

CSS LAB 2

INTRODUCTION TO CSS

What You Will Learn

- How to use CSS to style HTML documents
- How to use CSS selectors
- The CSS box model

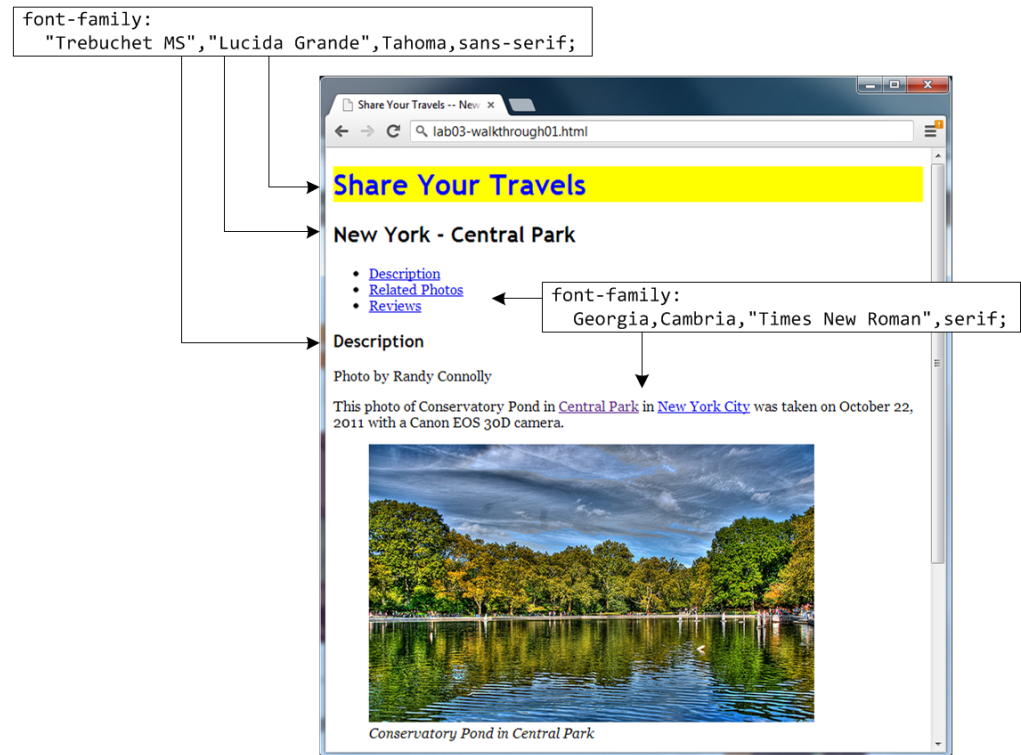


Figure 2.1 – Exercise 2.1 Complete

Exercise 1.. —EXTERNAL STYLE SHEETS

- 1 Create a new text document with the following content:

```
header, footer {
    color: white;
    background-color: #213643;
}
nav {
    background-color: #728B96;
}
h1, h2, h3 {
    font-family: "Trebuchet MS", "Lucida Grande", Tahoma, sans-serif;
}
body {
    font-family: Georgia, Cambria, "Times New Roman", serif;
}
```

Notice that this document contains no HTML. Also notice that the first style rule uses a grouped selector, meaning that both the header and footer elements will receive the same style.

- 2 Save your file as lab02-exercise01.css.
- 3 Open lab01-exercise01.html (which you have created last lab).

- 4 Add the following to the <head> element.

```
<head>
  <meta charset="utf-8">
  <title>Share Your Travels -- New York - Central Park</title>
  <link rel="stylesheet" href="lab02-exercise01.css" />
</head>
```
- 5 Save and test lab01-exercise01.html in browser.

CSS SELECTORS

When defining CSS rules you will need to first need to use a selector to tell the browser which elements will be affected by the styles. CSS selectors allow you to select individual or multiple HTML elements, elements that belong together in some way, or elements that are positioned in specific ways in the document hierarchy. The previous exercises used HTML selectors. The next exercises make use of other selectors.

Exercise 2.. — ELEMENT, CLASS, AND ID SELECTORS

- 1 lab01-exercise01.html.
- 2 Add the following to the markup.

```
<section>
  <h3 id="reviews">Reviews</h3>
  <div>
```
- 3 Open lab02-exercise01.css, add the following style, and test.

```
#reviews {
  border-bottom: solid 3pt #213643;
  color: #ed8030;
}
```
- 4 Change the previous selector to the following and test.

```
h3#reviews
```

In this example the selector in step 3 and 4 are functionally equivalent. However, the one in step 4 is more specific, so in some situations will be preferable.

- 5 Switch to lab01-exercise01.html and add the following.

```
<section>
  <h3 id="reviews">Reviews</h3>
  <div>
    <p class="first">By Ricardo on <time>September 15, 2012</time></p>
    <p>Easy on the HDR buddy.</p>
  </div>
  <hr/>
  <div>
    <p class="first">By Susan on <time>October 1, 2012</time></p>
    <p>I love Central Park.</p>
  </div>
  <hr/>
</section>
```

- 6 Switch to lab01-exercise01.css and add the following style.

```
.first {
  color: #728B96;
  font-style: italic;
}
```

- 7 Test.

Remember that whenever the lab says “test” it means view the relevant HTML file in a browser.

Attribute Selectors

An attribute selector provides a way to select HTML elements either by the presence of an element attribute or by the value of an attribute. This can be a very powerful technique, but because of uneven support by some of the browsers, not all web authors have used them.

Exercise 3.. — ATTRIBUTE SELECTORS

- 1 Open lab01-exercise01.html
- 2 Open lab02-exercise01.css, add the following style, and test.

```
[title] {
  cursor: help;
  text-decoration: none;
  padding-bottom: 3px;
  border-bottom: 2px dotted blue;
}
```

This will select every element that contains a title attribute. Examine the HTML to see which elements contain a title attribute.

- 3 Modify the attribute to the following and test.

```
[title="Calgary"] {
```

This selects only the one element whose title attribute is exactly Calgary.

- 4 Modify the attribute (add the asterisk) to the following and test.

```
[title*="Calgary"] {
```

*This selects all elements that **contain** the text Calgary within in.*

- 5 Modify the attribute (add the caret) to the following and test.

```
[title^="Calgary"] {
```

*This selects all elements that **begin** with the text Calgary.*

Pseudo-Class Selectors

The next exercise illustrates the use of pseudo-class selectors, which do not apply to an HTML element, but targets either a particular state or, in CSS3, a variety of family relationships. The most common use of this type of selectors is for targeting link states. By default, the browser displays link text blue and visited text links purple. Exercise 4.6 illustrates the use of pseudo-class selectors to style not only the visited and unvisited link colors, but also the hover color, which is the color of the link when the mouse is over the link. Do be aware that this state does not occur on touch screen devices.

Exercise 4.. — PSEUDO SELECTORS AND LISTS

- 1 Open lab01-exercise01.html.
- 2 Switch to lab02-exercise01.css and add the following style.

```
a:link {  
    font-weight: bold;  
    color: #47B3D7;  
}  
a:visited {  
    color: #BB78FF;  
}
```

- 3 Test in browser.
- 4 Add the following style to lab01-exercise01.css and test.

```
a:hover {  
    background-color: #FFFF99;  
}
```

To test this style, you must hover your mouse over a link.

- 5 Add the following style to lab01-exercise01.css and test.

```
nav ul li {  
    list-style: none;  
}
```

This removes the bullet from the navigation elements.

- 6 Add the following and test.

```
nav ul li {  
    list-style: none;  
    display: inline;  
}
```

By changing the elements from their default display: block value, the list items are no longer block elements (thus existing on their own lines) but are now inline elements.

- 7 Add the following and test.

```
nav ul li {  
    list-style: none;  
    display: inline;  
    margin: 1em;  
}
```

This adds space to the left and right of each list item. Why doesn't it also add space to the top and bottom as well? The answer is that top and bottom margins are ignored for inline elements.

- 8 Add the following and test.

```
nav {  
    padding: 0.25em;  
}
```

This adds padding space within the <nav> element.

- 9 Add the following and test.

```
nav a {  
    padding: 0.25em;  
}
```

This makes the size of the <a> element the same as the container, which makes the hover size the same as the container. The result should look similar to that shown in Figure 2.2. You will get more practice with margins and padding in the next exercise.

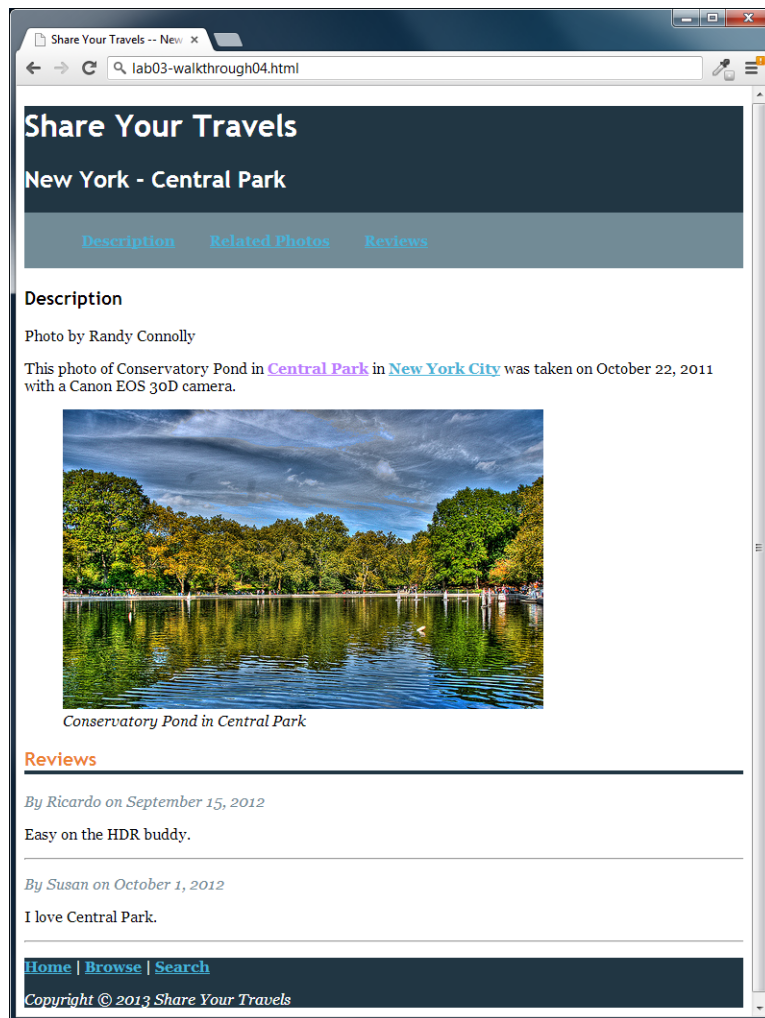


Figure 2.2 – Exercise 4.4 Complete

Contextual Selectors

A contextual selector allows you to select elements based on their ancestors, descendants, or siblings. That is, it selects elements based on their context or their relation to other elements in the document tree. While some of these contextual selectors are used relatively infrequently, almost all web authors find themselves using descendant selectors.

Exercise 5.. — CONTEXTUAL SELECTORS

- 1 Open lab01-exercise01.html.
- 2 Switch to lab02-exercise01.css and add the following style and test.

```
p {  
    color: #983C2A;  
}
```

This changes the text color of every occurrence of the `<p>` element.

- 3 Modify the style as follows. Before you test, see if you can figure out which text will be affected by the change to this descendent selector.

```
section p {  
    color: #983C2A;  
}
```

This selects all `<p>` elements that exist somewhere within a `<section>` element. That selects all of them except for the ones within the `<footer>` element.

- 4 Modify the style as follows and test.

```
div p {  
    color: #983C2A;  
}
```

This selects all `<p>` elements that exist somewhere within a `<div>` element.

- 5 Modify the style as follows and test. As with the previous steps, see if you can figure out which text will be affected by the change before you test it.

```
section>p {  
    color: #983C2A;  
}
```

This doesn't select the review `<p>` elements because they are contained within `<div>` elements, and thus are not direct children of a `<section>` element.

- 6 Modify the style as follows and test.

```
h3+p {  
    color: #983C2A;  
}
```

This selects any `<p>` elements that immediately follow an `<h3>` element.

- 7 Modify the style as follows and test.

```
h3~p {  
    color: #983C2A;  
}
```

This selects any `<p>` elements that share the same parent as an `<h3>` element.

CSS CASCADE AND BOX MODEL

Exercise 6.. — CSS CASCADE

- 1 Open and examine lab01-exercise01.html.
- 2 Add the following style to lab02-exercise01.css and test.

```
div {  
    font-weight: normal;  
    color: magenta;  
}  
  
p {  
    color: green;  
}
```

- 3 Add the following style to lab02-exercise01.css and test.

```
.last {  
    color: blue;  
}  
  
#verylast {  
    color: orange;  
    font-size: 16pt;  
}
```

Notice that class selectors take precedence over element selectors and that id selectors take precedence over class selectors.

In CSS, all HTML elements exist within an element box. It is absolutely essential that you familiarize yourself with the terminology and relationship of the CSS properties within the element box, which are shown in Figure 2.3.

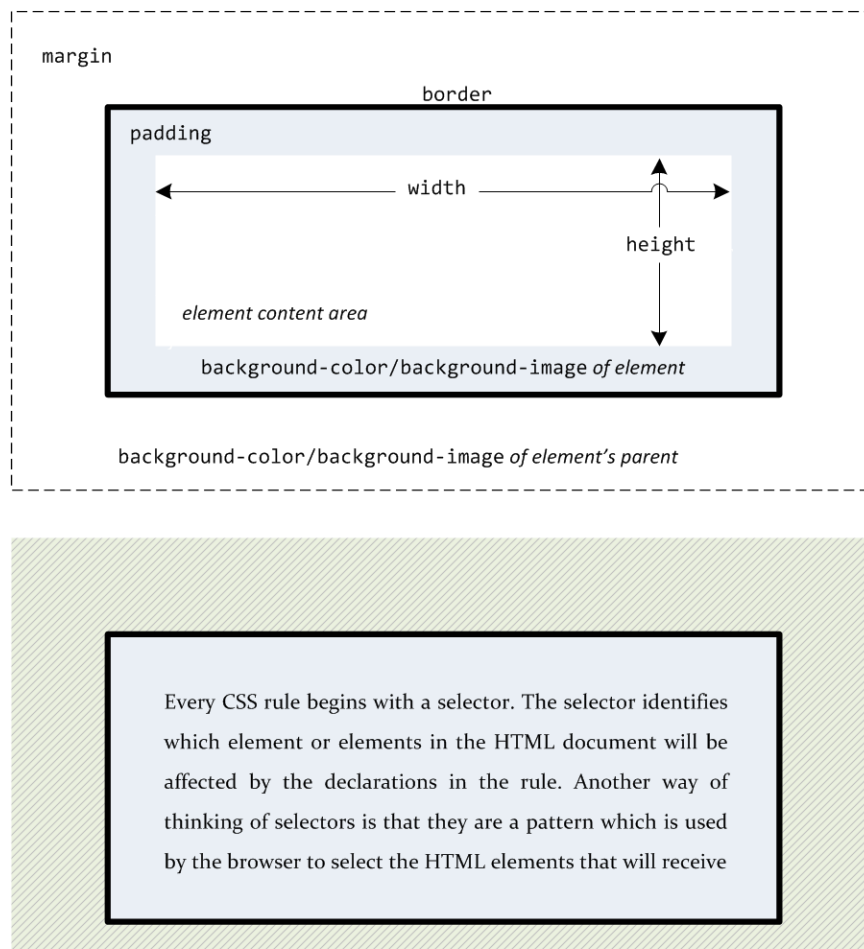


Figure 2.3 – CSS Box Model

Exercise 7.. — BORDERS , MARGINS, AND PADDING

- 1 Open lab01-exercise01.html.
- 2 Switch to lab02-exercise01.css and add the following style **to the very top** of the file. Test in browser.

```
header, footer, nav, main, article, section, figure, figcaption, h1, h2,
h3, ul, li, body, div, p, img
{
  margin: 0;
  padding: 0;
  font-size: 100%;
  vertical-align: baseline;
  border: 0;
}
```

This is an example of a CSS reset, that is, a way to remove any browser defaults. This way, any styling that is present is due to explicit styling rather than to browser defaults, which can potentially vary from browser to browser.

- 3 Add the following and test.

```
figure {  
  margin: 2em;  
}
```

- 4 Modify the style as follows and test.

```
figure {  
  margin: 2em;  
  background-color: #EEEEEE;  
}
```

- 5 Modify the style as follows and test.

```
figure {  
  margin: 2em;  
  background-color: #EEEEEE;  
  padding: 1.5em;  
}
```

- 6 Modify the style as follows and test.

```
figure {  
  margin: 2em;  
  background-color: #EEEEEE;  
  padding: 1.5em;  
  border: solid 1px #999999;  
}
```

The result should look similar to that shown in Figure 2.4.

- 7 Add the following and test.

```
h3+p+p {  
  margin: 2em;  
}
```

This adds margin space to the paragraph above the figure. Notice that the space between the paragraph and the figure does not change. Can you figure out why?

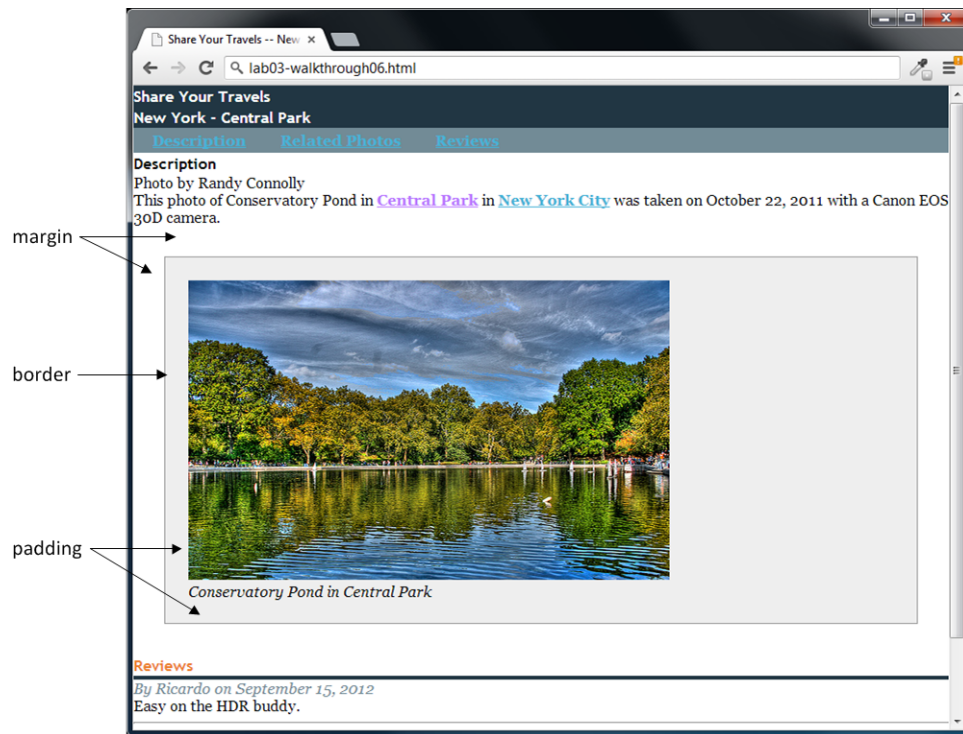


Figure 2.4 – Exercise 2.7 at step 6.

Exercise 8.. — BACKGROUND STYLE

- 1 Open lab01-exercise01.html.
Notice that it has added the share images within the `<figure>` element.

- 2 Add the following style to lab02-exercise01.css and test.

```
figure div {
    margin-top: 0.25em;
    font-size: 70%;
    padding: 0.5em;
    width: 485px;
    background-color: #9FAAB0;
    border: solid 1px #999999;
    border-radius: 4px;
}
```

This adds a rounded rectangle to the `<div>` within the `<figure>`. You may want to try saving and testing after entering each line.

- 3 Add the following style and test.

```
figure div p {
    background: url(images/share.png) no-repeat;
    padding-left: 18px;
}
```

Notice that the share icon is not aligned with the Share text. The next step will move it.

- 4 Modify the style as follows and test.

```
figure div p {
  background: url(images/share.png) no-repeat;
  padding-left: 18px;
  background-position: 0 4px;
}
```

- 5 Examine `glyphicons-halflings.png` in either a graphics editor or some type of graphics viewer. You can also simply drag and drop the image into a browser window.

It is common to package together a large number of images into a single image in order to improve performance (i.e., only one file to request and download rather than dozens of files). This typically involves using `background-position` in conjunction with widths and heights, as shown in the next step.

- 6 Modify the HTML as follows.

```
<figcaption><em>Conservatory Pond in Central Park</em></figcaption>
<div>
  <p><span class="share"></span>Share:
  ...
```

- 7 Remove or comment out the style rule created in steps 3 and 4.

- 8 Add the following style and test.

```
span.share {
  display: inline-block;
  background: url(images/glyphicons-halflings.png) no-repeat;
  width: 14px;
  height: 14px;
  background-position: -120px -72px;
  padding-right: 5px;
}
```

Notice the use of the `inLine-bLock` value for the `display` property. This value keeps elements inline, but preserves their block capabilities such as setting width and height, margins, and paddings.

Notice also the negative `background-position` values. To make sense of these values, see Figure 2.5.

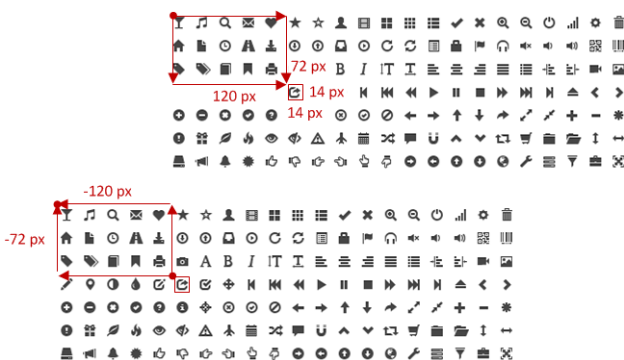


Figure 2.5 – Using `background-position`

Exercise 9.. — CSS FONT SIZES

- 1 Open lab01-exercise01.html.
- 2 Add the following style to lab02-exercise01.css.

```
body {  
    font-size: 100%; /* about 16px */  
}
```

This sets the base line size of textual content in the document.

- 3 Add the following styles and test.

```
h1 {  
    font-size: 24px; /* for older browsers */  
    font-size: 1.5rem;  
}  
h2 {  
    font-size: 18px;  
    font-size: 1.125rem;  
}  
h3 {  
    font-size: 16px;  
    font-size: 1rem;  
}  
p {  
    font-size: 16px;  
    font-size: 1rem;  
}
```

- 4 Add the following style and test.

```
.first, figcaption, h3+p {  
    font-size: 14px;  
    font-size: .875rem;  
}
```

Exercise 10.. — CSS FONTS

- 1 Open and view lab01-exercise01.html in the browser.
- 2 Use your browser to visit <http://www.google.com/fonts>.
- 3 Search for the **Lobster** font.
- 4 Click on the Quick Use button and copy the <link> element for the font that is provided onto the clipboard:

```
<link href='http://fonts.googleapis.com/css?family=Lobster'  
      rel='stylesheet' type='text/css'>
```

- 5 Paste the <link> element into lab01-exercise01.html.
- 6 Modify the following rule in lab02-exercise01.css.
- 7 Test in browser.
- 8 Switch back to <http://www.google.com/fonts> and search for the font **Lato**. Add the appropriate <link> element for this font and modify the following rule:

```
body {  
  font-family: Lato, Helvetica, Arial, sans-serif;  
}
```

- 9 Test in browser.

Exercise 11.. — CSS PARAGRAPHS

- 1 Open and view lab01-exercise01.html in the browser.
- 2 Add the following rule in lab02-exercise01.css.

```
p {  
  margin-bottom: 0.5em;  
}
```

- 3 Add the following rule and then test in browser.

```
p#first {  
  text-align: right;  
}
```

You may need to reduce the size of your browser window in order to see the right alignment of the text.

- 4 The following rule and test.

```
p#second {  
  text-indent: 2em;  
  line-height: 1.5em;  
}
```

- 5 Modify the following rule and test.

```
h1 {  
  font-size: 250%;  
  text-transform: uppercase;  
  letter-spacing: 10px;  
}
```

- 6 Add the following rule and test.

```
h1 {  
  font-size: 250%;  
  text-transform: uppercase;  
  letter-spacing: 10px;  
  text-shadow: 3px 3px 3px rgba(0,0,0,0.3);  
}
```