LAB 3

HTML 1: Introduction

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, 3rd Ed

Randy Connolly and Ricardo Hoar

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

Date Last Revised: February 7, 2022

Quick Tour of HTML

Preparing Directories

The starting `lab03` folder has been provided for you (within the zip file/folder downloaded from Gumroad).

1 If you haven't done so already, create a folder in your personal drive for all the labs for this book.

2 Copy the folder titled `lab03` from the zip file/folder to your course folder.

Note: these labs use the convention of `blue background` *text to indicate filenames or folder names and* ***`bold red`*** *for content to be typed in by the student.*

Now we are ready to create our first web page.

Exercise 3.1 — First Web Page

1 Using some type of text or HTML editor (such as Notepad, Visual Code, Brackets, Sublime, 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 `lab03-ex01.html` in your `lab03` folder (the folder you just created in the Preparing Directories step above).

3 Start up **FireFox**, **Chrome**, **Safari** or some other browser. Open the file `lab03-ex01.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 3.2 — Additional Structure Tags

1 Create a new HTML document with the following content:

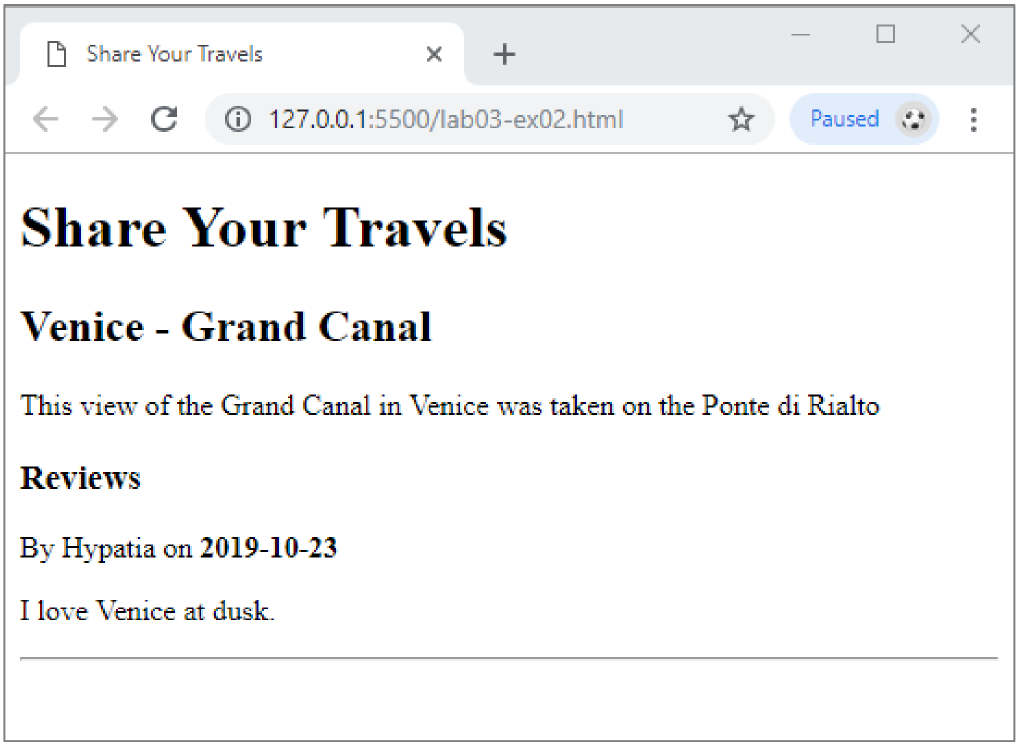
```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="utf-8">
   <title>Share Your Travels</title>
</head>
<body>
   <h1>Share Your Travels</h1>
   <h2>Venice - Grand Canal</h2>

   <p>This view of the Grand Canal in Venice was taken on the
      Ponte di Rialto
   </p>

   <h3>Reviews</h3>
   <div>
      <p>By Hypatia on <strong>2019-10-23</strong></p>
      <p>I love Venice at dusk.</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 `lab03-ex02.html` and test in browser. The result should look similar to that shown in Figure 3.1.



Share Your Travels

Venice - Grand Canal

This view of the Grand Canal in Venice was taken on the Ponte di Rialto

Reviews

By Hypatia on **2019-10-23**

I love Venice at dusk.

Figure 3.1 – Exercise 2 Complete

EXERCISE 3.3 — MAKING MISTAKES

1 Open `lab03-ex03.html` (which has the same content as the last exercise).

2 Before the text “`Ponte`” (in the paragraph tag), add the tag `<randy>`.

Your editor may automatically add a closing `</randy>` tag. No need to remove it since we are just testing things out.

3 Save and then test in browser. After testing, remove the `<randy>` tag (and end tag if your editor inserted it).

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, you might think the browser assumes that the content after it should continue being displayed as a first-level heading. When the browser encounters the `<h2>` tag, it implicitly closes the `<h1>` element.

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 3.4 — Linking

1 Open lab03-ex04.html and add the following bolded text:

```
<p>This view of the Grand Canal in Venice was taken on the
<a href="https://en.wikipedia.org/wiki/Rialto_Bridge">
Ponte di Rialto</a>
</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 view of the Grand Canal in <a
href="venice.html">Venice</a> was taken on the
<a href="https://en.wikipedia.org/wiki/Rialto_Bridge">
Ponte di Rialto</a>
</p>
```

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

Adding Images

Exercise 3.5 — Adding Images

1 Add the following tag to your file from the previous exercise and then test:

```
<img src="images/venice.jpg" alt="Venice" />
<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/venice.jpg" alt="Venice" title="Venice" />
```

The title attribute is used to display a tooltip.

3 Modify the image tag as follows and test.

```
<img src="images/venice.jpg" alt="Venice" title="Venice"
width="100" />
```

This changes the width of the displayed image to just 100 pixels. Notice that the height is changed as well to maintain its original aspect ratio. The size of an image can also be altered via CSS.

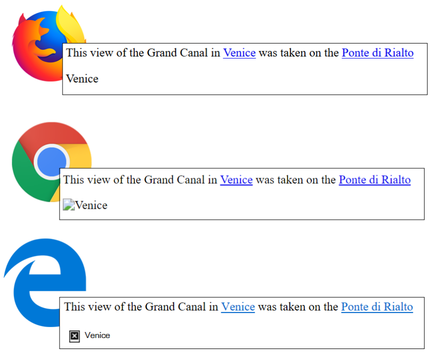
4 Change the `src` attribute to the following (i.e., add a slash before the folder name) and remove the width attribute then test.

```
<img src="/images/venice.jpg" alt="Venice" title="Venice" />
```

If you are viewing this file on a web server (or even localhost) then nothing will change; if you are viewing this file directly in browser, then you will likely no longer see the image. Why? Because the root reference does not work when tested locally from the file system.

Also, depending on the browser, you may or may not see a missing image icon, as shown in Figure 3.2. Notice that all three of the browsers in the Figure 3.2 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.



This view of the Grand Canal in Venice was taken on the Ponte di Rialto

Venice

This view of the Grand Canal in Venice was taken on the Ponte di Rialto

Venice

This view of the Grand Canal in Venice was taken on the Ponte di Rialto

Venice

Figure 3.2 – Missing image indication in different browsers

5 Remove the slash added in step 4.

6 Add the following and then test:

```
<a href="venice.html">
<img src="/images/venice.jpg" alt="Venice" title="Venice" />
</a>
```

This turns the Venice image into a link (in this case, a link to another page).

7 Modify the link as follows and then test:

```
<a href="images/large-venice.jpg">
<img src="/images/venice.jpg" alt="Venice" title="Venice" />
</a>
```

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

8 Add the following after the Venice image and test:

```
<a href="images/large-venice.jpg">
<img src="/images/venice.jpg" alt="Venice" title="Venice"  />
</a>
<div>
  <h3>Related Images:</h3>
  <img src="images/small-v1.jpg" alt="More Venice" />
  <img src="images/small-v2.jpg" alt="More Venice" />
  <img src="images/small-v3.jpg" alt="More Venice" />
</div>
```

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

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

```
<div>
  <h3>Related Images:</h3>
  <img src="images/small-v1.jpg" alt="More Venice" /><img
src="images/small-v2.jpg" alt="More Venice" /><img
src="images/small-v3.jpg" alt="More Venice" />
</div>
```

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 3.3.

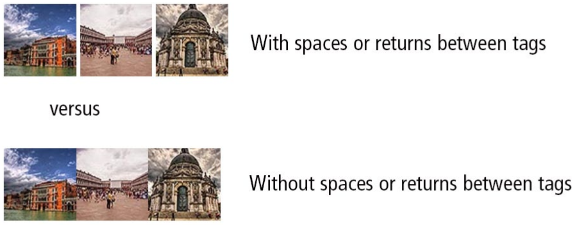

With spaces or returns between tags
versus
Without spaces or returns between tags

Figure 3.3 – Carriage return treated as a space

10 Add a line break as shown in the following and test.

```
<p>This view of the Grand <br>Canal in …
```

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 3.6 — Making a List

1 Open `lab03-ex06.html` and add the following bolded text:

```
<h1>Continents</h1>
<ul>
   <li>Africa</li>
   <li>Asia</li>
   <li>Australia</li>
   <li>Europe</li>
   <li>North America</li>
   <li>South America</li>
</ul>
```

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 3.7 — Linking with Lists

1 Continue working with `lab03-ex06.html` and add the following link to each element in the list and test:

```
<ul>
  <li><a href="#">Africa</a></li>
  <li><a href="#">Asia</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="#africa">Africa</a></li>
  <li><a href="#asia">Asia</a></li>
  <li><a href="#europe">Europe</a></li>
  <li><a href="#north">North America</a></li>
  <li><a href="#south">South America</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="#africa">Africa</a></li>
  <li><a href="#asia">Asia</a></li>
  ...
</ul>
<h2 id="africa">Africa</h2>
...
<h2 id="asia">Asia</h2>
...
<h2 id="europe">Europe</h2>
...
<h2 id="north">North America</h2>
...
<h2 id="south">South America</h2>
```

4 Test by clicking on links in the continent list.

You may need to shrink the vertical size of your browser to see these relative links work.

5 Add the following code and test.

```
<h2 id="africa">Africa</h2>
<h3>Countries</h3>
<ul>
   <li>Northern
      <ul>
         <li>Egypt</li>
         <li>Morocco</li>
         <li>Tunisa</li>
      </ul>
   </li>
   <li>Central
      <ul>
         <li>Ghana</li>
         <li>Nigeria</li>
         <li>Somali</li>
      </ul>
    </li>
</ul>
```

This adds a nested bulleted list.

6 As shown in the previous exercise, list items can contain other lists. In fact, a list item can contain other block and online elements. Modify your existing list as follows and test.

```
<li>Central
  <ul>
    <li>
      <h4>Ghana</h4>
      <img src="images/flag.svg" width="100"
           alt="flag of Ghana" />
      <p>Ghana is a country located along the Gulf of Guinea
and
         Atlantic Ocean.</p>
      <p>Ghana became independent of the United Kingdom on 6
         March 1957.</p>
    </li>
    <li>Nigeria</li>
    <li>Somali</li>
  </ul>
</li>
```

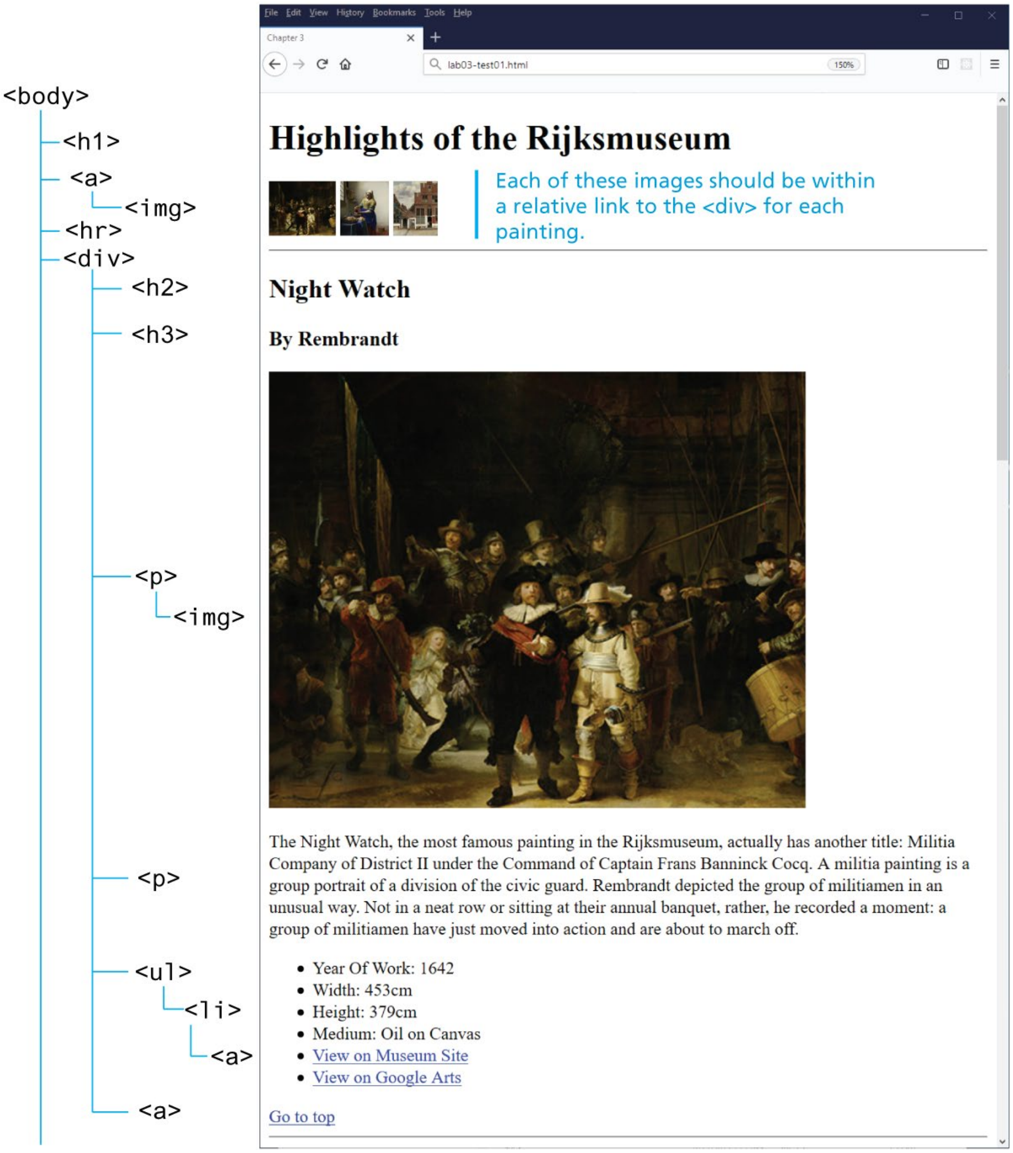
Test Your Knowledge #1

The file `ch03-test01.html` contains text content: you will be adding in HTML tags so that it looks similar to that shown in Figure 3.4.

1 Add in the appropriate structure tags (html, head, body).

2 Each painting is its own `<div>`. Figure 3.4 indicates the appropriate tags to use.

3 Finally, turn the small thumbnail images at the top into links to the `<div>` for that painting. At the end of each `<div>`, add another link that jumps to the top of the page. Examine Figure 3.15 in the book (or Exercise 3.7 in the lab) for guidance on this step.



<body>
<h1>
<a>
<img>
<hr>
<div>
<h2>
<h3>
<p>
<img>
<p>
<ul>
<li>
<a>
<a>
File Edit View History Bookmarks Tools Help
Chapter 3
lab03-test01.html
150%
Each of these images should be within a relative link to the <div> for each painting.

Highlights of the Rijksmuseum

Night Watch

By Rembrandt

The Night Watch, the most famous painting in the Rijksmuseum, actually has another title: Militia Company of District II under the Command of Captain Frans Banninck Cocq. A militia painting is a group portrait of a division of the civic guard. Rembrandt depicted the group of militiamen in an unusual way. Not in a neat row or sitting at their annual banquet, rather, he recorded a moment: a group of militiamen have just moved into action and are about to march off.

- Year Of Work: 1642
- Width: 453cm
- Height: 379cm
- Medium: Oil on Canvas
- View on Museum Site
- View on Google Arts

Go to top

Figure 3.4 – Completed Test Your Knowledge #1

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 3.8 — Header and Footer

1 Open `lab03-ex08.html` and test.

2 Add the following (some existing markup omitted) and test.

```
<body>
<header>
   <img src="images/logo.svg" height="100">
   <h3>Gallery Viewer</h3>
   <ul>
      <li><a href="#"><img src="images/home.svg"...
      <li><a href="#"><img src="images/catalog.svg" ...
      <li><a href="#"><img src="images/map.svg" ...
      <li><a href="#"><img src="images/search.svg" ...
   </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 add unique styles to the header.

3 At the end of our document, replace the final `<div>` with `<footer>` elements as shown in the following and test.

```
<footer>
<p>Copyright &copy; 2020 Gallery Viewer</p>
Icons made by ...
</footer>
</body>
```

Like the `<header>` element, the `<footer>` element has no built in style. Notice as well the `&copy;` character entity, which adds the copyright symbol.

4 Modify the footer as follows and test.

```
<footer>
  <p><strong>Copyright &copy; 2020 Gallery Viewer</strong></p>
```

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

5 Modify the footer as follows and test.

```
<p><em>Copyright &copy; 2020 Gallery Viewer</em></p>
```

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

Exercise 3.9 — Semantic Containers

1 Continue working with `lab03-ex08.html` and add the following and test.

```
<header>
   <img src="images/logo.svg" height="100">
   <h3>Gallery Viewer</h3>
   <nav>
      <ul>
         <li><a href="#"><img src="images/home.svg" ...
         <li><a href="#"><img src="images/catalog.svg" ...
         <li><a href="#"><img src="images/map.svg" ...
         <li><a href="#"><img src="images/search.svg" ...
      </ul>
   </nav>
</header>
```

Like with the other HTML5 semantic elements, there is no special browser formatting for the <nav> element (and the others in this exercise). They are used purely to make our markup clearer.

2 Modify your code as follows (some code omitted).

```
</header>
<article>
   <h1>J. Paul Getty Museum</h1>
   <section>
      <h2>Overview</h2>
      <p>The J. Paul Getty Museum at the ...
   </section>
   <section>
      <h2>Details</h2>
      http://www.getty.edu/museum/
      1200 Getty Center Drive
      ...
   </section>
   <section>
      <h2>Highlights</h2>
      <div>
         <img src="images/getty1.jpg" alt="by Chardin" /> ...
      </div>
      <div>
         <img src="images/getty2.jpg" alt="by David" /> ...
      </div>
      <div>
         <img src="images/getty3.jpg" alt="by Canaletto" /> ...
      </div>
   </section>
</article>
<footer>
```

3 Change the `<article>` tags to `<main>` tags and test,

In this example, it might make more semantic sense to use a `<main>` *element instead of* `<article>`*. As you can see, it doesn't affect what appears in the browser.*

4 Add the following markup to the details information and test.

```
<section>
   <h2>Details</h2>
   <address>
      <a href="http://www.getty.edu/museum/">
         http://www.getty.edu/museum/</a><br />
      1200 Getty Center Drive<br />
      Los Angeles, CA 90049-1687<br />
      <a href="tel:+310-440-7330">+1 (310) 440-7330</a><br />
      <a href="gettymuseum@getty.edu">gettymuseum@getty.edu</a>
   </address>
</section>
```

Exercise 3.10 — Figure and Captions

1 Continue working with `lab03-ex08.html` and then replace the `<div>` around the painting images with `<figure>` elements as shown in the following. Test.

```
<section>
   <h2>Highlights</h2>
   <figure>
      <img src="images/getty1.jpg" alt="by Chardin" />
      <figcaption><em>Still Life with Peaches, a Silver Goblet,
Grapes, and Walnuts</em> (1759) by Jean-Baptiste-Simeon
Chardin</figcaption>
   </figure>
   <figure>
      <img src="images/getty2.jpg" alt="by David" />
      <figcaption><em>Portrait of the Sisters Zenaide and
Charlotte Bonaparte</em> (1880) by Jacques-Louis
David</figcaption>
   </figure>
   <figure>
      <img src="images/getty3.jpg" alt="by Canaletto" />
      <figcaption><em>The Grand Canal in Venice from Palazzo
Flangini to Campo San Marcuola</em> (1738) by Giovanni
Canaletto</figcaption>
   </figure>
</section>
```

Here's a surprise ... there is in fact additional browser formatting for the `<figure>` *and* `<figcaption>` *elements. 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.*

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 3.11 — Validating HTML

1 Open a browser and go to `https://validator.w3.org`

2 In the **Validate By File Upload** tab, click the **Browse** or **Choose File** button and choose your `validate.html` file.

3 Click the **Check** button.

The site should indicate that this file contains several errors. Some are quite clear: for instance, the `randy` attribute and the `<ricardo>` element are at the time of writing not valid HTML.

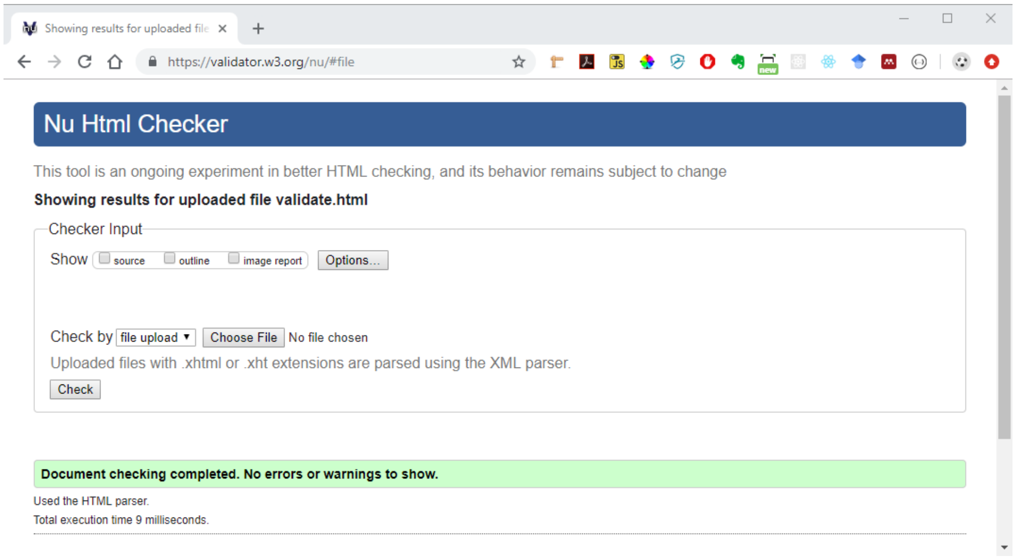
4 Remove the `randy` attribute and change the `<ricardo>` tags back to `<figure>` tags. Also add the `alt` attribute to the image indicated by the validator.

5 Try re-validating the edited file.

There will be two errors listed. The first error message is misleading. The real problem is indicated by the second error, namely, that the <ul> element is missing a close </ul>.

6 Fix the file by adding in the missing `</ul>` and then re-validate.

The file should have no errors (as shown in Figure 3.5).



Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for uploaded file validate.html

Checker Input

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 9 milliseconds.

Figure 3.5 – Using a validation service