What is HTML?

HTML is short for HyperText Markup Language. It’s the primary markup language for creating web pages that are to be displayed in a web browser. A “webpage” is a coded document, created using html tags and saved with the .html file extension.

HTML “tags” provide structure in creating web pages. These tags, combined with a plain text editor, can be used to create documents with the .html file extension. Text and images can be displayed within a webpage, and formatted using another style sheet language called CSS. All web pages are viewable with a web browser and publishable via the World Wide Web. The HTML layer is used for overall structure and CSS is used for presentation. We will first focus on HTML, and it’s basic elements.

A basic text-editor, such as Notepad, is all that you need in order to get started in creating a webpage. Dreamweaver is also a valuable tool, since it offers code syntax highlighting and file management. It’s important to learn the structure and syntax of HTML manually, and then use tools like Dreamweaver to expedite the production process.

Basic Tags:

A basic HTML document includes the following tags, which make up a framework that will be used to build out the rest of the page:

```
<!DOCTYPE html>  This is called the “DocType”
<html>          The HTML tag defines the page
<body>         The Body tag defines visible content
Some initial content goes here.

</body>

</html>         Notice the opening and closing tags: “<>” and </>
```

Create a basic html page with these tags. Save it using the .html extension, and open it in a web browser of choice.

The appearance of the page will be very minimal, but the purpose of HTML is to apply structure, not presentation or style, and this example defines the basic building blocks of a web page.

The first line of code, <!DOCTYPE html>, is a document type declaration that lets browser know which version of HTML you’re using. This specific Doctype is specified as HTML5. Previous versions of HTML may be declared by specifying a different (and much longer) doctype.

The doctype is a must - If you don’t include it, browsers will work in “quirks mode” and not deliver consistent results.

The <html> tag is the opening tag that represents the root of an HTML document. The </html> closing tag is an important part of an HTML document. The markup between the opening <body> tag, and the closing </body> tag is the visible content of the document that will appear in the browser window.

Most tags require a separate closing tags. Some are self-terminating, such as the line-break tag: <br /> and the horizontal rule <hr /> tag. You’ll notice that the Doctype tag is also self-terminating.

Attributes:

Tags can me modified with attributes, are modifier of HTML elements. Attributes appear within opening tags and their values are listed in quotation marks.

```
<tag attribute="value">content</tag>
```
Elements

Tags mark the beginning and end of an element. An HTML tag is composed of the name of the element, surrounded by angle brackets. An end tag also has a slash after the opening angle bracket, to distinguish it from the start tag.

Example: everything in between (and including) the <body> and </body> tags contain the body element. An example Title element would look like this:

<title>This is the Title of my first Webpage</title>

Page Titles

To add this title to your page, nest it within the <head> tag:

```html
<!DOCTYPE html>
<html>
<head>
<title>David Hardy: Graphic Designer</title>
</head>
<body>
Body Content Goes Here.
</body>
</html>
```

The Title tag defines the title of the document

The <title> element:
• defines a title in the browser toolbar
• provides a title for the page for bookmarks
• displays a title for search-engine results

The head <head> opening tag </head> and closing tag appears before the start of the body element <body> and contains information about the page.

If you save and preview this document in the browser (save and reload), you will see that whatever was indicated in the title tag will appear on a tab or the title bar of the window (browser chrome).

The text that you defined between the title tags has become the title of the document. If you were to bookmark this page, you would see that the title is also used.

Paragraphs

Type another line of content under the line that reads “Body Content Goes Here” and refresh the page.

You might have expected your document to appear as you typed it, on two lines. The reason they appear on a single line is because web browsers don't interpret white space in any significant way.

In order for the text to appear on different lines, they must be assigned some sort of semantic value.

Instead of arbitrary text inside a <body> element, inserting the copy inside of paragraph tags makes more sense.

```html
<p>Body Content Goes Here.</p>
<p>This is a second paragraph.</p>
```

The <p> tag is used for paragraphs. The two lines will now appear on two lines because the browser recognizes them as separate paragraphs.
**Emphasis**

You can emphasize text in a paragraph using `em` (emphasis) and `strong` (strong importance).

<p>The book <em>On The Road</em> is considered a defining work of the postwar <em>Beat Generation</em>.</p>

<p>The deadline for submissions is noon on <strong>October 1st</strong> and any applications received after this date will not be considered.</p>

Traditionally, browsers will display em in *italics* and strong in **bold** by default. However, the markup has changed in recent years, so `em` and `strong` don’t necessarily mean italic or bold. These are semantic connotations, separate from the visual realm associated with italic and bold type.

These two tags are holdovers from HTML 4:

The `<i>` tag defines a part of text in an alternate voice or mood. The content of the `<i>` tag is usually displayed in italic.

The `<b>` tag specifies bold text. According to the HTML 5 specification, the `<b>` tag should be used as a LAST resort when no other tag is more appropriate.

**Line breaks**

The line-break tag can also be used to separate lines of copy like this:

Body Content Goes Here. `<br />`

This is a second line.

There’s no content involved in breaking lines so as a result, there’s no closing tag. Line breaks should only be used to set type in the same paragraph to the next line, to format addresses, etc. `<br />` tags should never be used to separate two blocks of content. A `<p>` tag would be more suitable in that instance.

**Headings**

There are specific tags used to indicate page headings in HTML. They are, in order of importance: `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`

The `<h1>` tag should only be used once per page, as the main heading of that page. `<h2>` to `<h6>` can be used as often as desired, but they should always be used in order, as they were intended.

For example, an `<h4>` tag should be used as a sub-heading of an `<h3>` tag, which should be a sub-heading of an `<h2>` tag.

```html
<!DOCTYPE html>
<html>
<head>
<title>My first web page</title>
</head>
<body>
<h1>Web Design for Beginners</h1>
<h2>HTML Explained</h2>
<p>A page put together for the purpose of learning HTML</p>
</body>
</html>
```

**Lists**

There are three types of list: unordered lists, ordered lists and definition lists. We will initially examine the first two types.

Unordered lists and ordered lists work similarly. The former is used for non-sequential lists with list items preceded by *bullets*. The latter is for sequential lists, which are represented by *incremental numbers*.

The `<ul>` tag is used to define unordered lists and the `<ol>` tag is used to define ordered lists. Inside the lists, the `<li>` tag is used to define each list item.

```html
<ul>
<li>List Item One</li>
<li>List Item Two</li>
<li>List Item Three</li>
</ul>
```
If you look at this in your browser, you will see a bulleted list. Simply change the `<ul>` tags to `<ol>` and you will see that the list will become numbered.

```
<ol>
  <li>List Item Number One</li>
  <li>List Item Number Two</li>
  <li>List Item Number Three</li>
</ol>
```

Lists can also be nested within lists to form indented items. You can mix and match ordered/unordered.

```
<ul>
  <li>Bullet Point</li>
  <ol>
    <li>List Item Number One</li>
    <li>List Item Number Two</li>
  </ol>
</ul>
```

Links

An element in an a webpage that links to another place in the same document or to an entirely different document is known as a hyperlink.

Hyperlinks are the “web” that holds the world the internet together, linking sites to other sites.

An anchor tag `<a>` is used to define a link, but you also need to add something to the anchor tag - the destination of the link.

The destination of the link is defined in the “href” attribute that’s added to the tag. The link can be absolute (linking to another site) or relative (linking to text in the same webpage).

A link can link to any file on the worldwide web. The link below is nested within a paragraph `<p>` tag:

```
<p><a href="http://www.csn.edu">CSN</a></p>
```

A link can also link to another part of the same page. By adding an id attribute to a tag, i.e.

```
<p id="test">Test</p>
```

That tag can be accessed from an href link:

```
<a href="#test">Go to Test</a>.
```

Clicking the link will scroll to the element with that ID.

Images

In the early days of the world-wide web, the internet was text-only. In 1993, Marc Andreessen wrote:

I’d like to propose a new, optional HTML tag:

```
IMG
```

Required argument is src="url".

Mosaic was the first browser to process images, and the modern day internet was born.

The `<img>` tag defines an image in an HTML page. It has two required attributes: src and alt.

Images are not actually embedded into an HTML page, they are linked to HTML pages. The `<img>` tag creates a holding space for the referenced image.

```
<img src="http://www.csn.edu/images/csn/csnlogo.gif" />
```

To link an image to another document, simply nest the `<img>` tag inside `<a>` tags.

```
<p>
  <a href="http://www.csn.edu">
    <img src="http://www.csn.edu/images/csn/csnlogo.gif" />
  </a>
</p>
```

The most commonly used file formats used for images are JPEGs, GIFs, and PNGs. Images on the internet are typically saved as 72 dpi resolution.

Adobe software such as Photoshop has a “Save to Web” option for exporting optimized images for use on the internet.

The src attribute tells the browser where to find the image. This can be absolute (a full URL address) or relative (as in ... locally relative to the document you’re currently working in).
Example:

An **absolute** link looks like this

```html
<img src="http://www.csn.edu/picture.jpg" />
```

A **relative** link looks like this

```html
<img src="images/picture.jpg" />
```

The width and height attributes are recommended for optimal page-load speed (or to display a larger image at a smaller size).

```html
<img src="images/picture.jpg" width="100" height="50" />
```

The alt attribute is the alternative description. This is an accessibility consideration, providing meaningful information for users who are unable to see the image (if they are visually impaired, for example).

```html
<img src="images/picture.jpg" width="100" height="50" alt="tree" />
```

Note that, like the `<br>` tag, no closing tag is required for the `<img>` tag. It’s self-terminating.

---

**Tables**

HTML tables are still best known for being used to layout the overall design structure of webpages in the early days.

Now, we use standards-based HTML combined with Cascading Style Sheets (CSS), to layout the page.

Tables are still used on the web to structure tabular data; to display items in an excel-like grid of rows and columns. The markup for table rows and cells can get quite lengthy if the table is large.

The `<table>` element defines the table.

The `<tr>` element defines a table row.

The `<td>` element defines a data cell. These must be nested within the `<tr>` tag.

If you construct a 3x3 table (9 cells), there should be three `<tr>` elements to define each row and three `<td>` elements within each of the rows, totalling in nine `<td>` elements.

```html
<table>
  <tr>
    <td>Row 1, cell 1</td>
    <td>Row 1, cell 2</td>
    <td>Row 1, cell 3</td>
  </tr>
  <tr>
    <td>Row 2, cell 1</td>
    <td>Row 2, cell 2</td>
    <td>Row 2, cell 3</td>
  </tr>
  <tr>
    <td>Row 3, cell 1</td>
    <td>Row 3, cell 2</td>
    <td>Row 3, cell 3</td>
  </tr>
</table>
```

---

**Forms**

Forms are used to collect data inputted by a user. They can be used to send data across the web.

Forms are used in conjunction with a programming language to process the information inputted by the user. Forms on the web use a server-side language (such as PHP) to deliver the data inputted to a database or end user.

```html
<form action="script.php" method="post"></form>
```

The `<form>` element defines the form. If you are using a form for a user to submit information, an `action` attribute within this is needed to send the data.

The `method` attribute tells the form how the data is going to be sent.

The options are `get`, which is default, and pushes the form information (via a URL variable) to a web address, or `post`, which sends the form’s information (appends form-data inside the body of the HTTP request).

```html
<form action="script.php" method="post"></form>
```
**Input**

The `<input>` tag can take a multitude of “types,” the most common of which are outlined below:

- `<input type="text">` or simply `<input>` is a standard textbox. This can also have a value attribute, which sets the initial text in the textbox.

- `<input type="password">` using this input will result in the characters inputted shown as hidden (bullets).

- `<input type="checkbox">` is a checkbox, which can be toggled on and off by the user. This can also have a checked attribute `<input type="checkbox" checked>` The optional attribute doesn't require a value.

- `<input type="radio">` is similar to a checkbox, but the user can only select one radio button in a group. This can also have an optional checked attribute (boolean).

- `<input type="submit">` is a button that when selected will submit the form. You can control the text that appears on the submit button with the value attribute, for example `<input type="submit" value="Send">`.

**textarea**

The `<textarea>` tag results in a large, multi-line textbox. The anticipated number of rows and columns can be defined with rows and cols attributes, although you can also manipulate the size using CSS (recommended).

- `<textarea>Room for text</textarea>`

- `<textarea rows="5" cols="20">Lots of room for text</textarea>`

Any text you choose to place between the opening and closing tags will form the initial value of the text area.

**select**

The `<select>` tag works together with the `<option>` tag to make (drop-down) select boxes.

- `<select>`
  - `<option>Option 1</option>`
  - `<option value="third option">Option 3</option>`
- `<select>`

When the form is submitted, the value of the selected option will be sent. This default value is the text between the opening and closing `<option>` tag unless an explicit value is specified with the value attribute, in which case this will be sent instead.

In the previous example, if the first item is selected, “Option 1” will be sent, if the third item is selected, “third option” will be sent.

Similar to the checked attribute of checkboxes and radio buttons, an option tag can also have a selected attribute, to begin with one of the items already being selected:

- `<option selected>Test</option>` would pre-select “Test” from a list of other options

**name**

The name attribute is used to reference form data after a form is submitted.

They are necessary for using a form-handling script. The form data needs name fields to associate the corresponding data with.

```html
<form action="contactus.php" method="post">
  <p>Name:</p>
  <p><input type="text" name="name" value="Your Name"></p>
  <p><textarea name="comments" rows="5" cols="20">Please choose</textarea></p>
  <p><input type="radio" name="choice" value="red"> Male</p>
  <p><input type="radio" name="choice" value="blue"> Female</p>
  <p><input type="submit"></p>
</form>
```

These are the basics of a HTML document. In the next module, we will explore some intermediate elements such as `<span>` and `<div>`, Meta tags, Definition lists, and Sections.