**HTML5, CSS3 AND 960 GRID SYSTEM**

Required:

* A text editor
* Use of browser

What to do?

* Creating a different css folder for styling is a good practice.
* Always validate your html and css code because browser understands your mistakes and don’t ask for correction.
* Use of class selector rather than id selector because id selector is for a specific styling in a single area but for group styling class selector is more preferred.
* Do work on typography.
* Intent the code properly.
* Add meta statement and doctype.
* Use different folders for images, js and css.

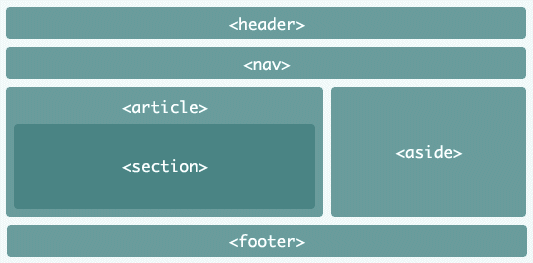
What not to do?

* Use inline or internal CSS.
* Copy any image url from the website and paste it.

HTML5

**Hypertext Markup Language** (**HTML**) is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language" \o "Markup language) for creating [web pages](https://en.wikipedia.org/wiki/Web_page) and [web applications](https://en.wikipedia.org/wiki/Web_application). HTML elements are tags which form the structure of the page and by this tags the browser understand.

Basically we have to first look into the basic structure of the webpage.



Now, the basic code of html or you can say the layout.

<!doctype html>

<html lang=”en”>

<head>

<meta charset=”utf-8”>

<title> \_\_\_\_\_\_\_\_\_\_ </title>

</head>

<body>

<h1>\_\_\_\_\_\_\_\_\_\_\_</h1>

<p>\_\_\_\_\_\_\_\_\_\_\_\_\_</p>

</body>

</html>

Now understand the command used above:

<!doctype html> declaration defines this document to be HTML5

<html> element is the root element of an HTML page

<head> element contains meta information about the document (meta data is data that provides information about other data).

<title> element specifies a title for the document

<body> element contains the visible page content

<h1> element defines a large heading

<p> element defines a paragraph

Now let’s go one by one to different types of tags and their purpose.

General format: First the opening tag then content and then closing tag.

<tagname>\_\_\_\_Content\_\_\_\_\_</tagname>

Some tags are self closed. Eg.

<tagname \_\_\_\_\_\_\_\_\_ />

Some elements with no content are called empty elements which don’t have an end tag. Eg: <br> for line break and <hr> for thematic break(displayed by a horizontal line)

Comments tags are <!------your comments----->

* For images:

Tag : <img>

Attribute: src for sourcefile, alt for alternative text(text written when due to some reasons images don’t get displayed), width and height.

Format: <img src=”\_\_\_\_\_” alt=”\_\_\_\_\_\_” width=”\_\_” height=”\_\_” />

* For links:

Tag: <a>

Attribute: href for link name

Format: <a href=”\_\_\_\_\_\_” >\_\_\_\_\_\_</a>

* For paragraphs

Tag: <p>

Format: <p>\_\_\_\_\_</p>

* For headings

Tag: <h1>, <h2>, <h3>, <h4>, <h5>, <h6>

Format: <h1>\_\_\_\_\_\_</h1>

* For preformatted text

Tag: <pre>

Format: <pre>\_\_\_\_\_\_</pre>

* For quotations from another source

Tag: <blockquote>

Attribute: cite for work title

Format: <blockquote cite=”\_\_\_\_\_”>\_\_\_\_\_</blockquote>

* For sidebar

Tag: <aside>

Format: <aside>\_\_\_\_\_</aside>

* For text direction

Tag: <bdo>

Attribute: dir for direction maybe rtl(right to left or ltr(left to right)

Format: <bdo dir=”\_\_\_”>\_\_\_\_\_\_\_\_\_</bdo>

More common tags with their description:

Tag Description

<b> bold

<em> emphasised text

<i> italic text

<small> smaller text

<strong> important text

<sub> subscripted text

<sup> superscripted text

<ins> inserted text

<del> deleted text

<mark> marked/highlighted text

<abbr> abbreviation as a toolkit

<address> contact information(in italics)

<q> short quotations

<caption> caption/title of the table

Now two important things which are in use a lot in html5 i.e. tables and lists

First,

TABLES:

We use <table> tag to define a table. Each table row is defined with the <th> tag and table data/cell by <td> tag. By default, table headings are bold and centered.

Format:

<table>

<tr>

<th>First name</th>

<th>middle name</th>

<th>Last name</th>

< /tr>

<tr>

<td>\_\_\_\_\_\_\_\_\_</td>

<td>\_\_\_\_\_\_\_\_\_</td>

<td>\_\_\_\_\_\_\_\_\_</td>

</tr>

</table>

LISTS:

There are three types of lists:

1. Unordered list
2. Ordered list
3. Description list

For unordered list:

Format:

<ul>

<li>\_\_\_\_</li>

<li>\_\_\_\_</li>

<li>\_\_\_\_</li>

</ul>

Attribute: In style, list-style-type: disc, circle, square, none

For ordered list:

Format:

<ol>

<li>\_\_\_\_</li>

<li>\_\_\_\_</li>

<li>\_\_\_\_</li>

</ol>

Attribute: type attribute: “1”, “A”, “a”, “I”, “i”

For description list:

Format:

<dl>

<dt>\_\_\_\_\_\_\_</dt>

<dd> -\_\_\_\_\_\_\_\_\_</dd>

</dl>

Nested list and horizontal lists is also possible in html

Format for nested list:

<ul>

<li>\_\_\_\_\_</li>

<ul>

<li>\_\_\_\_\_\_</li>

</ul>

</ul>

For horizontal list: se the CSS property **float: left** or **display: inline**to display a list horizontally.

BLOCK AND INLINE ELEMENTS

Block-level elements – always starts on a new line. Eg- <div> tag

Inline elements – doesn’t start on a new line. Eg - <span> tag

ID AND CLASS SELECTOR

Note: We use id to target one element and we use class to style a group of elements.

Format for id selector:

<li id=”demo”>

<style>

#demo{

}

</style>

Format for class selector:

<li class=”demo”>\_\_\_</li>

<style>

.demo{

}

</style>

LINKS

There are two types of links hyperlinks and local links(links to the same website).

Tag: <a>

Attribute: href for the name of the link, target for where to open the linked documents.

Target attribute:

\_blank: in new window/tab

\_self: in same window

\_parent: in parent frame

\_top: in full body of the window

IFRAMES

It is used to display a webpage within a web page. We can use embedded iframes of youtube and a lot other website and link the youtube videos in our website.

Syntax:

<iframe src=”url” target=”\_\_\_”></iframes

CSS3

**Cascading Style Sheets** (**CSS**) is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](https://en.wikipedia.org/wiki/Markup_language" \o "Markup language). CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

There are three types of styling:

1. Inline css – it uses style attribute in the html elements.
2. Internal css – it uses <style> tag inside <head> tag.
3. External css – it uses external css files.

Syntax for external css:

Selector{

Property: value;

}

Lets see some examples:

For inline css

<h1 style=”font-size:30px;”>\_\_\_\_</h1>

For internal css inside <head> tag

<style>

h1{

color: blue;

}

</style>

For external css

To link external css to the html code we have to add in the head section

<link rel=”stylesheet” href=”style.css” />

COLOR CODES

HTML supports 140 standard color names.

Color values:

1. RGB values (Red, Green, Blue): take values from 0 to 255
2. HEX values (Hexadecimal values): take values from 00 to ff
3. HSL values (hue, saturation, lightness): hue take values from 0 to 360, saturation and lightness take percentage values.
4. RGBA values alpha for transparency
5. HSLA values (take values from 0 to 1)

Some properties we should know:

Property Styling eg

Background color background-color: #333333

Text color color: white

Border border: 2px solid Tomato;

Font font-family: sans-serif;

Text size font-size: 30px;

Text alignment text-align: center;

Collapsed border border-collapse: collapse;

Width width: 30px;

Height height: 30px;

Background image background-image: url(\_)no-repeat

To learn what is difference between margin and padding refer to the css box model:



Margin has four properties:

* Margin-top
* Margin-right
* Margin-bottom
* Margin-left

Padding has four properties:

* Padding-top
* Padding-right
* Padding-bottom
* Padding-left

There are different ways by which we can mention these four properties in a single line.

Examples:

margin: auto; //equal spacing both from left and right

margin: 20px auto; //first spacing is for top and bottom and second for left and right

margin: 1px 2px 3px 4px;//first for top then right then bottom and the last for left

margin: 20px 5px 10px;//first for top then right and the last for bottom

BASICS OF TYPOGRAPHY

Let’s figure this out with examples

style.css file

body{

background-color: #ff7060;

}

h1{

font-size: 120px;

text-align: center;

margin-top: 50px;

font-family: sans-serif;

letter-spacing: -5px;

color: black;

text-transform: lowercase; (or uppercase)

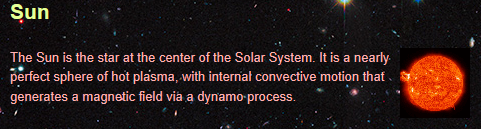
text-shadow: 0 1px 0 white;//first for x- axis then y-axis then blurness and at last color of the shadow

font-variant:

}

FLOATING

This property in the style attribute makes the content float to either left or right making space for other content. For example:



To make this type of layout use property

float: left; (or right)

HTML LINK COLORS

Html follow default link colors:

* An unvisited link is underlined and blue
* A visited link is underlined and purple
* An active link is underlined and red

We can do styling to the specific links by:

a: link{\_\_\_\_}

a: visited{\_\_\_\_}

a: hover{\_\_\_\_}

a: active{\_\_\_\_}

RELATIVE AND ABSOLUTE POSITIONING

Relative positioning is relation with the frame of the screen. Example:

position: relative;

top: 20px; this means the content is 20px from the top

bottom: 0; and 30px from left part of the screen

left:30px;

right:0;

Absolute positioning is with relation to the parent (Generally, body). This means absolutely position this within its nearest parent that is positioned. Example:

position: absolute;

left: 20px; from parent 20px to the left and 30px top

top: 30px;

NOTE: Absolute positioning works only when parent position is relative otherwise it work as relative positioning.

One important thing about css:

CSS is cascading styling. Here there is override of the property of the element. The lower it is in the code more likely it is to take precedence. Every element whether it is an id or a class or a tag name receives a specific amount of weight. The more is the weight the more precedence it will have. That is how cascading nature of CSS works.