About Software Training for Students

Software Training for Students is an organization on campus that provides free software training to all students and faculty. Our services include custom workshops, open-enrollment classes, one-on-one project help, and access to Lynda.com. For more information on the Software Training for Students (STS) program, visit our website at wisc.edu/sts.

STS is part of the Division of Information Technology (DoIT) - Academic Technology at UW-Madison. For more information regarding DoIT Academic Technology, visit at.doit.wisc.edu.

© University of Wisconsin Board of Regents.

This manual and any accompanying files were developed for use by current students at the University of Wisconsin-Madison. The names of software products referred to in these materials are claimed as trademarks of their respective companies or trademark holder.

If you are not a current member of the UW-Madison community and would like to use STS materials for self-study or to teach others, please contact sts@doit.wisc.edu. Thank you.
<table>
<thead>
<tr>
<th></th>
<th>Topics Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>The Dreamweaver Interface</td>
</tr>
<tr>
<td>3</td>
<td>Setting Up a New Site</td>
</tr>
<tr>
<td>4</td>
<td>Accessibility</td>
</tr>
<tr>
<td>5</td>
<td>Working With Text</td>
</tr>
<tr>
<td>6</td>
<td>Document Structure with Divs</td>
</tr>
<tr>
<td>7</td>
<td>Dreamweaver Breadcrumbs</td>
</tr>
<tr>
<td>8</td>
<td>Introduction to CSS</td>
</tr>
<tr>
<td>9</td>
<td>Working With Images</td>
</tr>
<tr>
<td>10</td>
<td>Page Layout with CSS</td>
</tr>
<tr>
<td>11</td>
<td>The Box Model</td>
</tr>
<tr>
<td>12</td>
<td>Bonus: Page Templates</td>
</tr>
<tr>
<td>13</td>
<td>Bonus: Embedding External Content</td>
</tr>
<tr>
<td>14</td>
<td>Bonus: Advanced CSS Effects</td>
</tr>
</tbody>
</table>
Introduction

Adobe Dreamweaver, part of the Adobe Creative Suite package, is a powerful application used to streamline website construction and maintenance. This extremely popular application is considered an industry standard, used by all shapes and sizes of organizations.

About this Class

This manual provides a basic introduction to the Dreamweaver interface and explores commonly used features by guiding students through the construction of a simple web page.

During this class students will:

- Become familiar with the Dreamweaver environment and workflow
- Gain a basic understanding of common web design practices
- Implement HTML and CSS concepts and web design practices by utilizing common Dreamweaver tools and features

Dreamweaver is a class intended to serve as an introduction to Dreamweaver, and to website design concepts in general. It also explores advanced techniques and how to transform the web page constructed in this class into a full-featured website.

Prerequisites

- Basic understanding of Hyper Text Markup Language (HTML) and Cascading Style Sheet (CSS) concepts

Other Requirements

- Access to Adobe Dreamweaver CC software
- Web browser
- Internet connection

The Dreamweaver Interface

This section provides a brief overview of the Dreamweaver interface and describes commonly used components.

Interface Overview

The portion of the Dreamweaver interface that is used to view and edit documents is known as the Workspace. A wide variety of windows, tools, and menus (known as Panels) can be displayed within the Workspace in order to provide quick access to the features needed for a given task.

1. Open the Dreamweaver CC application
2. At the Welcome Screen, choose Create New > HTML. If the Welcome Screen is not displayed, go to File in the top menu bar and choose New > HTML.

The resulting Workspace should look like the image on page 6, with the Application Bar, Document Window, Property Inspector, Insert Panel, CSS Styles Panel, and Files Panel displayed. These Workspace items are discussed on the next few pages.
Application Frame Mode

In Mac, the Dreamweaver Workspace behaves differently than in Windows. In Windows, documents, menus, toolbars, etc, are grouped together inside one window, but, in Mac, documents float freely from other items and can be maximized/minimized independently of the application itself.

Mac users who prefer the Windows approach can mimic it by using Application Frame mode. To turn on Application Frame mode, go to Window > Application Frame.

Common Interface Components

A. Document Window

The Document Window displays the document that you are currently working on. If you have multiple documents open, each document appears as a tab in the Document Window.

Document Windows each contain a Document Toolbar, which provides viewing options for the current document. Using the buttons on the Document Toolbar, you can view a document in several ways:

- **Code view** - Shows only the HTML code for your document
- **Design view** - Shows only an editable visual layout of your document
- **Split view** - Shows both HTML code and an editable layout
- **Live view** - Shows a non-editable visual layout of your document, as it would look in a particular web browser

B. Application Bar

On Mac machines, the Application Bar contains very few tools; menus are docked at the top of the screen, not within the Dreamweaver window.

On Windows machines, the Application Bar contains menus common to many applications, such as File, Edit, and View, as shown below.

C. Property Inspector

Located at the bottom of the screen, the Property Inspector shows settings and options for an object or section of text that you currently have selected. The options in the Property Inspector change depending on what type of object is selected.

D. Insert Panel
The Insert Panel is used to easily add elements such as links, images, and tables, to a document. Available elements are divided into categories such as Common, Layout, and Text. Use the drop-down menu in the upper left corner of the Insert panel to switch element categories.

**E. CSS Styles Panel**

The CSS Styles Panel shows CSS (Cascading Style Sheet) rules applied to your document, or to an object within your document. Click the All button at the top of the CSS Styles Panel to view all rules for the current document or select an object in the document and click Current to view only the rules that affect the object. The CSS Styles Panel can also be used to edit existing rules or add new rules.

**F. Files Panel**

The Files Panel allows you to access and manage your site files, both on your local machine and your remote server. The Files Panel is also used to transfer files between the local machine and remote server.

Use the drop-down menu in the upper right corner of the Files Panel to switch between viewing your local files (Local view) and your remote files (Remote server). To view local and remote files side-by-side, click the Menu Icon in the upper right corner of the Files Panel and choose View > Expand Files Panel.

**Showing/Hiding Individual Panels**

In addition to using pre-defined Workspace Layouts, you can show/hide individual panels.

1. To show a panel, go to the Window menu at the top of the screen and Click the name of the panel you'd like to display.
2. To hide a panel, Click the Menu Icon in the upper right corner of the panel and Click Close.
3. Once we're done discussing the workspace layout, Close the current document.

**Setting Up a New Site**

Regardless of its design or purpose, almost every Dreamweaver site begins with the same steps. In this section, we'll learn how to properly set up a new site.

**Creating a Folder for Site Files**

Before opening Dreamweaver, we'll set up a folder on the desktop to store the files we'll create.

1. Right-click on your computer desktop and choose New Folder.
2. Name your new folder "dreamweaver".
3. Within your dreamweaver folder, Create a new folder and Name it “images.”

**Site Definition**

In order for the program to function properly, each site must be "defined" within Dreamweaver. This process is called Site Definition.
Defining a New Site

1. In Dreamweaver, Go to the Site menu at the top of the screen and Click New Site...
2. In the Site Name field, Type "dreamweaver."
3. In the Local Site Folder field, Click the folder icon to the right of the field and browse to the dreamweaver folder on your Desktop that you created in the previous exercise.

4. Click Save. Notice that the Files Panel now shows the contents of the dreamweaver folder. This indicates that the site is properly defined.

Creating Pages

Every website consists of individual web pages, which are linked together. This section covers the basic steps for creating a page in Dreamweaver.

First Page - index.html

In this exercise, we'll create a very simple web page using text from our class files.

1. In Dreamweaver, go to the File menu at the top of the screen and Click New...
2. Set the Page Type to HTML and DocType to HTML5.
3. Click on the Bootstrap tab and make sure that the "Include a pre-built layout" is unchecked. This will give us a blank HTML document.
4. In the upper-left corner of the window, Click Split View so that you can see both the Design View and Code View for the page.
5. In the Title field, Type Dreamweaver. In a live site, text in the title field appears at the top of the web browser window and in search engine results.
6. Open the website-text.txt file from the class folder in a text editor.
7. Select all of the text in this file (Control /Command + A) and Copy it (Edit > Copy or Control/Command + C).
8. Click in the Design View, then Paste (Edit > Paste or Control/Command + V) your copied text. In the Code View, notice that Dreamweaver has automatically added some HTML code.
Let's take a look at our page in a web browser - from the File menu at the top of the screen, choose Preview in Browser > Firefox.

A dialog box asking you to save the page will appear; Click Save/Yes

Name the file index.html and save it in your dreamweaver folder.

Your page should now appear in a new Firefox window. Congratulations! You just made your first web page with Dreamweaver!

Accessibility

When creating a website, it is important to keep in mind that users may access your site from widely varying technological environments. Users with low/no vision, for example, may listen to your site’s content as read aloud by a device called a screen reader. Users may also visit your site using older browsers that behave very differently from modern versions. It is important for people of all abilities, and with all levels of browsing technology to be able to access the content on your site. This topic is referred to as Web Accessibility.

Web developers use a variety of techniques derived from best practice guidelines, such as those published by Web AIM, in order to make sites accessible to as many users as possible. Accessibility best practices will be introduced throughout this class and should be incorporated into all of your web development projects.

It is important to note that adherence to accessibility guidelines is required by many organizations. UW-Madison, for example, requires compliance with its Web Accessibility Policy, available at http://www.wisc.edu/policies/wwwap.php.

Working With Text

Text makes up most of the content on a web page. In order to format our text so it is neat and easy to read (both in a browser and with a screen reader), we need to enclose each section of text in HTML tags. This process, called structuring, also allows us to
apply consistent styles to our text using Cascading Style Sheets (CSS). In the following exercises, we'll structure our text using common HTML tags. We'll also add links and change font characteristics using basic CSS techniques.

**Structuring Text**

A large portion of web design work involves structuring text, which consists of adding HTML tags that group text into logical sections. This is tedious work, but Dreamweaver can make it go much more quickly. While adding these tags might seem dubiously useful at first, structural tags serve two very important purposes:

- Structural tags allow us to apply custom styles quickly and consistently throughout a page or an entire site.
- Structural tags allow visitors who are using screen readers or other assistive devices to navigate through a page easily.

**Adding Headings to Text**

1. In the Dreamweaver Design View, Highlight the top line of text that reads WELCOME TO MY PORTFOLIO WEBSITE! This text looks like a main heading, so we'll give it Heading 1 tag.

2. In the Properties Panel at the bottom of the window, Change Format to Heading 1. This changes the text size/weight in the Design View and, more importantly, it wraps the selected text in `<h1>` tags in the Code View.


   Oh no! The entire block of text changed to Heading 2! As mentioned at the start of class, Dreamweaver is not infallible. To fix the problem, we'll need to edit in the Code View.

4. In the Code View, Delete the `<br>` tag that follows ME in ABOUT ME.

5. Highlight the closing Header 2 tag `</h2>`, Cut it (Edit > Cut or Control/Command + X), and Paste it (Edit > Paste or Control/Command + V) right after ME in ABOUT ME.

6. Highlight the remainder of the text in the ABOUT ME section and, in the Properties Panel, change Format to Paragraph. Your finished text should look like the image below.

**Adding Lists to Text**

Instead of plain old paragraphs, let's add some lists to the Education and Work Experience sections.

1. First, we'll make EDUCATION a subheading. In the Design View, Highlight EDUCATION and, in the Properties Panel, change Format to Heading 2.

2. As in steps 5-6 above, Delete the `<br>` tag after EDUCATION and Cut/Paste the closing `</h2>` tag in its place.

3. Next, we'll turn the text in this section into a list. In the Design View, Highlight all of the text in the Education section (not including the EDUCATION subheading).

4. In the Insert Panel, make sure the menu at the top of the panel is marked HTML, and Click ul Unordered List.
Notice in the Code View that Dreamweaver has placed <ul> tags around the entire block of text and <li> tags around each item in the list.

Repeat the previous steps to format the Work Experience section so that it looks similar to the Education section. Your finished text should look similar to the image below:

Special Characters

HTML uses bits of code called Entities to represent special characters, like mathematical operators and other symbols. In this exercise, we’ll add a © symbol to our copyright statement using an entity.

1. In the Design View, Highlight the word Copyright at the bottom of the page.
2. Go to the Insert Panel and choose Common from the dropdown menu at the top of the panel.
3. In the Insert Panel, Click the arrow next to Characters ; a drop-down menu showing common special characters opens.
4. Click © Copyright.

Creating Links

Links allow users to easily move from one web page another. Links can direct users to external websites (external links) or to different pages within your own site (local links). In this exercise, we’ll create both External and Local links.

External Links

In this exercise we’ll link some of the text in our page to an external site.

1. In the Design View, highlight the text "extreme ironing", under the About Me section.
2. In the Properties Panel, type http://www.extremeironing.com /into the Link field (this is the page that our link will direct to).
Local Links

In this exercise we’ll create a simple navigation menu by using an Unordered List and Local Links. We don’t have any additional pages for our site yet, so the links won’t work yet.

First, we’ll add a list of page names using an unordered list element. In the Design View, Click before the W in WELCOME to place your cursor there.

1. In the Code View, notice that the cursor is inside the <h1> element that contains our main heading text. We don’t want our navigation menu inside the <h1> element, so, in the Code View, Click just before <h1> to move your cursor.
2. Go to the Insert Panel and make sure HTML is chosen in the drop-down menu at the top of the panel.
3. Click ul Unordered List to add a list element just above the main heading, then Click li List Item to add the first bullet point to the list.
4. Click the Refresh Design View button at the top of the Document Window to see the bullet appear, then type "Home."
5. Press Return to add another bullet point and type "Portfolio." Add one more bullet and Type "Contact Me."

Now we’ll create the links. In the Design View, Highlight Home.

6. In the Properties Panel, make sure that the view is set to HTML.
7. In the Link field, Type index.html. As long as our pages are all stored within the same folder, only the file name for the page is needed in order to create a local link.
8. In the Title field, Type Home. Leave the Target field blank.

9. Repeat steps 7-10 for Portfolio and Contact Me, using the file names portfolio.html and contact.html.

Document Structure with Divs

In addition to structuring text, we also need to structure our document into logical sections, so that we can position them on the page using CSS. Divs are HTML tags that have no default styles - they are simply containers that we can use to group HTML elements together.
Inserting Div Elements

Wrapping Divs Around Existing Elements

1. In the Design View, Select all of your text.
2. If the Insert Panel is not showing, Go to Window>Insert.
3. Set the drop-down menu at the top of the Insert Panel to Common.
4. Click Insert Div Tag.
5. In the window that opens, Leave Insert set to Wrap around selection, Type "wrapper" in the ID field, and Click OK. Our page content (the text) is now contained inside of one div named wrapper.
6. Highlight the Copyright Statement at the bottom of the page and repeat steps 4-5, using footer as the ID.
7. Highlight all of the text EXCEPT the Copyright Statement, then repeat steps 4-5, using content as the ID.

Inserting Divs At a Specific Point

1. One last div! For this one, we want to add a div at the very top of the page, where there is no content. Since there's nothing to select, we need to pinpoint the correct location in the code view. In the Code View, Click right before <div id="content"> to place your cursor there.
2. In the Insert Panel, Click Insert Div Tag.
3. In the window that opens, leave Insert set to At Insertion Point, type header in the ID field, and Click OK. In the Design View, you should now see dotted lines outlining your divs.
Dreamweaver Breadcrumbs

In Dreamweaver, it can be difficult to make accurate selections in the Design View. Dreamweaver Breadcrumbs help to overcome this problem by providing an alternate way to select different parts of your HTML document.

Selecting HTML Elements with Breadcrumbs

Using the Breadcrumbs Trail

In this exercise, we’ll learn to identify and select elements in an HTML document using Breadcrumbs.

If your Split View contains the Code View on the bottom, and the Live View on top, you must change it to display the Design View on top.

1 If Live View is displayed on the top portion of your split view, change it to display Design View. CC 2014 will automatically be in this setting. To do so, Click on the drop down arrow next to Live and Select Design, as shown in the image below.

2 In the Design View, Highlight the list of navigation links (Home, Portfolio, and Contact Me).

3 Notice that Dreamweaver has selected all of the List Items (<li></li>) in the list, but not the Unordered List (<ul></ul>) that surrounds them. This is problematic, because, in order to copy or move this list, we need to select all of its components, including the <ul></ul> tags. Fortunately, Dreamweaver’s Breadcrumbs feature can help to identify and select elements that are difficult to select in the Design View.

4 Locate the Breadcrumb "trail" in the lower-left corner of the document window, beneath the Code View. It is designed to show where your cursor is currently located within the hierarchy of your HTML tags. With our current selection, we can see that we are inside of the following tags: <body> <div#wrapper> <div#content> <ul>.

5 By clicking a tag name in the Breadcrumbs trail, you can select that tag, as well as any other tags that fall beneath it in the HTML hierarchy. To select our Unordered List and all of the List Items inside of it, Click the <ul> tag in the Breadcrumbs trail.

Moving the Links List

In this exercise, we will move the list that we selected in the previous exercise into the navbar div.

1 In the Design View, make sure that the Unordered List and its List Items are still selected.

2 Cut this selection (Edit > Cut or Command/Control + X).

3 In the Code View, Highlight the placeholder text inside <div id="navbar"></div> that reads 'Content for id "navbar" goes here' and Press Delete on your keyboard.

4 Leave your cursor inside the div in Design View, and Paste the Unordered List that you cut in step 2 (Edit > Paste or Command/Control + V).

5 The resulting code should look like the image below:
Now that we've moved the navigation menu into position, we'll work on styling it with CSS.

**Introduction to CSS**

CSS, or Cascading Style Sheets, are used to style and change the appearance of HTML elements.

If the CSS Designer panel is not open, it can be opened by going up to Window on the top menu bar and selecting CSS Designer.

**Using the CSS Designer Panel**

The CSS Designer panel can be used to access all of the CSS rules applied to your web pages through the linked CSS style sheet (or sheets). It can also be used to inspect the rules applied to a current selection in the design or code view.

**Adding New Styles**

Using the Dreamweaver breadcrumbs and the CSS designer panel, we are going to create a new CSS rule for the links in the navbar div that we just pasted from the content div. We are going to apply properties to remove the bullet points and underlines from the links.

**Styling the Navbar List**

In this exercise, we'll create a style rule to remove the bullets from our navbar list.

1. Select the entire Unordered List by either highlighting in the Code View from the opening <ul> tag to the closing </ul> tag or by choosing the <ul> tag in the breadcrumbs trail.
2. In the CSS Selectors Panel, Click the New CSS Selector button.
3. Dreamweaver will supply #wrapper #navbar ul as the selector. While technically accurate, this selector is longer than needed (see note at right). Delete the redundant selector to get #navbar ul.
4. Hit Enter to confirm your changes.
5. In the CSS Properties window to the right, Click the "Text" button in the Category options along the top side of the panel.
6. Click on the grayed out text next to list-style-type to get a dropdown menu to appear.
7. From the drop-down menu next to list-style-type, Choose None.
8. Select Center from the button options to the right of text-align.

---

**Decendant Selector**

The example: #wrapper #navbar ul is called a decendant selector in CSS. This selector will select every unordered list inside of an element with an ID of ‘navbar’ that is inside of an element with an ID of ‘wrapper’.
Click the Layout button in the Category options on the top side of the panel.

In the Padding section, Click the link in the center of the box diagram, and Enter 0px on any side of the box. With the link clicked, it should apply 0px to all sides.

After creating this CSS rule, you will see that the bullet points have been removed from the list of links in the navbar div. The padding from the list has also been removed and the links line up in the center of the div. Padding will be further explained later in this course. You should also see that a new rule has been added to the CSS Designer panel for #navbar ul.

Styling Links in the Navbar List

To finish our navigation menu, we'll add styles to the individual links in our list. In this exercise, we'll change the text color and remove the underline from the links.

Note that we'll target the <a> tag to make these changes, rather than the <li> tag. This is because the default style properties (blue color, text underline) that we would like to remove are applied to the <a> tag, not to the <li> tag.

1 In the Design View, Click one of the menu items in your navigation bar to place your cursor there. In the Breadcrumbs bar, you should see an <a> tag at the end of the trail.
2 Choose the <a> tag from the Dreamweaver breadcrumbs.
3 In the CSS Selector Panel, Click the "New CSS Rule" button
4 Change the selector to be #navbar ul li a. Then, Hit Enter.
5 First we'll change the text color. In the Text category, Click the box next to "Color" to open the Color Picker tool.
6 In addition to choosing from the color palette, you can also use the Color Dropper to sample any color you see on your screen. In this case, we will sample the brown color that we've used elsewhere on this page. Click on the Color Dropper and then Click a brown part of the header graphic to select that color. Alternatively, you can type #512B09 into the text box for the Color property.
7 Next, we'll remove the underline. Still working in the Text section of the CSS Rule panel, Click the "none" button next to the Text-decoration field.

Adding Style Rules to Divs

Setting a Width for the Content Area

We can easily specify the size of our page's content area by adding a style rule to the wrapper div. Since the wrapper div contains all of our content elements, style rules applied to it will affect those content elements as well.

1 In the Design View, Click the dotted outline that wraps around all of your text - this is <div id="wrapper">. When you have it selected, the outline should turn into a solid line.
2 Note that the Properties Panel doesn't give us too many options for styling this element. In order to style this element, we'll use the CSS Styles Panel in the Panel Dock instead. If the CSS Styles Panel is not visible, Click the CSS Panel button in...
At the bottom of the CSS Panel, Click the New Rule button

In the New CSS Rule window that opens, set the options as listed below.

- **Selector Type:** ID (applies to only one HTML element)
- **Selector Name:** Wrapper
- **Rule Definition:** screen.css

This means that we’re creating a new style rule in our stylesheet that applies to just the `<div id='wrapper'>` element.

Click OK. A CSS Rule Definition window opens that will allow you to define various properties for the element.

1. In the CSS Rule Definition window, Click Box on the left-hand side.
2. In the Width field, Type 960 and Leave the units set to pixels (px). Don’t click OK yet!

Adding Background Colors to Divs

Next we’ll add some color to our layout by adding a Background-color property to the style rules for some of our elements.

1. Continuing on from step 7 of the last exercise, we’ll set the color of wrapper div. In the CSS Rule Definition window, Click Background on the left-hand side.
2. In the Background-color field, Type fff - this is the web color code for white. We’re choosing white for this element, because we’ll be changing some of the other div colors in a moment, and we want the background behind our text to remain white.
3. Click OK to apply your changes to the wrapper div.
4. Next, we’ll add a color to the footer div. In the Design View, Click the dotted outline around © 2012 Firstname Lastname.
5. In the CSS Styles Panel, Click the New CSS Rule button.
6. In the New CSS Rule window that opens, Set the options as listed below.

- **Selector Type:** ID (applies to only one HTML element)
- **Selector Name:** Footer
- **Rule Definition:** screen.css
This means that we're creating a new style rule in our stylesheet that applies to just the `<div id="footer">` element.

1. In the CSS Rule Definition window that opens, Click Background on the left-hand side.

2. Click the white square next to Background-color - this opens the color swatch panel. We could choose a color here, but this panel provides relatively limited color options. Instead, Click the Globe Icon in the upper right corner of the Color Swatch panel; this opens a Color Picker menu specific to the operating system that you are currently using.

3. Use the Color Picker to choose a dark brown color - in the example files, we used web color code #352017.

4. While we're here, let's change the text color and alignment for the footer div. Click Type in the left-hand side of the CSS Rule Definition window. In the Color field, Type fff, the web color code for white.

5. Now Click Block in the left-hand side of the CSS Rule Definition window and Change the Text-align field to center.

6. Click OK to apply the changes.

Finally, we'll set a color for the area outside of our page content. We can do this by applying a style rule to the `<body>` element.

Since we already created one style rule for the `<body>` element (the text color), we can add our background color by editing the existing rule using the CSS Styles Panel.

7. In the CSS Styles Panel, Click All at the top of the panel to show all of the elements that currently have style rules applied to them.

8. Double-Click body to open the CSS Rule Definition window.
Repeat steps 14-17 to choose a background color for the body element, but select a slightly lighter shade of brown. In the example files, we used web color code #4e3b34.

### Working With Images

In web design, we use images not only to place photographs into pages, but to add graphic elements of all sorts. In this final activity, we'll discuss best practices for creating web images, and we'll add a decorative banner image to our web page.

#### Image Optimization

Images used on the Web must meet specific file format and size requirements. Often, it’s necessary to resize an original image and convert it to a different file format in order to use it on a website. This process is called Image Optimization.

#### Image Size/Resolution

Since image file sizes affect the amount of time it takes to load a web page into a browser, it’s important to keep file sizes as small as possible while still maintaining reasonable quality.

- Always set image resolutions to 72ppi (pixels per inch) - this is the default resolution of most device screens.
- Always scale images to the exact size needed - if the space on your web page is 960px x 180px, your image should be 960px x 180px. While browsers can scale images to fit a particular space, this results in distortion and/or unnecessarily large image files.

#### Web Image Formats

Web browsers are only able to display a few types of images, therefore, we often need to convert original images into a web-friendly format. Different formats are used for different types of images:

<table>
<thead>
<tr>
<th>Format</th>
<th>Characteristics</th>
<th>Best For</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIF</td>
<td>Great at compressing flat color, but can only display 256 colors. Can display simple on/off transparency.</td>
<td>Illustrations/text with flat colors</td>
</tr>
<tr>
<td>JPEG</td>
<td>Can display millions of colors; smaller file size than PNG-24</td>
<td>Photographs, graded colors (i.e.: gradients)</td>
</tr>
<tr>
<td>PNG-8</td>
<td>Great at compressing flat color, but can only display 256 colors. Can display simple on/off transparency. (Similar to GIF)</td>
<td>Illustrations/text with flat colors</td>
</tr>
<tr>
<td>PNG-24</td>
<td>Able to display partial transparency (shadows, etc)</td>
<td></td>
</tr>
</tbody>
</table>

#### Inserting & Optimizing Images

1. In the Design View, Highlight the placeholder text in the header div that reads "Content for id "header" Goes Here" and Delete it. Leave your cursor where it is!
2. In the Insert Panel, Click Images: Image.
3. Navigate to the class files folder and Choose banner.jpg, then Click Open.
A message appears warning us that banner.jpg is not in our dreamweaver site folder, and asks if you'd like to copy it there. Since it's important to keep all of the site files in one place, Click Yes.

In the Save window that opens, Leave the file name set to banner.jpg, Navigate to dreamweaver > images (the images folder inside the site folder that we created earlier in the class) and Click Save.

In the Image Tag Accessibility window that opens, Type Firstname Lastname Portfolio Website Banner Graphic in the Alternate Text field.

Click OK to apply the changes.

Whoa! This image is huge! In the Properties Inspector, you can see that the image size is currently 3994 x 748 pixels - that's more than 3 times the width of our wrapper div.

In the Properties Inspector, Type 960 in the W field and leave the units set to px (pixels). This changes the width so that it matches the wrapper div. Notice that the height changes automatically in proportion to the width.

We've changed the size that the image is displayed at on our page, but this does not alter the size of the banner.jpg file. In order to reduce the size of the file, we need to optimize it. With the image selected in the Design View, Click the Edit Image Settings button in the Properties Inspector.

In the Image Optimization window that opens, Set the Preset field to PNG 8 for Logos and Text, then Click OK. This will create a new, optimized file called banner.png.

In the Files Panel, Click the arrow next to the images folder to show its contents. We can see that the banner.png file we just created is significantly smaller than the un-optimized banner.jpg file.
11 Since we are no longer using banner.jpg it's safe to delete it. Right-Click the file name and choose Edit > Delete. A window opens to ask whether we really want to delete the file; Click yes.

12 Save your file (File>Save or Command/Control + S), then check the finished page in a web browser by choosing File > Preview in Browser > Firefox.

Your page should look something like the image below.

---

**Page Layout with CSS**

In this section, we will use CSS to make modifications to our website's layout. Before we can dive into the exercises, we need to understand several key concepts used in web page layout: Normal Flow, Block and Inline Elements, and Floats.

**Normal Flow**

If we created a web page with no CSS rules, how would a web browser know where to place different elements in the browser window? Earlier in the class, we learned that default styles for text elements, like `<p>`, `<h1>`, and `<a>`, take over when no CSS styles are present. Similarly, when there are no CSS rules defining the layout of a page, the browser determines the layout according to its own built-in set of rules. These default layout rules are known as Normal Flow.
Block and Inline Elements

Every HTML element is inherently either a block or inline element. Block elements always start on a new line; in Normal Flow, block elements stack on top of each other. Most elements, such as `<p>`, `<h1>` and `<div>` are block elements.

In contrast, Inline elements do not start a new line. In Normal Flow, inline elements line up from left to right across the screen. If a row of inline elements is longer than the screen width, elements will wrap onto a new line and continue flow left to right. There are only a few Inline elements in HTML. Some common examples include `<a>`, `<img>`, and `<span>`.

Normal flow is evident from our current index.html page. Nearly all of the elements on the home page are block elements, so they stack vertically on top of one another.

Floats

While the stacked layout that Normal Flow produces is often fine for simple pages, there are cases when you may wish to have elements sit side-by-side. For example, in our project, we might want to place our navigation menu in a vertical column on the left side of the page and have the main content area line up right next to it. This can be accomplished using the CSS Float property.

For any block element, the Float property can be set to left, right or none (none is the default). When the Float property is set, the element is shifted as far to the right or left on the current line as possible. If there are other floated elements already on the same line, additional floated elements will line up next to them until the width of the line is filled. When the width of the line is filled, the next floated element will drop down to a new line.

It is important to remember when working with floats that any content without a float applied will continue to follow Normal Flow, and will flow around floated elements. This can sometimes produce unexpected results.

Float the Navbar Div

Floating the navbar will be the first step in achieving our desired layout. We are going to float the navbar div by creating a new CSS rule.

1. In the Design View, Click one of the menu items in your navigation bar to place your cursor there. In the Breadcrumbs bar, Click the breadcrumb corresponding to the navbar.

![CSS Selectors Panel](image)

2. In the CSS Selectors Panel, Click the “New CSS Selector” button

3. Delete “#wrapper” from the selector because this is redundant. The selector should be just be #navbar.

4. Select the Layout category on the top side of the CSS Properties panel.

5. First we set a width for the navigation bar. In the Width field, Enter “200 px”. You must first Click on the grayed out area to the right of the width property to bring up a dropdown menu. From here, Choose px for pixels, and Enter 200 for the value. Once done, Hit Enter.

6. Now, we'll float the div. In the Float field, Choose "left" from the buttons to the right.

![Float Property](image)

The navigation menu should now appear on the left side of the page. Since the content div is not floated, it now flows around the navbar div following Normal Flow. In the next exercise, we will float the content div as well.
Float the Content Div

In this exercise, we will float the content div so that it lines up next to the navbar div.

1. In the Design View, Click anywhere within the content div to place your cursor there. In the Breadcrumbs bar, Click <div id="content">

As with the navbar div, we'll first set the width. Remember that the content and navbar divs are both inside the wrapper div, which is currently set to 960px. Since we set the navbar width to 200px, the maximum width that we can set for the content div and have it remain on the same line as the navbar is 760px.

2. In the Width field, Enter "760px". To do so, Click on the grayed out area to the right of the Width label, Select px from the drop-down menu, and Enter 760 for the value. Once done, Hit Enter.

3. In the Float field, Select "left" from the buttons.

This gives us an unexpected result! If you examine the navbar and content divs, you will see that they are stacked next to each other as expected, but they are no longer white. Instead they've taken on the brown color of the footer. This issue is remedied by using another CSS property: clear.

Clearing Floats

Floating both the navbar and content divs resulted in an issue: the divs are no longer white. Why is this happening? The footer may have something to do with it.

The footer is currently flowing around both the navbar and content divs. Floated elements are not recognized by the normal flow of HTML elements. Since the footer is not floated, it thinks that it begins right after the header div and then flows around the floated divs. This can be shown by selecting the footer div via breadcrumbs. You will see that the content and navbar divs are floating on top of the footer div.

This issue with floating elements can be remedied by the use of another CSS property: clear. The clear property pushes an element below floated elements that appear before it in the HTML code. Applying this property to the footer will alleviate the background color issue.

Apply Clear to the Footer

By adding a clear to the footer, we will ensure that the footer sits below the floated navbar and content divs. We will do this by editing the CSS rules the footer div.

1. Under the Sources section on the right side, Select screen.css.
2. In the CSS Selector panel, Click #footer.
3. Select the Layout category.
4. Choose Clear: left from the buttons.

The issue with the background color was remedied by applying a clear: left to the footer div. This means that the footer will clear any elements with a float: left and begin on the next line below any floated elements before it.

The next steps to styling our page will be to adjust the background color of the navbar and to apply some box model elements to the page.
Working with Layout Colors

The next step to style the page is to apply a different background-color for the navbar div. We are going to use a complementary color to the page, such as the green that is located in the header image.

Changing the Navbar Background Color

Distinguishing the navbar is made easier by choosing a separate color for the div. The goal of the following exercises is to create a layout with a green navbar that stretches the entire page and a white content div that sits to the right of the navbar.

We are going to change the background color of the navbar div to green by editing its CSS rules.

1. Under the Sources section on the right side, Select screen.css.
2. In the CSS Selector panel, Select #navbar with a single click, since rules have already been defined for the navbar div.
3. Find the background-color property from the Background category of the CSS Properties panel.
4. Using the color dropper, Choose a shade of green from the header image at the top of the page. Alternatively, you can enter #848930 as the hex code for the color.

You will see that changing the navbar div background color will not give us the results we want. The goal was to create a continuous color that flows down the entire side of the page in-line with the content div. The white below the navbar div is coming from the wrapper div that contains all of the other divs. The background color for the wrapper div is currently set to white. By playing with some CSS rules, we will be able to get the green column that we are looking for.

Changing the Wrapper Background Color

Changing the background color of the navbar did not give us the desired result. Instead, we are going to change the background color of the wrapper to green to get the desired color scheme.

1. In the CSS Designer panel, Select the background-color property for #navbar and Delete the property by pressing the delete key on your keyboard.
2. Select #wrapper in the CSS Designer panel.
3. Change the background-color property from white to green by using the color picker or by entering the color hex code of #898E33.

This will change the background color of the wrapper to green, which seems to affect both the navbar and content divs. By changing the background-color of the content div to white, we can achieve our desired page color scheme.

Checking the Content Div Background Color

Next we are going to change the background color of the content div to get the desired color scheme.

1. Select #content in the CSS Styles panel.
2. Double-Check that the background-color of the element is white (#FFF) in the CSS Properties panel.
3. Save your HTML and CSS files by Navigating to File > Save All.
4. Preview your site pages in a browser (File > Preview in Browser or Click the Globe Icon at the top of the Document Window).

Changing the background colors of the wrapper and content divs has given us the desired color scheme for our web site. This worked because the navbar and content divs are inside of the wrapper div, so, according to CSS rules, they will inherit the
The properties of the wrapper unless they are given their own background colors.

The Box Model

All HTML elements can be considered as boxes. The CSS box model is essentially a box that wraps around an HTML element and has properties that can be modified to adjust spacing: margin, border, padding, and content.

![Box Model Diagram]

The box model brings up an important question: how big is an element on the page? The answer is not always clear, but it is critical for laying out your page. The true size of your element isn't simply the width you give an element, but the sum of the border, padding, margin, and content properties of the box model.

Content

The Content portion of the box model is the area where text and images appear. This is considered to be the foundation of the box model. The other box model elements build outward from the content. It is important to note that when you use CSS properties to set the height and width of an element, you are setting the width and height of the content area. The width of our divs have been set with CSS rules, but they will need to be modified for the box model.

Padding

Padding is used to clear an area around the content. It is used to create a buffer between the edge of the content and the border. Padding helps improve readability. It is important to note that padding is affected by the background color of the box. Padding can be applied to the left, right, top, bottom, or any combination of those four options at once.

Add Padding to the Content Div

Adding padding to the content div will make our text more readable. This will be accomplished by editing the CSS rules for the content div.

1. Under the Sources section on the right side, Select sreen.css.
2. Choose #content in the CSS Selector panel.
3. Enter 10 pixels for Padding Left and Right. To do so, Navigate to the box diagram for the padding section. Click on the gray area for the left side padding. Enter 10px. Do the same for the right side padding.
Adding the padding to the content div has made the text more readable. The text now sits 10 pixels away from the edge of the content div on both of the left and right side. However, now our content div sits below our navbar div. This is because our content div is now too wide to sit next to the navbar.

Let's do the math. The content div is 760 pixels with 20 pixels of padding for a total with of 780 pixels. The navbar is 200 pixels wide and will sit next to the content div for a total of 980 pixels. Our wrapper div is only 960 pixels wide, so we must reduce the width of the content div to account for the padding!

### Reduce the Width of the Content Div

Adding padding to the content div also increased the size of the content div. Therefore, we must reduce the size of the content div so that our floats work properly.

1. Under the Sources section on the right side, Select screen.css.
2. Choose #content in the CSS Selector panel.
3. Change the width of the div to 740 pixels by editing directly in the CSS Properties panel.
4. Save all of your changes by going to File > Save All.
5. Preview your site pages in a browser (File > Preview in Browser or Click the Globe Icon at the top of the Document Window).

This will bring our content div to the correct size and keep our floats in the correct position.

### Border

The Border is the line immediately outside of the padding and is specified by a thickness, pattern, and color. If the pattern of the border is dotted or dashed, the background color or image of the element can be seen under the border. We will be using the border element later when we introduce additional pages to our website.

### Margin

The Margin clears an area around the border of the element. It creates space between the border and adjacent elements, essentially pushing elements away from each other. Many HTML tags have margins applied by default. For example, `<p>` (paragraph) tags margins on the top and bottom, which is used to visually separate them. The margin is transparent.

There are some additional CSS elements that can be considered standard practices for styling a page. When you examine websites on the internet, most sites are centered in the middle of the screen and adjust as necessary based on the browser size. As you shrink or increase the browser window, the page renders as needed. This is accomplished by using the margin element from the box model. By adding margin-left and margin-right CSS rules to the wrapper, our web page will automatically be centered in the middle of the browser.

### Add Margin to the Wrapper Div

We want to make sure that our page sites in the middle of the browser when it is being viewed. This will be done by adding an auto-margin CSS property to the wrapper div.

1. Choose #wrapper in the CSS Selector panel.
2. Navigate to the margin box diagram under the Layout category.
3. Click on the units "px" and Choose Auto for Margin from the drop-down menu for both the Right and the Left sides.
4. Save all of your changes by going to File > Save All.
Preview your site pages in a browser (File > Preview in Browser or Click the Globe Icon at the top of the Document Window).

You should see that our page is centered in the middle of the web browser. As you adjust the window size of the browser, the margins will automatically be refined so that the page stays in the middle of the browser window.

Congratulations! You've just finished building a complete website using Dreamweaver! While we covered most of the techniques needed in order to create and style a basic site, there's so much more that we could do. Since we don't have time to cover it all in class, we've included a few bonus sections in this manual:

- Page Templates, p. 37-44
- Embedding External Content, p. 45
- Advanced CSS Effects (web fonts, hover effects), p. 46-47
- Getting Your Site Online, p. 48-58

**Bonus: Page Templates**

On many websites, elements such as the header, navigation menu, and footer are repeated on each page. In order to simplify page creation and site maintenance, Dreamweaver offers a helpful template feature. With templates, common elements can be stored in a single, central file, and linked to pages in a site. Changes made to the template file are reflected in all pages linked to it.

### Creating a Page Template

Once you have perfected the layout for a single page in your site, it's easy to create a template that can be used to generate the remaining pages in your site. In this section, we will create a template based on our index.html page and use it to create several new pages.

### Saving an Existing Layout as a Template

In this exercise, we will save our index.html page as a Dreamweaver Template (.dwt) file.

1. In the top menu bar, Navigate to File > Save as Template...
2. In the Save as Template window that opens, Enter the following settings:
   - **Site:** Dreamweaver
   - **Existing Templates:** (no templates)
   - **Description:** optional - Type a brief description of the template for your own personal reference
   - **Save As:** main-template (this is the template file name - web file naming rules apply)
3. Click Save
4. A window asking whether you would like to update links will open; Click Yes. Dreamweaver will now create a template file named main-template.dwt, which will be placed inside a new folder named Templates.

### Adding Editable Regions to a Template

After creating a template file, "editable regions" must be added. Editable regions define which parts of a page created from a template can be modified. In this exercise, we will add an editable region to our main-template file.
We want to be able to edit the content in the white area, inside the “content” div, so we will place an editable region there. First, however, we need to remove the existing text.

1. In the Code View, Highlight the code between `div id="content"` and its corresponding `</div>` tag.

2. Press the Delete key on your keyboard.

3. Leave your cursor positioned between `div id="content"` and `</div>`. In the top menu bar, Navigate to Insert > Template > Editable Region.

4. In the New Editable Region window that opens, Type “Main Content” in the Name field and Click OK.

5. In the Design View, the Editable Region appears with the region name as placeholder text.

6. In the Code View, the Editable Region appears as an HTML comment

```
<%-- Editable region name: Main Content --%>
```

7. Save your template file (File > Save or Command/Control + S)

Creating New Pages from a Template

After a template has been created, it can be applied to existing pages or used to generate new pages. In this section, we will create the remaining pages for our site using the template we built in the previous section.

Creating the Portfolio Page

In this exercise, we will use our template to create new page that will contain portfolio content.

1. In the top menu bar, Navigate to File > New...

2. In the New Document window that opens, Click “Site Templates” on the left side.
.dwt files stored in the Templates folder of your site are displayed in the center area of the New Document window. In this case, we only have one template (main-template), so Dreamweaver automatically selects it.

Important! Make sure that “Update page when template changes” is Checked, then Click Create. Dreamweaver will create a new HTML file that looks just like the template file, except that all areas not marked as Editable Regions are locked.

In the top menu bar, Navigate to File > Save. In the Save As window, Enter the following settings:

- **Save As:** portfolio.html
- **Where:** dreamweaver (your site files folder)

Leave the remaining settings as they are and Click Save. Verify that you have saved the page in the correct location by checking the Files Panel; a new file named portfolio.html should appear there.

Dreamweaver adds a generic title to pages created from a template; it’s important to change this title to something more meaningful. In the Title Box at the bottom of the Document Window, inside of the Properties panel, Select the existing text and Type “Portfolio.”

Next, we’ll add some text to the main content area of this page. In the Design View, Highlight the placeholder text that reads “Main Content” and Type “My Great Portfolio of Work.”

This line of text will be the main heading for the page, so we will change its format to Heading 1. In the Design View, Highlight “My Great Portfolio of Work.”

Navigate to the Properties panel at the bottom of the screen and make sure HTML is selected on the left side of the panel. This is the default.

From the Format drop-down menu, Choose Heading 1.

Save your file (File > Save or Command/Control + S). Leave the file open; we will come back to it in a few minutes to add more content.

Creating the Contact Page

In this exercise, we will follow the same general process as the previous exercise to create a new page that will contain contact information.

In the top menu bar, Navigate to File > New...

In the New Document window that opens, Click “Site Templates” on the left side.

As in the previous section, Dreamweaver automatically selects the main-template file. Make sure that “Update page when template changes” is Checked, then Click Create.

As in the previous section, Dreamweaver automatically selects the main-template file. Make sure that “Update page when template changes” is Checked, then Click Create.

- **Save As:** contact.html
Where:
dreamweaver(your site files folder)

Leave the remaining settings as they are and Click Save. Verify that you have saved the page in the correct location by checking the Files Panel; a new file named contact.html should appear there.

In the Title Box at the bottom of the Document Window, inside of the Properties panel, Highlight the existing text and Type “Contact Me.”

In the Design View, Highlight the placeholder text that reads “Main Content” and Type “Contact Me.”

Again, this line of text will be the main heading for the page, so we will change its format to Heading 1. In the Design View, Highlight “Contact Me.”

Navigate to the Properties panel at the bottom of the screen and make sure HTML is selected on the left side of the panel.

From the Format drop-down menu, Choose Heading 1.

On the design side, Click to the right of “Contact Me” to place your cursor right next to the “e” in “Me” and Press Enter to go down to the next line.

In the Insert panel, Choose Structure from the dropdown menu at the top of the panel, and Click to insert an unordered list.

After the first bullet, Type: “Email me at your-email[at] address.com,” then Press Enter to create another bullet.

After the second bullet, Type: “Follow me on Twitter @ YourTwitterNameHere,” then Press Enter to create another bullet.

After the third bullet, Type: “Come visit me at my office using the provided map.”

Save your file (File > Save or Command/Control + S). Leave the file open; we will come back to it in a few minutes to add more content.

Applying Templates to Existing Pages

Though we created our template using the index.html page, the template is not yet linked to that file. This means that any changes to the template would appear on the Portfolio and Contact Me pages, but not on the Home page. To fix this problem, we need to apply our template to index.html. In this section, we will learn to apply a template to an existing page.

Applying Templates Using the Assets Panel

There are several methods of applying a template to an existing page. In this exercise, we will apply a template to our index.html using the Assets Panel.

If your index.html is not already open, Navigate to the Files Panel and Double-Click the index.html file.

On the right side, next to the Files tab, Click the tab labeled “Assets,” to switch to the Assets Panel. If the tab is not present, navigate to Windows>Assets and drag it into place.

Applying Templates to Existing Pages

Though we created our template using the index.html page, the template is not yet linked to that file. This means that any changes to the template would appear on the Portfolio and Contact Me pages, but not on the Home page. To fix this problem, we need to apply our template to index.html. In this section, we will learn to apply a template to an existing page.

Applying Templates Using the Assets Panel

There are several methods of applying a template to an existing page. In this exercise, we will apply a template to our index.html using the Assets Panel.

If your index.html is not already open, Navigate to the Files Panel and Double-Click the index.html file.

On the right side, next to the Files tab, Click the tab labeled “Assets,” to switch to the Assets Panel. If the tab is not present, navigate to Windows>Assets and drag it into place.
On the left side of the Assets Panel, Click the Templates button (2nd button from bottom). Any files stored in the Templates folder for the site will appear.

main-template should be selected (highlighted in gray) by default, since it is the only file in the Templates folder. If it is not, Click main-template to select it.

To apply the template to the current HTML document, Click Apply in the lower–left corner of the Assets Panel.

Assigning Existing Content to Template Regions

In this exercise, we will specify how different parts of the index.html document will be placed into the template.

1. After applying the template, a window asking where in the template you would like to place different parts of the index.html document will appear.

2. Click Document Body, then, from the “Move content to new region” drop-down, Choose Main Content. This tells Dreamweaver to move everything between the tags in index.html to the Main Content Editable Region of the template. Since some items in the of index.html are already included in the template, we will need to do a little editing once we finish applying the template.

3. Click Document Head, then, from the “Move content to new region” drop-down, Choose Nowhere. This tells Dreamweaver to disregard everything between the tags in index.html and use only the already included in the template.

4. Click OK.

5. As mentioned in step 6, there is some duplication that we need to fix. In the Code View, Highlight all the code between >!- - InstanceBeginEditable name="Main Content" -- and >hl<WELCOME TO MY PORTFOLIO WEBSITE!/hl< (not including these two lines), then Press the Delete key on your keyboard.

6. In the Code View, Scroll down to the bottom of the page.

7. Highlight all the code between >/ul< and >!-- InstanceEndEditable --< (not including these two lines), then Press the Delete key on your keyboard.
Editing Templates

Now that we have created a template and applied it to all of the pages in our site, we can make site wide changes to the header, navigation menu, and footer by editing just the template file. In this section, we will learn how to edit a template file.

1. If your main-template.dwt file is not already open, Navigate to the Files Panel, Click the black arrow next to the Templates folder, and Double-Click the main-template.dwt file.


3. Type “2015”

4. Save your template file (File > Save or Command/Control + S).

5. A window prompting you to update files that use this template will appear; Click Update.

6. The update process may take a few seconds. When complete, the Update Pages window will display “Done” in the lower left corner.

7. When updates are complete, Click Close.

8. Important! When files are updated based on a template, changes to the individual files are not saved automatically. To save the changes, Navigate to File > Save All in the top menu bar.

9. Save your index.html file (File > Save or Command/Control + S).
To verify that your template changes have been applied, Preview your site pages in a browser (File > Preview in Browser or Click the Globe Icon at the top of the Document Window).

**Bonus: Embedding External Content**

Many popular web services offer content that can be embedded into your own site, just by copying and pasting some code. In this section, we will embed a Google Map into our contact page. The same general process can be followed to embed content from other sites/services into your site, but be sure to read the directions provided by the sites/service in case additional steps are needed.

**Google Maps**

**Embedding a Google Map**

We are going to embed a Google Map into the page. This will be done by pasting code that was provided by Google on their Google Maps site.

1. In embed.txt (located in the downloaded class files), Select all of the code for the Google Map starting with >iframe width.... > and ending with >/small<.
2. Copy the code and switch back to contact.html.
3. On the code side, Click your cursor after the closing >/ul< tag and Paste the code.
4. Save and Preview your change in the browser.

You should have a functional Google Map on your contact.html page. If you wanted to change how this map sits on your page, you should create a div for the map and apply CSS rules. Embedding content is very easy as long as you have a basic understanding of HTML code. You could follow these same steps outlined in this course to embed other content such as a Facebook Like button or a YouTube video.

**Bonus: Advanced CSS Effects**

CSS, particularly the latest version - CSS3, allows web designers to incorporate many advanced effects into their designs. In this section we’ll explore just a few of these features: Web Fonts, Pseudo-classes, and Transitions.

Note that CSS3 features are not yet supported consistently by all web browsers. Check the W3 Schools CSS Reference to determine which browsers support CSS3 features that you’re interested in using:

http://www.w3schools.com/cssref

Another good reference for web development is the Mozilla Developer Network, or MDN.

https://developer.mozilla.org

**CSS3 Web Fonts**

CSS3 has created additional flexibility for web designers with web fonts. Prior to CSS3, web designers had to rely on fonts that were already installed on the users’ computers. With CSS3, designers have many more font options. Web fonts can be purchased
online; however, Google provides a wealth of free web fonts that can be used for your design purposes: http://www.google.com/fonts

Adding a Google Web Font

In this exercise, we'll add a CSS3 Web Font provided by Google to modify our navbar links.

1. In embed.txt (located in the downloaded class files), Select all of the code for the CSS 3 Web Font starting with >link href="... and ending with type='text/css'".
2. Copy the code (Edit > Copy or Command/Control + C)
3. At the top of the Document Window, Click main-template.dwt to switch to your template file. If your template is not already open, Navigate to the Files Panel, Click the black arrow next to the Templates folder, and Double-Click the main-template.dwt file.
4. On the code side, Find the beginning of the >head< tag at the top of the page.
5. Paste the code copied from embed.txt in the first line after the start of the >head< tag.
6. In the upper-left corner of the document window, Click screen.css
7. Find the CSS rules for #navbar ul.
9. After that, Add “font-size: 24px;”

```css
#navbar ul {
    list-style-type: none;
    padding: 0px;
    text-align: center;
    font-family: 'Gorditas', cursive;
    font-size: 24px;
}
```

10. Save all your files by navigating to File > Save All. If your template updates all the HTML files, be sure to Save again.
11. Preview your changes in the browser.

Additional Web Font Sources

In addition to Google, many other sources for web fonts are also available (although some cost money!). A few popular providers are:

- **Font Squirrel**: http://www.fontsquirrel.com/
- **The League of Moveable Type**: http://www.theleagueofmoveabletype.com/
- **Adobe Typekit**: https://typekit.com/foundries/adobe

Hover Effects with Pseudo-classes

CSS pseudo-classes are used to add special effects to some types of tags and/or selectors. For example, a pseudo-class is commonly used to apply a hover effect when a user places their cursor over a link.

Adding a Hover Effect
In this exercise, we'll use a pseudo-class to add a hover effect to the links in the navbar div. This will cause the links to turn white when a user hovers over them.

1. Under the Sources section, Select screen.css.
2. Create a new CSS rule by selecting the icon in the CSS Selector panel.
3. Type #navbar a:hover to select the hover state of links inside the navbar div.
4. Choose a font color of white.
5. Save and preview your changes in a browser by hovering over the links in the navbar div.