Creating Accessible Web Pages with Adobe Dreamweaver

Introduction

The following tutorial takes a *structure-first* approach to web design. Although visual formatting is essential to every publication, in this tutorial it is treated as a secondary consideration to be addressed *after* the content has been marked up with structural HTML tags.

A structure-first approach, which may seem unintuitive to visually-oriented designers, offers several benefits. The first is pedagogical: learners are forced to conceptualize web design as two separate but interrelated steps. The second benefit is practical: good web design is built on a foundation of solid HTML code, and valid HTML requires a separation between document structure using HTML tags and visual presentation using Cascading Style Sheets.

Approaching web design from a structural perspective ensures that content will remain readable even when styling rules are disabled. Handheld devices, like cell phones, typically ignore visual formatting, as do assistive technologies like screen readers. A structural focus also leads to the creation of more flexible web pages, which can be viewed by users (and reformatted by designers) in multiple ways.

This tutorial is comprised of a series of lessons which build on one another to culminate in a simple, accessible website. Along the way, many principles of universal design and best practices are employed and explained. The tutorial was developed in collaboration with the office of Computer Training and Support Services (CTSS) at Colorado State University. It grew out of a desire to incorporate universal design principles and emerging web design standards into the University's basic training program for web design using Dreamweaver software.

Technical Requirements

These lessons were designed for use with Dreamweaver MX (version 8). Other, especially older, versions of the software may not correspond to the instructions and illustrations provided.

At key points, you will be asked to work with a new set of files to ensure that your work matches the descriptions and illustrations in the text. To switch to a new set of files, simply modify Dreamweaver's site settings to point to the appropriately numbered folder. The folder numbers are indicated in the text with red icons as follows:

①②③, etc.

Objectives

Upon completion of this module, the user will:

- Understand the importance of adhering to web standards and universal design practices
- Understand the importance of designing for diverse technologies and user requirements by adhering to web standards and universal design practices to maximize accessibility and usability
- Understand the importance of separating content from appearance (structure and function vs. visual design)
- Create a simple, standards-compliant website using XHTML, <div> tags, and cascading style sheets (CSS)
- Create rules for a cascading style sheet and understand the following:
    - CSS rule construction (syntax)
    - Three possible locations for CSS rules
    - The meaning of the term "cascade"
    - CSS positioning of page elements
    - Media-specific CSS, such as "screen" and "print"
- Review guidelines for using JavaScript and the importance of the <noscript> tag
- Create a Dreamweaver template and generate a website from it
- Understand the difference between fixed-width and "liquid" layouts, and the pros and cons of each approach
- Create a more complex 3-column layout using the above techniques
- Know where to find web development resources at CSU and elsewhere on the Web
- Understand the importance of validating HTML and CSS code prior to duplicating it throughout your website

Table of Contents

(1)

Download and Extract Work Files

Begin by downloading and extracting the work files used for this tutorial. The files, which are stored in the compressed, .zip file format, are located on the ACCESS Project website at http://accessproject.colostate.edu/udl/modules/html/tut_dw.cfm?display=pg_5. (Depending on your browser, you may need to open the .zip file(s) manually and extract their contents to a new location on your hard drive.) Once the files have been extracted, you may begin the tutorial.

Define a New Website

1. Begin by choosing Site > New Site…
2. Click on Advanced tab at the top
3. Enter "Practice 1" for the site name.
4. Click on the Browse button (the folder icon) to point to the folder titled "01_Begin" that contains the work files you downloaded and extracted. This will become the "local root folder" of your website.
5. Click on the Browse button (the folder icon) to point to the default images folder (this the folder titled "images" located inside the local root folder (01_Begin\assets\images).
6. Make sure the "Links relative to: Document" option is selected.
7. Make sure the "Enable cache" option is checked.
8. Click OK to complete the site definition.

Site Definition for Practice

| Basic | Advanced |

Category
- Local Info
- Remote Info
- Testing Server
- Cloaking
- Design Notes
- Site Map Layout
- File View Columns
- Contribute

Local Info

Site name: Practice

Local root folder: C:\Userdata\DW8_Practice\

☑ Refresh local file list automatically

Default images folder: C:\Userdata\DW8_Practice\assets\image:

Links relative to: ◉ Document ○ Site root

HTTP address: http://

This address is used for site relative links, and for the Link Checker to detect HTTP links that refer to your own site

Case-sensitive links: ☐ Use case-sensitive link checking

Cache: ☑ Enable cache

The cache maintains file and asset information in the site. This speeds up the Asset panel, link management, and Site Map features.

[OK] [Cancel] [Help]

In Dreamweaver's Files panel, note the new site structure:

Practice (Root)
```
├─ assets
│      ├─ css
│      ├─ images
│      └─ javascript
├─ Resources
│      └─ various Word and Excel documents
└─ Templates
       └─ currently empty
```

Create the basic page layout

Create a new web page.

1. Choose File > New…
2. Select the General tab
3. Select Category: "Basic page," and Basic page: "HTML"
4. Make sure the Document Type (DTD) is "XHTML 1.0 Transitional"
5. Click Create

Give the page a title

Each web page has a `<title>` tag. This tag is used by web browser software to identify and bookmark the page, so it's important how you name your document. The best practice for naming pages is to put repeating information at the end of the title. For example, the title

Faculty Information – Civil Engineering Department – Colorado State University

will yield a more useful bookmark than

Colorado State University Department of Civil Engineering – Faculty Information

1. In the Title box at the top of the screen, enter the following:
   Computer Training and Support Services - Colorado State University
2. Press the Enter key.

Save the document

Save the page as "design.html" This page will become the template; it will not be used *per se.*

Create the four primary "divisions"

It is important to recall that HTML is a "markup language." Markup provides structure and order to your content. For example, the `<p>` tag breaks text up into blocks that are interpreted by the web browser as paragraphs. Similarly, the <h1> tag indicates a level 1 heading, the `<a>` tag creates an anchor or link, and so on.

Another markup tag of great importance is the `<div>` tag. A "div" divides your document into logical divisions or sections. For training purposes, we will be using four main `<div>` tags. Keep in mind that a web page can contain any number of `<div>` tags and/or nested `<div>` tags.

We will create a `<div>` tag for each of the following:

1. header
2. navigation
3. content
4. footer

A `<div>` tag can be inserted in several ways. One method involves Dreamweaver's Insert menu:

Insert > Layout Objects > Div Tag.

Another method is to click the Insert Div Tag button on the Layout tab or menu of the Insert toolbar.
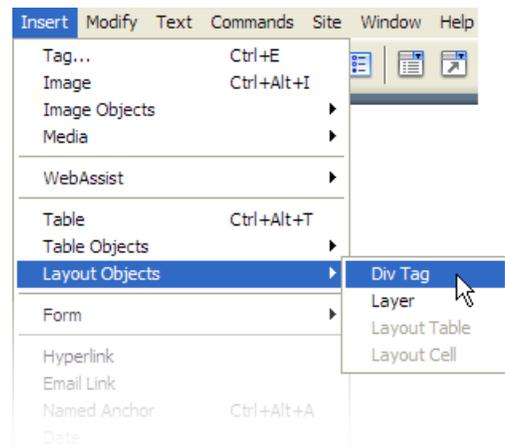
For this example, let's use the Insert Div Tag button from the Insert Toolbar.

Insert div "header"

Click on the Insert Div Tag button and enter the following information in the dialog box:

> **Insert: At insertion point**
> **ID: header**
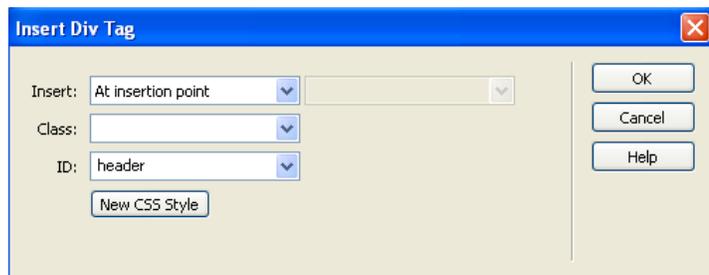
Click OK.

Note: The `<div>` tags we will create all have a unique identifier or "ID." This ID is used to connect page elements with cascading style sheet (CSS) rules. IDs can also be used by JavaScript to create interactive behaviors, and by hyperlinks to target a specific part of the page.

At this point, your page should look like this:
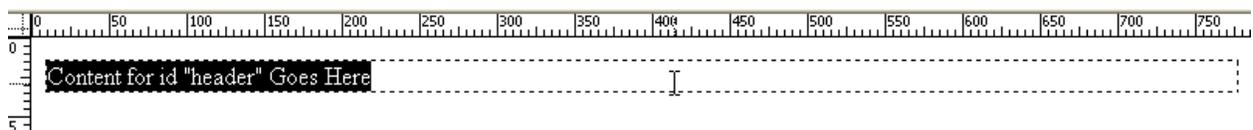
Insert the "navigation" div tag

Click on the Insert Div Tag button and enter the following information in the dialog box:

> **Insert: After tag <div id="header">** ◄┄┄┄┄┄┄┄┄┄┄ Note the "After tag" option!
> **ID: navigation**

Insert the div tags for "content" and "footer"

Repeat this process for the remaining two `<div>` tags, "content" and "footer."

For each new div, be sure to select **Insert: After tag** along with the previous ID name. For example, when creating the div "content," choose "Insert: After tag <div id="navigation">."

Once all four divisions have been created, your page should look like this:

```
Content for id "header" Goes Here
Content for id "navigation" Goes Here
Content for id "content" Goes Here
Content for id "footer" Goes Here
```

Insert additional <div> tags into the header

The header will contain two graphics: the CTSS logo and the CSU logo. Each graphic requires its own formatting, so we will wrap each it in its own `<div>` tag.

Click on the Insert Div Tag button and enter the following information in the dialog box:

> **Insert: before end of tag <div id="header">**
> **ID: ctsslogo**

Note "before end of tag" insert option!
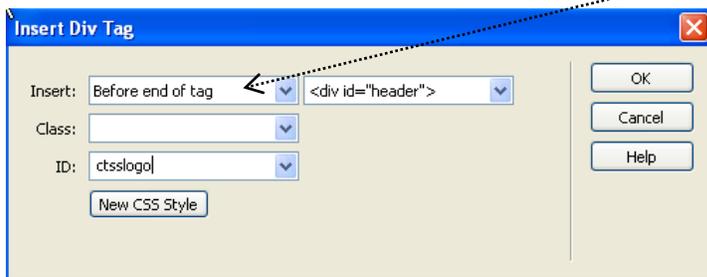
For the second logo, repeat with the following information:

> **Insert: before end of tag <div id="header">** ◄┄┄ Note the "before end of tag" option!
> **ID: csulogo**

When complete, your page should look like this:

```
Content for id "header" Goes Here
Content for id "ctsslogo" Goes Here
Content for id "csulogo" Goes Here
Content for id "navigation" Goes Here
Content for id "content" Goes Here
Content for id "footer" Goes Here
```

In Dreamweaver's Design view it is impossible to see how the divs "ctsslogo" and "csulogo" are nested inside the div "header." To see the hierarchy of `<div>` tags more clearly, switch to Code view, as shown below:

```
<> Code    Split    Design
```

The `<div>` tags should look like this:

```
<div id="header">Content for  id "header" Goes Here
  <div id="ctsslogo">Content for  id "ctsslogo" Goes Here</div>
  <div id="csulogo">Content for  id "csulogo" Goes Here</div>
</div>
<div id="navigation">Content for  id "navigation" Goes Here</div>
<div id="content">Content for  id "content" Goes Here</div>
<div id="footer">Content for  id "footer" Goes Here</div>
```

Notice how the two logo `<div>` tags appear between the opening and closing tags for the header.

Switch back to Design view to continue.
Save the page
Choose File > Save (Ctrl+S). Give the page the name "design.html"

Hey, where's all the *style*?

Admittedly, the page doesn't look like much at this stage! That's because we have focused our efforts on creating a useful structure rather than an attractive visual appearance. Aesthetics certainly are important, and in the next section we will spend time working with cascading style sheets (CSS) to make the page look more pleasing. Bear with us as we complete this phase of the tutorial!

Adding Content to the <div> tags

Insert logos into the header

1. Select the text "Content for id "ctsslogo" Goes Here" (triple-click to select the text)
2. Press the Delete key to remove the text
3. Select "Common" from the Insert toolbar menu (or the tab "Common" if in tab view)

4. Click the Insert image icon ▣
   a. Image: assets/images/ctss_logo.gif
   b. Alt text: "Computer Training and Support Services home page"
5. Repeat same steps for CSU logo
   a. Image: assets/images/csu_logo.png
   b. Alt text: "Colorado State University home page"

Your page should now look like this:

Note on alternative text: Each of the two images will serve as a link to another page. Therefore our alternative text refers to the links' destinations rather than to the graphics themselves. It's the *intent* of the images on the page that matters, and in this case they are pointers to other pages.

Add links to header graphics

The CTSS logo
1. Open browser and copy the URL of the CTSS home page: http://www.ctss.colostate.edu/.
2. Return to Dreamweaver
3. Select the CTSS logo
4. Click in the link field
5. Paste the URL
6. Press Enter
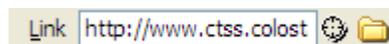
The CSU logo
1. Repeat these steps to obtain the link for the CSU logo. The URL is: http://welcome.colostate.edu/

Add the navigation list

1. Select the text "Content for id "navigation" Goes Here" (triple-click to select the text)
2. Type the following:
   a. Home
   b. Services
   c. Staff

       d. Training

       e. FAQ

       f. Final

3. Select the navigation items and turn them into an unordered (bulleted) list.

       a. Click on the Unordered List button in the Properties Inspector:

Create links for the unordered list items

Give each menu item a link to a future page:

1. Select each word in the list
2. Locate the Link field in the Properties Inspector
3. Enter the corresponding file name for the selected menu item from the following list:

       a. index.html

       b. services.html

       c. staff.html

       d. training.html

       e. faq.html

       f. final.html

4. Press the Enter key and repeat for each item in the menu (Note the .html file extension)
5. Save changes

Add the footer text

1. Return to the browser and navigate to the CSU home page: http://welcome.colostate.edu/
2. Copy the following line of text from the page footer:

Apply to CSU | Contact CSU | Disclaimer | Equal Opportunity | Privacy Statement

Colorado State University-Pueblo | Colorado State University-Global Campus | Colorado State University System
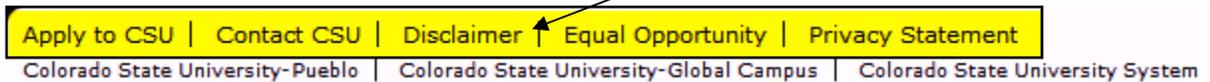
3. Return to Dreamweaver
4. Select the text "Content for id "footer" Goes Here" (triple-click)
5. Paste the text from the CSU home page. (Notice how Dreamweaver preserves the links for you.)
6. Save changes

Add dummy content

1. Open the Word document "Lorem ipsum.doc" in the folder "Resources"
2. Copy the first two paragraphs
3. Close the Word document
4. Return to Dreamweaver
5. Select the text "Content for id "content" Goes Here" (triple-click)
6. Paste the text from Word
7. Insert the cursor after the words "Lorem ipsum" and press Enter to start a new paragraph
8. Position the cursor in the first line—"Lorem ipsum"—and select "Heading 1" from the Format menu of the Properties Inspector

Format Heading 1
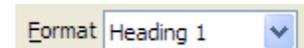
Add the "Skip to main content" link

1. Select the text "Content for id "header" Goes Here" (triple-click)
2. Press Enter (By pressing the Enter key you wrap the text in a `<p>` tag, which help us format the text.)
3. Type "Skip to main content"
4. Select the new text (triple-click)
5. Enter "#content" into the Link field.

   Link [#content   ▼]

   This is the same as linking to an anchor on the page, except in this case we're linking to a specific `<div>` element with an ID of "content."
6. Press the Enter key.
7. Save changes

Preview in a web browser

Preview the page in either Internet Explorer or Firefox (F12 or Ctrl+F12). Remember, we're focusing on structure, not style!

Link to default style sheet

1. Click the "Attach Style Sheet" button in CSS Panel
2. Browse for file: assets/css/default.css. Click OK.
3. Make sure "Add as Link" is selected
4. Set media type to "screen"
5. Click OK
6. The resulting link will appear as follows in Code view:
   ```
   <link href="../assets/css/default.css" rel="stylesheet" type="text/css" media="screen" />
   ```

Link to navigation style sheet

1. Repeat steps above to link to file: assets/css/navigation_vertical.css
2. Make sure the same "Add as Link" and "Media: screen" options are selected
3. In Code view, the resulting link should look like this:

   ```
   <link href="../assets/css/navigation_vertical.css" rel="stylesheet" type="text/css" media="screen" />
   ```

4. Let's give this style sheet a title. In the new `<link>` tag, locate the words '`media="screen"`' and click after the last quotation mark. Type the letter "t" and notice how Dreamweaver prompts you with a list of possible attributes for the `<link>` tag. Press Enter to accept the title attribute. The cursor should be within the double quotation marks. Type the words "Vertical Menu." The complete tag should look like this:
   ```
   <link href="../assets/css/navigation_vertical.css" rel="stylesheet" type="text/css" media="screen" title="Vertical Menu" />
   ```
5. Save changes

Finally, switch back to Design view. Your page should now look like the image at right:

Combining <div> tags and CSS

Creating a website using `<div>` tags and CSS helps keep a page's structure separate from its appearance. This is an important principle of modern web design because it simplifies the design process and facilitates subsequent formatting modifications. It also allows for maximum flexibility in how your page can be displayed using different web technologies.
Remember, your page should still be understandable without CSS formatting!

Metadata

Meta tags are the information inserted into the "head" area of a web page. Except for the title tag, information in the head area of a web page is not seen by viewers. Instead, meta information is used by search engines to catalog your page and by the browser to adjust its display settings. For example, meta tags can tell a browser what "character set" to use (e.g., Western European languages, for example) or a search engine how your page should be described in its summary.

To add metadata to your page, choose Insert > HTML > Head Tags > Meta.
Here's an example of a meta tag:
- Attribute: Name
- Value: description
- Content: A practice website modern, standards-compliant design techniques

Save changes and preview in a web browser

"Validate Before You Replicate"

Our basic design page is complete, but before we convert it into a template, and thereby replicate its code to all of the pages of our website, we need to check for any errors in the code that might prevent it from displaying properly. We can check the code in Dreamweaver using the following command:

File > Check Page > Validate Markup (Shift+F6)

The results should say "No errors or warnings found…" However, if you receive an error, simply double-click on it and Dreamweaver will take you to that line of code in the document. Make the necessary changes and save your page.

There are additional tests to check the validity of the XHTML and cascading style sheet code, but these require your page to be posted to a web server.
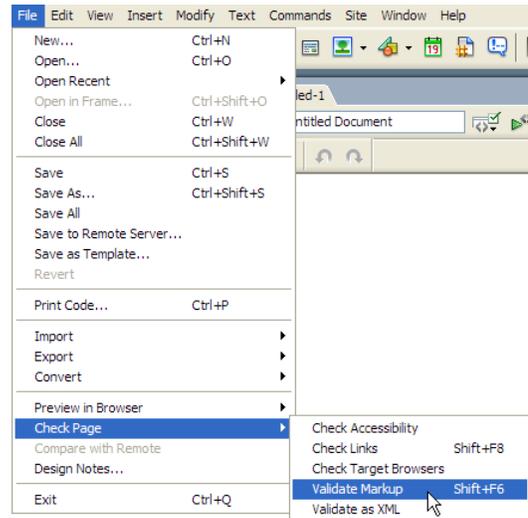
② Templates

Dreamweaver Templates can perform several important functions. Perhaps most important is their ability to "push" design changes out to all of the pages of your website, saving the designer a great deal of time. Another useful feature of Templates is their ability to preserve design consistency from page to page by locking down areas of content that might be changed accidentally. This is especially helpful when multiple "authors" are working on the site.

The process begins by saving a page as a Template. All subsequent pages are created from that template, and they remain linked to it. It is a parent-child relationship. Changes made to the Template (the parent) are passed along to its children (the pages of the website).

Save design.html as a Template

1. File > Save as Template…
2. Update Links: Yes
3. Note the new file name: "design.dwt"
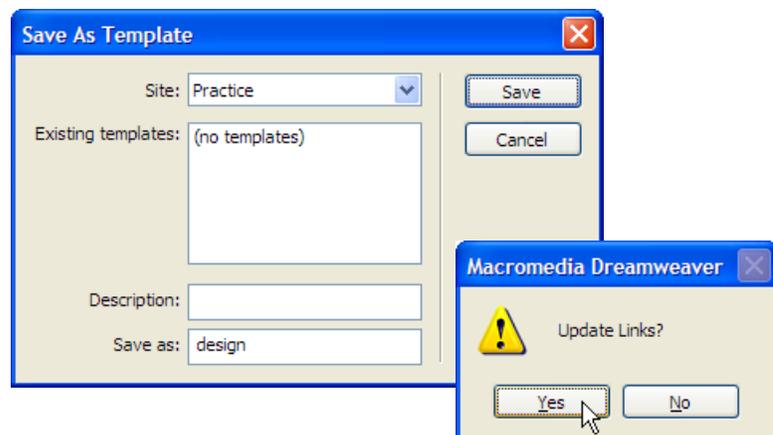
Create an editable region within the template

1. Insert cursor anywhere within the main content area
2. Click on <div#content> on the Tag Inspector bar to select the div "content"
3. Choose Insert > Template Objects > Editable Region (Ctrl-Alt-V)
4. Give the editable region the name "content"
5. Click OK
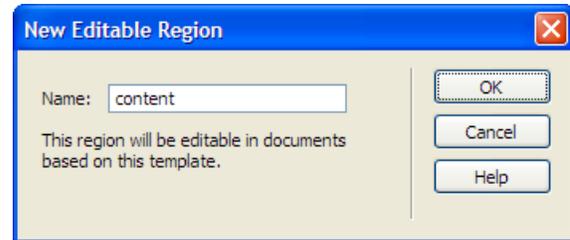
Save and close the Template

Generate pages of website

You will create the following five pages from Template

1. services.html
2. staff.html
3. training.html
4. faq.html
5. index.html

Create a new page from the Template

1. Go to File > New…
2. On the Templates tab, select "Site 'Practice'" from the first list and "design" in the second.
3. Make sure "Update page when template changes" is checked
4. Click "Create"

For each page of the site…

1. Insert the name of each page before the existing title tag

   For example, for the Services page, enter the following:

Services - Computer Training and Support Services - Colorado State University
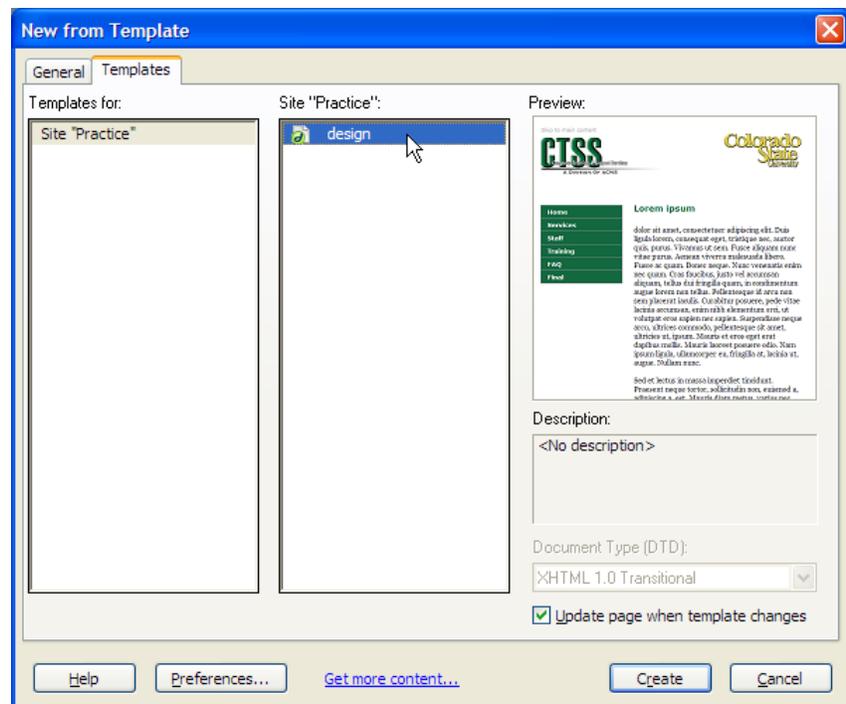
2. Change "Lorem ipsum" (the level 1 header) to match the first word of the page title (e.g., Services)
3. Save the page with the appropriate name (e.g., services.html)
4. Close the page

Repeat for all pages (services.html, staff.html, training.html, faq.html) **except the home page, for which you will use the steps below.**

The home page
1. For the home page, leave the title tag as is. It already says what it needs to.
2. Replace Header 1 with "Computer Training and Support Services"
3. Save as "index.html"
4. Don't close the home page
5. Preview the home page in the browser
6. Test all of the links, including header and footer links. (Note that the link "Final" will not work because we haven't yet created that page. This will come later in the class.)
7. Close the browser window

3 Modify individual pages

Home Page

Replace text
1. Open the page "index.html"
2. Open the Word document "content for web pages.doc"
3. In the Word document, select and copy the text in gray box under "Home Page Content"
4. Return to Dreamweaver, then select and delete the two paragraphs of "greeking"
5. Paste the text from Word into the content area of the web page.

Staff Page

Replace text
1. Open the page "index.html"
2. The Word document "content for web pages.doc" should still be open.
3. Scroll down the page to locate "Staff Page Content."
4. Select and copy the text in gray box under "Staff Page Content"
5. Return to Dreamweaver, then select and delete the two paragraphs of "greeking"
6. Paste the text from Word into the content area of the web page.

Add Photos
1. Insert (i.e., click) the cursor in front of the words "David Greene…")
   a. Take care not to click on the border of the content div area; you will know because diagonal stripes will appear over the navigation menu.
2. Choose Insert > Image and select the file david_thumbnail.jpg
3. For alternate text, enter "David Greene"

4. Repeat for paragraph two, using the photo "Jodie_thumbnail.jpg." For alt text, enter "Jodie Hanzlik"
5. Repeat for the next two paragraphs, using the remaining images in any order
6. The alternative text for each is as follows:
   a. cathy_thumbnail = Cathy Schelly
   b. patty_thumbnail = Patty Stutz-Tanenbaum

Apply custom class (styling rules) to make photos float left
1. Select the first photo, then choose "photo_left" from the Class menu in the Properties Inspector
2. Repeat for remaining photos

Save changes

Preview page in browser

Close page

Q: What makes the text wrap around the photos?
A: The "float" property

When applied to an HTML element, the float property takes the element out of the normal flow of text and aligns it to one side or the other of its parent element. For example, if we apply "float: left" to an image, the image snaps to the left edge of its parent element (in our case, the "content" div) and allows the text that follows to "float" up into the space previously occupied by the image. The text cannot occupy the space taken up by the image's width, however, so the effect is of an image with text wrapped around it, as shown in the example below.
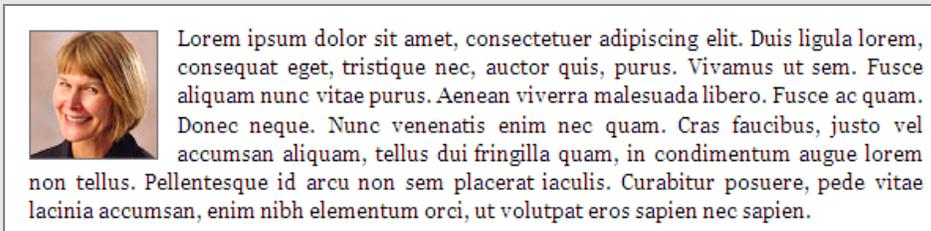
The CSS rule for this image uses a custom class selector called .photo_left. If we wanted the image to appear right-aligned, the selector would be .photo_right. Both are shown below:

```
.photo_left {
   border: 1px solid #666666;
   float: left;
   margin: 4px 12px 0 0;
   padding: 0;
}
```

```
.photo_right {
   border: 1px solid #666666;
   float: right;
   margin: 4px 0 0 12px;
   padding: 0;
}
```

Note: The margin values are important, particularly the left margin setting of 0 (for the left-aligned example). This ensures that the image will snap to the left edge of the page (or any other enclosing element). The margin values on the other side of the photo (12 pixels) control the distance text is "repelled" from the image (the text-wrap effect).

Q: What makes the footer stay at the bottom of the page?
A: The "clear" property

The opposite property of "float" is "clear." The clear property determines whether an element will allow other elements to appear alongside it. For example, `clear: left` will prevent an element from floating up alongside other elements that are set to "float." The effect is to provide "clearance" on one or both sides of the element, thus forcing it down below the floated elements that precede it. The clear property, applied to the div "footer," forces it to appear below the navigation div, which has the float: left rule applied to it.

Training Page

Add table
1. Open file "training.html"
2. Select and delete text below header "Training"
3. Expand folder "Resources" and locate file "data table.xls"
4. Drag and drop the file to the line below the header "Training"
5. At the Insert Document dialog box, check the following selections:
    a. Insert the contents
    b. Text with structure plus basic formatting (bold, italic)
    c. Click OK

Insert Document

How would you like to insert the document?

- ● Insert the contents
    - ○ Text only
    - ○ Text with structure (paragraphs, lists, tables, etc.)
    - ● Text with structure plus basic formatting (bold, italic)
    - ○ Text with structure plus full formatting (bold, italic, styles)
    - ☐ Clean up Word paragraph spacing
- ○ Create a link

Help   OK   Cancel
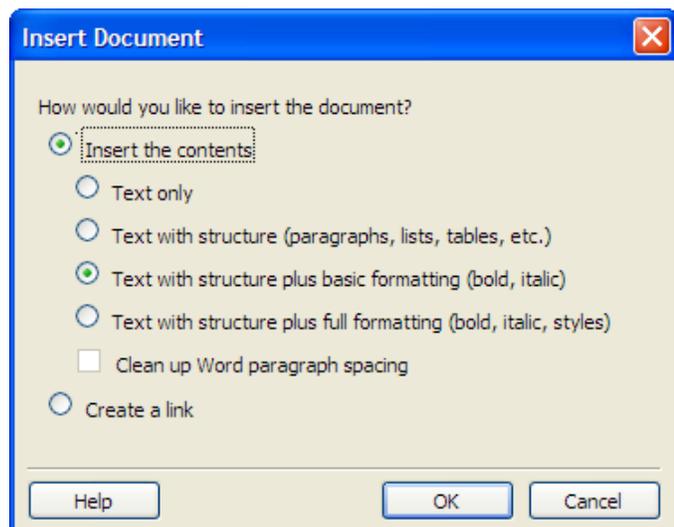
Clean up table
1. Click in first row, first column: "Classes"
2. Note the tag information in the Tag Inspector below.
3. Click on the tag `<table>` to select the entire table
4. Clear height and width settings for the table (these will be governed by the style sheet and should not appear in the HTML code)
    a. Click the button "Clear Column Widths" ⟶
    b. Click the button "Clear Row Heights" ⟶

5. Designate the first row of the table as a header row
   a. Click in the cell "Classes"
   b. Locate the `<td>` tag in the Tag Inspector
   c. Right-click on the `<td>` tag
   d. Select "Quick Tag Editor…"
   e. Press the Backspace key once to remove the letter "d"
   f. Type the letter "h"
   g. Press Enter twice—once to select the `<th>` tag from the list, and again to accept the new tag.
   h. Repeat for the remaining two fields: "Time" and "Location"
   i. Note: the color change is a function of a CSS rule, which has already been created for the `<th>` tag. Likewise, the table expands to the full width of the content area due to a CSS rule for the `<table>` tag, which reads: `width: 100%;`

Save changes and close page

FAQ Page

Replace text
1. Open the page "faq.html"
2. The file "content for web pages.doc" should still be open.
3. In the Word document, select and copy the text in gray box under "Frequently Asked Questions"
4. Return to Dreamweaver, then select and delete the two paragraphs of "greeking"
5. Paste the text from Word into the content area of the web page.

Apply markup
1. Apply Heading 2 markup to first line of text ("Training and Help FAQs")
2. Questions at top become an unordered (bulleted) list (essentially a table of contents for the page)
   a. Select the next 6 lines (each of which ends with a question mark. Do not select the line that begins with the letter "Q.")
3. Click the button "Unordered List" in the Properties Inspector.
4. Apply Heading 3 to each question (lines that begin with "Q.")

Tip: you can speed up this repetitive process by choosing Edit > Redo Text Format (Ctrl+Y) after inserting the cursor in each new line.

5. Add anchors (the `<a>` tag) to each question
   a. Insert the cursor before the letter "Q" of the first level 3 heading created in the previous step.
   b. Select the Common tab from the Insert bar
   c. Click on the button "Named Anchor"
   d. Enter the name "howoften" (no space) and click OK.

e. Continue down the page, inserting a named anchor before each level 3 heading. Use the following names:
- register
- eid
- handouts
- recommendation
- help

6. Link list items in table of contents at top of page to the anchors on the page below.
   a. Select each list item and type the following link location in the Link box of the Properties Inspector:
   - #howoften
   - #register
   - #eid
   - #handouts
   - #recommendation
   - #help

   Tip: Don't forget the pound sign (#) before each anchor name!

7. Link "back to top" lines to top of page (the #header div)
   a. Select each "back to top" line of text
   b. Enter "#header" in the Link box
   c. Press Enter.
   d. Repeat for all "back to top" lines.
   (Tip: You can speed up this repetitive process by selecting "#header" from the drop-down menu of the Link Box.)

8. Apply a custom class (i.e., a CSS style) to each "back to top" paragraph
   a. Click in the "back to top" text
   b. Select the <p> tag in the Tag Inspector
   c. Choose the style "backtotop" from the Style menu in the Properties Inspector
   d. Repeat for each "back to top" paragraph. (Be sure to select the <p> tag for each line of text.)

Note: The little up-arrow graphic that appears automatically in front of the words "back to top" results from a little trick of CSS called "pseudo-elements." See page 3 of the handout "cascading style sheets." (Unfortunately, Internet Explorer 7 still does not support pseudo-elements. It simply ignores them, which in this case works just as well.)

Save and close the page.

Services Page

Replace text

6. Open the page "services.html"
7. The file "content for web pages.doc" should still be open.
8. In the Word document, select and copy the text in gray box under "Services page content"
9. Return to Dreamweaver, then select and delete the two paragraphs of "greeking"
10. Paste the text from Word into the content area of the web page.

Save and close the page.

Modify the template

There are several techniques used to display a page's date of last modification. Some solutions use scripting languages like PHP, ASP, or ColdFusion, which run remotely on the server. The following method, however, runs in the browser using a few lines of JavaScript.

1. First, close all pages that may be open in Dreamweaver
2. Open the template, "design.dwt"

Add an email link to the footer

1. Begin by inserting the cursor after the last word in the footer.
2. Press Enter and type the following: Questions or comments about the website? Please send us an email.
3. Select the text "send us an email" and create a link to `mailto:ctss21@eagle.colostate.edu` using the Link box in the Properties inspector.
   Note: An alternative approach would be to select the text, choose Insert > Email Link, and enter just the email address (i.e., ctss21@eagle.colostate, without the mailto: prefix).

Add a "date stamp" in the footer

1. Switch to Split View and double-click on the word "email" in the last sentence. You should be able to locate the corresponding piece of HTML code in the upper pane.
2. Click after the word "email." in the Design View pane and press Enter. Notice that `<p>` tags were generated in the code above. You should see the following: `<p> </p>`
3. In the Code View pane, select this new line of code. Open the Word document "javascript_mod_date.doc" and copy all of the text. Close the Word document and return to Dreamweaver. Paste the code over the highlighted text. Your code should now look something like the image below:

```
<div id="footer">
  <p><a href="http://search.colostate.edu/index.asp?url=contact_information">Contact CSU</
 | <a href="http://welcome.colostate.edu/index.asp?url=info_csu-equalop">Equal Opportunity
Statement</a></p>
  <p>Questions or comments about the website? <a href="mailto:ctss21@eagle.colostate.edu">
  <script language="JavaScript" type="text/javascript">
    // <!-- Display date stamp and hide javascript
        document.write("<p>This page was last updated " + showDate() );
    // -->
  </script>
  <noscript>
  <p>This script creates a &quot;last update&quot; stamp.</p>
  </noscript>
</div>
</body>
</html>
```

In the Design View pane, the footer should now look like the image below. (Note: You may have to click the Refresh button to view the changes to your page.)

Contact CSU | Disclaimer | Equal Opportunity | Privacy Statement
Questions or comments about the website? Please send us an email
This script creates a "last update" stamp.
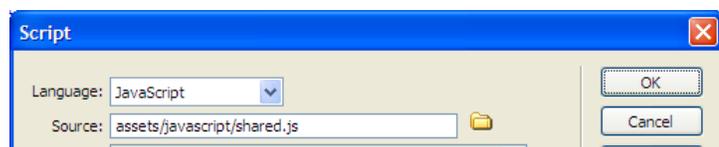
The <noscript> tag

Note the phrase "This script creates a 'last update' stamp." The text comes from the <noscript> tag, which is read by the browser in the event that JavaScript is turned off or is unavailable to the user's device. The <noscript> text is important. It tells users what they're missing if JavaScript isn't running. Note that many handheld devices still do not support JavaScript, and some user who can use it choose not to.

Best practice for JavaScript: Your page should operate well without relying on the capabilities of JavaScript. The rule of thumb is this: Let JavaScript's functionality be the "icing on the cake"—it's nice, but it mustn't be required to use your website.

Link to JavaScript source file

There's still another operation we must perform before our JavaScript code will operate: we must link our page to a JavaScript source file where our script—the "program" that will run when the page is loaded—is located.

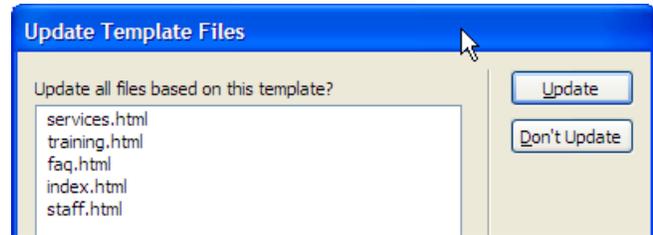1. In the Code View pane of Dreamweaver, scroll to the top of the code and locate the last <link> tag. Insert the cursor at the end of that line and press Enter. Now, from the HTML category of the Insert menu, click on the Script icon.
2. Click on the Browse button (the folder icon) and navigate to assets/javascript/shared.js.
3. Select that file and click OK. The Script dialog box should look like the image at right.
4. Click OK.

The following code should now appear in the head section of your page (ignore the text wrapping in the example):

```
<script language="JavaScript" type="text/javascript"
src="assets/javascript/shared.js"></script>
```

5. Save your page. You will be prompted to update linked pages (Remember, this is the template, and all pages created from it are still linked to it.)

6. Preview it in Firefox (Ctrl+F12). The footer should now show the actual page modification date.

Add links to additional style sheets

Attach the Horizontal Navigation style sheet to the template

1. Click the "Attach Style Sheet" button in CSS Panel
2. Browse for file: assets/css/ navigation_horiz.css
3. Set media type to "screen"
4. Make sure "Add as Link" is selected
5. Click OK
6. Switch to Code view to check the new <link> tag. It should appear as:

```
<link href="../assets/css/navigation_horiz.css"
rel="stylesheet" type="text/css" media="screen" />
```

7. Add the word "alternate" before the word "stylesheet" so that it looks like this:

```
<link href="../assets/css/navigation_horiz.css"
rel="alternate stylesheet" type="text/css" media="screen" />
```

8. Finally, add "title="Horizontal Menu"" before the end-of-tag bracket, like so:

```
<link href="../assets/css/navigation_horiz.css" rel="alternate
stylesheet" type="text/css" media="screen" title="Horizontal Menu" />
```

Note: The change to `alternate stylesheet` tells the browser not to apply the new rules until you say so. Unfortunately, Internet Explorer doesn't support alternate style sheets.

9. Switch back to Design view.

Attach the Print style sheet to the template

1. Using the same steps as above, link to the file labeled "print.css."
2. Important: Select "Print" from the Media drop-down menu

3. It should look like this in the code:

```
<link href="../assets/css/print.css" rel="stylesheet" type="text/css"
media="print" />
```

Note the new media specification: `media="print"`. This tells the browser to invoke the rules of this style sheet when the page is printed. In any other context, the rules from the print style sheet are ignored. This style sheet is an example of "media-specific CSS. Other types of media include screen (computer screens), projection, handheld, aural, and braille.

You can test this new style sheet by previewing the page in either Internet Explorer or Firefox (F12 or Ctrl+F12) and choosing Print Preview. Note the absence of the navigation links and the width of the content div, which now spans the full page.

Return to Dreamweaver and open the print.css file. Notice the selectors it contains and how they were modified from their original forms. These rules supersede those established in earlier style sheets, but they take effect only when the page is printed.

Attach 3-D menus

1. Continue adding the following style sheets, making sure to add the new attributes indicated in bold:

```
<link href="../assets/css/navigation_3-D_vertical.css" rel="alternate
stylesheet" type="text/css" media="screen" title="3-D Vertical Menu" />
```

```
<link href="../assets/css/navigation_3-D_horiz.css" rel="alternate
stylesheet" type="text/css" title="3-D Horizontal Menu" />
```

Save changes and test page

1. You will see an "Update Template Files" dialog box. Click "Update" to apply changes to the pages of the website.
2. Close the template.
3. Open the page "index.html."
4. Press Ctrl+F12 to preview the document in Firefox. (As noted above, this step works only in Firefox, not Internet Explorer.)
5. In the Firefox browser, choose View > Page Style, and select one of the new alternate style sheets to see the effect of applying different styling rules on the same HTML code.
6. Check the Page Preview as well.

Note: Traditionally, this rollover effect would have required JavaScript. We have achieved the same effect using only cascading style sheets!

The horizontal menu style sheet contains many of the same selectors (#content, #navigation, etc.) that appeared in the default.css and navigation.css style sheets. By repeating these selectors and redefining their properties, we have effectively overruled their previous settings.

Tip: Rules contained in the most recent style sheet supersede those that came before. This is how styles can be customized for both on-screen display and for printing, to name just a few of the media-specific options.

We will take advantage of the cascade again in the following exercises.

⑤ Two types of layouts: "Fixed-width" and "liquid"

Let's take a look at the following examples.

"Fixed-width" designs:
- The CSU home page (http://welcome.colostate.edu/)
- Dept. of Food Science and Human Nutrition (www.fshn.cahs.colostate.edu/)
- NPR (www.npr.org/)
- CNN (www.cnn.com/)
- Barnes & Noble (http://www.barnesandnoble.com/)

"Liquid" (or elastic) layouts
- Wikipedia (http://en.wikipedia.org/wiki/Census_Act_1800)
- CSU Occupational Therapy (www.ot.cahs.colostate.edu/)
- Amazon.com (www.amazon.com/)
- Center for Applied Special Technology (www.cast.org/)

Note: Website designs generally fall into one of these two categories. Fixed-width designs tend to emphasize *designer control* over the appearance of the page. Liquid layouts emphasize *user control* of the viewing experience. A few websites do both.

Create a "fixed" layout

Add "wrapper"

1. Open the template "design.dwt," if it is not already open
2. Select the <body> tag on the Tag Inspector bar
3. Insert new div using the following:

   Insert: Wrap around selection
   ID: wrapper

Link to style sheet "fixed_width.css"

1. Click on the Attach Style Sheet button in the CSS panel, then browse for the file "fixed_width.css."
2. Select "screen" from the Media menu to indicate when this style sheet should be invoked by the browser (i.e., when displayed on screen).
3. Click OK. (You should see a change in the appearance of your page!)
4. Switch to Code view and modify the link as follows:

```
<link href="../assets/css/fixed_width.css" rel="stylesheet"
type="text/css" media="screen" title="Fixed-Width" />
```

5. Switch back to Design view

Save and test

1. Save template and update linked pages
2. Test pages in Firefox; be sure to resize window and notice the background gradient that sweeps a dark green at the top to a light gray-green at the bottom. Notice also the static width of the main content area of the page, despite any changes in window size.

The new fixed-width style sheet contains rules for only two selectors, the HTML body element and the ID wrapper:

```
body {
        background: url(../images/background.gif) #BFCAC1 repeat-x;
        margin: 0;
        padding: 0;
}

#wrapper {
        background: #FFFFFF;
        border: 1px solid #000000;
        margin: 0 auto;
        padding: 0 2% 2%;
        width: 700px;
}
```

Tip: How was such a dramatic change achieved with so little code? First, remember that we wrapped the contents of our page in a new div called "wrapper." This div has a fixed width of 700 pixels. It also has left and right margins set to "auto," which effectively centers the div element on the page. The wrapper div has a one-pixel black border and a 2% padding space between the border and the contents within. Finally, it has a white background to "cover over" the green background of the body element beneath it.

The body selector, which originally appeared in the default style sheet, has reappeared in this style sheet in order to redefine its appearance. This rule changes the background color to #BFCAC1 (the light green seen at the bottom of the page) and adds a background image (the dark-to-light green gradient), which repeats as many times as needed to fill the canvas from left to right. Finally, because we want the body color and image to be flush with both left and right sides of the canvas, we have reset the margins and padding to zero.

Create a "liquid" layout

Just for fun, let's add a alternate style sheet that converts the new fixed-width layout to a "liquid" design.

1. In the File Inspector panel, locate the folder of style sheets (assets/css/)
2. Right-click on the file named "fixed_width.css" and choose Edit > Duplicate (Ctrl+D) from the pop-up menu.
3. Rename the duplicate file "liquid.css"
4. Locate selector `#wrapper`
5. Change the "width" value as follows: `width: 80%;`
6. Add a "min-width" property as follows: `min-width: 543px;`
7. When complete, the rule should look like this:

```
#wrapper {
      background: #FFFFFF;
      border: 1px solid #000000;
      margin: 0 auto;
      min-width: 543px;
      padding: 0 2% 2%;
      width: 80%;
}
```

8. Link to this new style sheet, "liquid.css"
9. Switch to Code view and modify the new link as follows:

```
<link href="../assets/css/liquid.css" rel="alternate stylesheet"
type="text/css" media="screen" title="Liquid Layout" />
```

10. Locate and delete the links to the following style sheets:
    a. navigation_vertical.css
    b. navigation_horiz.css
    c. navigation_3-D_vertical.css

   The template should now contain only the following css links (although they may appear in a different order:

```
<link href="../assets/css/default.css" rel="stylesheet" type="text/css"
media="screen" />

<link href="../assets/css/print.css" rel="stylesheet" type="text/css"
media="print" />

<link href="../assets/css/navigation_3-D_horiz.css" rel="stylesheet"
type="text/css" media="screen" />

<link href="../assets/css/fixed_width.css" rel="stylesheet"
type="text/css" media="screen" title="Fixed-Width"/>

<link href="../assets/css/liquid.css" rel="alternate stylesheet"
type="text/css" media="screen" title="Liquid Layout" />
```

11. Switch back to Design view
12. Preview the page in Firefox and select the Liquid Layout style sheet.
    (View > Page Style > Liquid Layout).

Create a 3-column layout

So far our page design has featured only a single column of content. While this may suffice for some needs, it will doubtless be too simplistic for others. A multi-column layout is often required to display a larger array of content on a page. Fortunately, such a layout, at least in principle, is not difficult to create.

Familiarization exercise

1. Using handout provided, circle and label the primary divisions of content on the page, including the following:
   a. header
   b. navigation
   c. column1
   d. column2
   e. message box
   f. column 3
   g. footer

Create a new template based on our previous work

1. Open the template "design.dwt"
2. Choose File > Save as Template…
3. Give it the name "3_column"

Alter the markup (the structure) of the new template

1. Make sure you are in either Design or Split View
2. Click in the main content text area
3. Locate the `<div#content>` tag in the Tag Inspector
4. Rename the `<div>` element
   a. Right-click on the `<div#content>` tag and select Quick Tag Editor… from the menu
   b. Type "column1" (no spaces) and press Enter

Add new div elements, one for each column

1. Column 2 (contains main content)
   a. Insert: After tag <div id="column1">
   b. ID: "column2"
2. Column 3 (contains important links)
   a. Insert: After tag <div id="column2">
   b. ID: "column3"

Insert the "messagebox" div that will contain special announcements

1. Insert: After start of tag <div id="column2">
2. ID: "messagebox"

Disable style rendering in Dreamweaver

1. Notice how the text in column1 is covered by the navigation menu. This is because the styling rules that governed the placement and appearance of the div "content" no longer apply now that it has been renamed "column1."
2. View > Style Rendering > Display Styles. (This is the equivalent of disabling style rendering in Firefox. It helps us see the basic structure of the content.)

Get new text for page

1. Open Word file "3-column content.doc"
2. Select and copy the text (the portion with the gray background) for div element "column1"
3. Return to Dreamweaver and select the existing text for column1.
    a. Insert the cursor in the div "content" and choose Edit > Select All (Ctrl+A).
4. Choose Edit > Paste (Ctrl+V) to replace the text
5. Repeat for each of the four divs on the page.

   Note that the div "messagebox" is within the div "column2." This means that when selecting the contents of div "column2," the "Select All" technique will select both divs. Instead of "Select All," simply triple-click on the text "Content for id "column2" Goes Here."

Apply markup to the text

1. Select the first line of each division and set to Heading 1.

Add photos to the section "Department in the News!"

1. Insert the cursor in front of the first word of paragraph one (the text begins "Lorem ipsum…"). Be careful not to select the border of the div!
2. Choose Insert > Image and select the file david_thumbnail.jpg
3. For alternate text, enter "David Greene"
4. Repeat for paragraph two, using the photo "Jodie_thumbnail.jpg." For alt text, enter "Jodie Hanzlik"
5. Repeat for the next two paragraphs, using the remaining images in any order. The alternative text for each is as follows:
    a. cathy_thumbnail = Cathy Schelly
    b. patty_thumbnail = Patty Stutz-Tanenbaum

Apply a style rule (called a "custom class") to each image

1. Select the first photo, then choose "photo_left" from the Class menu in the Properties Inspector
2. Select the next photo and choose "photo_right"

3. repeat and alternate
   Note: You will not see the effect of the applied style until style rendering is re-enabled.

Create some "null links" for Column 3

1. Select a line of text (the first line is "Lorem ipsem dolor")
2. Enter # in the link field and press Enter. (This creates a link to nowhere.)
3. Repeat for 5 or 6 more lines in Column 3

   Tip: You can speed up this repetitive process by doing one of the following after selecting each new line of text:

   a. Select the previously entered link in the drop-down menu of the Link box
   b. Select Edit > Redo Make Hyperlink (Ctrl+Y)

Add sub-menu items

1. Locate the bulleted list that forms the navigation menu
2. Insert the cursor after the word "Training" and press Enter
3. Add the following text:
   a. Workshops
   b. Handouts
   c. Schedule
4. Click on the word "Workshops" and press the Tab key to indent that line
5. Click on the word "Handouts" and press the Tab key twice, to indent the line to the right of "Workshops"
6. Repeat the same for "Schedule"
7. When complete, the list should look like the image at right:

- Home
- Services
- Staff
- Training
  - Workshops
    - Handouts
    - Schedule
- FAQ
- Final

Apply links to each of the new list items

1. For the link locations, simply enter the name of the link in lowercase, plus ".html"
   a. For example, "workshops.html," "handouts.html," etc.

   Note: These links point to pages that do not exist; they're there just to help create the effect of a working menu.

Re-enable Style Rendering

1. View > Style Rendering > Display Styles

Update links to style sheets

1. Link to new style sheet containing predefined rules for each of our new divs. (We will examine these rules in a moment.)
   a. Link to style sheet "3_column.css"
   ```
   <link href="../assets/css/3_column.css" rel="stylesheet"
   type="text/css" media="screen" />
   ```

2. Link to style sheet for 3-D fly-out menu
   a. Link to style sheet "flyout_horizontal.css"

```
            <link href="../assets/css/ flyout_horizontal.css"
            rel="stylesheet" type="text/css" media="screen" />
```

3. In Code view, the link tags should look like this:
```
<link href="../assets/css/3_column.css" rel="stylesheet"
type="text/css" media="screen" />

<link href="../assets/css/flyout_horizontal.css" rel="stylesheet"
type="text/css" media="screen" />
```

Switch back to Design view
Save the template (File > Save or Ctrl+S) and close it

Create a new page from the template

1. Choose File > New…
2. Select the Templates tab, then select the template "3_column"
3. Make sure the "Update page when template changes" option is checked
4. Click Create
5. Give the new page a title: "3-Column Layout"
6. Save the page as "3_column.html"

Preview the page in the browser

1. Experiment viewing it as a static (fixed) layout and a liquid layout.

For more examples of multi-column layout techniques, please refer to the list of resources at the end of this document. Many of these websites offer complete tutorials and links to other resources on related topics.

That's it!

Congratulations. If all went according to plan, you have now completed each of the lessons and built the sample website. You may want link to the final set of files to compare your work to the examples.

Appendix 1: Cascading Style Sheets

What are cascading style sheets?

A cascading style sheet (CSS) is a set of rules that define the appearance of a single web page or an entire website. When you create a style sheet, you specify rules to control the appearance of elements on your page, such as tables, hyperlinks, page margins, and fonts. Style sheets give you more control over the appearance of your page and allow changes to be applied consistently across your site.

"Cascading style sheets [are an] elegant and efficient way to add and manage elements of style (fonts, color, layout) within Web documents"
—Lie and Bos, *Cascading Style Sheets: Designing for the Web*

CSS rules

A *rule* is a statement about one stylistic aspect of one or more HTML elements. A *style sheet* is a set of one or more rules that apply to one or more HTML documents.

Example: The rule below sets the color of all first-level headings `<h1>` to green:

```
H1 { color: green }
```

Breaking it down

Anatomy of a rule

A rule consists of two parts:
- Selector – the part to the left of the curly brackets
- Declaration – the part within the curly brackets

The *selector* is the link between the HTML document and the style. It specifies what elements are affected by the declaration.

The *declaration* is that part of the rule that describes what the effect will be.

Anatomy of a declaration
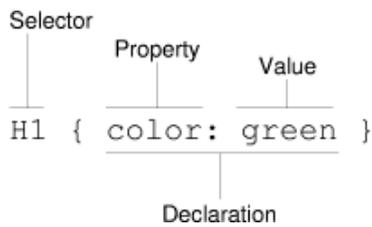
A declaration has two parts separated by a colon:
- Property – the part before the colon
- Value – the part after the colon

The *property* is a quality or characteristic that something possesses. In the previous example, the property is color. Other examples might include line-height, text-align, and font-family.

The *value* is a precise specification of the property. In the example, the value is "green," but it could just as easily have been "blue," "red," "yellow," or some other color.

The diagram below shows a rule with all of its ingredients.

.

Grouping selectors and rules

Style sheets can be reduced in length by grouping selectors and declarations. For example, consider the following rules for headings level 1-3:

```
H1 { font-weight: bold }
H2 { font-weight: bold }
H3 { font-weight: bold }
```

All three elements — `<h1>` through `<h3>` — are set to a font weight of "bold." Since all three declarations are identical, we can group the selectors into a *comma-separated list,* like so:

```
H1, H2, H3 { font-weight: bold }
```

A selector may have more than one declaration. For example, we could write a style sheet with these two rules:

```
H1 { color: green }
H1 { text-align: center }
```

Thus `<h1>` elements will display with a text color of green and will be centered within their enclosing elements (such as `<div>` or `<body>`). However, we can achieve the same effect more easily by grouping the declarations under a single selector, like so:

```
H1 {
  color: green;
  text-align: center;
}
```

Note the semicolon which marks the end of each declaration. All declarations must be contained within a pair of curly brackets.

Categories of CSS selectors

Type selectors

Redefine the appearance of HTML elements (tags)

Examples: `body { background-color: white; }`

Class selectors

A class selector is a custom style rules that can be applied to any HTML element

Example: `.graytext { color: #CCCCCC; }`

ID selectors

An ID selector applies to only one element on the page. ID and class work the same, but ID is more specific.

Example: `#footer { font-size: x-small; }`

The Universal selector

Applies to every element on the page (or at least those that possess the properties defined). The universal selector is the asterisk (*).

Example: `* { font-family: Verdana; }`

Pseudo-elements

Pseudo-elements refer to sub-parts of elements, such as the first line of a paragraph. They can also refer to less tangible attributes, such as the space before and after an element.

Example 1:   `p:first-line { padding-left: 1.5em; }`

Example 2:
```
p.backtop:before {
      content: url(assets/images/up_arrow.gif) " ";
}
```

which (with the appropriate graphic) produces the following effect:

Pseudo-classes

Like pseudo-elements, pseudo-classes refer to characteristics of elements, such as underline and hover attributes of links. These characteristics take their default appearance from the style rules built into the browser, but you can take control of them using the following pseudo-classes:

`a:link` – the standard color of linked text

`a:visited` – the color of linked text once the link has been visited

`a:hover` – the color of text and background while the cursor is hovering over the link

`a:active` – the color of text and background during the brief instant when the mouse is pressed down on the link

Example 1:   `a:link { color: blue; }`

Example 2:   `a:hover { background-color: yellow; }`

Editing style sheets

Cascading style sheets are simply text files with a file extension of .css. Style sheets can be edited in simple text editors like Notepad (for Windows), or in full-featured CSS editors like TopStyle, Style Master, and others. For most purposes, the CSS editing tools built into Dreamweaver are more than adequate.

The "cascade"

Fundamental to understanding CSS is the principle that more than one style sheet can influence the presentation of a document. CSS rules can come from three sources: the author, the user, and the browser application itself. The "cascade" refers to the mixing and prioritizing of style rules from these sources to achieve a single "style sheet." The cascade formula uses something called "selector specificity," to determine which style will be applied to each element of the document.

Three Locations for CSS

Style rules can be found in three locations:

1. External file
> External style sheets are collections of rules stored in separate, external CSS documents. Every page of a website can be linked to a single external style sheet.

2. Embedded CSS
> Style rules that appear in the <head> element of the document are often referred to as "embedded style sheets." Embedded styles affect only the document in which they appear. When referring to the same selector, embedded rules supersede external rules.

3. Inline CSS
> Rules that appear inside HTML tags are called "inline styles." Inline style rules trump both external and embedded rules.

Which location is best?
Inline rules are the powerful, in the sense that they supersede both embedded and external style rules. But because inline styles are so specific—referring not only to a particular web page, but even to particular HTML elements—they are actually less useful! By contrast, external style sheets offer maximum efficiency by controlling the appearance of potentially all of the pages of your website. External style sheets are recommended for most web designs.
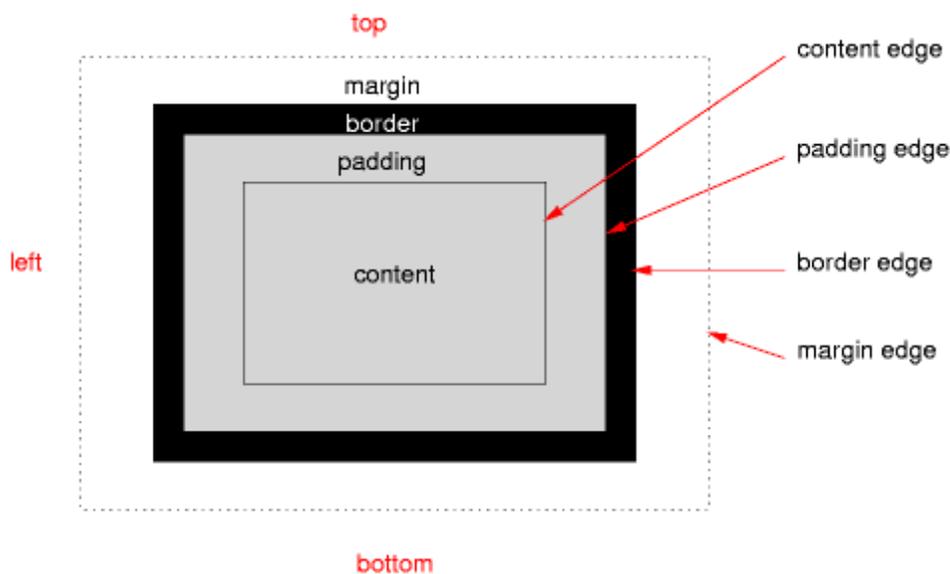
Tip for designing with style sheets

Designing with CSS may be simple in principle, but it can become complicated and frustrating when multiple rules begin to overlap and conflict. One helpful trick is to assign each `<div>` or HTML element in question a unique background color during development. The color allows you to see the location and size of your design blocks, and helps clarify tricky margin and padding settings. This is especially handy when trying to troubleshoot the position of text, graphics, and form elements (see CSS Box Model below).

The CSS Box Model

"…Cascading Style Sheets describe the rendering of documents on various media. When *textual* documents (like HTML) are laid out on *visual* media (such as a computer screen or paper), CSS represents the elements of the document with rectangular boxes (usually invisible). These boxes are laid out one after the other, or nested inside each other, in an ordering that is called a *flow…*"
– From the W3C CSS specification (http://www.w3.org/TR/css3-box/)

The box model applies equally to graphics, text, and form elements. For example, each paragraph element `<p>` has padding, margin values, and potentially a border as well.

If values are not explicitly defined in the style sheet, the browser looks to its own internal CSS rules for displaying page elements.

CSS syntax

Standard notation

```
p {
   background-color: #CCCCCC;
   color: #0000FF;

   border-color: Red;
   border-style: solid;
   border-width: 1px;

   font-family: Verdana;
   font-size: medium;
   font-style: italic;
   font-variant: small-caps;
   font-weight: bold;
   line-height: 130%;

   margin-top: 12px;
   margin-right: 24px;
   margin-bottom: 12px;
   margin-left: 24px;

   padding-top: 0em;
   padding-right: 0em;
   padding-bottom: 0em;
   padding-left: 0em;
}
```

Shorthand notation

```
p {
   /* This is a "comment" — a note to yourself or others */
   background: #ccc; /* not case-sensitive; some hex color values may be
      abbreviated to 3 characters */
   color: blue; /* common colors have English names */
   border: 1px solid red; /* 3 rules combined */
   font: italic small-caps bold medium/130% Verdana; /* 6 rules combined */
   margin: 12px 24px; /* top and bottom, left and right combined */
   padding: 0; /* same value repeated for all 4 sides of the box */
}
```

Shorthand notation for box dimensions

The four box dimensions are read in the following order: **T**op, **R**ight, **B**ottom, **L**eft.
Note the mnemonic acronym: TRBL. (Remember: Design with CSS and stay out of TRouBLe!)

For example, the following expression:

```
padding: 1em 0em 1.5em 2em;
```

is simply a more succinct way of saying:

```
padding-top: 1em;
padding-right: 0em;
padding-bottom: 1.5em;
padding-left: 2em;
```

Note: The sides of each CSS "box" come in pairs: top and bottom, right and left. Any missing value is assumed to duplicate the remaining value for that pair. Thus:
- If only 3 dimensions are listed, they are read in the following order: Top, Right/Left, Bottom.
- If only 2 dimensions appear, the order is Top/Bottom, Right/Left.

Appendix 2: Resources

CSU resources for web developers

(all links valid as of January 1, 2007)

- **CSU Web Accessibility Policy**
  http://accessibility.colostate.edu/policy/webaccesspolicy.html
- **CSU Web Access Standards**
  http://www.colostate.edu/Depts/ATRC/CSUstandards.htm
- **CSU Web Central** (Resources for Web Developers/Administrators)
  http://webdev.colostate.edu/

XHTML and CSS references, tutorials, and tools

Books

> *Cascading Style Sheets: The Definitive Guide*, 2nd Edition.
>   Eric Meyer. O'Reilly Media, 2004.
> *Cascading Style Sheets: Designing for the Web*, 3rd Edition.
>   Håkon Wium Lie and Bert Bos. Addison-Wesley, 2005.

Web Links (by category)

XHTML Specification

> http://www.w3.org/TR/xhtml1/ (complete descriptions of all elements, syntax, etc.)

Cascading Style Sheets (CSS)

> http://academ.hvcc.edu/~kantopet/old/css/index.php (Tutorial)
> http://www.w3.org/Style/LieBos2e/enter/ (Tutorial and examples to accompany book)
> http://www.w3schools.com/css/default.asp (CSS Tutorial)
> http://www.cssbasics.com/ (Explanation and tutorial)
> http://www.learn-css-tutorial.com/ (Tutorial)

Liquid Layout/multiple columns

> http://www.maxdesign.com.au/presentation/liquid/index.htm ("Liquid layouts the easy way")
> http://www.maxdesign.com.au/presentation/liquid-layouts/ (presentation w/ useful links)
> http://www.maxdesign.com.au/presentation/liquid-layouts/liquid.htm (example of resizing forms and images in liquid layout)

Multi-Column layout

http://www.fu2k.org/alex/css/ (multi-column recipes)
http://www.fu2k.org/alex/css/layouts/3Col_OrderedAbsolute.mhtml (multi-column tester)
http://wellstyled.com/index.html (CSS & Web design workshop)
http://www.treetrybe.com/imm/tests/nn4_3cols_x.html (3-column, Liquid, CSS-based layout)
http://www.themaninblue.com/experiment/ResolutionLayout/ (A demonstration of a JavaScript-based horizontal-to-vertical layout conversion depending on viewport width)

CSS menus

http://meyerweb.com/eric/css/edge/menus/demo.html (Eric Meyer, Pure CSS Menus)
http://www.kalsey.com/tools/csstabs/ (Adam Kalsey, CSS tabs with submenus)
http://css.maxdesign.com.au/listamatic/ (A list of menu recipes from various designers)
http://www.alistapart.com/articles/horizdropdowns/ (Drop-down menus, horizontal style)
http://www.alistapart.com/articles/dropdowns/ (Suckerfish Dropdowns: A List Apart)
http://www.htmldog.com/articles/suckerfish/dropdowns/ (Son of Suckerfish Dropdowns)
http://www.xs4all.nl/~peterned/csshover.html (whatever:hover menu tutorial)
http://www.xs4all.nl/~peterned/examples/cssmenu.html (whatever:hover menu example)
http://www.tanfa.co.uk/css/examples/menu/tutorial-v.asp#vstart (CSS pop-out menu tutorial)
http://www.positioniseverything.net/ordered-floats.html (Source Ordered Columns)

CSS Table formatting

http://icant.co.uk/csstablegallery/

CSS Tutorials

http://css.maxdesign.com.au/floatutorial/index.htm (float tutorial)
http://wellstyled.com/index.html (table formatting)
http://www.w3.org/Style/Examples/007/ (W3 CSS tips & tricks)
http://www.westciv.com/style_master/house/tutorials/index.html (tutorials and reference guide)
CSS Zen Garden (a demonstration of the flexibility of CSS)

HTML validator

http://validator.w3.org/

CSS Validator

http://jigsaw.w3.org/css-validator/