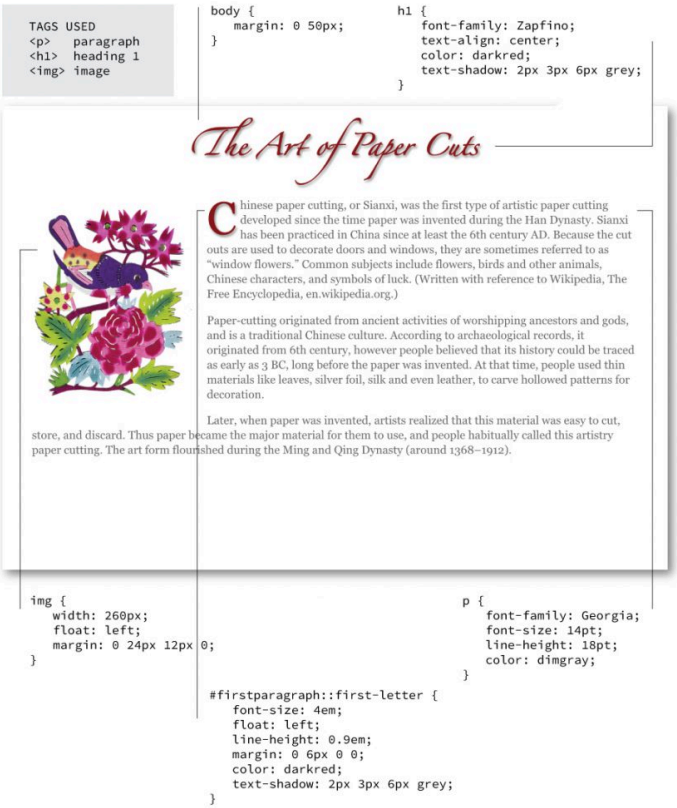
Commonly Used HTML Tags	
tag	**meaning**
	paragraph
<h1></h1> to <h6>	heading, h1 largest and h6 smallest
<img />	image (self-closing)
	division, used to create a text or picture box or to group items
	unordered list (bullet points, each point is a Note: and are sometimes used to create menus)
	ordered list (numbered items, each is)

CSS. Styles can be specified directly in the tag ("local" or "inline" styles), in the head of the document ("global" or "embedded" styles), or in a separate, linked file of type .css ("cascading style sheet"). "CSS" has two meanings: (1) the hierarchy of local/global/css and (2) a file type of .css. Styles can include size (width, height), spacing (margin, padding), font, color, background, and many other characteristics.

Inline and Block-Level Tags. "Inline" describes tags that flow one

after the other, like characters in a word or words in a paragraph. Example: <img> tag. “Block-level” refers to tags that have an automatic “paragraph return” after them, so subsequent tags go to separate lines. Examples: <p>, <h1>, <li>, .

W3Schools Reference. The site “w3schools.com” is a handy reference for HTML tags, CSS styles, and lots of other information about the web. If you are not sure how to create a specific look, try searching the site. Example: You want to create a drop-shadow around an image. Searching for “drop shadow” will lead to an explanation of the “box-shadow” and “text-shadow” styles. Also try searching the web for “CSS drop shadow”; the first entry will probably be from W3Schools.



Tags and styles used to create a simple web page with a heading (h1), three paragraphs (p), an image (img) with text wrap (float), and drop-cap.

Let's Create a Web Page

1. Open the file "greywolf_START.html" in Dreamweaver and Save As "greywolf.html."
2. Insert and center Header image: images/_DSC0067_1004.jpg
3. Link to Home (index.html) and The Grey Wolf (greywolf.html) using "a href" tags (e.g., Home).
4. Make "The Grey Wolf" into a *heading* <h1...h6>, where h1 is the biggest.
5. Designate each paragraph with a <p> tag (opening and closing tags).
6. Add a *local* or *inline* style to the first <p> tag and select a font you would like to use (e.g., <p style="font-family: Georgia;">
7. Inside the <head> tag, add *global* or *embedded* opening and closing <style> tags. Inside the tags, write selectors for the tags, and style specifications in braces {}. E.g., p {font-family: Georgia;}.
8. Insert image "_DSC0061.jpg" in front of paragraph, float left with 20px margin.
9. Indent the text block left and right.
10. Choose and measure light RGB colours from the image, and colour the header and body.

Home · The Grey Wolf

The Grey Wolf

According to this tale, the Grey Wolf was an intelligent character with magical powers. The tsar sent his youngest son, Ivan, to find out who was stealing golden apples from a magical tree in his orchard.

Spending the night in the orchard, Ivan observed that the Firebird came and took an apple. He tried to catch it but only got one feather. The firebird did not return, so Ivan set off on horseback in search of it.

Ivan came to a three-way fork in the road. A stone marker proclaimed that whoever took the first road would know hunger and cold; the second road, his horse would die; and the third road, he would die but his horse would live. Ivan took the second road. The Grey Wolf came out of the woods and ate his horse. Ivan walked until he was exhausted. The Grey Wolf appeared and offered to carry him.

The wolf brought Ivan to another kingdom with a magical garden. The Firebird was sitting in a golden cage. The wolf told Ivan he could take the Firebird, but don't touch the cage. Ivan got the Firebird but thought, "Why not take the cage?" An alarm went off and he was captured by the resident tsar. The tsar told Ivan he could have had the bird for the asking, but now he could be spared only if he would bring the legendary Horse with the Golden Mane.

Ivan returned and apologized to the Grey Wolf. The wolf then brought Ivan to a stable where the Horse with the Golden Mane lived. The wolf told him he could take the horse, but he should not touch the golden bridle. Again Ivan was tempted by the bridle. Bells sounded and he was captured by a second tsar. This tsar told Ivan that he would have given him the horse, but now he would be spared only if he brought the legendary Helen the Beautiful to be his bride.

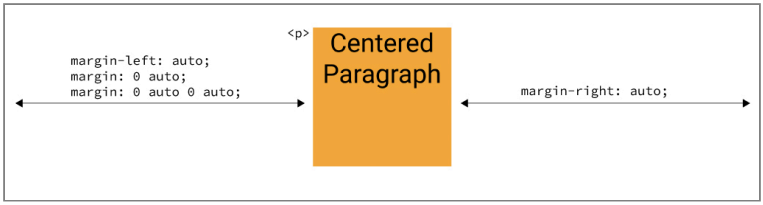
Again Ivan apologized to the wolf, who brought him to Helen's castle. Ivan persuaded Helen to come with them. When they arrived at the second king's, Ivan wept because he and Helen had come to love each other. The wolf turned himself into a princess and exchanged himself for the Horse with the Golden Mane. The wolf later escaped, and the four returned home and lived happily everafter.

Some Useful Styles	
style	**description**
margin: 0 auto;	center an item (tag must be block-level and browser must know width in pixels or %)
margin: 0 12px 6px 0;	margin puts space around objects, numbers go clockwise—top, right, bottom, left
padding: 0 12px 6px 0;	padding adds space from inside the object, whereas margin is space outside the object. E.g., if a tag has background: yellow, padding will expand the yellow color, while margin will not.
text-align: center;	centers text or anything inside <p> paragraph tags
font styles	• font-family: Helvetica; • font-size: 12pt; • font-style: italic; • font-weight: bold; • line-height: 1.2em; • color: dimgray;
max-width: 100%;	scales items to 100% of window width for "responsive design"

color: colorname;
color: rgb(127,127,127);
color: rgba(127,127,127,0.5);
	• specify color by name
	• specify by RGB (red, green, and blue values, 0–255), or
	• specify by RGBA (A is opacity, 0–1.00, i.e., 0 is transparent and 1.00 is opaque.)

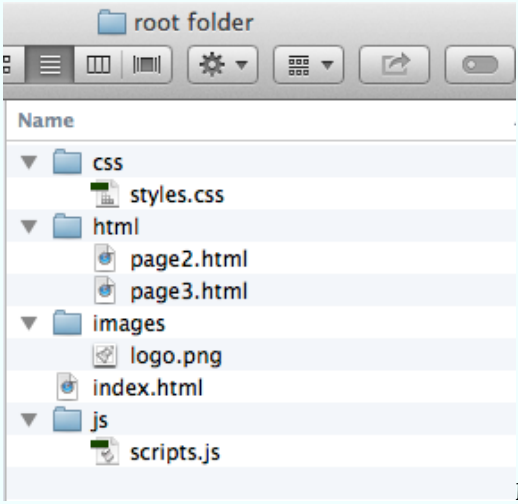
body {
width: 80%;
margin: 0 10% 0 10%;
}
	centers the body and puts at 10% margin around it

How to center a tag. A common question is how to center something in a page or within another tag. The "margin: auto;" style can be used to center a block-level tag, that is, a tag with a built-in paragraph return, and of known width. These include most text tags (p, h1…h6, ul, ol) but not the img tag, which is inline. To center an image with margin: auto, the image tag must first be styled to "display: block."



Block-level tag of known width (256px) centered with the "margin: auto;" style. All three of the styles do the same thing. "Block-level" means that the paragraph tag has an automatic carriage return at the end, whereas an image tag would not and would need to be set to "display: block;" before it could be centered with the auto margin. Alternatively it could be placed inside a paragraph tag set to "text-align: center;."

Organizing Files



1. Make a "gold master" folder on your desktop, such as "site," and place final

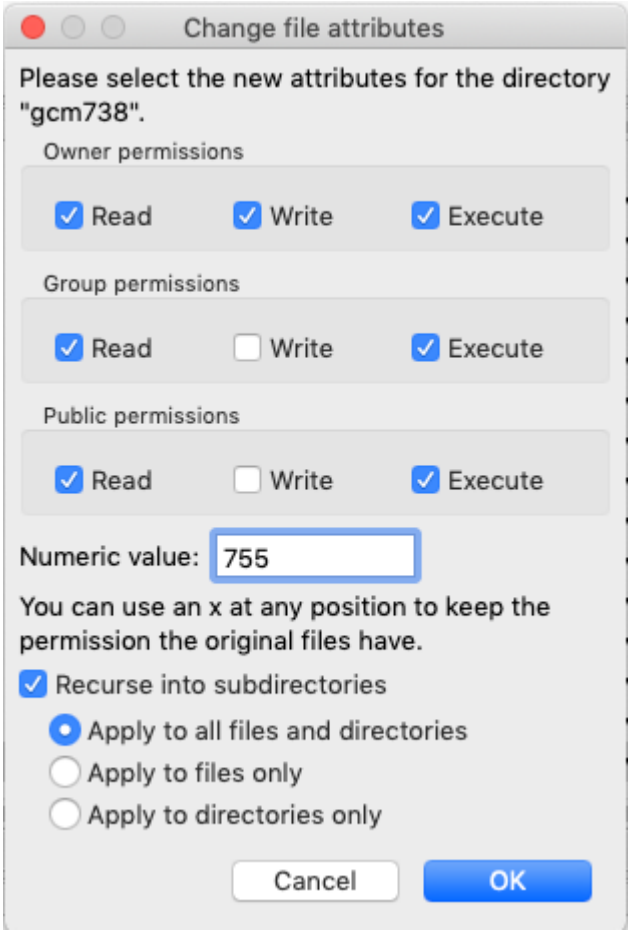
files in it.
2. Place the required .html, .css, and image files for your site into the folder.
3. Follow naming conventions for convenience:

    - Use all lowercase letters.
    - No spaces in file or folder names – use hyphen (-) or underscore (_) to separate words.
    - No absolute links to images, .css, .js, or other files (e.g., "file///MacintoshHD/images/image.jpg") in your documents.

4. Make sure your home page is called "index.html" so the file will load automatically when the folder is entered.
5. If placing image, .css, and other files in folders, be sure to cite the folder name in the link. E.g., "images/image1.jpg"
6. If placing .html files in a folder, remember you will have to go up one directory level to get to the index file and other folders. Use "../" before the folder name.
7. Test the files to make sure all links work and there are no missing images.
8. Don't rename any files after organizing the folder.
9. Upload the files to the STW > public_html folder.
10. Right-click the folder and set permissions to "755" so the public can read and execute future JavaScript but not write to your files.
11. Test your site from a browser to be sure your files can be read from the web.

FileZilla Settings for File Upload to the Server

Host: stw.ryerson.ca
Port: 22
User: your short ID (see my.ryerson.ca > Manage My Online Resources > Update Owner Information)
Password: same as for my.ryerson.ca



Submitting Assignments on the Server to D2L

1. Place your project on the student web server (STW) and record the link.

2. To submit and assignment on D2L, you have to give D2L a file, so you can:

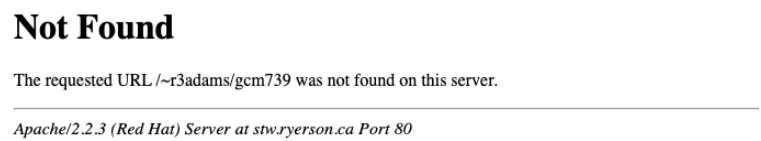
    - Create an HTML file with a link to your STW server, and submit this HTML file to D2L.
      Example: Challenge 1
    - Give D2L a "dummy file" (such as a screen capture of your project), then write the link in the Comments section.
    - If all else fails (not preferred), compress your "site" folder with the project and upload the resulting .zip file to D2L.
    - Please **refrain from emailing** files to the instructors or TAs, as this causes the files to get separated from the D2L assignment and your project will not get marked.

Troubleshooting

Below are three common errors encountered when attempting to view a web page on the server.

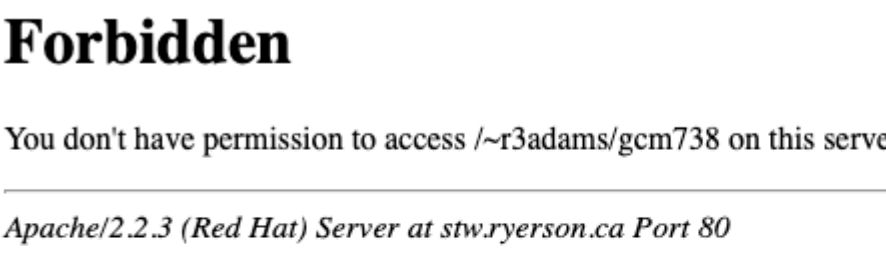
"Not Found." The browser cannot find the page you are looking for. Check for:

- If the page(s) is/are in a folder, include the folder name in URL.
- Make sure the first file is named "index.html" (do not capitalize the "I" in "index').
- Don't put any spaces in any of the file names, as this confuses the server. If necessary, separate words with hyphens or underscores.



"Forbidden." Permissions have not been set to enable the public to see your files.

- In FileZilla, right-click > Set Permissions as shown in the screen capture, i.e., set Public permissions to "Read" and "Execute" (JavaScript).



Set File Permissions to allow the public to view your files and interact with any Javascript th

Missing Photo. The path to the photo has not been properly described in the <img> tag.

- If you put images in a folder, specify the folder name in the image link, e.g., <img src="folder/image.jpg">.
- Be sure photos are not set with absolute links, which will default in Dreamweaver if you haven't saved your HTML document before importing images, e.g., file:///MacHD/Users/richardadams/Desktop/image.jpg. Instead, set image links to be relative, e.g., image.jpg or folder/image.jpg.

Missing photo is indicated by the small blue icon below the title.

The 4 "Ws" of Web Design

A work-in-progress, someone with a sense of humor determined that there are four "Ws" of web design, a reference to "www" or "W3":

- Width—Design for width, let scrolling handle the height.
- Who—We do not know who will visit our site, but we want everyone to feel welcome and be able to use it, even with any sensory challenges they may face.
- What—We don't know what equipment and software our users have, such as the computer platform or device, monitor or screen, operating system, browser, or browser window size. However we try to design for the common scenarios, such as large-screen monitors and cell phones.
- Where—We don't know where our audience comes from, but we would like to keep them on our site as much as possible.

Four "Ws" of Web Design		
"W"	**Description**	**Example(s)**
Width	Design for width, let scrolling handle the height	• Define page width in px, %, or vw (viewport widths)
Who	We do not know who will visit our site, but we want everyone to feel welcome and be able to use it, even with any sensory challenges they may face.	• Design for accessibility • Provide user feedback, e.g., color links differently when mouse is over them and/or when clicked.
What	We don't know what equipment and software our users have, such as the computer platform or device, monitor or screen, operating system, browser, or browser window size. However we try to design for the common scenarios, such as large-screen monitors and cell phones.	• @media device queries to set styles for specific resolutions • Inclusion of multiple fonts in styles • Quality checks on different browsers and platforms
Where	We don't know where our audience comes from, but we would like to keep them on our site as much as possible.	• Google Analytics • Use of new tab for links (target="_blank")

Tutorial 2 • Planning Your Site

This tutorial is about designing your web site, including planning for the user experience, laying out the pages, designing for different devices with responsive design, doing a design critique, and implementing analytics.

User Experience Design (UXD) and "DesignAgility"

Steve Jobs of Apple Computer once said something like, "We must always think about the customers first and develop technology to meet their needs; we cannot start by inventing cool tech and then look for a way to sell it."

Likewise, when designing web pages, designers have to think about the purpose of the site, the audience, and the readers' experience as they use it – a process commonly called "user experience design" (UXD). "DesignAgility," a systematic approach to UXD for media, was created by Okke Schlüter from HdM-Stuttgart and Stefanie Quade (reference: Quade, Stefanie, and Okke Schlüter, *DesignAgility–Toolbox Media Prototyping*. Amazon Kindle Direct Publishing, 2019). DesignAgility combines "design thinking" with "agile manufacturing."

The first step in DesignAgility, "Discovery," is to think of a challenge or problem that people want to solve. For a web site, this could include questions like: What is the site's purpose–to inform, persuade, entertain, or receive a product or service? What is the dominant media–text, photos, videos, animation? Who will the

audience be, and what are their interests and possible sensory challenges they may face in reading and/or listening to the site?

The second step, "Interpretation," means to develop an understanding of how people will meet their challenge by using your site. A valuable step in understanding users is to define several "personas," who form a representative sampling of presumed users of the site, and their characteristics.

The third step, "Ideation," means thinking of solutions to the design challenge. The personas' perception and use of a site can be expressed in "user stories."

Some additional DesignAgility steps are described in the table below.

DesignAgility Step	**What to Do**
Discovery	Define the purpose of your web site: • inform, persuade, entertain, receive product or service? • dominant media—text, photos, videos, animations?
Interpretation	Define several "personas," or presumed users, their ages, o
Ideation	Write "user stories" about why each persona will be interes
Specification	Sketch out a sample page layout for your site: • fixed-width page for text-heavy documents • variable-width page where photos are the most importan
Implementation	Design the site in HTML & CSS.
Evaluation	Design Critique • show your design to another class member and ask for th
Deployment	Make the site public and assess impact • Google Analytics—Enables tracking of users to your site

DesignAgility Example

Discovery and Challenge Identification. Define a challenge or problem that people want to solve. A web designer wants to create an informational site about cat breeds. The site will feature color

photos of each breed, along with information about the breed, such as history, origin, characteristics, temperament, medical conditions, nutritional requirements, food preferences, and grooming needs. The photos and text are equally important. The designer wants the site to appeal to readers "from 6 to 106."

Interpretation and Personas. "Granny Smith" is an 86-year-old grandmother who loves cats and reads everything she can find about them. She visits doctor's offices and complains about the wait times. Her grandchildren bought her a low-cost B&W eReader with WiFi so she could read eBooks and web sites while waiting. Granny has some visual challenges, including presbyopia (needs reading glasses to see up close) and the beginnings of cataracts (cloudy vision).

"Junior Jones" is a 6-year-old boy who likewise developed a strong interest in cats. He has an iMac 21″ computer and an iPad Mini with WiFi that he uses to read web pages and eBooks at home and when traveling. Junior loves to see pictures of cats and could benefit from audio on the web site that could read the text to him.

Ideation and User Stories. On this site the pictures and text are equally important. The photos should look good in color and when rendered in black-and-white. The text should be high-contrast and enlargeable and be readable with a screen reader or included audio. The pages should be designed responsively so they look good on both desktop computers and tablets and cell phones.

Granny Smith likes to look at pictures of cats so the photos should look good in both color and B&W. The photos should be set up as links so that when Granny clicks or taps on an image, the link brings up a larger image. Being able to enlarge the type will make it easier for Granny to read.

Junior likewise enjoys seeing enlarge photos of the cats. As he learns to read, it's helpful for him to tap a button that reads the text to him.

Writing Your Web Site Proposal

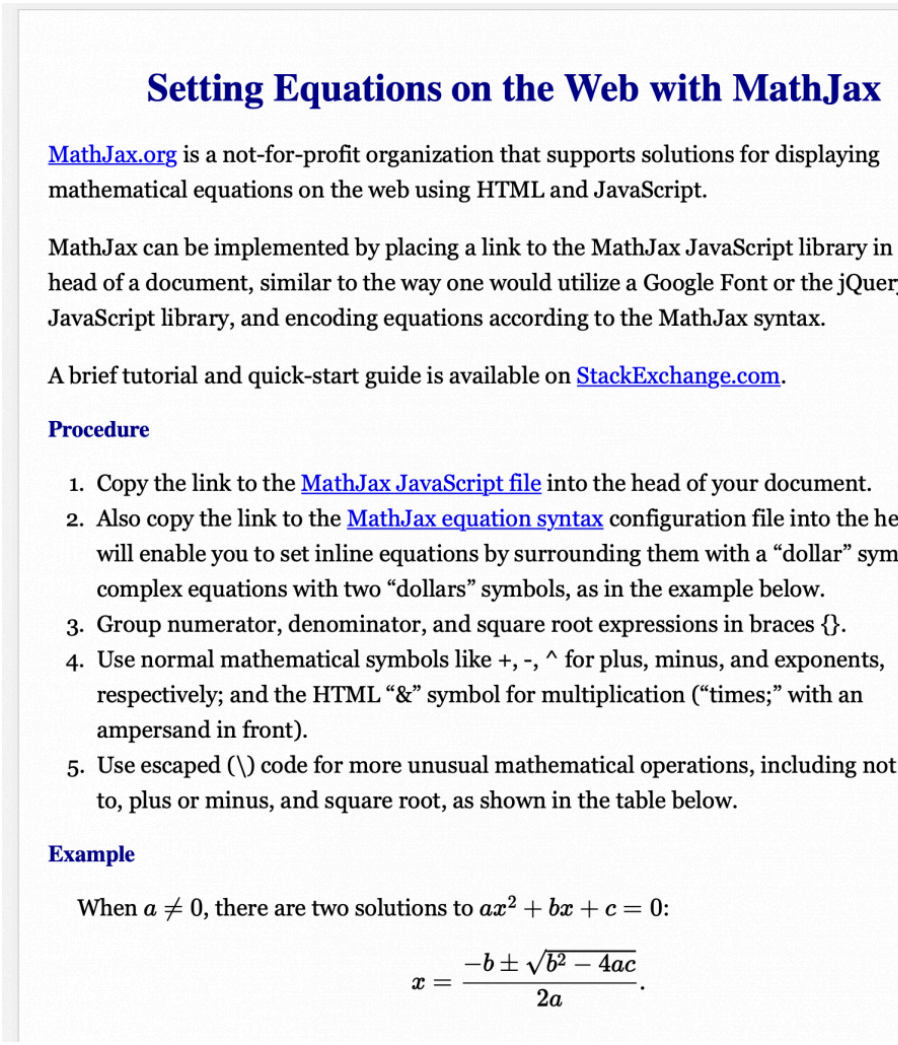
1. Think about a challenge that you could address with a web site. What will be more important—text or photos? Describe how you will build the site.
2. Define some personas of prospective site users and their characteristics.
3. Write a user story for each persona, describing what will bring them to your site and how they will use it.
4. Draw a basic layout of one of your content pages, including (if present) banner, menu, photos, text, and other information. Will the site use a fixed or variable layout?

Page Layout

A web site's purpose and audience are important to know before designing pages. As discussed in Tutorial 1, page layout for the web is different than it is for print because the designer never knows what type of computer, device, operating system, monitor, screen, browser, or window size the readers will use. Regardless, we want everyone to feel included and be able to use the site.

Web pages can be designed with fixed or variable sizes. Fixed size is easiest for text-heavy pages, such as news stories, while variable sizes are good when photos are the dominant media.

Below are examples of fixed-width and variable-width pages.



Setting Equations on the Web with MathJax

MathJax.org is a not-for-profit organization that supports solutions for displaying mathematical equations on the web using HTML and JavaScript.

MathJax can be implemented by placing a link to the MathJax JavaScript library in head of a document, similar to the way one would utilize a Google Font or the jQuer JavaScript library, and encoding equations according to the MathJax syntax.

A brief tutorial and quick-start guide is available on StackExchange.com.

Procedure

1. Copy the link to the MathJax JavaScript file into the head of your document.
2. Also copy the link to the MathJax equation syntax configuration file into the he will enable you to set inline equations by surrounding them with a "dollar" sym complex equations with two "dollars" symbols, as in the example below.
3. Group numerator, denominator, and square root expressions in braces {}.
4. Use normal mathematical symbols like +, -, ^ for plus, minus, and exponents, respectively; and the HTML "&" symbol for multiplication ("times;" with an ampersand in front).
5. Use escaped (\) code for more unusual mathematical operations, including not to, plus or minus, and square root, as shown in the table below.

Example

When $a \neq 0$, there are two solutions to $ax^2 + bx + c = 0$:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

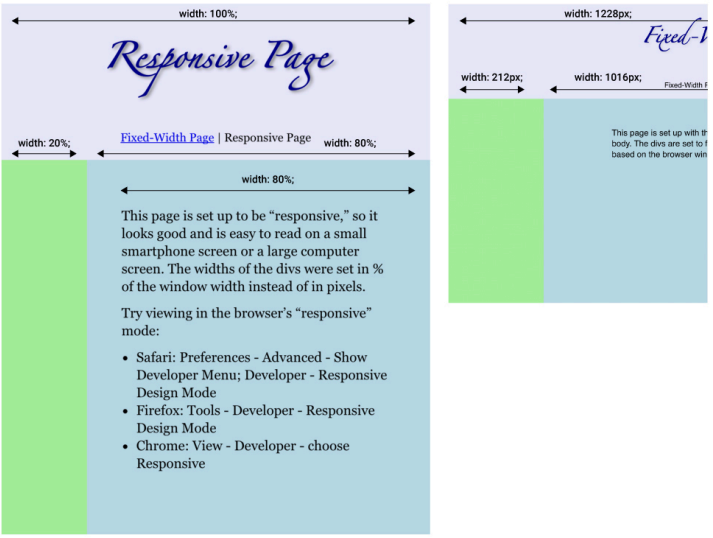
Example of a fixed-width page for a text-heavy document, designed to look like a U.S. Letter The "page" centers itself in the browser window and remains the same size. Sizes were defin (pixels).

Responsive Design

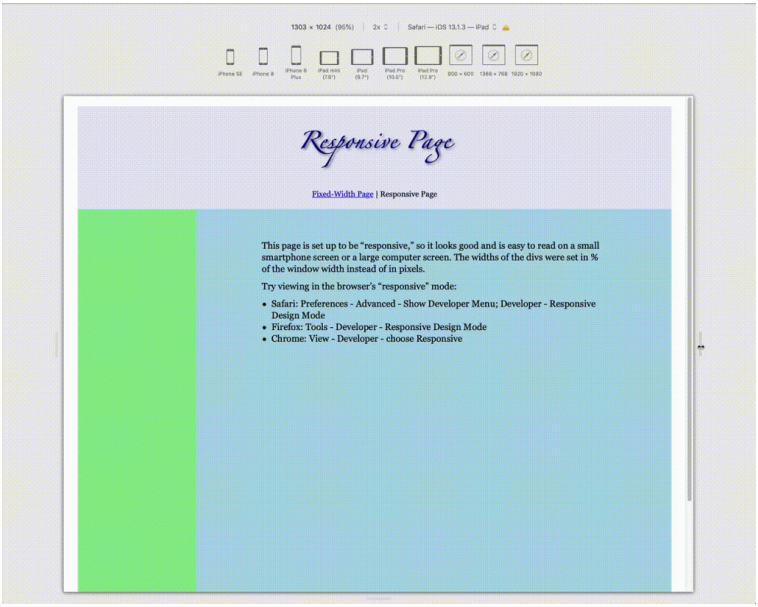
Does your web page need to be readable and look good on a large-screen monitor, cell phone, and iPad or tablet? If so then it should be designed to be "responsive," or adaptable to different screen sizes.

Styles for responsive design include "width" set in percent instead of pixels, and limiting the width to that of the browser window by using the "max-width" style in percent.

In the two examples below, the widths of the page elements in the responsive page are set as 100%, 20%, and 80%, respectively; while in the fixed-width page they are set in pixels (px).



Examples of responsive and non-responsive pages. The responsive page expands and contracts to fit the browser window, while the non-responsive page remains the same size regardless of the browser window size.



This brief video shows responsive design in action. (Click to play.)

Further refinements to responsive design can be made using the media query rule (@media), in which the resolution of the applicable styles is defined. Example: @media only screen and (max-width: 750px) {}. Inside the braces would go the styles applicable to screen resolutions of 750px or less. Details in Tutorial 4.

Design Critique

This section is based on a design critique contributed to class by Prof. Bettina Tabel, HdM-Stuttgart. When designing a web page, it can be useful to get a design critique from an objective third party. Following is a list of questions to consider:

Home Page

1. What is the purpose of the site? Audience? Is this clear from the title and design?
2. From the title and main photo (if any), are the subject and theme of the site clear?
3. Does the site's color scheme fit the target group and the subject? Should the color scheme be neutral, pastel, bright, complementary, analogous, or other theme?
4. Is the font selection appropriate to the site's subject and audience?
5. Are the fonts legible (size, contrast)? Could they benefit from a drop shadow, complimentary-colored or lighter or darker background?
6. If the site has one or more photos, have the image rights been clarified?
7. Are photos sized appropriately to the pages and for viewing on different devices?

Other Pages

1. Do special effects (drop caps, callouts, sidebars) contribute to the theme of the site without being distracting?
2. Are photos large enough for readers to see detail? If the reader clicks on or touches a photo, does it link to a larger version that can be viewed in more detail?
3. Are sections (if present) separated clearly?

Original Site


Home
This site is a technol
boxes and the stories
Russian lacquer art
in pre-1900s imperia
Common subjects to
Box Theme

Sample Critique • Questions

1. Why are the top and bottom borders not the same height?
2. Why are the links not vertically aligned in the menu bar
3. Why are the links set in all caps?
4. Why is a serif font used for the links?
5. Why are the line lengths so long?
6. Why was a script cap initial used when it's difficult to read?

Common Design Flaws—Don't Do It!

The graphic below shows several examples of common web design flaws, which are listed below and corrected in the second graphic.

Poorly designed page showing common design flaws.

1. Itty-bitty images—can't see! Make 'em bigger because people want to see!
2. Excessively large image files (>1 MB) reduced in width in CSS. They take too long to download. Size appropriately in

Photoshop or Preview.

3. Use of bold paragraph <p> for headings. Instead use heading tags <h1>, <h2>, ..., <h6>.
4. Use of underline for headings, reminiscent of typewriter days. Use a heading tag and a complimentary font.
5. Use of default Times Roman font (boring!).
6. Itty-bitty font size (can't read!). Make it 14pt or bigger and 1.4em line leading (line-height).
7. Insufficient vertical space between elements. Use padding and/or margin to adjust vertical height, not carriage returns
.
8. Type lines too wide (can't read!), set paragraph width to be 70 characters or less and center on the page.

Redesigned page: <h1> tag with Zapfino font; text 16pt Georgia with 1.5em line height, paragraph width 768px and centered on page; image width 512px and float left and with 12px right margin and 6px bottom margin to separate it from text.

Google Analytics

"Get to know your customers" is the theme of Google Analytics. This site provides a code that you can place in your web site to get information about customers – where are they from, how often do they visit your site, how long do they stay, and what pages do they visit?

1. Log in to google.com/analytics
2. Create a login with a non-Ryerson email. (Ryerson already uses Google Analytics so you won't be able to access the service using an @ryerson.ca email.)
3. You will receive an 8-line code snippet to place in the head of your web pages. The code snippet will capture information about visitors to your site, such as their location, browser, and platform, and report this in a page on the Google Analytics site.

Tutorial 3 • Styles, Fonts, and Tables

CSS Styles

It's commonly said that HTML gives structure to your page, while cascading style sheets (CSS) determines its appearance. CSS has two meanings: (1) a file of type .css, and (2) the hierarchy of styles, i.e., that a local/inline style overrides a global/embedded style, which in turn overrides an external .css file.

Three ways of writing styles:

Local or inline style. A style="" statement inside a tag, e.g., <p style="font-family: Roboto;">. Local/inline styles override global/embedded and CSS files.

Global or embedded style. A <style> tag with statements in the document head. Global/embedded styles override CSS files.

CSS file. A separate file of type .css that can be applied to multiple pages, e.g. file "styles.css" attached to a page, <link href="styles.css" type="text/css" rel="stylesheet">

Making a CSS File

1. Copy the global/embedded styles from one of your pages into a separate file, without the <style> tags, and save with file extension .css, e.g., "styles.css."

2. Link your .css file to both pages using the <link> tag:

   ```
   <link href="styles.css" type="text/css" rel="stylesheet">
   ```

3. Verify that the styles take effect and that the .css file appears as a separate Dreamweaver tab in your document.

Using Selectors

Selectors are like the strings on a marionette—they connect the tags with the styles you want to apply. You can select all tags by writing the tag in a style statement and the styles in braces, e.g., p {font-family: Georgia;} selects all paragraphs and sets them in the font Georgia. But what if you want to style different paragraphs differently? In that case you can label the paragraph with an ID (use once on a page) or with a class (used multiple times). Select the ID in CSS using a hash tag # and a class by preceding it with a dot.

Pseudoclasses are a type of class that's built in to the HTML5 and does not need to be labeled by the designer. Example: p:nth-child(1) selects the first paragraph.

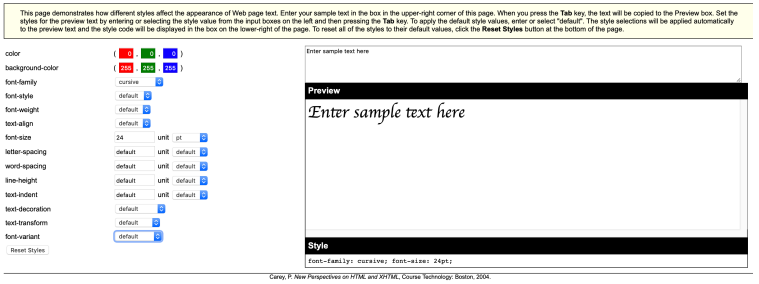
Another way to select tags for styling is to use the structure of the HTML. Example: you used an unordered list of bullet points to make a navigation menu, and you placed the list within a navigation or <nav> tag. You can select only that list by writing "nav ul." Other lists on the page will not be affected.

A great exercise on selectors is the "CSS Diner" at http://flukeout.github.io. To select one or more items out of several:

Methods of Selecting Tags for Styling		
Selection method	*Example in HTML*	*Example in CSS*
Use an ID (for one item)	<p id="para1">	#para1 {}
Use a class (for multiple items)	<p class="para1">	.para1 {}
Use a pseudoselector (for first, second, or a repeating pattern of connected items, known as "children")	– (built into HTML5)	p:nth-child(1) 1, 2, 3, ..., odd, even 1n+3 (every 3 starting with 1st)
Use the HTML structure, e.g., with an <img> tag in it	<img>	div img {}

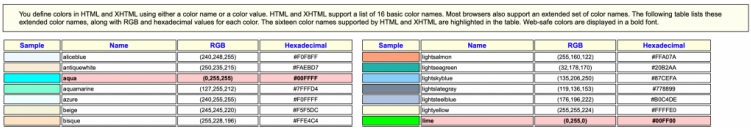
What to Style

Two demo files from Patrick Carey's textbook, *New Perspectives on HTML* (Thompson) show examples of how you can style text and color. (A third shows the web-safe palette.)



demo_text_styles. Notice all the characteristics of type that you can style. Some common ones include:

- font-family–the typeface
- font-weight–bold
- font-style–italic



You define colors in HTML and XHTML using either a color name or a color value. HTML and XHTML support a list of 16 basic color names. Most browsers also support an extended set of color names. The following table lists these extended color names, along with RGB and hexadecimal values for each color. The sixteen color names supported by HTML and XHTML are highlighted in the table. Web-safe colors are displayed in a bold font.

Sample	Name	RGB	Hexadecimal
	aliceblue	(240,248,255)	#F0F8FF
	antiquewhite	(250,235,215)	#FAEBD7
	aqua	**(0,255,255)**	**#00FFFF**
	aquamarine	(127,255,212)	#7FFFD4
	azure	(240,255,255)	#F0FFFF
	beige	(245,245,220)	#F5F5DC
	bisque	(255,228,196)	#FFE4C4

Sample	Name	RGB	Hexadecimal
	lightsalmon	(255,160,122)	#FFA07A
	lightseagreen	(32,178,170)	#20B2AA
	lightskyblue	(135,206,250)	#87CEFA
	lightslategray	(119,136,153)	#778899
	lightsteelblue	(176,196,222)	#B0C4DE
	lightyellow	(255,255,224)	#FFFFE0
	lime	**(0,255,0)**	**#00FF00**

demo_color_names. A palette of the 140 named colors and four ways to specify them:

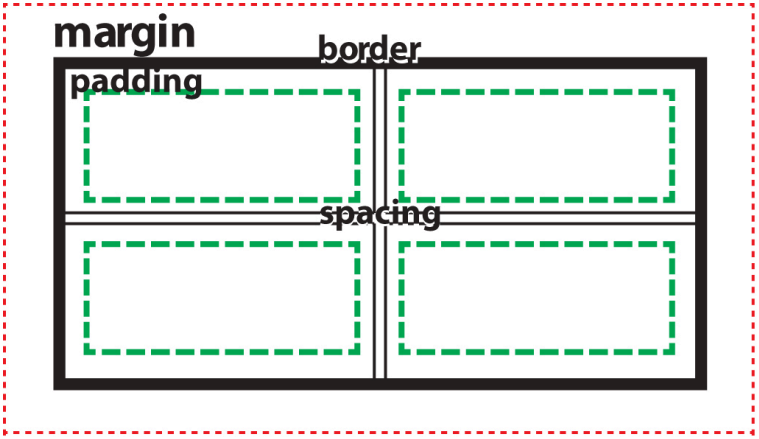
- named colors—red, green, blue, lightblue, azure, chartreuse, … 102 names
- rgb—red, green, and blue values, scale 0–255, e.g., color: rgb(255,0,0);
- rgba—"a" adds the "alpha channel," or transparency, from 0-1.00, where 0 is completely invisible and 1.00 is opaque.
- hex—3 or 6 numbers and letters on a scale of 0-16, where 10=A, 11=B, etc. Difficult to comprehend if you are trying to figure out what color it is.

What to color

- color: red; —will color text
- background: red; —colors objects, like "fill" in Illustrator or Photoshop

Space

- margin: all around; margin: top bottom; margin: top right bottom left—determines space around an object; margin: 0 auto will center an object if its width is specified and it's a block-level element.
- padding: expands an object (including background color)



Demonstration showing the difference between margin, padding, border, and table cell spacing.

Size

- width
- height

Working with Fonts

Two ways to include an unusual or decorative font on a web page are to place the font on the server and access it using the @font-face rule, and use a Google or other free font.

Including a Font on Your Server

If you want to use an unusual font and you are unsure if users will have this font on their computers, you can put the font file on your server and link to it using the "@font-face" rule. In this example, the font candara.ttf was placed on the server in the same folder as the styles.css file and linked similar to a background image.

```
@font-face {
    font-family: Candara;
    src: url(candara.ttf);
```

Using a Google Font

Google has numerous free fonts available on its web site, www.google.com/fonts. The font information can be linked to your web page using the <link> rule. To use a Google font:

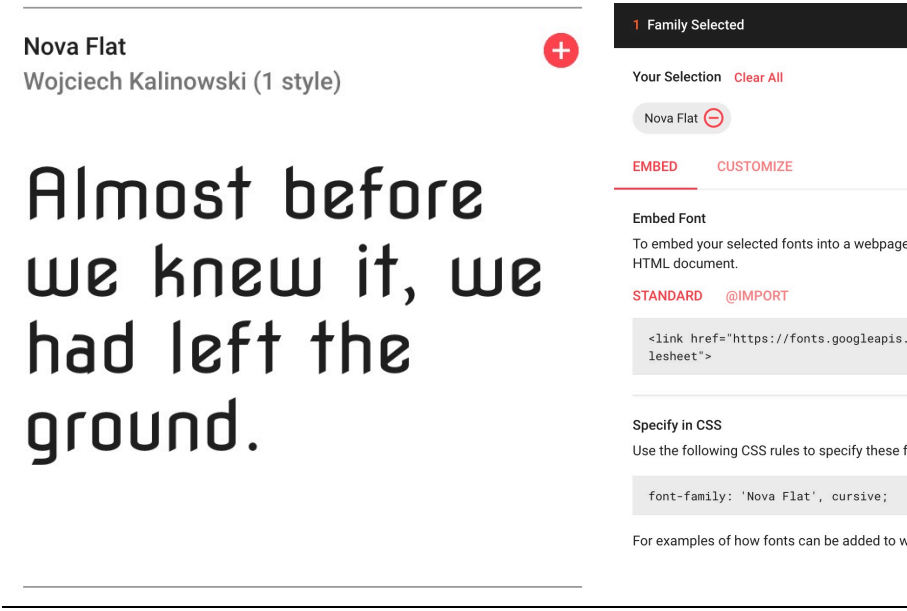
1. Browse the available fonts and add one you want to your collection.
2. Access your font collection to get the link to the font and the syntax to cite the font.
3. In this example, the font "Nova Flat" was accessed:

   Embed code:

   ```
   <link href="https://fonts.googleapis.com/css?family=Nova
   ```

4. To use the font:

   ```
   font-family: 'Nova Flat', cursive;
   ```



Tables

Tables are a nice way to organize information that's easy for readers to comprehend. Basic table tags are table <table>, table rows <tr>, and table data <td> which refers to the cells and therefore the columns. (There is no tag for table columns, they are determined by the cells.)

Various styles can be applied to the table, tr, and td tags including the border, color (of font), and background, including an image background. "Pseudoselectors" including nth-child(odd) can be used to shade alternating rows to differentiate them more imaginatively than with the traditional borders.

Anatomy of a Table

<table>	<td>	<td>
<tr>	Row 1, Column 1	Row 1, Column 2
<tr>	Row 2, Column 1	Row 2, Column 2

Sample table (above) and the tags for the table <table>, rows <tr>, and columns (cells, <td>). Columns are determined by the cells. Code below.

```
<table>
    <tr>
        <td>Row 1, Column 1</td><td>Row 1, Column 2</td>
    </tr>
    <tr>
        <td>Row 2, Column 1</td><td>Row 2, Column 2</td>
    </tr>
</table>
```

Table Tags	
Tag	**Description**
<table></table>	table
<tr></tr>	table row merge–number of columns to span
<td></td>	table data–columns • there is no "column" tag because the columns are determined by the cells
<th></th>	table header • first row(s), type will be bold centered

CSS Styles Useful to Table Tags	
Style	*Description*
border	px color, e.g., border: 1px solid gray;
padding	text inset, e.g., padding: 12px;
spacing	space between cells
background: color; background-image: url(image.jpg);	color of table, rows, and/or cells; could include background image; • color: rgb(0,0,0); • color: #fff; • background: (url "image.jpg");
pseudo-selectors :nth-child(1) :nth-child(odd) :nth-child(even) :first-child :last-child :nth-child(3n)	first item in a group of multiple all odd-numbered items all even-numbered items first item last item every 3rd item

Importing Table Text

1. Import the table text (table.txt) into Dreamweaver: File >

Import > Tabular Data.

2. Style the table by:
    - inserting an embedded or global <style> tag in the <head>
    - selecting the table elements (<table>, etc., see Table 1 below)
    - applying the specifications you want (Table 2 below)

Russian Fairytale Boxes		
Box	**Characters**	**English Counterpart**
The Grey Wolf	Ivan Tsarevitch, Helen the Beautiful, the Grey Wolf	none
The Horse with the Golden Mane	Ivan Tsarevitch and the Horse with the Golden Mane	The Magic Pony
The Firebird	Ivan Tsarevitch and the Firebird	The Phoenix
Father Frost	Father Frost, the Snow Queen, and a troika of horses	Santa Clause

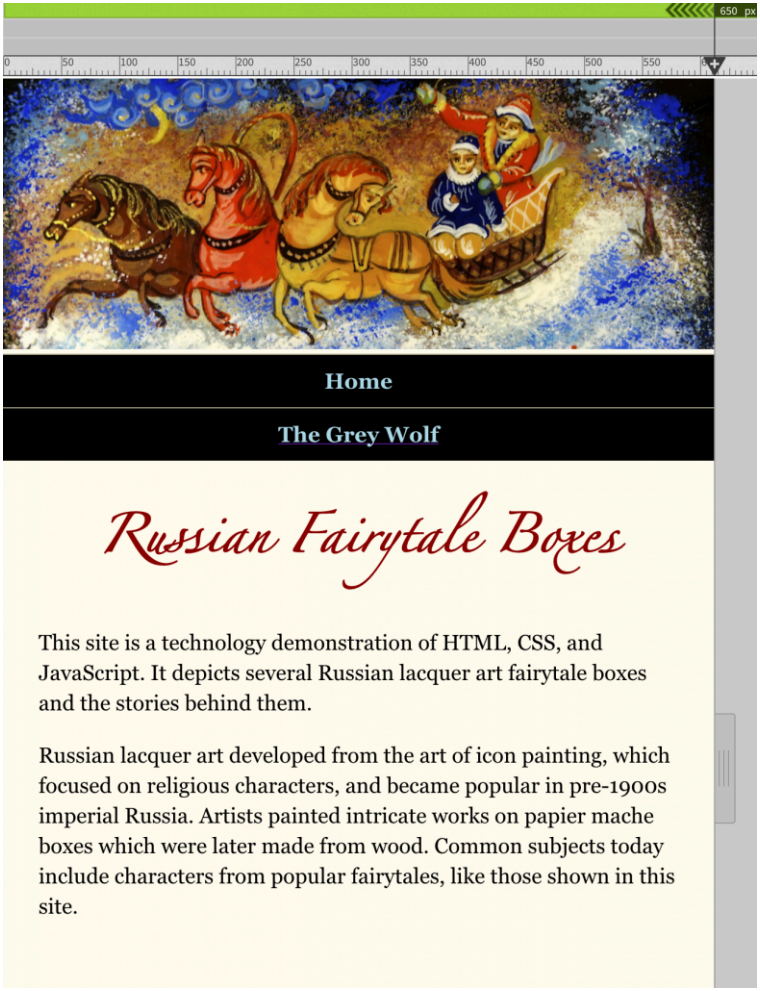
Tutorial 4 • Navigation Menus and Responsive Design

Menu designed for screen resolutions 750px and above.

In this tutorial we will turn an ordinary-looking menu into a menu bar like the one shown. The menu bar is made using the unordered list <ul> tag, where each item is a "bullet point," or list item <li>, with the "bullet" removed. The list items are styled to "float: left," which turns them from block to inline. (Block items act as if they had paragraph returns at the end; inline items, like spaces between.) Other settings include "color" for type, "background: color," "border-bottom," "line-height," "padding," and "width." The width is set so that all the list items fit across the page (i.e., 2 items occupy 50%, 3 would be 33%, etc.).

Making the Menu Bar

1. Open the file “index_nav_START.html. Save As each file remove the “_START” from the name.
2. Make the Menu into a Menu Bar:
    - Remove the bullet point from between “Home” and “The Grey Wolf,” remove the paragraph tag <p>, place the items into an unordered list <ul>, and make each item into a list item <li>.
    - Style the ul to be 1004px wide and also have max-width: 100%;.
    - Style the li to remove the bullet points (list-style: none;), arrange horizontally (float: left;), and set width to 50%. Add padding or set height to make the menu items into rectangles.
3. Add a CSS <style> selector, li:hover, that applies when the mouse is over the item. Style the list items different so users will know they put the mouse in the right place to click a link. E.g., set a different color background, (font) color, drop shadow, and/or other style.



Menu designed for screen resolutions 650px or less.

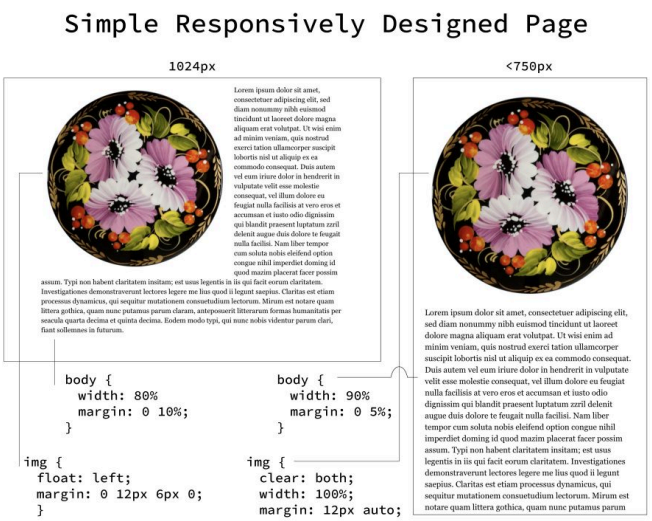
Adding Responsive Design

Increasingly viewers are using smartphones to look at web pages. The goal of responsive design is to have a web page "respond"

to different size viewing screens by changing styles to promote legibility. Responsive design can be implemented with media queries. "Responsive" could also be interpreted as improving accessibility for readers with visual challenges.

Media queries (@media) make it possible to style something, like a menu, differently on a smartphone than on a large screen.

When designing for cell phones, keep in mind that people want to see the images as large as possible (especially if they have a lot of detail) and want to read the text easily. Image the size of postage stamps and text that runs off the page and requires horizontal scrolling will frustrate readers.

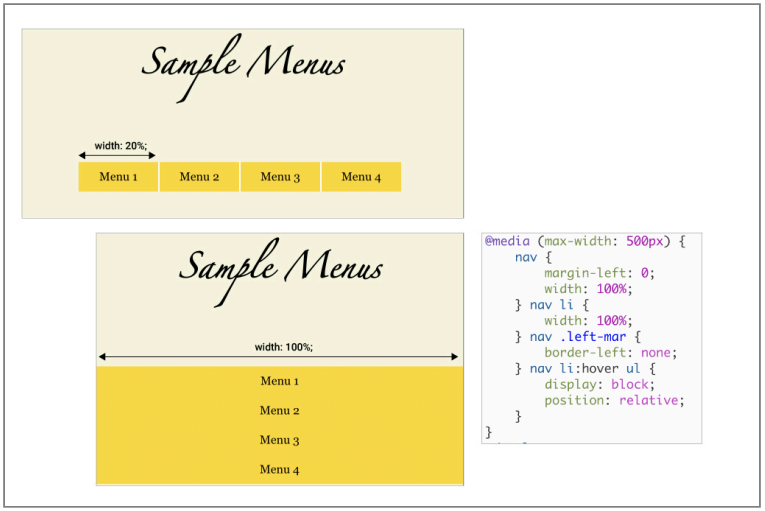


Simple responsively designed page, shown on a 1024px-wide browser window (left) and a cell phone screen of less than 750px (right). For the large screen, the body is centered in the page with 10% margin on each side, and the image is styled with text wrap ("float: left;"). On the cell phone screen, the body is widened to 90% of the screen with a 5% margin, type size and leading are slightly enlarge for legibility, the text wrap is removed from the image with the style "clear: both;" and the image is set to 100% of the body width.

If you have a menu of several items across your page, it may be easy to see and select on a 21-in. monitor but difficult on a 4-in. phone screen.

iPhone Screen Resolutions

Model	Screen Size	Resolution (px)
iPhone 5, 5s, SE	4 in. diag.	640
iPhone 6, 6s, 7, 8	4.7	750
iPhone X	5.8	1125
iPhone 6Plus, 7Plus, 8Plus	5.5	1282



Menus with responsive design, styled to appear horizontally (top screen capture) on large-screen monitors and vertically (bottom) on smartphones. The @media query code makes the menu items stack vertically on screens with less than 500px resolution.

To make menus easier to see, web designers typically use a media query to structure menus vertically on the page for smartphones.

Add responsive design to your menu

1. Add a media query to your embedded/global <style> or external .css:
   @media screen and (max-width: 750px) {}
   This query will apply to screens that are 750px or less in width.
2. Style the nav ul selector:
   - max-width: 100%;
   - Style the nav li selector:
     - width: 100%;
     - float: none;

Checklist for Responsive Design

? menu and buttons

? menu position, e.g., horizontal on large screen, vertical on small screen

? :hover – change style when mouse is over tag

? :active – change style when tag is clicked

? page layout

? make images occupy the full screen width by changing the image display from text wrap (float) to block

? make the text occupy 95% of the screen width

? change color (color, background)

? change contrast

? change font

? size (font-size)

? leading (line-height)

? bold

? contrast

? black/white vs. white/black

? alt tag – use to describe images

Viewing Your Responsive Design

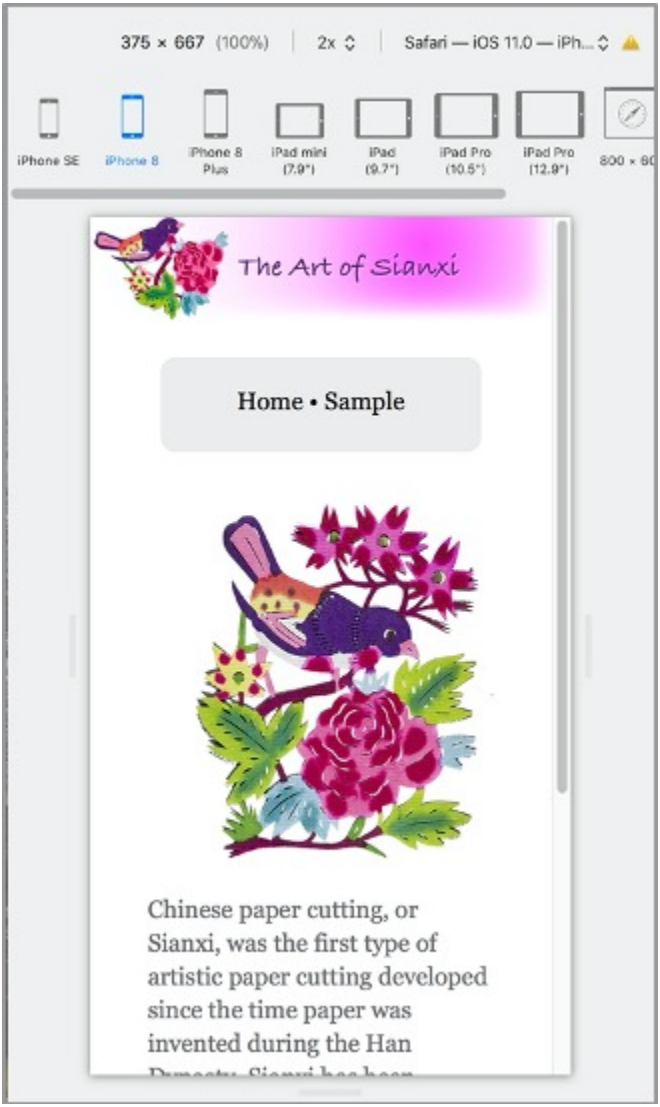
Most popular browsers have a "responsive design mode" for viewing your page as it would appear on various smartphones and tablets.

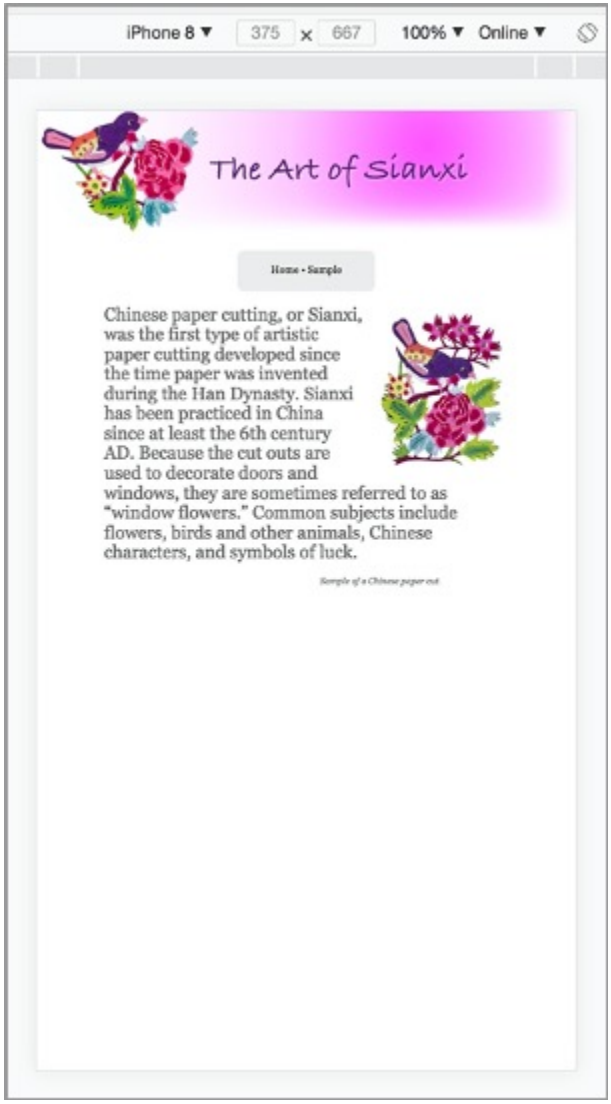
Safari

- Preferences > Advanced > check Show Developer Menu
- Developer > Enter Responsive Design Mode > select device

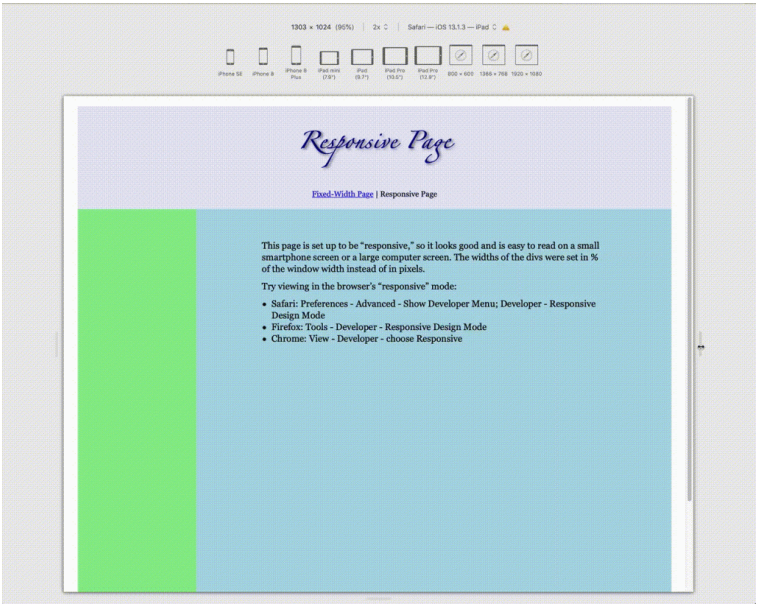
Chrome

- View > Developer > Developer Tools > click on Responsive Design button


375 × 667 (100%)
2x
Safari — iOS 11.0 — iPh...
iPhone SE
iPhone 8
iPhone 8 Plus
iPad mini (7.9")
iPad (9.7")
iPad Pro (10.5")
iPad Pro (12.9")
The Art of Sianxi
Home • Sample
Chinese paper cutting, or Sianxi, was the first type of artistic paper cutting developed since the time paper was invented during the Han



Page viewed in Safari (left) and Chrome (right) in responsive design mode with iPhone 8.



This animated GIF shows how a responsively designed page works. (Click to play.)

Accessibility

Accessibility and responsive design are interrelated because their goal is to promote universal access as much as possible to a web site. Accessibility refers to readers with visual or tactile challenges that may make it difficult to read or interact with a site. For example, visually-impaired readers may find it difficult to read small or low-contrast type, while those with tactile impairments may find it difficult to click the mouse on a small link. Accessibility and responsive design are interrelated.

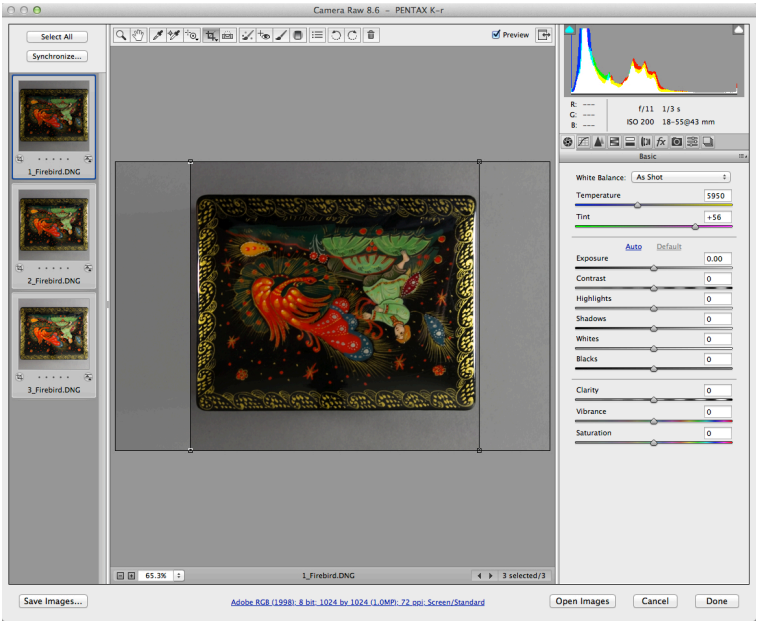
In addition to the responsive design checklist on page 1, web designers can include image information in the form of "alt" tags that display when an image does not load, and in "title" tags that appear as popups.

Tutorial 5 • Image Optimization

Bitmapped images are use-specific, meaning the resolution has to be set for the intended use and reproduction size, so as to avoid "the jaggies," or pixelization, and also to avoid excessively large file size, which slows output for print and download time on the web.

A good place to start thinking about web images is whether the image should occupy a full page, half page, or quarter page. Resolution for web should be 72 ppi. Higher resolution does not produce finer detail unless the image is to be enlarged, e.g., if a smaller image is linked to a larger view.

Transparency is often useful for web images. GIF and PNG file types support transparency, but not JPEG.



Photoshop's Camera Raw open dialog box.

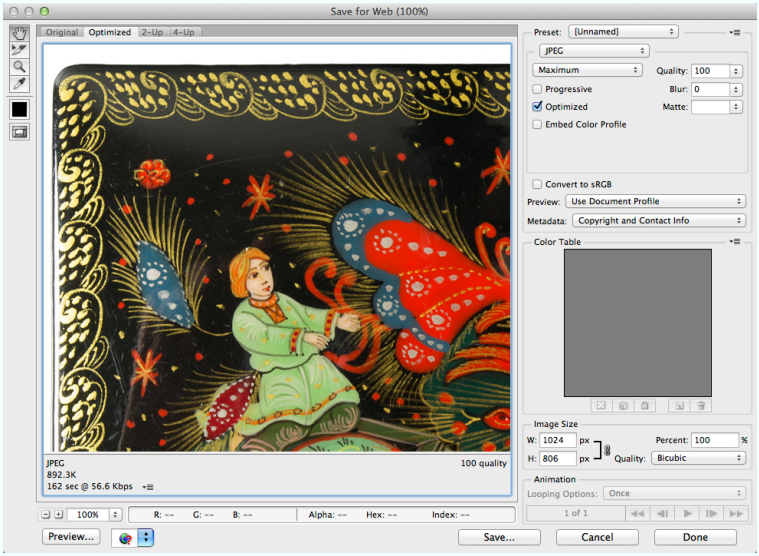
Let's Optimize Some Images

DNG Images

1. In Photoshop, simultaneously open the three .DNG "digital negative" images, which will open as Camera Raw.
2. Note that the images are a bracketed exposure series of ±⅓ exposure value (EV). When making an exposure, bracketing

only takes a few extra seconds and enables you to compare exposures later on a large screen.

3. Note that the three images can be identically cropped if they are all selected when the cropped area is drawn.
4. Final image size and resolution can be set in the blue link below the preview.
5. Select all 3 images, flip 180°, and crop to a square 1024×1024 px at 72 ppi, and open.
6. Choose File > Save for Web > JPEG. Note that the dialog box includes settings for quality and size, and also displays the final file size and expected download time at different network speeds.



Photoshop's Save for Web dialog box.

PNG Image

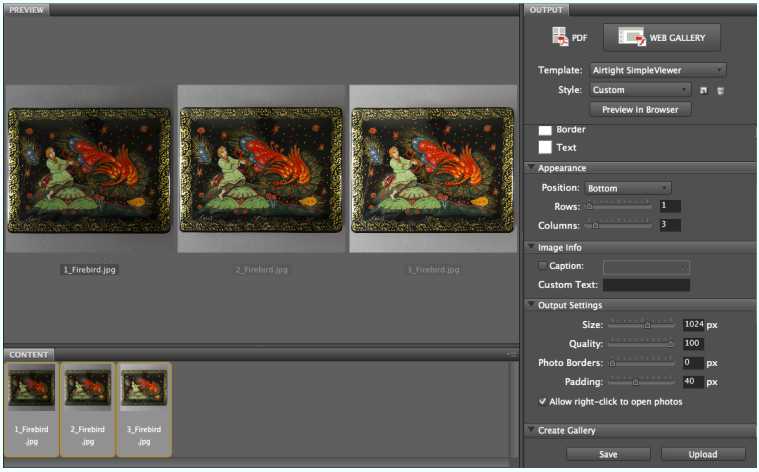
1. Open the image "4_BlackFlowers.jpg."
2. With the Circular Selection Tool, hold SHIFT+OPT and draw a circle around the box, starting from the centre. Reposition if necessary.
3. In the selection, right-click or CNTRL-click "New Layer from Cut." This places the box on a new layer, leaving the Background empty.
4. In the Layers palette (Window > Layers), delete the Background layer.
5. File > Save for Web, select PNG-24 format, and check Transparency. The background should show as a checkered pattern.

Place in Web Page

1. Place both images into a web page styled with body {background: gray;}.
2. Note that the background of the PNG image is transparent while that of the JPG is not.

Image Gallery

1. Place the 3 bracketed "firebird" JPEG images into a separate folder and open in Adobe Bridge.
2. Make an image gallery using the Output tab.
3. Preview in browser.



Bridge's Image Gallery dialog box.

Tutorial 6 • MailChimp for Email Campaigns

Email Campaigns

Thanks to Mark Corrigan (GCM '13) for contributing this tutorial. MailChimp and similar programs enable the creation of email campaigns ("blasts"). Functions include creating:

- HTML/CSS-formatted emails with text and images
- mailing lists of people to whom to send the emails
- sign-up forms for web sites that enable readers to automatically sign up to receive email updates

1. Create a Free Account

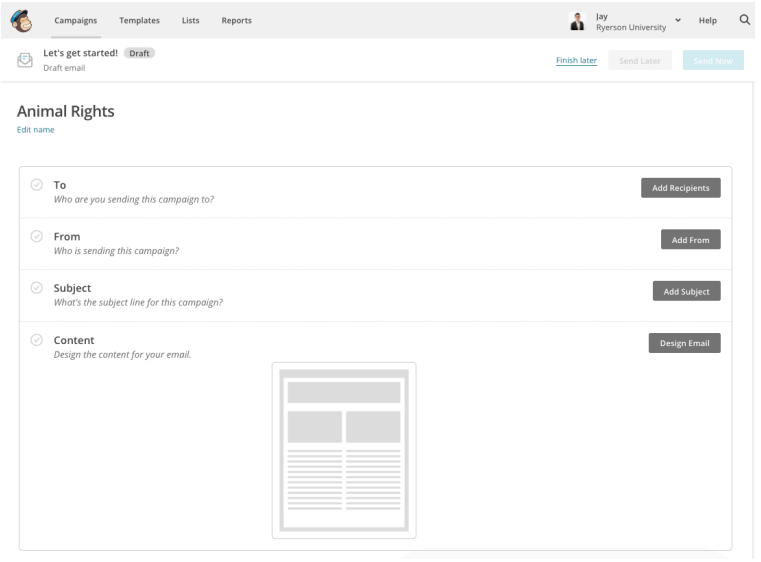
Visit https://login.mailchimp.com/signup/ to create your free account using your Ryerson email address.

2. Create a Mailing List

Once you have created your account, you will need to create a mailing list of recipients who will receive your emails. To start, add your personal email address.

3. Create a Campaign

Once you've created a recipient list, select Campaign and Create Campaign. You will need to Create an Email and name your campaign. Additional info that is required includes the subject line & from fields.



4. Customize your Email

To start customizing your email you must select a Layout or Theme. You will find that MailChimp's interface is very user-friendly and intuitive. You can view MailChimp's design guide by clicking here: https://mailchimp.com/email-design-guide/

The layout of your email can be customized using MailChimp's drag-and-drop features. You can drop content for text, dividers, images and more. Content can be edited by clicking on the pencil while hovering over your email preview.

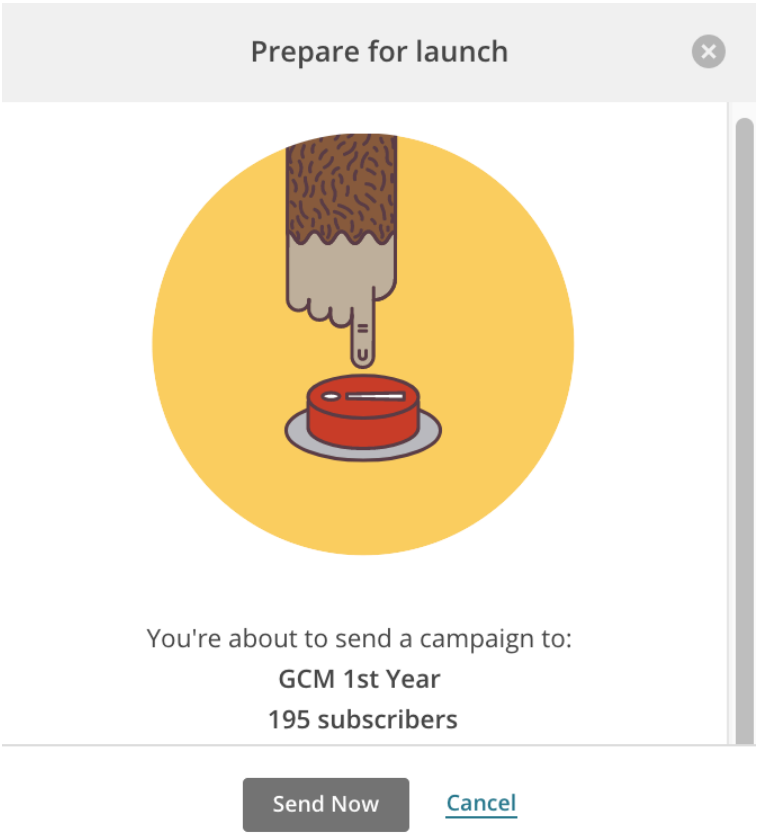
With your newly learned HTML skills, you can "crack into the code" using the Source button  on the toolbar. Try adding an HTML table to your email.

5. Preview & Test your Email

Be sure to utilize the Preview & Test tab in the top menu to ensure your email is looking as desired. You can even send a test email to one of your accounts to simulate what it would look like in the inbox of your receiver.

6. Send your Email

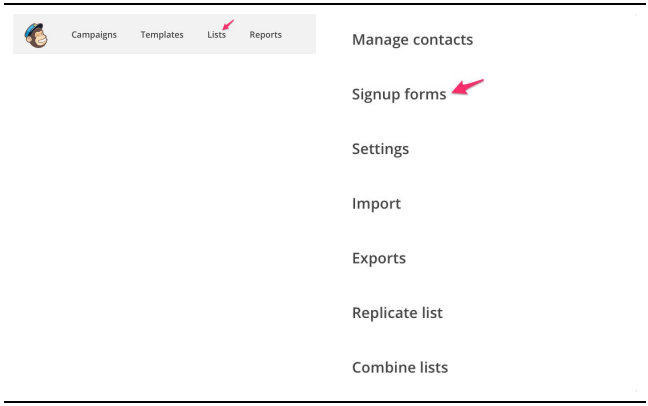
Once you are 100% confident in your email, it's ready to send! Simply select Next, confirm the settings and you can Send the email to all recipients on your list. Be sure to triple check your content as there is no turning back once your email has been sent!



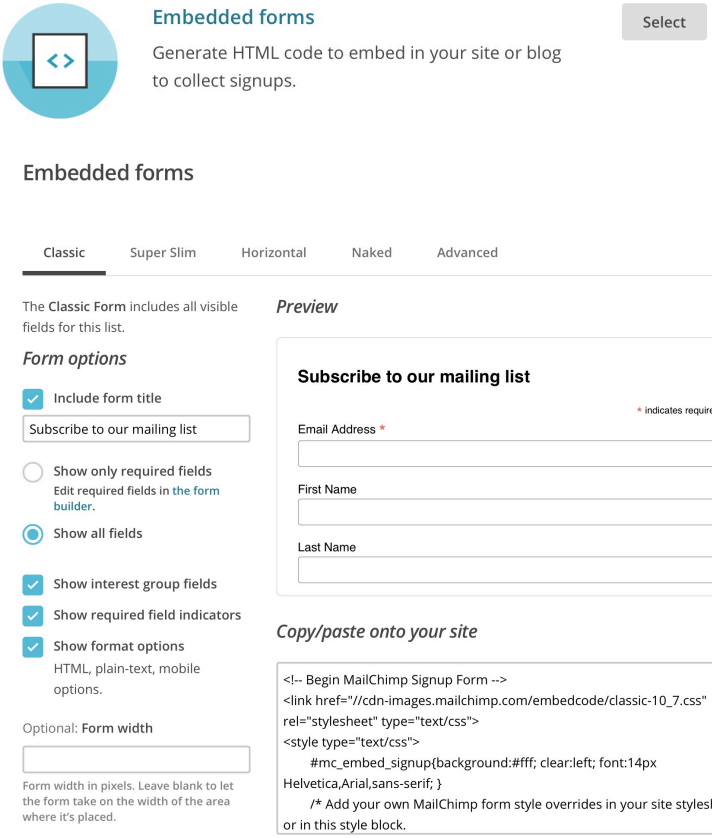
7. Use MailChimp to Make a Signup Form

With MailChimp you can create a form to place in your site where readers can sign up to receive your emails. The MailChimp generates the HTML and CSS for the form and includes the interactive elements to automatically add subscribers to your mailing lists.

1. Select a campaign and then click on “Lists” from the MailChimp menu.



1. Click the down arrow next to “Stats” and select “Signup forms.”
2. Select “Embedded forms.”
3. There are many options for customizing the information in the form and its design. Once you have selected all options, highlight the code that says “Copy and paste into your site,” and place in Dreamweaver or other coding program.


Embedded forms
Generate HTML code to embed in your site or blog to collect signups.
Select
Embedded forms
Classic
Super Slim
Horizontal
Naked
Advanced
The Classic Form includes all visible fields for this list.
Form options
Include form title
Subscribe to our mailing list
Show only required fields
Edit required fields in the form builder.
Show all fields
Show interest group fields
Show required field indicators
Show format options
HTML, plain-text, mobile options.
Optional: Form width
Form width in pixels. Leave blank to let the form take on the width of the area where it's placed.
Preview
Subscribe to our mailing list
* indicates required
Email Address *
First Name
Last Name
Copy/paste onto your site
<!-- Begin MailChimp Signup Form -->
<link href="//cdn-images.mailchimp.com/embedcode/classic-10_7.css" rel="stylesheet" type="text/css">
<style type="text/css">
#mc_embed_signup{background:#fff; clear:left; font:14px Helvetica,Arial,sans-serif; }
/* Add your own MailChimp form style overrides in your site stylesheet or in this style block.

Terms

List	A list of subscribers who have opted in to hear from you.
Groups	A category within your list that organizes your subscribers by their interests and preferences.
List Fields and *|MERGE|* tags	Extra information about your subscribers is stored in list fields. Just like Groups, this data can be hidden or visible and can be used in your content via the *|MERGE|* tag. You can also add your own custom fields.
Segments	A specific set of your list subscribers with common qualities. Segments can be created to target subscribers by location, engagement, activity, and more – with a maximum of five filters selected.

Tutorial 7 • JavaScript and jQuery

This tutorial demonstrates the basics of using the JavaScript programming language and jQuery library to add interactivity to a web site. Thanks to Ahmed (Am) Sagarwala (GCM '08) for contributing the section on jQuery.

"Plain" JavaScript

Using a Variable, Conditional Statement, and Function

In this section we will see how to write JavaScript code, include a conditional statement, and place the code into a function so it can be used multiple times.

1. Open the starter document for the "Fireside Chat."
2. Inside the , place opening and closing <script> tags and add the code:

   ```
   document.write("Welcome to the Fireside Chat web page.");
   ```

   Note that the text is written below the heading.
3. Add opening and closing <p> tags inside the quotes. Add a style statement and style the text as you like.
4. Above the "document.write" line, add a variable statement to hold the reader's name:

```
var name=prompt("Please type your name.");
```

5. Add the variable name to the statement by placing it outside the quotes and connecting it to the text with "+" signs:

```
document.write("<p style="font-family: Georgia;">Welcome to
```

6. If the reader does not type a name, the statement will print with a blank. To prevent the statement from printing if the user does not enter a name, place the "document.write" statement in a conditional "if" statement:

```
if (name!="") {

document.write("p style="font-family: Georgia;">Welcome to

}
```

7. To make a function out of the statements, add opening and closing <script> tags to the <head>. Create the function by writing:

```
function writeName() {

place code here

}
```

8. Call the function by keeping the <script> tags in the <body> and writing:

```
writeName();
```

Making a Print Button

In this section we will create a function to print the page and link

it to a button. Adding an “@media” query will enable fitting the document on a letter-size page and omitting the buttons from the print.

1. Program the button at the bottom of the page to print the page.
2. Add a function to the <script> tag in the <head>:

```
function printPage() {

this.window.print();

}
```

3. Call the printPage function by linking it to the “Print” button:

```
<button onClick="printPage()">Print</button>
```

4. Add an @media print {} query to confine the page to 8.5×11″ and hide the buttons in the print view:

```
@media print {

body {

transform: scale(0.85);

} #page {

width: 8.25in;

height: 10.0in;

} footer {

display: none;
```

```
}
```

Using Buttons to Change Color

In this section we will link a button to a function to change color of the text.

1. In the <script> tag in the head, add a function to change color of the div “livearea”:

```
function colorGrey() {

document.getElementById("livearea").style.color="dimgray";

}
```

2. Link the function to the “Grey” button:

```
<button onClick="colorGrey()">Grey</button>
```

3. Create a similar function to change the text color back to black.

jQuery JavaScript Library

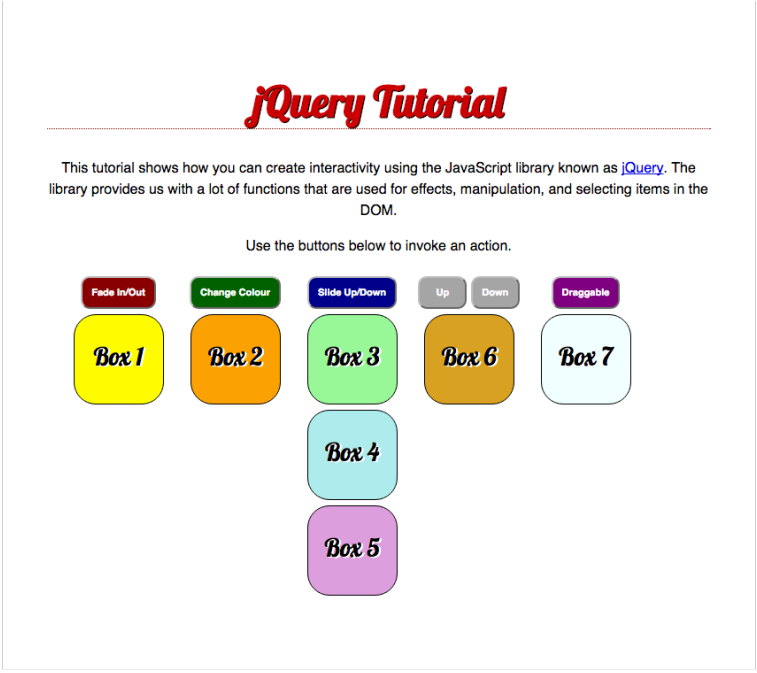
jQuery is a curated library of functions that was started by a group of computer science professors at MIT. jQuery simplifies the application of JavaScript by providing pre-coded functions. Instead of scripting all of the steps to select an element and make it do something, we can usually achieve this in a few lines of code using jQuery’s functions.

The jQuery library is only a few K in file size and can be

downloaded and hosted on your site or linked-to on the code.jquery.com site.

To create an interaction using jQuery, think of a sequence of events—an action and the resulting reaction. For example:

- The user moves the cursor to a button.
- A click action on the button takes place.
- An element on the page reacts by fading out.



jQuery exercise

1. Create Custom Style to Change Color

Create a custom style that will be applied to change the color.

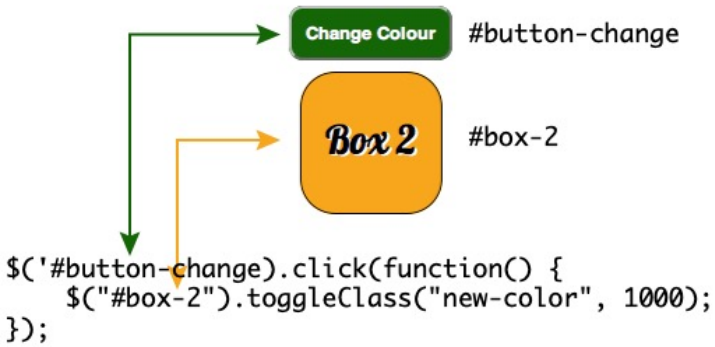
```
#box.new-color {
background: lightblue;
}
```

2. Link to jQuery

Links to the jQuery and jQueryUI libraries need to be inserted. You can download both libraries and host them on your site, or link to them on Google's hosted libraries site, GoogleAPIs (application programmer interfaces). Place the jQuery link anywhere before your JavaScript code, but it is best to place this before the end of your </body> tag. This helps with page load times and there improves user experience. Around line 49 in our tutorial is ideal for this (before the line of code starting with $).

```
<script src="https://code.jquery.com/jquery-3.3.1.min.js"></scr

<script src="https://code.jquery.com/ui/1.12.0/jquery-ui.min.js
```

```
3. Add Selectors to Code
```



Add your selectors to the JavaScript code:

```
$('#button-change').click(function() {

$("#box-2").toggleClass("new-color", 1000);

});
```

4. fadeToggle

Use the fadeToggle function to fade the box in and out.

```
$('#fade').click(function() {
$("#box-1").fadeToggle(1000);
});
```

5. slideToggle

Use the slideToggle function to make the boxes slide up and down.

```
$('#slide').click(function() {
$("#box-3, #box-4, #box-5").slideToggle(1000);
});
```

6. animate

Use the animate function to move Box 6 down and up with the respective buttons. Animation can be done in the directions top, bottom, left, and right. The operators "+=" and "-=" mean add or subtract the specified value (215px) to the current value.

```
$('#down).click(function() {
```

```
$("#box-6").animate({
top: '+=215px'}, 1000);
});

$('#up).click(function() {
$("#box-6").animate({
top: '-=215px'}, 1000);
});
```

7. draggable

Use the draggable function to enable readers to click-and-drag Box 7 around the page.

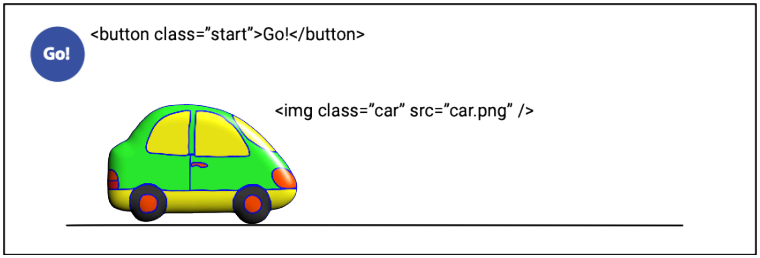
```
$('#box-7').draggable();
```

Tutorial 8 • Animation Methods

Animation generally refers to movement of vector graphics. Objects that move have a great deal of impact and educational value, and vector graphics have small file sizes. User interaction and control can make animations that much more engaging.

Four ways to add animations to web pages (and ePub eBooks) include (1) CSS, (2) CSS with JavaScript, (3) jQuery, and (4) an animation program.

Animation with CSS



The CSS @keyframes rule and animation-name style enable you to animate objects in web pages, however there is minimal user control. If you want to have a button start the animation (as in the example above) or otherwise have user control, you would have to use JavaScript to connect the button with the object. For CSS animation the object must have position: relative; or position: absolute;, then you can control the left and top coordinates, along with other styles like size (width, height), color, and opacity.

The @keyframes rule specifies major change points as “from” and

“to” or as percentage points from 0% to 100%, accompanied by the styles applicable to each point.

```
.car {

     animation-name: letsgo;

     animation-duration: 3s;

     animation-fill-mode: forwards;

     animation-iteration-count: 1;

     position: absolute;

} @keyframes letsgo {

     0% {left: 100px;}

     100% {left: 700px;}

}
```

Animation with JavaScript

In the example above, a “Go!” button can be added and used to start the animation using JavaScript.

1. For the car, set “animation-play-state: paused;”
2. Write a JavaScript function to select the ID, “car,” and set the animation-play-state to ?“running” (see below).
3. Add the function to the button: <button onClick=”animateCar()”>Go!</button>

```
<script>

     ?function animateCar {      document.getElementById("car")
```

```
</script>
```

Animation with jQuery

User interaction can also be implemented with jQuery, an opensource library of JavaScript functions that was started by a group of MIT computer scientists. To use jQuery you have to either download and import the jQuery library (jquery-3.3.1.min.js) and the jQuery user interface (UI, jquery-ui.js), or connect to both from Google's GoogleAPIs hosted site. jQuery uses the ID assigned to the button and the object to link the two.

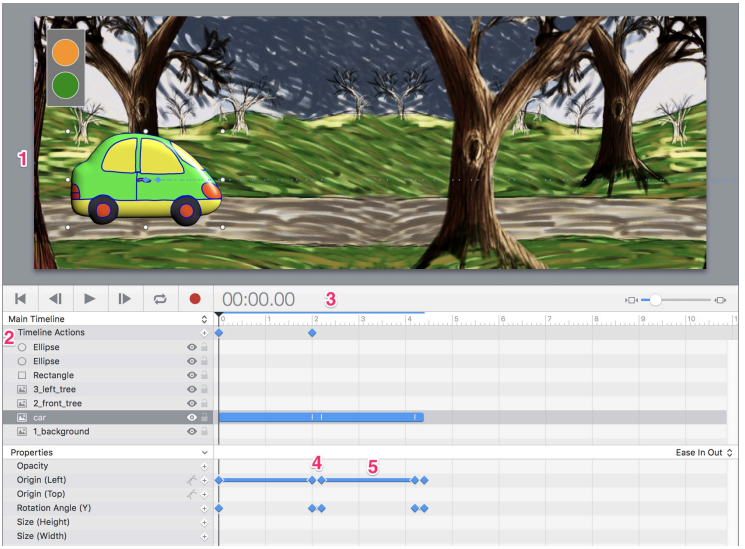
```
$(function() {

          $('start').click(function() {

                $('car').animate({left: '+=1200px'}, 1000);

          });

});
```

In this example, we are animating the left coordinate of the "car" class by adding 1200px to the current value, which moves it off the screen, and doing the animation in 1000 microsec., or 1 sec. The animation is triggered when the user clicks the button, class 'start.'

Animation with Tumult Hype

Tumult Hype (www.tumult.com/hype) is a vector-graphic animation program for MacOS similar to Adobe's Flash and Animate. Hype has an easy-to-use interface and simple procedure for placing

an animation into a web page or iBook. The publisher offers a fully functional demo for 15 days.



Screen capture of an animation in Tumult Hype. (Car from pngtree.com.)

Components in Tumult Hype	
Component	**Function**
1 stage	document or pasteboard for the animation
2 layers and properties	each object goes on a separate layer and is affected by CSS properties in the list
3 timeline with playhead	controls timing of the animation
4 keyframes	main points of change in the animation
5 tweens	transitions between keyframes

New Document

1. Make a new document in Hype. Set the stage size to the width and height of the background image (864324px).
2. Drag-and-drop the background image onto the stage.
3. Drag-and-drop the car image onto the "road," size appropriately, and place at the left edge of the scene.
4. With the Elements, draw a rectangle with coloured circles that will be made into buttons.

Animation

1. The animation sequence in Hype is: timeline > keyframe > position. To animate the car, first select it. Put the Play Head at 00:00.00 sec. and place a Keyframe for the Origin (Left) Property at that point.
2. Move the timeline to 2 sec. (or the length of time you want the car to take to get off the screen) and place another Keyframe > Origin (Left) at that time.
3. Move the car to the right, off the stage.
4. Play the animation in Hype to see how it works.

Buttons

1. To give users control over the animation, select the green circle that you drew. In the Actions Inspector at right (hand symbol), click on On Mouse Up (Touch End) > Action > Continue Timeline.
2. To keep the animation from starting by itself, bring the Play Head to the 00:00.00 position. Set Timeline Actions > Action > Pause Timeline.

3. To test your animation, click the Preview button, which will open it in the browser of your choice. Check that the button works and that the animation plays as expected.

Additional Steps

1. To make the car turn around at the end of the animation, add a Keyframe for Rotation Angle (Y) and set to 180°.
2. Add another Timeline Action > Pause Keyframe to make the animation stop before the car turns around.
3. Add more seconds to the timeline to make the car drive back to the starting point and turn around again.
4. Program a second button to bring the car back.

Export Animation

A Hype animation can be placed into into a web page by copying the indicated tag and accompanying resources folder.

The animation can be placed into an Apple iBook by exporting to Apple Widget (.wdgt) format. Just drag-and-drop the .wdgt file into iBooks Author.

The animation can be placed into a fixed-layout ePub by exporting as Animated GIF, however this format offers no user control.

Tutorial 9 • Video

Videos add an element of interactivity to web sites, whether to show how something works, what it looks like, how it behaves, or how it feels to be there. Videos are sequences of bitmapped images, usually 24 to 30 per second, and therefore. The higher the resolution, the larger the file. Therefore it's important to consider how much resolution is required, based on whether the video is intended to inform, persuade, or entertain.

iMovie

1. Start iMovie. Note the three buttons at top left: Import, Create, Share.
2. Import the MP4 video of the i1 iSis scanning spectrophotometer.
3. Create ("+" symbol) a new movie called "i1 iSis."
4. In the Clips window, Select All and move the iSis clip to the stage.

Detach Audio

1. Detach the audio from the video (right-click > Detach Audio or Modify > Detach Audio).
2. Select the audio track and delete.

Add Time Lapse Effect

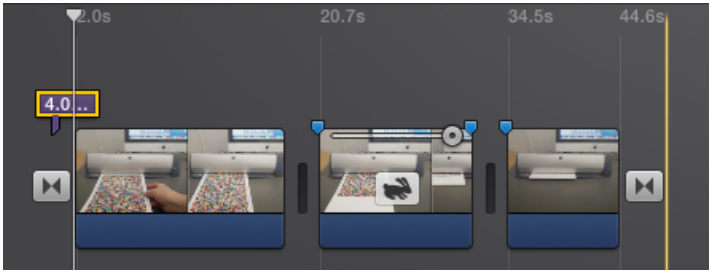
1. Locate the frames and place markers (Mark > Add Marker) where the iSis starts and finishes reading the chart.
2. Place the play head at each marker and split the clip (Modify > Split Clip).
3. Select the clip and adjust the speed (turtle icon) to Custom > 2000% to speed up the movie 20 times.

Add Fade-In and Fade-Out

1. From the lower left corner of the screen, select Content Library > Transitions.
2. Add a “Fade to Black” transition at the beginning and end of the movie.

Add Titles

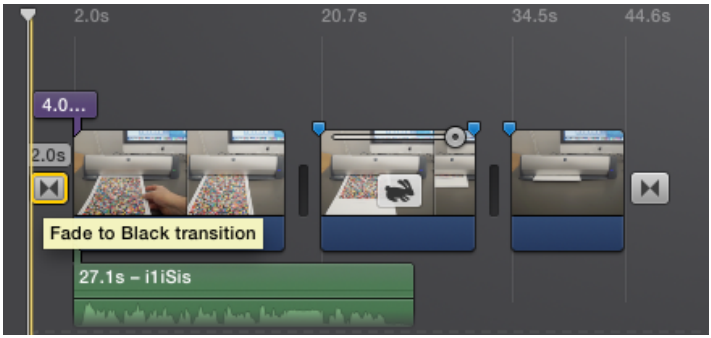
1. From the Content Library select Titles.
2. Choose a title format and drag-and-drop the title onto the top of the movie. Add title text.
3. Note that the title can start with or after the Fade-to-Black (fade from black) effect.



iMovie with 3 clips, 2-sec. fade-in from black, 4-sec. title, and 1-sec. fade-out to black

Write a Script

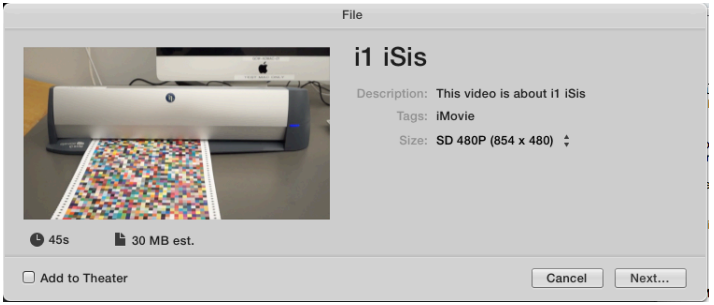
1. Using the supplied template, write a script for a narrative about the i1 iSis, what it does, and how it works.
2. Using a 3-column layout, write the scene in the left column, narrative in the middle column, and time in the right column.
3. Record your script using Quicktime Player (File > New Audio Recording) and the built-in microphone on the iMac.
4. Drag-and-drop your sound recording onto the movie.



iMovie with added voiceover narrative

Export

Export your clip to MP4 format (Share > File > SD). Note the estimated file size, download time, and resolution.



Export to standard-definition (SD) movie.

Insert in Web Page

1. In Dreamweaver make a new web page and insert the video using the <video> tag:
   <video src="i1iSis.mp4" width="854" height="480" controls></video>
2. Make a screen capture (CMD+SHFT+4) of the movie to use as a poster, and add this to the <video> tag:
   poster="iSisMovie.png"

Appendix • Photo Credits

"Adobe® Photoshop® and Dreamweaver® screenshots reprinted with permission from Adobe Systems Incorporated. Adobe® and Photoshop® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Screen captures cannot be cropped further."

Tutorial 3

Partial screen captures of "demo-text-styles" and "demo-color-names" files from *Creating Web Pages with HTML and CSS* by Patrick Carey, Thompson Courseware.