

HTML & CSS

An Introduction to Markup and Stylesheets

Class Guide



Author - Silas Bamidele

HTML stands for Hypertext Markup Language and it's used to create site layouts.

HTML is largely made up of Markups and Divs that helps you create the layouts that you want.

For example, a div could contain the site logo and another div contains a gallery of photos -- all within the Body tags.

**Note that HTML helps you create your site layout as well as helps you hold content in the divs that you create.

CSS on the other hand, Cascading Stylesheets, helps you style the layout and content on your webpages. For example, adding a background color to a div



The HTML skeleton

`<html>`

`<head> <title> </title> </head>`

`<body>`

`</body>`

`</html>`

******These are all individually tags but all form the markup

-- All Divs are created within the body tags

-- The site title goes within the title tags

-- The link to your stylesheet (e.g style.css) goes in the head tags

-- Divs have names -- divs are named with the **id** or **class** attribute



Let's create an html project and start marking up, shall we..

Grab a copy of your html_starter zipped project in your Tools folder and unzip into your document root, www.

Notice that, for organization purposes, you have an **img** folder, a **js** folder and a **css** folder -- all within your html_starter project.

The img folder holds your site images, the css folder holds your stylesheets and the js folder holds your javascript files

Notice also that you have an index.html file, which loads as the home page of your html project.

Now looking closely in the head markup, we see that a CSS framework called 12c.css has been linked. This CSS framework helps us lay out our webpages quickly and does some initial heavy lifting for us.



Class Practice : Create divs and understand how the framework works. Each row can have a maximum of 12 divs when added together.

Let's learn some tags we can use in our divs next

-- Paragraph `<p>` - See usage below;

`<p>` content goes here..`</p>`

-- Bold `` - See usage below;

`` content goes here..``

-- Heading `<h1>`, `<h2>`, `<h3>` `<h6>` - See usage below;

`<h2>` heading goes here..`</h2>`



-- Creating lists with versus

 is used for an unordered list.

See usage below;

```
<ul>
```

```
<li> list item one </li>
```

```
<li> list item two </li>
```

```
<li> list item three </li>
```

```
</ul>
```

 is used for ordered lists -- this will show numbering in the list you create.

See usage below;



```
<ol>
```

```
<li> list item one </li>
```

```
<li> list item two </li>
```

```
<li> list item three </li>
```

```
</ol>
```

-- The table tag <table>. See usage below to create rows and columns (or table data) ;

```
<table>
```

```
<tr> <th>Column 1</th> <th>Column 2</th> </tr>
```

```
<tr> <td>Data or item</td> <td> Data or item </td> </tr>
```

```
<tr> <td>Data or item</td> <td> Data or item </td> </tr>
```

```
</table>
```



-- The image tag - See usage below;

```

```

*the src attribute stands for source and defines where the image is located

-- The link tag <link /> - The link tag helps us link our external css files and js files to our current project or website. See usage below

```
<link rel="stylesheet" type="text/css" media="screen" href="css/12c.css" />
```

```
<link rel="stylesheet" type="text/css" media="screen" href="css/style.css" />
```

*the href attribute defines the location of the css stylesheets



-- The anchor tag `<a>` - the anchor tag helps us create links to other pages in our website or to external pages or links outside of our website. See usage below;

```
<a href="about.html"> About Us </a>
```

```
<a href="https://google.com.ng"> Google search page </a>
```

** Notice that the `img` and `link` tags are both self-closing

*Hint : When you're writing tags and divs, you're writing HTML



Enter, CSS!

Styles created in your stylesheets or stylesheet files can be used to make your web pages appear prettier and add even neater structure.

The syntax of a style:

```
target { property: value; }
```

f.example

```
h1 { color : black; }
```

Naming and targeting the element to style:

.class

#id



Class Practice:

Get hands on with these CSS properties;

color	display	margin-top
background	position	width, max-width
padding	list-style	height, max-height

Examples

```
p{ color : purple; }
```

```
div.footer{ display: none; }
```

```
div.logo{ margin-top: 25px; }
```



```
div.content{  
    position: relative;  
    top: 45px;  
    left: 15px;  
}  
  
body{ color: #eee; }  
  
body{  
    background: url('../img/bgee.jpg');  
    background-attachment: fixed;  
    background-repeat: no-repeat;  
}  
  
.logo{ width: 80%; }
```



```
ul li{  
    list-style: none;  
    padding: 6px;  
    list-style-type: square;  
}
```

****Learn and understand the DOM and CSS selectors**

```
<ul class='fruits'>  
<li class='rotten'>mangoes</li>  
<li class='fresh'>oranges</li>  
<li class='rotten'>guavas</li>  
</ul>
```



Author - Silas Bamidele

Styling the fruits list above needs a solid understanding of the DOM - Document Object Model

Make the color of the rotten fruits red

```
ul.fruits li.rotten{ color: red; }
```

Make the color of the fresh fruits green

```
ul.fruits li.fresh{ color: green; }
```

****Notice how in HTML the tag <ul class='fruits'>**

in CSS becomes ul.fruits

that is, ul is the target and the specific ul you're targeting is the one named fruits, which changes to .fruits in CSS



******Notice also that to target the `<li class="rotten">`tag in CSS, you declare it thus;

`ul.fruits li` --- the `li` here is written on the next space because it is another tag of it's own

`ul.fruits li.rotten` -- the complete CSS interpretation to target the rotten fruits

`ul.fruits li.fresh` -- the complete CSS interpretation to target the fresh fruits

