LAB 2

Introduction To HTML

What You Will Learn

- How to create HTML documents
- Basic HTML structure
- How to creating hyperlinks
- How to add images to a web page
- HTML5 semantic tags

Approximate Time

The exercises in this lab should take approximately 30 minutes to complete.

Fundamentals of Web Development

Randy Connolly and Ricardo Hoar

Textbook by Pearson
http://www.funwebdev.com

QUICK TOUR OF HTML

PREPARING DIRECTORIES

1. If you haven't done so already, create a folder in your personal drive for all the labs for this book.
2. From the main `labs` folder (either downloaded from the textbook's web site using code provided with book or in a common location provided by your instructor), copy the folder titled `lab02` to your course folder created in step one.

Now we are ready to create our first web page.

EXERCISE 2.1 — FIRST WEB PAGE

1. Using some type of text or HTML editor (such as Notepad, Notepad++, PSPad, etc), type in the following:

```
<!DOCTYPE html>
<title>A Very Small Document</title>
<p>This is a simple document with not much content</p>
```

Note: these labs use the convention of ***`red bolded`*** *text to indicate content to change/enter.*

2. Save the file as `lab02-exercise01.html` in the `lab02` folder on your personal drive (the folder you just created in the Preparing Directories step above).
3. Start up **FireFox**, **Chrome**, **Internet Explorer** or some other browser. Open the file `lab02-exercise01.html`. To do this, you could use the Open command in the menu, drag-and-drop the file from the file manager of the operating system, or double-click the file from the operating system file manager.

   This will display the file created in step one in the browser window.

4. Switch back to your text editor. Position the cursor before "`This is a simple`" and then press **Enter** three times. Position cursor after the word "`much`". Press space five times.
5. Save the changes and then switch back to browser. Refresh the page.

   Notice that the browser ignores extra spaces and paragraph returns.

6. Remove the extra spaces and returns added in step 4. Save changes.

EXERCISE 2.2 — ADDITIONAL STRUCTURE TAGS

1 Create a new HTML document with the following content:

```
<!DOCTYPE html>
<html>
<head lang="en">
   <meta charset="utf-8">
   <title>Share Your Travels -- New York - Central Park</title>
</head>
<body>
   <h1>Share Your Travels</h1>
   <h2>New York - Central Park</h2>

   <h3>Description</h3>
   <p>Photo by Randy Connolly</p>
   <p>This photo of Conservatory Pond in Central Park New York City was
   taken on October 22, 2011 with a Canon EOS 30D camera.
   </p>

   <h3>Reviews</h3>
   <div>
      <p>By Ricardo on September 15, 2012</p>
      <p>Easy on the HDR buddy.</p>
   </div>
   <hr/>
</body>
</html>
```

Notice that this document has additional structure tags (`<head>`, `<body>`, `<html>`) that were required in XHTML but are now optional in HTML5.

2 Save your file as `lab02-exercise02.html` and test file in browser. The result should look similar to that shown in Figure 2.1.

Figure 2.1 – Exercise 2 Complete

EXERCISE 2.3 — MAKING MISTAKES

1. Open `lab02-exercise03.html` (which has the same content as the last exercise).
2. Before the text "`Conservatory`" (in the second paragraph tag), add the tag **`<randy>`**.
3. Save and then test in browser. After testing, remove the `<randy>` tag.

   Sadly there is no `<randy>` tag in HTML. Your browser will simply ignore any tag it does not recognize.

4. Remove the trailing `</h1>` end tag, save and then test.

   Since the `<h1>` tag is never closed, the browser assumes that the content after it should continue being displayed as a first-level heading.

5. Put back the trailing `</h1>` end tag (i.e., after "`Share Your Travels`").
6. Change the `<h1>` tag to `<H1>`, save and then test.

   Notice that HTML5 is case insensitive.

LINKING

Hyperlinks are an essential feature of any web page. Links are created via the anchor (`<a>`) element.

EXERCISE 2.4 — LINKING

1. Open `lab02-exercise05.html` and add the following bolded text:

```
<p>This photo of Conservatory Pond in
<a href="http://www.centralpark.com/">Central Park</a> in New York City was
taken on October 22, 2011 with a Canon EOS 30D camera.
</p>
```

   This will create an external link.

2. Save changes and test in browser.
3. Modify the document by adding the following link and test.

```
<p>This photo of Conservatory Pond in
<a href="http://www.centralpark.com/">Central Park</a> in
<a href="newyork.html">New York City</a> was
taken on October 22, 2011 with a Canon EOS 30D camera.
```

   This will create a relative link (i.e., a link to another page in the same web site).

ADDING IMAGES

EXERCISE 2.5 — ADDING IMAGES

1 Add the following tag and then test:

```
<img src="images/central-park.jpg" alt="Central Park" />
<h3>Reviews</h3>
```

This instructs the browser to display the file `central-park.jpg` which is found in the `images` subfolder.

2 Modify the image tag as follows and test (be sure to move your mouse over the image).

```
<img src="images/central-park.jpg" alt="Central Park"
     title="Central Park" />
```

The title attribute is used to display a tooltip; Internet Explorer, also displays the content of the `alt` attribute in a tooltip if there is no `title` attribute specified.

3 Change the `src` attribute to the following (i.e., add a slash before the folder name) and test.

```
<img src="/images/central-park.jpg" alt="Central Park"
     title="Central Park" />
```

You will no longer see the central park image. Why? Because the root reference does not work when tested locally.

Also, depending on the browser, you may or may not see a missing image icon, as shown in Figure 2.2. Notice that all three of the browsers in the Figure 2.2 will also display the `alt` attribute, but Firefox does not display a missing image icon.

What would we see in Firefox if the missing `<img>` did not have an `alt` attribute defined? The answer is nothing. While this makes sense perhaps from an end-user perspective, from a developer's perspective this behavior can be frustrating. This is one of the many reasons why we strongly recommend testing your pages in multiple browsers.

Figure 2.2 – Missing image indication in different browsers

4 Remove the slash added in step 3.

5 Add the following and then test:

```
<a href="images/large-central-park.jpg">
<img src="images/central-park.jpg" alt="Central Park" />
</a>
```

This turns the Central Park image into a link (in this case, a link to a larger version of the Central Park image).

6 Add the following after the Central Park image:

```
<a href="images/large-central-park.jpg"><img src="images/central-park.jpg"
    alt="Central Park" title="Central Park"/></a>
<p>Share:
  <img src="images/social/email_16.png" alt="Email this to someone" />
  <img src="images/social/rss_16.png" alt="Syndicated content" />
  <img src="images/social/twitter_16.png" alt="Share this on Twitter" />
</p>
```

Notice that images are by default inline content in that they exist in the same flow as text.

7 Remove the returns between each `<img>` tag, as shown below, and then test.

```
<p>Share:
<img src="images/social/email_16.png" alt="Email this to someone" /><img
src="images/social/rss_16.png" alt="Syndicated content" /><img src=
"images/social/twitter_16.png" alt="Share this on Twitter" />
</p>
```

Notice that the browser interprets each (or multiple ones in a row) carriage return in the HTML as a single space, as shown in Figure 2.3.

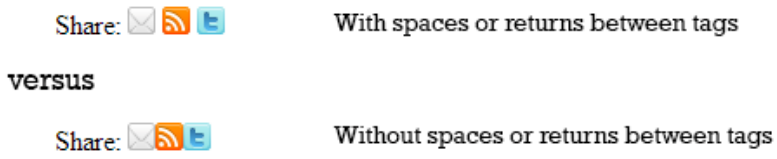

Figure 2.3 – Carriage return treated as a space

8 Modify the following and test.

```
<p>Share: <br/>
```

The `<br/>` tag adds a line break.

LIST BASICS

Lists are a way of organizing information. HTML supports several different types of list: definition lists, ordered lists, and unordered lists.

EXERCISE 2.6 — MAKING A LIST

1 Open `lab02-exercise04.html` and add the following bolded text:

```
<body>
   <h1>Share Your Travels</h1>
   <h2>New York - Central Park</h2>
   <ul>
      <li>Description</li>
      <li>Reviews</li>
   </ul>
   <h3>Description</h3>
```

*Remember: these labs use the convention of **`red bolded`** text to indicate content to change/enter/insert.*

*This will add an unordered list to your page. **Notice that it is a lowercase L not the number 1 in these new tags**.*

Also, the indenting shown in the list above doesn't affect the output in the browser. It is added to make the markup more readable for us, the developers.

2 Save and test.

3 Change the `<ul>` and `</ul>` to `<ol>` and `</ol>` and then test in browser.

This will change the list to an ordered list.

4 Change the list back to an unordered list.

It is common practice to create a list of related links. The next exercise demonstrates this technique.

Exercise 2.7 — Linking with Lists

1 Continue working with `lab02-exercise05.html` and add the following to the list and test:

```
<ul>
  <li><a href="#">Description</a></li>
  <li><a href="#">Reviews</a></li>
</ul>
```

Notice the target for the links (i.e., `href="#"`). The `#` simply indicates the current page (i.e., it goes nowhere). This is a common technique for showing links whose destinations are not yet known.

2 Modify the list as follows:

```
<ul>
  <li><a href="#description">Description</a></li>
  <li><a href="#reviews">Reviews</a></li>
</ul>
```

These are now references to anchors on the existing page, which we will add in the next step.

3 Add the following anchors to your document as shown below.

```
<ul>
  <li><a href="#description">Description</a></li>
  <li><a href="#reviews">Reviews</a></li>
</ul>
<h3 id="description">Description</h3>
<p>Photo by Randy Connolly</p>
<p>This photo of Conservatory Pond in
<a href="http://www.centralpark.com/">Central Park</a> in
<a href="newyork.html">New York City</a> was
taken on October 22, 2011 with a Canon EOS 30D camera.
   </p>

<h3 id="reviews">Reviews</h3>
<div>
   <p>By Ricardo on September 15, 2012</p>
   <p>Easy on the HDR buddy.</p>
</div>
```

4 Test by clicking on links in bulleted list.

HTML5 Semantic Elements

HTML5 introduced a number of new semantic elements that can make your markup more understandable and thus easier to maintain. The next set of exercises introduces several of these elements.

Exercise 2.8 — Header and Footer

1 Open `lab02-exercise09.html` and test.

2 Add the following and test.

```
<body>
   <header>
      <h1>Share Your Travels</h1>
      <h2>New York - Central Park</h2>
      <ul>
        <li><a href="#">Description</a></li>
        <li><a href="#">Reviews</a></li>
      </ul>
   </header>
```

You will notice that the browser does not add any formatting or spacing for the `<header>` element. It is used purely to make our markup more understandable. Later, once we learn CSS, we can give the header a particular look.

3 At the end of our document, add the following and test.

```
   <footer>
      <p>Copyright &copy; 2013 Share Your Travels</p>
   </footer>
</body>
</html>
```

Like the `<header>` element, the `<footer>` element has no built in style.

Notice as well the `©` character entity, which adds the copyright symbol.

4 Modify the footer as follows and test.

```
<footer>
  <p><strong>Copyright &copy; 2013 Share Your Travels</strong></p>
</footer>
```

The `<strong>` element is an inline text element.

5 Modify the footer as follows and test.

```
<p><em>Copyright &copy; 2013 Share Your Travels</em></p>
```

Exercise 2.9 — Navigation, Articles and Sections

1 Open `lab02-exercise10.html`, add the following and test.

```
<header>
  <h1>Share Your Travels</h1>
  <h2>New York - Central Park</h2>
  <nav>
    <ul>
      <li><a href="#">Description</a></li>
      <li><a href="#">Reviews</a></li>
    </ul>
  </nav>
</header>
<article>
   <section>
       <h3>Description</h3>
       <p>Photo by Randy Connolly</p>
       ... [content omitted]
       </p>
   </section>
   <section>
       <h3>Reviews</h3>
       <div>
          <p>By Ricardo on September 15, 2012</p>
          <p>Easy on the HDR buddy.</p>
       </div>
       <hr/>
   </section>
</article>
<footer>
```

Like with the other HTML5 semantic elements, there is no special browser formatting for these elements. They are used purely to make our markup clearer.

Exercise 2.10 — Figure and Captions

1 Open `lab02-exercise11.html`, view in browser, then add the following to the large image and test.

```
<figure>
 <a href="images/large-central-park.jpg"><img src="images/central-park.jpg"
   alt="Central Park" title="Central Park"/></a>
 <figcaption>Conservatory Pond in Central Park</figcaption>
</figure>
<p>Share: <br/>
```

Here's a surprise ... there is in fact a little bit of additional browser formatting for the `<figure>` HTML5 semantic element. Also notice that we are not wrapping the share icon images in a `<figure>` element. As discussed in the text, the `<figure>` element should be used only for images (or other content) that is essential but whose position on the page could change. The share icons are not really essential so they are not contained within a `<figure>`.

VALIDATING HTML

In the next exercise, we will use an external validation service to verify that our web page contains HTML that is valid according to the HTML5 DTD.

EXERCISE 2.11 — VALIDATING HTML

1. Open a browser and go to `http://validator.w3.org`
2. In the **Validate By File Upload** tab, click the **Browse** or **Choose File** button and choose your `lab02-exercise05.html` file.
3. Click the **Check** button.

   The site should eventually verify that your page is valid (as shown in Figure 2.4). You may or may not get a warning, but some of the warning are relatively unimportant.

4. Remove the closing `</ul>` element, save, and then redo steps 1-3 of this exercise.

   *The page will **not** be valid and the service may find not one but many errors. At the time of writing, the validator lists the missing `</ul>` element as error number 10. Thus, while a validator can help you find an error in your markup, the error messages do take some interpretation.*

5. Put the closing `</ul>` tag back in, save, and re-validate.

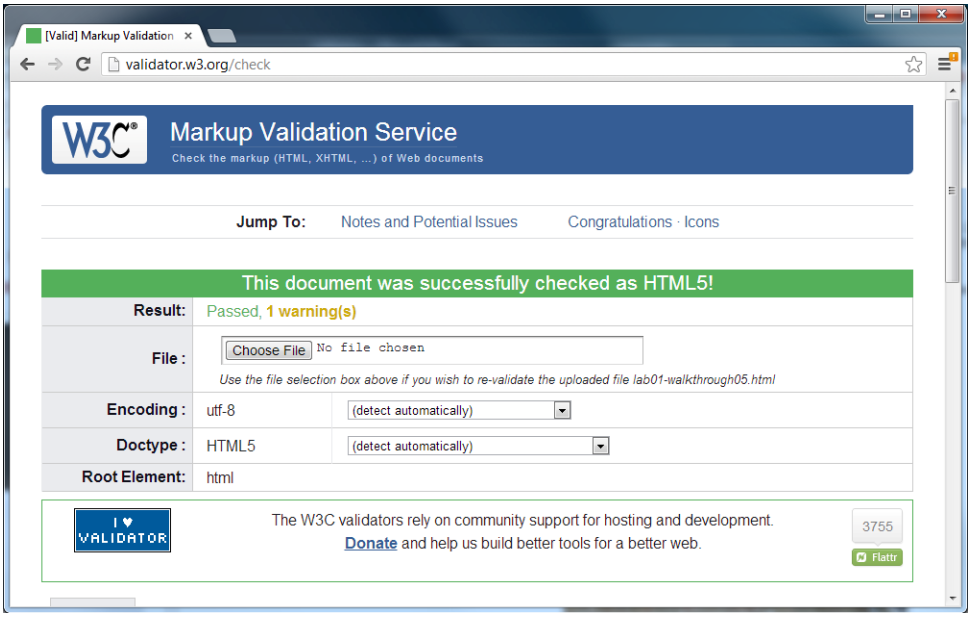

Figure 2.4 – Using a validation service