

# Standard HTML Page elements – none of this is the visible content of the page:

<!doctype html>

**<html** **lang="**en**">**

**<head>**

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

**<title>**Page Title**</title>**

**<link rel="**stylesheet**" type=**”text/css” **“href="/**css/stylesheet.css**">**

**</head>**

**<body>**

**…**

**</body>**

**</html>**

# Elements that can be used in the body of a Webpage:

/\* Use div as a blocking element to group elements \*/

**<div>**

/\* header element \*/

**<h1>Heading 1</h1>** /\* and h2 through h6 \*/

/\* paragraph tag with nested span tag \*/

**<p>...<span>...</span>...</p>**

/\* image element – empty tag \*/

**<img src=**"picture.jpg" **alt="alternate text"/>**

**<p><em>***Emphasized***</em></p>**

**<p><strong>Important</strong></p>**

/\* line break \*/

**<br/>**

/\* link can be to another place in this file, to another file on the same website or to other website \*/

**<a href="url">**Link**</a>**

**/\***Lists **\*/**

**<ul>**

**<li>**unordered list item**</li>**

**</ul>**

**<ol>**

**<li>**ordered list item**</li>**

**</ol>**

**/\*** Table with one row and column**\*/**

**<table>**

**<tr>**

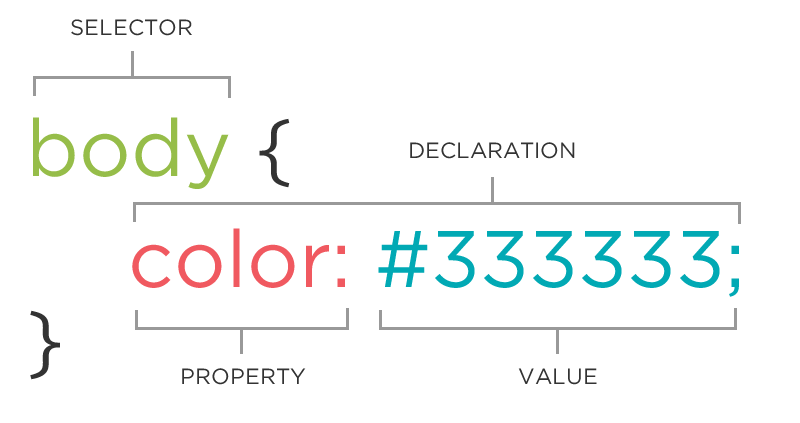
**<td>**Row 1 Col 1**</td>**

**</tr>**

**</table>**

**</div>**

# CSS – Cascading Style Sheets



Cascading Style Sheets are a different language from HTML, although HTML and CSS work together.

CSS is a set of rules with a selector and a declaration.

Declarations are property-value pairs.

Sometimes there are multiple values.

Text styling – or typeface is controlled by the font rules. They can be determined by individual rules or by a single rule with multiple values for the property **font.**

p{

font-style: italic;

font-weight: bold;

font-size: 10px;

font-family: Arial, Verdana, sans-serif;

}

p {

font: italic bold 10px sans-serif;

}

Colors can be designated in three ways, rgb values, the Hex equivalent, or a standard name:

p {

color: rgb(102,205,170);

}

p {

color: #66cdaa;

}

p { color: MediumAquaMarine; }

# What Cascading Means

 Inheritance

* Auto inheritance: *like* font-family, color
* No inheritance: *like* background-color, border
* Forced inheritance: inherit value keyword

 Precedence

1. !important keyword
2. Inline styles
3. Selector specificity
   1. Id **<p id=** “footer”**>**  #footer {…}
   2. Class **<p class=**“warning”**>** .warning{…}
   3. Element **<p>** p{…}

# Other Selectors

\* {…} universal selector – apply this rule to every element

div, p, img {…} multiple selector – apply this rule to these selectors

div p img {…} positional – apply this rule to image inside a paragraph after an h1 header

.warning.urgent multiple class selector – apply this rule to an element that has both warning and urgent classes

Three ways to add CSS to HTML

Inline:

**<p style=**”color:red;”**>**Some text**</p>**

In the header:

**<head>**

**<style type=”text/css”>**

p {

color: blue;

font-size: 12px;

}

**</style>**

**</head>**

In an external file:

<head>

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

# Pseudo-classes

a { color: FFFFFF; } /\* anchor\*/

a:link {color:#FF0000;} /\* unvisited link \*/

a:visited {color:#00FF00;} /\* visited link \*/

a:hover {color:#FF00FF;} /\* mouse over link \*/

a:focus {color:#FFFF00;} /\* holding pointer over link \*/

a:active {color:#0000FF;} /\* selected link \*/

# Box Model

Margin – distance between elements

Border – could be visible or not

Padding – distance between border and content

# 

# Positioning Schemes

**h1{position:static;}**

**Normal flow** every block-level element appears on a new line, even if there is room for two objects to appear on the same line.

**h1{position:relative;}**

**Relative positioning** moves an element from the position it would have had in normal flow by the amount specified. Doesn’t affect other elements. Becomes a “positioning context”

**h1{position:absolute;}**

**Absolute positioning** positions the element in relation to it’s containing element. It is taken out of normal flow and doesn’t affect surrounding elements.

**h1{position:fixed;}**

**Fixed positioning** a form of absolute positioning that positions the element in reference to the windows. The element stays fixed on the page, even during scrolling.

**h1 {float:left;}**

**Float**  takes the element out of normal flow and positions it to the far left or right of the containing box. The floated elements becomes a blocking element around which other content can flow.

# Z-index

**h1{z-index:10;}  
p{z-index:5;}**

When you move an element from normal flow, there’s a possibility that elements will overlap, z-index determines the order of overlap – the highest z-index number is on top.

# Clear

The clear property says that no element within the same containing element should touch the left or right (or both) sides of a box.