

# BASICS | The html structure layout

## Important concepts:

- HTML stands for **H**ypertext **M**arkup **L**anguage
- HTML is a series of "elements" or parts of a page. Each image, text block, link, button, and everything else on the page is a separate element. These elements have opening and closing tags.
- HTML is built by a browser from the top to the bottom. This is less important until you start using JS.

## Important Vocabulary:

- Element | Part of a page
- CSS | The code used to style the page

## Code:

```
1 | <!DOCTYPE html>
2 | <html>
3 | <head>
4 | <title>What shows up in the tab</title>
5 | <link href="style.css" rel="stylesheet">
6 | anything that you want to load BEFORE the content
7 | </head>
8 | <body>
9 | The content of the site
10| </body>
11| </html>
```

## Explanation:

```
1 | Tells the browser what type of file it is loading - Not required in most modern browsers,
   but older versions of Internet Explorer need it.
2 | First tag - your text editor will provide autofill when you are in html tags.
3 | First part of the page - the head loads before all of the rest of your page. If you are
   using any external libraries, they will be identified
   here (line 5).
4 | The text that shows up in the tab at the top of the page. If not specified, it defaults to
   the file name.
5 | Calls your stylesheet we will make later. This allows your page to be pretty
6 | Any external libraries or files that need to be loaded (stylesheets, javascript, favicons)
7 | Tells the page that it is done with the head tag
8 | Opens the second part of the page - the body is where you will put all of the content of
   your site.
9 | This is where almost everything will go
10| Closing out the body element
11| Telling the page you are done, and the site is ready for full use.
```

# BASICS | Tags

Code:

```
<tag class="class" id="id" name="name" src="source">stuff to show<tag>
```

## Explanation:

We are going to break it up

**<tag** | Open the tag with "<" then specify the tag.

**class="class"** | Specify the class, which controls styling. More on this later. (note the class can be named whatever you want)

**id="id"** | Gives the element a handle that Javascript can grab. You won't need this until you start using Javascript (note the id can be whatever you want it to be)

**name="name"** | Specify the name, which is used in forms. More on this much later. (note the name can be whatever you want it to be)

**src="source"** | Specify the source. This is used in javascript, videos, images, and datalist tags. More on this later. (note the source can be whatever you want)

**>** | Closes the tag opener.

**stuff to show** | The inside of the tag. Usually this is text. Not all elements use it (E.G. images, inputs)

**</tag>** | Closes the tag. The "/" differentiates it from the opener

# BASICS | Tags Definitions

`<div>content</div>` | A container to put other elements in. This is great for code organization and applying styles to multiple elements.

`<h1>content</h1>` | Largest header

`<h2>content</h2>` | Second largest header. Continues to h6, with size getting smaller as you go. **Should be only used for headers**

`<p>content</p>` | Paragraph tag - put your paragraphs in it.

`<span>content</span>` | Inline text block. Great for highlighting and different colors in a paragraph.

`<a href="location">content</a>` | Link. The location you are going to goes in the href. NOTE: if you are linking to an external source, you must include http:// in the href.

`<button>content</button>` | Button. Used for interaction with Javascript.

`<b>content</b>` | Bold

`<i>content</i>` | Italics

`<u>content</u>` | Underline

`<s>content</s>` | Strikethrough - puts a line crossing out the text.

`<hr>` | Horizontal line, great for page organization and separating different sections.

`<ul>content</ul>` | Bulleted (unorganized) list LOOK AT DEFINITION FOR `<li>`

`<ol>content</ol>` | Numbered (organized) list LOOK AT DEFINITION FOR `<li>`

`<li>content</li>` | List element - you need to put each bullet or number in one

`<script>content</script>` | A method for imbedding Javascript in the page. Don't mess with it untill you know js.

# GETTING STARTED | Divs

The div tag defines a division or a section in an HTML document. It is often used with css to layout a web page. <sup>[1]</sup>

It can be used to create sections of a page with different background colors, font weights, sizes, font colors, and heights and widths.

```
<div>content</div>
```

Anything you want can be put in a div.

# GETTING STARTED | Text nodes

There are many ways to create text in HTML. Here are the most popular.

## Choice 1: none

HTML supports just text. It doesn't need to be in a tag.

## Choice 2: p

```
<p>content</p>
```

The P tag is mainly used for paragraphs.

## Choice 3: Span

```
<span>content</span>
```

The span tag is usually used for inline styling, like setting the color of a certain word.

## More on spans

While usually bad practice, it is ok to use inline styling for spans. Inline styling is injecting css to a specific element without the use of classes. More on css later.

```
<span style="color:red">content</span>
```

This example will make the color of the word "content" to red.

STARTING | Divs & Text nodes

# GETTING STARTED | Images

To add an image, you need to know the file name and the file extension.

This is case sensitive. This means the file "Image1.PNG" is not the same as "iMagE1.png"

```

```

# GETTING STARTED | Videos

Videos can be imported 2 ways: Embedding from Youtube or some other streaming service, or uploading a file yourself.

This will provide the description for the second of the 2.

```
<video width="320" height="240" controls>  
  <source src="file.extension" type="video/extension">  
  description if something goes wrong  
</video>
```

The width of 320 and height of 240 are arbitrary, but they work well for most videos. The units are pixels, and you can experiment with them.

The modifier "controls" tells the page to load a play button, among other basic controls.

If you want the video to play when the page is loaded, include the modifier "autoplay"

# GETTING STARTED | Links

Links in HTML can be used to go to other pages

For linking to other sites, you need a path. If you are not using folders, it is just the filename.extension.

```
<a href="path.extension">content</a>
```

# GETTING STARTED | Titles

HTML has a few tags that are meant for titles (headers).

## The h1 tag

By default, the h1 tag is the largest, and should be used for main headers

```
<h1>content</h1>
```

## The h2 tag

By default, the h2 tag is the second largest, and should be used for subtitles.

```
<h2>content</h2>
```

## The h3, h4, h5, & h6 tags

These are more subtitle options. As the number in the tag increases, the size decreases.

```
<h3>content</h3>  
<h4>content</h4>  
<h5>content</h5>  
<h6>content</h6>
```

```
<h7>content</h7> Actually no. the H tags only go up to 6.
```

*H tags should only be used for titles. Refer to the text nodes reference for page content.*

# GETTING STARTED | Page Organization

Most programmers like to add comments to their code to remove a line from action without actually deleting it. It can also be used to create notes to yourself.

```
HTML | <!-- content -->
```

```
CSS | /*content*/
```

```
Javascript | // content
```

# STYLING | Getting started

In HTML, we use a language called CSS (Cascading Style Sheets) to change the styles of the page. CSS Allows you to change almost everything visual.

## CSS format

CSS uses *classes* to organize its efforts. The same class can be applied to multiple elements, allowing you to create consistent pages.

To use CSS with an element, you must specify a class within it (refer to BASICS | Tags). In css we must insert the basic following code:

```
.classname {  
}
```

The classname is what we told the element in the `class=""` attribute.

We must include a "." before the classname to tell CSS to look for elements with the class "classname".

In between the curly brackets "{}" is where we put our styling code. It is important to add a semicolon ";" or the following lines *will not work*.

## Preparing an environment

Create a new file. It's extension must be `.css`

In HTML add the following line:

```
<link href="filename.css" rel="stylesheet">
```

This will call in your external stylesheet.

You can use the "style" tag to avoid having to import a stylesheet, but it is bad practice. Most tutorials (w3, stackoverflow, documentation) will use the style tag, but it is just to simplify the formatting of their tutorials.

# STYLING | Font

HTML:

```
<tag class="myclassname">content</tag>
```

Replace the tag with h1, a, span, p, or many other tags.

## Font color

CSS:

```
.myclassname {  
  color:red;  
}
```

## Highlighting / background color

CSS:

```
.myclassname {  
  background-color:yellow;  
}
```

## Font size

CSS:

```
.myclassname {  
  font-size:15px;  
}
```

## Font outline

CSS:

```
.myclassname {  
  -webkit-text-stroke: 1px black;  
}
```

## Other details

For colors, CSS can use names `red`, `blue`, `pink`, `lightgrey`, etc , hex codes `#a1a1a1` ,  
transparent `transparent` , or RGB `rgb(255,255,255)`

STYLING | Getting Started



# EXCERCISE | Title banner

The title banner - a staple in every website. We can make a title over an image or a video. But first we need a image or video, and a title.

Lets get started!

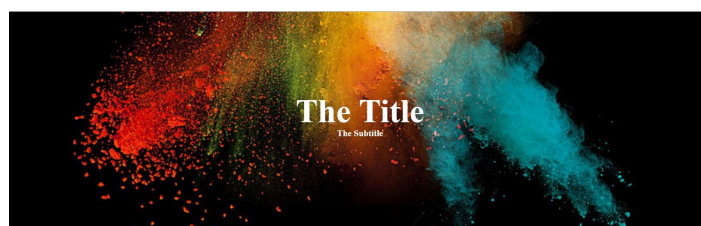
HTML: (image)

```
<div class="header-container">
  
  <div class="title">The Title</div>
  <div class="subtitle">The Subtitle</div>
</div>
```

CSS:

```
.header-container {
  position: relative;
  text-align: center;
  color: white;
}
.title {
  position: absolute;
  top: 45%;
  left: 50%;
  transform: translate(-50%, -50%);
  font-size: 45px;
}
.subtitle {
  position: absolute;
  top: 55%;
  left: 50%;
  transform: translate(-50%, -50%);
}
html, body { /*we do not use the . before the class name because we are styling tags, not
classes. Usually a bad practice, but excusable for the HTML and Body tags.*/
  padding: 0;
  margin: 0;
}
```

And the result:



# EXCERCISE | Title banner continued

The title banner - a staple in every website. We can make a title over an image or a video. But first we need a image or video, and a title.

Lets get started!

HTML: (video)

```
<div class="header-container">
  <video autoplay loop muted width="100%">
    <source src="banner.mp4" type="video/mp4">
    Text to load in the face of an error
  </video>
  <div class="title">The Title</div>
  <div class="subtitle">The Subtitle</div>
</div>
```

CSS:

```
.header-container {
  position: relative;
  text-align: center;
  color: white;
}
.title {
  position: absolute;
  top: 45%;
  left: 50%;
  transform: translate(-50%, -50%);
  font-size: 50px;
  -webkit-text-stroke: 1px black;
}
.subtitle {
  position: absolute;
  top: 55%;
  left: 50%;
  transform: translate(-50%, -50%);
}
html, body { /*we do not use the . before the class name because we are styling tags, not
classes. Usually a bad practice, but excusable for the HTML and Body tags.*/
  padding: 0;
  margin: 0;
}
```

Most browsers cannot run the autorun argument if the muted argument is there. If you want the user to be able to control the video, add the control attribute.

# EXCERCISE | Image and description

Here is a simple and popular means of including images and describing what they are. This is a great way to share content.

## HTML:

```
<div class="master">
  
  <div class="right">content</div>
</div>
```

## CSS:

```
.master {
  display:inline-block;
  width:100%;
  border:1px solid black;
}
.left {
  float:left;
  width:30%;
  padding-right:1%;
}
.right {
  float:right;
  width:68%;
  padding-left:1%;
}
```

This can be used for way more than images though. Replace the img with a div and put anything you want in it, and you will have 2 divs next to eachother.

# EXERCISE | Creating a contact us

Crucial for creating an effective web presence, there are 2 primary ways to create a contactus form.

## Option 1:

A mailto link

A mailto link is just like any other link, but it opens an email window.

```
<a href="mailto:you@example.com">Send us an email!</a>
```

## Option 2:

A form.

Forms work with server-side code called PHP, which I will not be going into, as GitHub does not support it. Instead, we will use a free service called FormSpree.

```
<form action="https://formspree.io/your@email.com" method="POST"> <!-- REPLACE your@email.com  
with your email. OTHERWISE LEAVE THIS LINE BE -->  
  <input type="text" placeholder="The text to show in the box before any input" name="detail" >  
  <!-- Replace detail with what you want the field to be named in your results. -->  
  <input type="email" placeholder="your email" name="_replyto"> <!-- DO NOT CHANGE ANY OF THIS  
LINE except for the placeholder= attribute -->  
  <input type="submit" value="Send"> <!-- DO NOT CHANGE ANY OF THIS LINE -->  
</form>
```

One more task: send a test response, and verify your email. (Formspree must be run on a web server, so don't do this until it is on the web.)

Now when someone fills out the form on your website, you will get an email with their response, and you can reply directly to them!

# Some final notes

Personally, I find the default font to be ugly. To overcome that, I go to [fonts.google.com](https://fonts.google.com) and choose one I like. My current favorite is a simple open sans. To import add this line to the head element of your page:

```
<link href="https://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet">
```

And in your stylesheet:

```
font-family: 'Open Sans', sans-serif;
```

And your done!

## Recources

Ask any developer, and they will tell you that a lot of research is required. Don't expect yourself to know all you will ever need.

A great resource is [w3schools.com](https://w3schools.com). They have tutorials for anything you could want to do. For styling tricks, check out [css-tricks.com](https://css-tricks.com).

## Closing

For today's lesson plan, powerpoint, and everything else, including code to copy and paste, go to [github.com/htmllessons/htmllessons.github.io](https://github.com/htmllessons/htmllessons.github.io)

Thank you, If you need any help at any point in your development, styling, html, javascript, and anything else, feel free to write me an email at [3jbc22@gmail.com](mailto:3jbc22@gmail.com).

Jack Crane