

Introduction to HTML

HTML Basics: Tags and Elements

HTML stands for **Hyper Text Markup Language**. We are now on the fifth revision of the HTML specification (hence **HTML5**). Although often referred to as “code” when used alongside CSS and other web scripting languages, HTML is not a programming language. It instead is a markup language composed of **markup tags** that describe the content of a webpage. HTML therefore is said to “structure” otherwise unstructured text and image content, while CSS then “styles” this structure. The majority of HTML files are stored in plain text files with an *.html* extension (e.g., *index.html*).

HTML **tags** are keywords composed of reserved words surrounded by angle brackets, such as `<html>` or `<body>`. Often these tags come in pairs designating a **start tag** and an **end tag** (also referred to as an **opening tag** and a **closing tag**). The end or closing tag is written with a forward slash before the tag name ([Example 1](#)). An HTML **document** is composed of HTML tags written in plain text. An HTML document is therefore synonymous with a **webpage**.

```
<h1></h1> <!-- a start and end heading tag -->
<p></p> <!-- a start and end paragraph tag -->
```

Example 1: Example HTML Tags

Tags are written in lowercase (e.g., `<p>` instead of `<P>`). In [Example 1](#), there is text written between the sequence of `<!-- -->`. This is an HTML **comment** and will not be displayed by the browser. Comments are used sparingly in HTML, as opposed to other scripting and programming languages.

An HTML **element** includes the HTML tags, the content contained within the tags, and other element **attributes** (more about attributes below). [Example 2](#) shows two HTML elements that does not include any attributes.

```
<h1>This is a heading</h1>
<p>This is a paragraph tag</p>
```

Example 2: Example HTML Elements (No Attributes)

Text Editors and Web Browsers

You compose HTML using a **text editor** and then preview your markup using a **web browser**. Different editors include various features, such as color-coding specific to different code languages, automatic indentation and closing tags, and live preview. Since these are updated with new features regularly, now is a good time to review your choice of editor and compare it to other available editors to see if it may be worth switching. Some popular editors as of this writing are:

- [Notepad++](#)

- [Sublime Text](#)
- [Aptana Studio](#)
- [Brackets](#)
- [Atom](#)

You may use the text editor with which you are most comfortable. To view your markup, simply drag your *.html* file into a web browser!

Setting Up Your Directory

A webpage is just a collection of files that sits on a server that is accessed through a **URL (Uniform Resource Locator)** using a web browser. Thus, the first step in designing a website is setting up a working **directory** (i.e., a folder) to organize these files, which eventually will be published to a web server (e.g., GitHub pages).

The landing page of your website must be named *index.html*, as modern browsers will read this file by default when navigating to your URL. For instance, navigation to <https://geography.wisc.edu/cartography/> reads the *index.html* file in the */cartography/* folder on the geography server. The *index.html* file must be located at the root, or top level of your web directory. Web directories often have the following **subdirectories** (i.e., subfolders) at the root level:

1. *css* for holding stylesheets that define the look and feel of your content (e.g., *style.css*);
2. *js* for holding scripts that define the custom behavior of interactive elements on your page (e.g., *portfolio.js*);
3. *lib* for holding external code libraries leveraged by the webpage (e.g., *jquery.js*);
4. *assets* for images, documents, and other files that you are embedding in or distributing through your website (here, combining *img* and *docs* folders)

The HTML Boilerplate

The *index.html* document is built from a standard **boilerplate** or required set of tags needed for a valid HTML5 document. Example 3 provides the Begin by typing the code included in **Example 3** within your Notepad++ Editor. This HTML includes the minimum required tags for a valid HTML5 document, plus placeholder content (the `<h1>` and `<p>` elements).

The odd looking tag in **Example 3, Line 1** (`<!DOCTYPE HTML>`) is known as a **DOCTYPE** declaration and defines the document type as HTML (HTML5 is implied by this DOCTYPE declaration). Versions of HTML prior to HTML5 had different DOCTYPE declarations, and were often much more complicated. When looking at the code of other websites on the web, be careful to note the DOCTYPE declaration and be aware that some elements valid for previous versions of HTML may no longer be valid in HTML5.

The HTML5 boilerplate first defines an **html** (`<html>`) element. The head (`<head>`) and body (`<body>`) tags then are enclosed, or **nested**, within the html (`<html>`) element. This idea of nesting implies that heading (`<head>`) and paragraph (`<body >`) elements are **children** of the **parent** element body (`<html>`) within the overall **document tree**. Nesting is an important way in which content is organized in the layout of your webpage.

```
<!DOCTYPE HTML>
<HTML>

<head>
  <title>Title of the document</title>
</head>

<body>
  <h1>My First Heading</h1>

  <p>My first paragraph. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  Ut et lorem in ante vestibulum fringilla a at sapien. Quisque porta dolor orci,
  vel convallis est. In accumsan pretium tempus. Etiam malesuada, nisl non
  posuere scelerisque, massa urna pulvinar metus, at euismod velit nulla quis
  risus.</p>

</body>

</HTML>
```

Example 3: The HTML Boilerplate with Placeholder Title, Heading, and Paragraph Tags

The head (`<head>`) and body (`<body>`) tags perform different roles in the HTML boilerplate. The required `<head>` element contains references to CSS stylesheets, JS scripts, and metadata about the HTML document (refer to Tutorial 2 for details about linking to stylesheets and scripts). In [Example 3](#), a `<title>` element is defined, which sets the title displayed for the browser toolbar and search engine results. In contrast, the `<body>` elements structure the content that will be rendered in the webpage. Here, heading `<h1>` and paragraph `<p>` element are defined, which will render the included text content differently in a browser ([Figure 1](#)).

My First Heading

My first paragraph. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut et lorem in ante vestibulum fringilla a at sapien. Quisque porta dolor orci, vel convallis est. In accumsan pretium tempus. Etiam malesuada, nisl non posuere scelerisque, massa urna pulvinar metus, at euismod velit nulla quis risus.

Figure 1: The HTML Boilerplate Rendered in a Browser

The whitespace between HTML elements is for human readability only. The web browser ignores whitespace; this means that the browser would render the html in [Example 5](#) the same as the HTML boilerplate in [Example 3](#). An additional blank lines is added using the *line break* (`
`) while additional blank character is added inline to text using the ` ` escape character.

```
<!DOCTYPE HTML><HTML><head><title>Title of the document</title></head><body><h1>My
First Heading</h1><p>My first paragraph. Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Ut et lorem in ante vestibulum fringilla a at sapien. Quisque porta
dolor orci, vel convallis est. In accumsan pretium tempus. Etiam malesuada, nisl non
posuere scelerisque, massa urna pulvinar metus, at euismod velit nulla quis
risus.</p></body></HTML>
```

Example 4: The HTML Boilerplate with Whitespace Stripped

Attributes and Common HTML Tags

Many HTML tags include *attributes* that allow you to structure the element in multiple ways. When combining HTML and CSS (see Tutorial 2), some attributes become properties that can be consistently restyled through CSS declarations. Attributes are defined within the opening tag using an equals sign to set the attribute value and double quotations around the attribute value to distinguish it from other text (e.g., `<p align="left">`). A commonly used attribute across HTML tags is the *unique identifier* (`id`) attribute, which gives an element a name of your choosing for reference with an *id selector* (i.e., the hashtag, `#`, notation) in linked stylesheets. You should be familiar with referencing unique identifiers from CartoCSS.

The following provides a description of commonly used HTML tags and their attributes. [Example 5](#) makes use of these tags in a functional website that can be explored in a browser.

Paragraphs

Paragraph length text in your website is structured using the *paragraph* (`<p>`) tag. Its primary attribute is `align`, which sets the justification to `left`, `right`, `center`, or `justify`.

Headings

Titles are created using the *heading* (`<h1>`) tag. The HTML5 specification defines six levels of headings, ranging from `<h1>` to `<h6>`. The six heading levels therefore are a method for structuring a hierarchy of content within your document. It is convention to use the first level heading (`<h1>`) element only once in a document, with the `<h1>` element being the most important and `<h6>` elements the least important. All heading tags also have an `align` attribute.

Lists

Lists are created using the *ordered list* (``) and *unordered list* (``) tags. Ordered lists use numbers, while unordered lists use bullets. The type of number or bullet used for the lists is defined through the `type` attribute. Items within ordered and unordered lists are structured using the *list item* (``) tag.

Anchors and Hyperlinks

Hyperlinking among text files is *the* foundational concept of the web (i.e., it is a web of hyperlinks in text files!); a website often consists of multiple *.html* files that link to each other. Hyperlinks are added to text passages using the *anchor* (`<a>`) tag. The hyperlink URL is set using the `href` attribute. The hyperlink is applied to text within the anchor element.

Embedding Images

Images are embedded into the webpage using the **image** (``) tag. Similar to the anchor tag, the image is embedded by identifying the image's URL through the `src` attribute (not `href`). While many anchor elements set the `href` attribute to an **absolute path** (e.g., `https://geography.wisc.edu/cartography/people/img/directors/rob_roth.jpg`), it is good practice to link to images and other files on your webpage using a **relative path** (e.g., `img/directors/rob_roth.jpg` when setting the image `src`), as it makes the website easier to migrate to a new server. You also can link to images and files in a folder above the current folder using the `../` syntax.

Text Formatting

Some of the text formatting options provided in a word processor can be created through HTML tags. Commonly used text formatting tags include **bold** (``) versus **italic** (`<i>`), **strong** (``) versus **emphasis** (``) (these pairings will look similar unless you style them differently in CSS), and **superscript** (`<sup>`) versus **subscript** (`<sub>`), in addition to many more. Finally, a unique CSS style can be applied to section of text using the `style` attribute of a **span** (``) element.

Dividers

Layout organization of the main body of your website is organized using the **divider** (`<div>`) tag. The divider tag often is used as a **wrapper** element to enable the application of a consistent set of styles for all content within the wrapper (**Example 5**). Divider elements are essential to the **CSS Box Model** (see Tutorial 2), with the divider margin and padding styles updated automatically through the Bootstrap responsive framework (see the *Primer on Bootstrap*).

Additional Layout Organization

There are several additional layout tags new to HTML5 that are intended to provide a more meaningful structure to specific divider elements in the page. The **header** (`<header>`, and not to be confused with the `<header>` element) and **footer** (`<footer>`) tags are available to structure persistent content at the top and bottom of all webpages on your site (e.g., links to social media). The **navigation** (`<nav>`) tag organizes hyperlinks or buttons that enable navigation between website pages. Finally, the **section** (`<section>`) tag provides a higher level distinction among dividers, if needed to maintain clear page organization.

Explore the full set of HTML tags at the w3schools reference: <http://www.w3schools.com/tags/>!

```
<!DOCTYPE HTML>
<HTML>

<head>
  <meta charset="UTF-8"/>
  <title>My name | home</title>
  <meta name="keywords" content="buzzword, buzzword, buzzword"/>
</head>

<body>
  <div id="wrapper">
    <header></header>

    <h1>My name</h1>

    <nav>
      <ul>
        <li>home</li>
        <li>publications</li>
        <li>portfolio</li>
        <li>curriculum vitae</li>
        <li>contact</li>
      </ul>
    </nav>

    <div id="content">

      <p>My first paragraph. Lorem ipsum
dolor sit amet, consectetur adipiscing elit. Ut et lorem inante
vestibulum fringilla a at sapien. Quisque porta dolor orci, vel
convallis est. In accumsan pretium tempus. Etiam malesuada, nisl non
posuere scelerisque, massa urna pulvinar metus, at euismod velit
nulla quis risus.</p>

      <p>My second paragraph. Sed ligula lorem, mollis ut sagittis non,
sollicitudin id odio. Phasellus sit amet leo ante, eget porta ipsum.
Vestibulum eu hendrerit diam. In faucibus mi sit amet diam auctor
luctus. Nullam suscipit aliquam nunc, at cursus dui euismod in.
Pellentesque interdum lorem et purus scelerisque et euismod lectus
ullamcorper.</p>

    </div><!-- end #content -->

    <footer>
      <p>&copy; my name</p>
      <p>last updated: August 30, 2015</p>
    </footer>

  </div><!-- end #wrapper -->

</body>

</HTML>
```

Example 5: The Foundational Structure Used for Building Your Web Portfolio