Introduction to HTML

HTML (**H**yper**T**ext **M**arkup **L**anguage) is a markup language used to define the structure of and to display the content of a web page. It essentially defines the content the browser should display. It is made up of *elements* which define what the content is (i.e. the meaning of the content). For example, everything within the `<body>` element is the body of the page.

Note

The **body** of the page is where the content to be displayed to the user is placed.

Each HTML element is formed with one or more *tags* with each tag enclosed within angle brackets (in the example above, "**<**`body`**>**"). Most tags need to be closed and will have a corresponding closing tag denoted by a slash (e.g. "`</body>`"). Elements should be properly nested or strange errors or behaviours can occur.

Example:

```
<body>
  <p>This is a paragraph. <strong>This is bolded text while <em>this
is bolded and italicized.</em></strong></p>
</body>
```

The above HTML is only the body portion of an HTML page. It defines that within the body of the page, there is a paragraph, denoted by the `<p>` tag. The text of the paragraph is within the `<p>` tags. Note the slashes in the closing tags. Two other tags are present: `<strong>` and `<em>`. The `<strong>` text is used to bold the text while `<em>` is used to italicize or *emphasize* the text. In this case, because the `<em>` tag is within the `<strong>` element, the words within the `<em>` tags are both bolded *and* italicized. Note that because the `<strong>` tag was opened first the `<em>` tag must be closed before the strong tag. This is what is meant by proper nesting. The closing `</em>` tag cannot be placed after `</strong>`.

Caveat!

All HTML element tags have a meaning and should be used according to the context of the content. The `<strong>` and `<em>` tags have semantic meanings and are usually meant to be used for statements of importance or to emphasize text. These tags are generally not recommended for bolded and italicized text that are stylistic and do not denote importance. In these cases, some people may use the older tags `<b>` and `<i>`; however, these are deprecated tags. There are a variety of methods to bold and italicize text of no importance. In blocks of text or headings you can use CSS instead to style text. In other words, `<b>` and `<i>` should only be used when there are no other possible tags to use. They are for typographic styling only. Words you wish to stress the importance of or emphasize should use `<strong>` or `<em>`.

Note

You can view the HTML of any page by viewing the source in the browser ("`<Ctrl> + U`" in Firefox, Chrome, IE and Opera for Windows, "`<Command> + U`" for the same browsers for Mac).

The Basic Structure of an HTML Page

An HTML page, at its most basic, is made up of the following parts:

- a declaration as the first line to declare that the document is an HTML document
- an `<html>` tag to enclose the HTML document
- a `<head>` tag to enclose page information and meta-data
- a `<body>` tag to enclose the content

HTML is in plain text so you will need to use a plain text editor to edit the file. If you are using Windows, you can use Notepad or download Notepad++. If using Mac, you can use TextEdit (but set it to plain text mode via **Format > Make Plain Text** or the keyboard shortcut **<Shift> + <Command> + T**) or you can try installing another plain text editor such as TextWrangler or Brackets. If the option is available, make sure you select UTF-8 encoding when you save the file. (If you have a choice of "without BOM" choose "without BOM".)

Document type declaration statement

As of HTML5, the very first line of an HTML document (a file ending with ***.html***) should be:

```
<!DOCTYPE html>
```

This declares that the document is of type HTML.

<html>

The second line of the document should be the opening `<html>` tag where you will define the HTML document itself. Everything within the `<html>` tags are the HTML document itself. It's a good idea to put the closing tag too so that you will not forget to put it in later.

By now you should have:

```
<!DOCTYPE html>
<html>
</html>
```

Note: For better compatibility, you should also add `lang="en"` to your opening <html> tag, like so:

```
<!DOCTYPE html>
<html lang="en">
</html>
```

<head>

Not to be confused with a header within a web page (that would be in the `<body>` section), this is where all of the metadata is defined and where you will link your external style sheets and/or Javascript files.

By default, HTML5 uses the UTF-8 character set which is standardized unicode, but this default can also be changed in the server so it is best practice to explicitly declare the character set. This ensures that the browser interprets and shows the characters correctly. This declaration needs to be set as close to the beginning of the document as possible, so it should be the first line after the opening `<head>` tag. You can declare the character set with the following line:

```
<meta charset="utf-8">
```

This is the newer and shorter way to declare character sets and should be backwards compatible back to IE6. You should also make sure that you check the encoding for your plain text editor and ensure that UTF-8 is selected and not ANSI. If you come across the meta charset tag in the format below, that's just an older version (you don't need to use it).

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

The basic `<title>` tag is found here to define the page's title. This is the title which appears in your **browser bar** or in your **browser tab**. The `<title>` tag is required for valid HTML.

You can also add a meta-data description tag to describe your page. This is the text used in search engines to describe a page or website. For example, in a Google search, a search result consists of a linked page title, the URL, and a description. The description tag is used here. If there is no description tag, the search result will display the first few words of the page instead.

Add the following within your `<html>` section.

```
<head>
  <meta charset="utf-8">
  <title>Basic HTML page</title>
  <meta name="description" content="My basic HTML page">
</head>
```

Note that in the above example, the `<meta>` tag is self-closing hence there being no separate closing tag. (For XHTML, you will see self-closing tags ending with a slash—e.g. `<meta charset="utf-8" />`. For HTML, do not use the closing slash for self-closing tags.)

At this point you should have the following in your HTML file:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Basic HTML page</title>
    <meta name="description" content="My basic HTML page">
  </head>
</html>
```

<body>

This area is where your content goes. This element goes between the `<html>` tags but after the `<head>` section.

Taking the body example from page 1, you should now have the following for your HTML page.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Basic HTML page</title>
    <meta name="description" content="My basic HTML page">
  </head>
  <body>
    <p>This is a paragraph. <strong>This is bolded text while <em>this
    is bolded and italicized.</em></strong></p>
  </body>
</html>
```

Save this file as ***basic.html*** and view it in a browser. Make sure you save it with UTF-8 encoding and **not** ANSI. If possible and the choice is available, choose UTF-8 encoding without BOM. BOM stands for **B**yte **O**rder **M**ark and this "mark" is a special set of bytes which appear (invisibly) at the beginning of a file.

A little about the DOM

You may read about or hear references to the DOM (Document Object Model). Essentially, the browser interprets your HTML code (also for XML) and defines the HTML document as a tree of elements (the model). This tree structure defines how you can access the various elements of the HTML page. These elements can be found by traversing the tree and then be manipulated with a script, such as Javascript. The DOM looks very similar to the HTML code but is not quite the same. It represents the **current state** of the page

In simple terms, your HTML code is what is viewed when you "view source" in your browser. The DOM shows the current state of the page and is what you'll see if you "inspect element" or view the page in the browser's developer tools. This is why, if you have content (say, a paragraph) that is displayed via Javascript, "view source" will not show that content while the DOM will.

Common basic HTML elements

The following are some basic HTML elements which are used to define content in the body. Keep in mind each of the elements' semantic meanings.

<h1></h1>...<h6></h6>

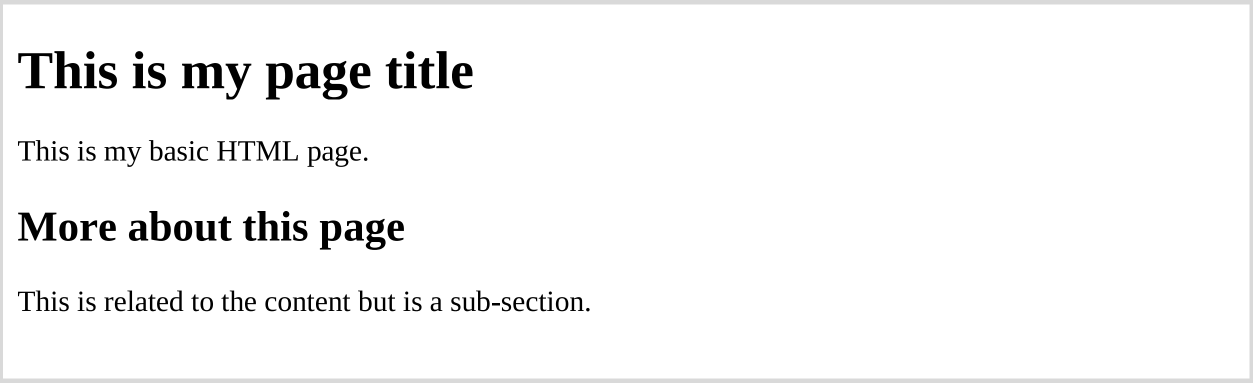
These elements are used for headers. H1 is the largest header (the smallest is H6). This is used for page titles or website names in general.

Note

Semantically, headers (with their numbers) have meanings, so these should be ordered appropriately. H1 means that it is the header for the main section (e.g. a page title). A subsection should use H2 for its header. In other words, you should *not* have a lone `<h2>` element without an `<h1>` element. You can have multiple `<h2>` elements after an `<h1>` element but if you want a header for a sub-subsection, you should use `<h3>`.

Although headers range in size from largest (heading 1, H1) to smallest (heading 6, H6), you should not use the headers indiscriminately without thought to semantics. If you only want to use the headers based on size, the font size and styling should be done using CSS rather than by header type (H1 to H6). In other words, do not jump from `<h2>` to `<h4>` just because you want the smaller size. Semantically, a sub-heading of `<h2>` is `<h3>`. Use CSS to change the size. Although the browser will display it, it is semantically incorrect and changes the meaning of the content. It will also cause issues with screen readers and messes up the structure of the page.

For example, let's write out the correct markup for the following text:



This is my page title

This is my basic HTML page.

More about this page

This is related to the content but is a sub-section.

This is **correct**:

```
<h1>This is my page title</h1>
<p>This is my basic HTML page.</p>
<h2>More about this page</h2>
<p>This is related to the content but is a sub-section.</p>
```

The following is **incorrect**:

```
<h2>This is my page title</h2>
<p>This is my basic HTML page.</p>
<h4>More about this page</h4>
<p>This is related to the content but is a sub-section.</p>
```

<p>...</p>

The `<p>` element is used for paragraphs.

<strong>...</strong>

This makes bolded text to imply strong importance of the text.

<em>...</em>

This italicizes text to imply an emphasis (e.g. when you speak with emphasis on a certain word or phrase).

<a>...</a>

The <a> is used for external and in-page links. The "a" stands for "anchor." To use as a link, you will need to use the following format:

`<a href="`***[page_location]***`">My link text</a>`

where the `href` value is the path for your link. This path can be an absolute path (e.g. "*http://www.google.ca*") or it can be relative to the HTML page (e.g. "*mypage.html*"). The second example means that the link points to ***mypage.html*** which is located in the same directory as the current HTML file.

As an in-page link, this requires you to use one `<a>` element as the link and another `<a>` element in the page as an anchor. Anchors do not need an `href` value.

The link will have the following format:

```
<a href="#faq">FAQ</a>
```

where the "#" says that the link is pointing to an anchor with an ID of "faq". Upon clicking this link, the browser will jump to the location of the element or anchor with the given ID. Note that if you want to jump to an anchor or element in another page you would simply put the page location followed by "#" followed by the anchor or element `id`. The anchor should be placed at the beginning of the content it is referring to and should use the following format:

```
<a id="faq"></a>
```

Previously, `<a>` used `name` instead of `id` but this has been deprecated. Note that because you are using an `id` and IDs are **unique**, you can point the link to any element with an `id`.

Example: `<h2 id="faq">FAQ</h2>`

<img>

The `<img>` element is used to display an image.

To use this tag, type: `<img src="yourpicture.jpg" alt="My picture">`

You can also specify the height and/or width of the image by adding `height="`*[height in pixels]*`"` and/or `width="`*[width in pixels]*`"`.

Attributes

The opening tag of an element may define additional information. This additional information is an attribute, which consists of the attribute's **name** followed by an equals sign followed by the attribute **value**. In the above image tag, two attributes are defined: `src` and `alt`.

The `src` attribute (the attribute name is `src`) defines the source or location of the image. In this example, the location is defined in relation to the HTML page's location, so that means that the ***yourpicture.jpg*** file is located in the **same** directory as the HTML page. To define the location of an image in a *subdirectory*, give the `src` as ***subdirectory_name/yourpicture.jpg***. If the image is in a directory *outside* of the current HTML page's directory, you will need to define the `src` as ***../external_directory/yourpicture.jpg***. The two dots ("**..**") mean to go up a directory, so in other words the location is up one directory then look in the directory named ***external_directory*** for ***yourpicture.jpg***.

The `alt` attribute defines **alternate text** in case the image cannot be viewed. This is also used by screen readers to determine what the image displays. The `alt` attribute should describe what the image is and not be a generic line of text.

<br>

This element represents a line break. This is a simple line break. Unlike `<p>`, it has no semantic meaning so you should use `<p>` where appropriate and should not use `<br>` in place of `<p>` if the content is a paragraph. As this is usually used for more stylistic purposes, you should avoid its use in most cases. (For XHTML, the slash is required. It is not required for HTML.)

<ul>...</ul>

This element defines an **unordered list**. These would be lists without a numbering system, such as bullet lists. List items are enclosed in `<li>` tags. By default, the list type is a bullet. This can be changed by using CSS to change the list style type. The default type chosen by the browser may also depend on the nesting level of the list.

Lists should not be nested within a paragraph according to HTML specifications. Doing so will throw an error in a validator. Terminate the paragraph before adding a list.

For example:

The list

- Item one
- Item two

Corresponding HTML

```
<ul>
  <li>Item one</li>
  <li>Item two</li>
</ul>
```

<ol>...</ol>

This element is used for an **ordered list**. This would be a numbered list. Similar to the `<ul>` element, list items are enclosed in `<li>` tags. The default list type is a numbered list. Ordering by upper- or lower-case letters or Roman numerals can be done by using the type attribute with the following values:

Attribute declaration	*List type*
type="a"	a. Item one b. Item two
type="A"	A. Item one B. Item two
type="i"	i. Item one ii. Item two
type="I"	I. Item one II. Item two
type="1" * this is the default	1. Item one 2. Item two

HTML5 also includes a `start` attribute which you can use to declare the starting number for a list as opposed to beginning the list at number 1.

Lists can be nested as long as you remember to nest the tags properly. (As mentioned above, lists should not be nested within a paragraph.) When nesting a list, the sub-list is nested *within* the parent `<li>` tag.

For example:

The list

- Item one
- Item two
    1. Another item
    2. Yet another item
- Item three

Corresponding HTML

```
<ul>
  <li>Item one</li>
  <li>Item two
    <ol>
      <li>Another item</li>
      <li>Yet another item</li>
    </ol>
  </li>
  <li>Item three</li>
</ul>
```

Special characters

There are some characters that you cannot type plainly in an HTML document because they have special meaning. For these characters, you will need to use escape characters.

Left angle bracket (<)

Since this is used in HTML tags, you cannot use this character plainly or the browser may try to interpret whatever follows as a tag rather than just content. For this, use `&lt;` which stands for "less than."
Right angle bracket (>)

Use `&gt;` instead which stands for "greater than."

Ampersand (&)

Ampersands are used in escape characters so you cannot use them alone in your content. The escape character for ampersand is `&amp;`.

Copyright symbol (©)

This is not a special character in that it will not interfere with HTML parsing, but it is a special character not on the keyboard. Using Unicode (UTF-8), you can type this character directly, which is usually recommended because it makes the HTML code more readable. The Unicode number for this character is 0169.

To type this directly in Windows, you will need to use the number pad so turn on **<Num lock>** then hold down **<Alt>** and type "0169".

To type this directly in Mac, use the keyboard shortcut **<Option> + G**.

If you don't want to type the character in directly or if you don't have the number pad, you can use the HTML entity `&copy;` or the following hex or decimal codes: `&#x000A9;` or `&#169;`. Of course, if your document has many escaped characters, the HTML code can quickly become a nightmare to read so the best practice is to use the character directly whenever possible.

Example

In this example, you will be building a basic page containing text and some lists. The page title which appears in the browser tab will be: "My bio". The page content will also appear as below.

My bio

My name is Joe Schmoe. I am a developer based in Toronto and I have five years of experience coding websites using **PHP** and open-source content management systems. I also have experience with the following technologies:

- ASP.NET
- Javascript
- C
- Photoshop

More about me

I also have a variety of hobbies, such as reading. The following are books I have read recently.

1. War and Peace
    - Leo Tolstoy
2. The Art of War
    - Sun Tzu

© Joe Schmoe, 2018.

1. Create a new file in your choice of plain text editor (e.g. Notepad++, VSCode, Notepad, TextWrangler, etc).
2. Type out the basic HTML empty page. Don't forget to add the character set declaration.

   ```
   <!DOCTYPE html>
   <html lang="en">
     <head>
       <meta charset="utf-8">
     </head>
     <body>
     </body>
   </html>
   ```

3. Since we want the page title in the browser tab to read, "My bio", add the following line in the `<head>` section after the character set declaration.

   ```
   <title>My bio</title>
   ```

4. Next fill out the body content. First identify the parts:
   a. The page title is "My bio". This should be an `<h1>` element.
   b. There are two paragraphs.
   c. There is a list. Don't forget that it should *not* be nested in the paragraph.
   d. There is a sub-section. The title should be an `<h2>` element because, structurally, it is the sub-section directly under or within the main section. Remember that using `<h3>` means that the title is the title of a sub-section of another sub-section.
   e. There is a nested list with an outer ordered list and inner unordered lists.
   f. There is a special character (the copyright symbol). You can type this directly using the shortcut (**<Num lock> + <Alt> + 0169** for Windows, **<Option> + G** for Mac) or just use the escaped character.
5. Using the parts you identified above, type out the body. (Be sure to put the body content *within* the body open and close tags.)

   ```
   <h1>My bio</h1>
   <p> My name is Joe Schmoe. I am a developer based in Toronto and
   I have five years of experience coding websites using
   <strong>PHP</strong> and open-source content management systems.
   I also have experience with the following technologies:</p>
   <ul>
     <li>ASP.NET</li>
     <li>Javascript</li>
     <li>C</li>
     <li>Photoshop</li>
   </ul>
   <h2>More about me</h2>
   <p>I also have a variety of hobbies, such as reading. The
   following are books I have read recently.</p>
   <ol>
   ```

```
    <li>
      War and Peace
      <ul>
        <li>Leo Tolstoy</li>
      </ul>
    </li>
    <li>
      The Art of War
      <ul>
        <li>Sun Tzu</li>
      </ul>
    </li>
  </ol>
  <div>© Joe Schmoe, 2018.</div>
```

6. Now save your file as ***bio.html***. While saving, select the "UTF-8" encoding option otherwise your copyright symbol will display incorrectly (if the character was typed directly). If available, select "UTF-8 without BOM".

> *Tip!*
>
> Your plain text editor of choice may have a different way for you to change the encoding. For example, in Notepad++, you can set in **Settings > Preferences** that you want all new documents to be set as UTF-8. Choose UTF-8 without BOM if available. Although browsers can read UTF-8 with BOM (you have no choice of "without BOM" with Notepad), choosing "without BOM" is preferable.

7. Open your saved file in a browser to view it.

> *HTML Validator*
>
> The W3 organization has an HTML validator at *http://validator.w3.org*. There you can upload your HTML file to be validated. It should automatically jump to the HTML5 validator at *http://validator.w3.org/nu*. It is important for you to ensure your code is valid with the validator because browsers will still show your content regardless, but if your code is invalid this can impact accessibility and can lead to unexpected behaviour.

HTML tables

Tables in HTML are used for displaying data. It should **never** be used for page layouts. Every table is composed of a bunch of rows with data cells per row. A table uses the <table> element.

<table>...</table>

This element wraps around the WHOLE table (i.e. it's the outermost element around the rows of data).

Before HTML5, there was a `border` attribute (e.g. `border="1"`). This is useful to quickly check your table structure, but should be removed because it will throw an error with an HTML5 validator. It is now deprecated because table styling should be done with CSS. By default, tables will have no borders.

As with lists, tables should **not** be nested within paragraphs (i.e. a table is not a paragraph).

<thead>...</thead>

This is used to define which row(s) is the **header row**. Within this element, there should still be a table row element (`<tr>`).

<tr>...</tr>

This element is used to define a **table row**. Nested within, it should contain either table header cell elements (`<th>` if `<tr>` is within `<thead>`) or table data cell elements (`<td>` if `<tr>` is within `<tbody>`).

<th>...</th>

This is used to define each **header cell**.

<tbody>...</tbody>

This element holds the table data rows and cells and defines the **body of the table** (does not include the header, which is enclosed in `<thead>`).

<td>...</td>

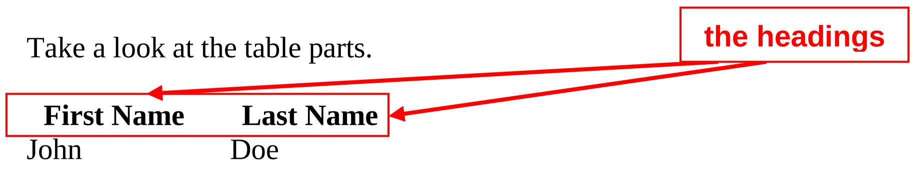
This holds the **table data**. Each `<td>` element is a data cell.

Note

In a simple table, you really only need the `<table>`, `<tr>`, and `<td>` tags but **it is best to use `<thead>`, `<tbody>` and `<th>` for proper semantics and accessibility**. Using the proper semantics and structuring helps keyboard users as they tab through a page.

An example of a table:

Take a look at the table parts.


the headings

First Name	Last Name
John	Doe

The corresponding HTML will be:

```
<table>
  <thead>
```

```
    <tr>
      <th>First Name</th>
      <th>Last Name</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>John</td>
      <td>Doe</td>
    </tr>
  </tbody>
</table>
```

colspan and rowspan

You can use the `colspan` and `rowspan` attributes to define the number of columns or rows a cell spans. This essentially merges the cell across columns and rows, respectively. Since the cells are merged, you do not need to specify `<td>` elements for those cells that have been merged in.

Example: (In the following example, the table has a dotted border to show the spanning, but by default, there is no border to see.)

The table

Group	**Name**
Team 1	John Doe
	Bill Dawson
	Jane Lee
Team 2	Hannah Foster
	Sean Hayes

Corresponding HTML

```
<table>
  <thead>
    <tr>
      <th>Group</th>
      <th>Name</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td rowspan="3">Team 1</td>
      <td>John Doe</td>
    </tr>
    <tr>
      <td>Bill Dawson</td>
    </tr>
    <tr>
      <td>Jane Lee</td>
    </tr>
    <tr>
      <td rowspan="2">Team 2</td>
      <td>Hannah Foster</td>
    </tr>
    <tr>
      <td>Sean Hayes</td>
    </tr>
  </tbody>
</table>
```

Can you view table borders?

Yes, you can! For diagnostic purposes only, you can do this with HTML only (not for production).

As a quick visual check **only to help you check if you're structure is correct** you can add a `border="1"` to the `<table>` element. This is only for a quick check as the border attribute is now obsolete (it will fail validation). Once you know your structure is correct, remove the border attribute. This works in a pinch since we haven't talked about CSS yet.