

# Introduction

## What is HTML?

HTML or Hyper Text Markup Language is the building blocks of all web pages. HTML uses tags to create element that are filled with content like text and images. The elements are instructions to the web browser how to display the content on the screen.

Another way to think about it is HTML is a series of start and stop commands for the web browser. The browser then renders that content to the specs of HTML.

## The Three Buckets

Modern web development uses the three bucket approach. Each bucket adds another layer to the display of the page.

- HTML
  - CSS
  - JavaScript
- 
- HTML is the building blocks or scaffolding of the page.
  - CSS or Cascading Style Sheets is the presentation or the look and feel of the page.
  - JavaScript is the action layer of the page. We will not cover any JavaScript in this class.

## The Tag

There are over 80 different HTML tags that can be used. They all have the same basic format. Most tags have an opening and a closing tag.

```
<tag>content</tag>
```

All tags have brackets around them < and > that are used by the browser to know what is a tag and what is content. Right after the opening bracket < is the tag name like h1 or p then the closing bracket >.

```
<tag>
```

If the tag has a closing tag it will match the opening tag but have an / after the opening bracket. This is the indication to the browser to close the tag that was opened. With out it the browser would never know when to stop.

```
</tag>
```

There are a few number of tags that do not have a closing tag because they do not wrap any content. Older versions of HTML required that all tags be opened and closed. This is no longer the case although you may still see a self-closing tag `<tag />` in code examples.

## Attributes

Sometimes a tag will need additional information to perform correctly. Adding an attribute to the tag allows the developer to set the correct information on each tag.

- Attributes will always go on the opening tag after the tag name. There must be space between the tag name and the attribute name.
- Attributes do not need to be closed. In other words when you close a tag you close its entire attribute.
- If an attribute has a value the value must be quoted and you should use the double quote "
- If an attribute has a value there must be an equal sign = between the attribute name and the value.
- The order of the attribute does not matter.
- The number of attributes that are used has decreased with newer versions of HTML.

### One Attribute

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

### Two Attributes

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

## Block-level vs inline

Most all tags fall in to one of two types of tags, inline or block. Understanding how an element displays will help you choose the correct tag to use.

Block-level elements will

- Fill the full width of the parent element
- Force the content to its own line
- Allows for padding and margin to be set on the top and bottom.

### Examples

`<p>`, `<div>`, `<header>`, `<nav>`, `<ul>`, `<li>`, and `<h1>`.

Inline will

- Only take up the space of the content
- Wont force the content two its own line i.e. stay "inline"

- Can't have padding on the top and bottom
- Can't have a width or height set

Examples

`<a>`, `<span>`, `<strong>`, `<em>` and `<code>`.

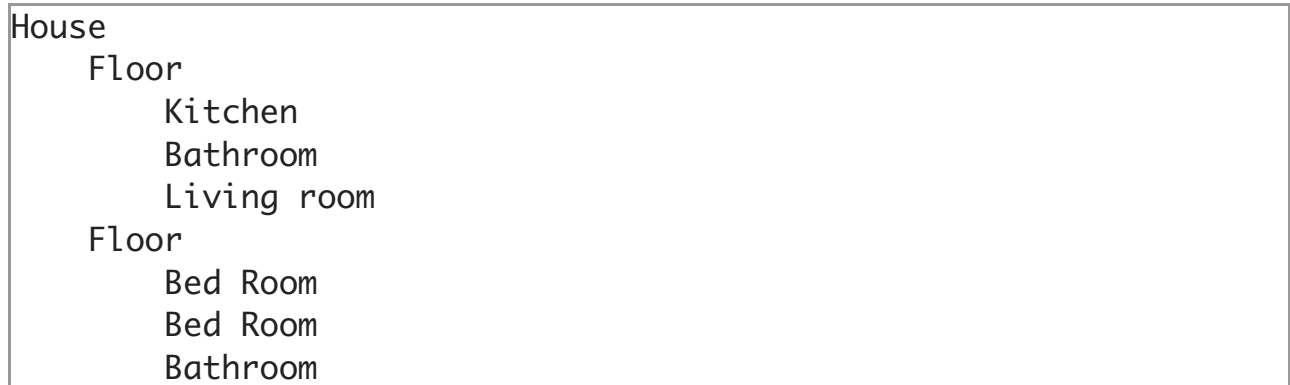
The block examples are structural elements, while the inline elements are text-based. This is an easy way to remember which is which.

Generally speaking, you can put any block element inside another block element. You can also put any inline element inside a block element, as well as any inline element inside any other inline element. But you cannot put a block element inside an inline element. The paragraph tags also must be the inner most block element or in other words you cannot have a paragraph tag nested in a paragraph tag.

## Nesting Elements

If we think about html as the building blocks of a webpage we will have to nest elements within in each other to build the page correctly.

Think about it this way. If we were to build a house we might start to build the ground floor. We might add a kitchen, bathroom and living room all with in the ground floor. We would not finish the floor until all the rooms are done.



If this was HTML it might look something like this

```

<house>
  <floor>
    <kitchen></kitchen>
    <bathroom></bathroom>
    <living></living>
  </floor>
  <floor>
    <bed></bed>
    <bed></bed>
    <bathroom></bathroom>
  </floor>
</house>
  
```

You can see we open the house tag but do not close it until both floors are done.

First in, last out. The first tag you open is the last tag you close.