



CSS Type Design

Some people don't get coding. Others don't like writing content for their sites (we often call those people "clients"). However, I haven't met anyone that won't offer an opinion on the visual aspects of Web design.

The topics in this last lecture— associating CSS to a valid HTML document, using selectors, using CSS to control typography, and the box model—are all key concepts to have under your proverbial belt as a professional Web developer.

This exercise will give you practice in using CSS for typography. First you'll focus on making a long article readable, exploring CSS type



Practice your typography skills in this exercise.

controls. Then you'll experiment with designing box-level elements.

Performance Objectives

In this exercise, you're expected to:

Create an HTML5

page from a sample article provided in the Word document.
Use an internal CSS stylesheet to apply basic padding,

borders, and margins

to the text column.

- Choose font-family styles/sizes for headers, paragraphs, lists, and pullquotes.
- Create an attractive, readable layout by adjusting letter and line spacing and styling lists.
- Set a background and text color and style links.
- Demonstrate an ability to imitate two

Let's get started!

Project Brief

Part I: Typography Tour-de-Force

In the first part of this assignment, take the content from the Word document provided below and transform it into an HTML5 document with a fixed width layout.

The document is an article I wrote a couple of years ago about measuring the success of a Web site. The article has paragraphs, headers, quotes, pullquotes, links, and lists, so it that will challenge you to explore the concepts you've been learning. I can't wait to see what you'll do with it!

Click here to access the article in Word document form.

If you like, you can use the following CSS layout:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Untitled Document</title>
<style type="text/css">
#frame {
width: 800px;
margin: 0 auto;
```

box designs using border, margin, and spacing.



Assignments are evaluated for understanding of concepts covered in the lecture.



Special characters can be created with a specific sequence of symbols. The symbol for an em dash is — for example.



You'll notice that this is an *internal* style sheet. Since you're only formatting one page, you'll write the CSS rules into the top of the document. In other projects, you'd use an external style sheet.

We'll cover detailed steps for how to proceed in a moment, but first refresh your memory on CSS text properties in this Review Kit:



1. Mark up the document

Start by applying HTML5 semantic markup to the content. You may wish to print out a paper copy of the article and mark up the components you'll want to style. In your markup, you'll want to identify headers, sections, paragraphs, lists, and technical terms. Recall that HTML5 provides <header>, <nav>, <section>, and <footer> tags, and that you can still use <div> and tags as needed.

There are three levels of headers in the document: $\langle h1 \rangle$, $\langle h2 \rangle$, and $\langle h3 \rangle$. Enclose every paragraph in $\langle p \rangle$ tags. Next, mark up the lists. To keep it simple, make the first list ordered $\langle o1 \rangle$, and the second unordered $\langle u1 \rangle$.

Mark up the four long "pullquotes" in the text as blockquotes. **Pullquotes** are longer lines that repeat key phrases or tips that are stated in the body text. In magazines, they are often "called out" of the text and italicized with quotation marks to attract the attention of a browsing or casual reader:

"You can't assign hard numbers to a development goal, but they are great for laying down the groundwork for more specifics."

One of the quotes is a Web user quote (mark that up as a blockquote for now, but we'll come back to that later).

Also italicized in the Word document are seven technical terms, such as:

The first place many Web developers and businesses look when evaluating a site's success is the *log files*.

I want you to link those terms to their Webopedia definitions. That way, you'll have the opportunity to style those links later. Here are the terms and URLs listed in the order they appear:

```
log files
http://www.webopedia.com/term/l/log.html

servers
http://www.webopedia.com/term/s/server.html

browser
http://www.webopedia.com/term/b/browser.html

IP address
http://www.webopedia.com/term/I/IP_address.html

hit
http://www.webopedia.com/term/h/hit.html

clients
http://www.webopedia.com/term/c/client.html

traffic
http://www.webopedia.com/term/t/traffic.html
```

Mark these up using regular anchor tags, adding the following title attribute to each one: title="Webopedia definition" and a target="_blank" attribute to launch the definition page in a new window.

For example:

```
<a href="http://www.webopedia.com/term/l/log.html"
title="Webopedia definition" target="_blank">log files</a>
```

Validate your page before you go any further: http://validator.w3.org. Remember to choose the correct Doctype when running the validation (click "More Options" on the page).

2. Create an internal stylesheet

To style your page, you will be adding CSS rules (selectors, properties, and values) to the head of your document, as shown at the beginning of this exercise.

3. Adjust column width and margins

We've already set up an ID attribute to style all the page content. The first thing you might do is adjust column width and margins to your liking. The page is set to 800 pixels wide. Wide text columns can be tiring on the eye, so experiment with a narrower column. Also, you'll note that margins are set to 0 at the top and bottom and auto (equal reset) on the left and right. Experiment with those settings and use individual margin settings if you wish to position your text column differently.

4. Choose font-family styles/sizes for headers and paragraphs

Now it's time to select your fonts. You might try contrasting styles for headers and paragraphs (using serif for one and sans serif for the other). Remember that you can specify more than one font-family and combine selectors if you want to:

```
h1, h2, h3 {
  font-family: Verdana, Arial, sans-serif;
}
p {
```

```
font-family: Georgia, "Times New Roman", serif;
}
```

Check out the default fonts that are available on your system and try some interesting combinations, as well as standard ones.

Next, select your font size. You'll recall this can be done many different ways, but that designers often use pixels or ems, like so:

```
h1 {
font-size: 24px;
}
```

Make sure your headers are set to contrasting size and don't wrap around like this:

Measuring Web User Experience

Also apply styles to your lists and pullquotes. Start by setting them in the same font-family and size as your body text, then experiment. Pullquotes should be larger than the body text to attract attention.

You'll also recall that you can set font-weight, font-style, font-variant, and text-decoration. Pullquotes might be italicized using the font-style: oblique; option. Since pullquotes should look separate from the text, you might also want to set them to text-align: right;

5. Letter-spacing and line-height

Now let's make sure that the letter-spacing in headings is effective and the line-height in our paragraphs makes the text readable. You'll recall that the spacing between letters in headings can be globally adjusted:

```
h1 {
letter-spacing: 2px;
}
```

Experiment with positive and negative values until your headings look punchy. Then apply line-height to various pieces of body text, starting with paragraphs:

```
p {
line-height: 16px;
}
```

Experiment with different values until the text has enough space around it to "breathe" without losing the visual coherence of each paragraph.

6. Style lists

Next, it's time to apply some styles to lists. You'll recall that lists can have unconventional bullets:

```
ul {
list-style-type: square;
}
```

7. Style pullquotes

OK, time to use some divs to make the pullquotes stand out. Notice the class attribute:

```
<div class="quote">
  <blockquote>A log file evaluation based on traffic can help
if you can compare traffic to some other temporal reference
point.</blockquote>
  </div>
```

Next, define the class with the . sign in the CSS. Because <div> tags create a block-level element, you can define widths, alignment, borders, margin and padding for your pullquote:

```
.quote {
width: 400px;
border: 1px dashed;
margin-left: 100px;
margin-top: 5px;
margin-bottom: 5px;
padding: 10px 40px 10px 10px;
}
```

The above approach is good when there is more than one pullquote. If you only have one pullquote, I would recommend changing the class attribute to an id attribute.

Id attributes should be used once and mark specific areas of a document, like content or a footer or a header. Class attributes can be reused often and cake take the same value within the same document without errors—so if you have several paragraphs out of many you want to style, you should use class.

Recall that <code>span</code> tags are inline tags like images and a tags. So if you want to apply an extra bit of style within elements that are inline, use a <code>span</code>. If you want to setup a block level element, use a <code>div</code>. <code>Div</code> tags and <code>id</code> attributes go hand-in-hand in setting up certain sections of a document, so you will often see them used for marking out areas like header, footer and content.

8. Background and text color

Now it's time to work on the background color and text color. Remember—you can set the background color and text color for the page body, headers, paragraphs, as well as the ID tags (pullquotes).

```
p {
  color: #ff0099;
  background-color: white;
}
```

You can also use background graphics in block-level elements. I would encourage you to experiment but avoid the usual traps: bright colors, clashing colors, too much contrast, or insufficient contrast. You may experiment with background graphics too, but it's essential that you keep your page easy to read.



Remember, if you're stuck for color combinations, try looking up hex colors at one of the many color resource sites on the Web. You might start by choosing a dominant color then thinking about two or three complementary colors that will work well with it.

Two good places to start are http://www.colorcombos.com. and http://www.colorcombos.com.

9. Style links

Lastly, you'll need to style your links, particularly if you used a page background. You can target this specific *state* of a link by using **pseudo-classes**. There are four pseudo-classes for the anchor element:

- a:link (unvisited link)
- a:visited (visited link)
- a:hover (moused-over link)
- a:active (selected link)

You can set the attributes for each link state at the top of your document, like so:

```
a:hover {
text-decoration: none;
color: red;
background-color: #fffff99;
}
```

<u>Click here</u> to see one possible version of this page. I can't wait to see what *you* will come up with! What I'm looking for in your assignment is solid technical coding combined with the ability to create an engaging, readable layout.

10. Test your site

Post this assignment to your Web host to test it live. Remember to test your work in multiple browsers.



Test your work in Firefox and Safari before Internet Explorer.

12/12/2019 CSS Type Design

Still unable to get part of your Web design working? Check out my <u>Web</u> <u>Development Troubleshooting guide</u> for more step-by-step advice on getting your CSS working.

Part II: Boxing Clever

For the second part of the assignment, you will experiment with different margins, padding, and border styles by imitating two different sections of Web pages that you admire. Asking yourself "how did they do that?" and trying it for yourself is a great way to learn CSS.

Visit some of your favorite Web sites. Notice how the different sections of page content are in "boxes." Find two boxes (ideally from two different sites) to recreate using the CSS knowledge you've gained so far.



Insound.com uses boxes for its new music releases. How do you think they got that cute dotted border on the top and bottom? What about the padding?



Woxy.com has a box for its upcoming shows. How might the background image sit in the corner and not repeat?

On a new HTML page, use HTML and CSS to recreate the two boxes that you have identified. Name this page boxes.html and upload it to your Web host along with any associated image files. When you submit this assignment, please mention the Web sites on which you modeled your boxes so we can compare and discuss.

When you are finished, be sure both assignments are on your Web host and working properly, then submit both direct links to the Dropbox for evaluation.

What your instructor expects you to do: Create a valid HTML5 page from a sample article provided in the Word document. Use an internal CSS stylesheet to apply basic padding, borders, and margins to the text column.

12/12/2019 CSS Type Design

	☐ Choose font-family styles/sizes for headers, paragraphs, lists, and pullquotes.
	☐ Create an attractive, readable layout by adjusting letter and line spacing and styling lists, and making effective background color and text color choices.
	☐ Demonstrate the ability to imitate two different box designs using border, margin, and spacing properties.
	Important reminder: Templates from HTML editing software—such as Dreamweaver—are not allowed in this course. This is a course about learning the fundamentals of HTML and CSS, and hence we stick strictly to hand coding. Any assignments based off any kind of template (including Dreamwever templates) will receive a 0.
	Once you're done, go to the Dropbox for this exercise and post links to both exercises with a brief comment.
	If you have a question before sending your completed exercise for grading, use the Send Mail area to contact your instructor.
	I look forward to seeing your work!