# HTML

## Basic Page Structure

<!DOCTYPE html> <!—Specific the version of html-->

<html>

<head>

<title>

</title>

</head>

<body>

</body>

</html>

## Hyperlinks

<a>: hyperlink

href (attribute): tells the browser where to link to

text between the <a> is how the link will appear on page

<a href=”http://www.google.com”>Google<a>

### To create a link to another section on same page

1. Create the anchor point on the section you want to reach

<a name=”section2”></a> or

<h2 id=”section2”>2</h2>

1. Use # symbol to tell the browser that the link is on the same page

<a href=”#section2”>Go to Section 2</a>

### To file

<a href=”page2.html”>Page 2<a>

### Opens email client

<a href=”mailto://me@myemail.com”>Email Me<a>

### Download

<a href=”/images/image.jpg” download>IMAGE DOWNLOAD</a>

## Images

<img src=”filename.jpg” alt=”Title” />

Alt: provides a text description of the image

# HTML/Tables

<table>

<!--Row--> <tr>

<th>Header/label elements of a table</th>

<th>First Name</th>

<th>Surname</th>

<th colspan=”2”>Name</th> <!--span multiple columns-->

</tr>

<tr>

<td>cell data</td>

<td>John</td>

<td>Doe</td>

<td rowspan=”2”>153</td> <!--span multiple rows-->

</tr>

</table>

# HTML/Forms

To gather user input

<form method=”get” action=”form.php”>

* Method: how form data is sent
  + Post: invisible
* Action: where to send the form

# HTML/Form, Inputs

## Form

<form action=”processingscript.php” method=”post”></form>

## Text

To collect short pieces of text from user

<input type=”text” name=”fname” />

## Password

Input password (displays input as dots)

<input type=”password” name=”pword” />

## Radio Buttons

<input type=”radio” name=”gender” id=”male” value=”M” />

<input type=”radio” name=”gender” id=”female” value=”F” />

* Value: send to php
* Id: association
* Name: use name to group them together

## Checkbox

<input type=”checkbox” name=”drink[]” value=”tea” id=”tea” />

<input type=”checkbox” name=”drink[]” value=”coffee” id=”coffee” />

## Reset

<input type=”reset” />

## Submit

<input type=”submit” name=”submit” value=”Submit” />

## Select

A drop-down list of options

<select name=”typeofblue”>

<option value=”blue1” selected=”selected”>Light Blue</option>

<option value=”blue2”>Medium Blue</option>

<option value=”blue3”>Bright Blue</option>

</select>

## Long Text Input

<textarea name=”message” rows=”4” cols=”50”>

</textarea>

## Label

<label for=”fname”>First Name:</label>

# HTML/Text Formatting

* Bold: <b> </b>
* Italic: <i> </i>

# HTML Description list

<dl>

<dt>Coffee</dt><dd>Black hot drink</dd>

<dt>Milk</dt><dd>White cold drink</dd>

</dl>

<dt>: defines terms/name

<dd>:describes each term/name

# CSS

## Applying CSS

Link your css file to html file:

If this file is saved as “style.css” in the same directory as your HTML page then it can be linked to in the HTML like this:

### Inline CSS

<p style=”color: red”>text</p>

### Internal

<style type=”text/css”>

….

</style>

### External

<head>

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

</head>

rel: relationship to page

type: type of document

Set the colour of the text for your webpage using hexadecimal notation

h1 {

color: #ffc;

background-color: #009;

}

# CSS/Class&ID Usage

## Class

### CSS:

.yellow-border{

…

}

### HTML:

<p class=”yello-border”>…</p>

## ID

### CSS:

#main-nav{

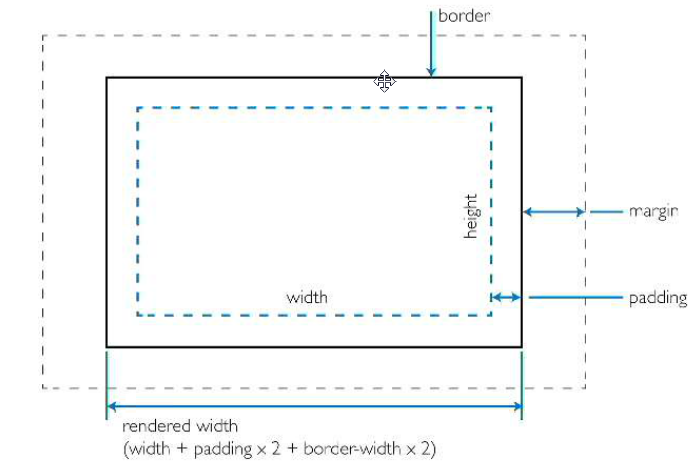
color: #336633;

}

### HTML:

<ul id=”main-nav”>…</ul>

Box Model in CSS



# CSS/Selectors

\*Pro Tip: In cases where there is only one element, that element counts as the first-child, only-child and last-child

## Descendant selector

* All element within ~
* #top h1 : any h1 element in id#top

## Child selector

* Parent>child
* #top>h1 (all immediate child h1 in #top)

## Adjacent selector

* h1+p (any p element that immediately follows h1)

## Only Child Pseudo-selector

* Select an element that are the only element inside of ~
* :only-child
* span:only-child : selects the <span> elements that are the only child of ~
* ul li:only-child selects the only <li> that are in a <ul>

## Last Child Pseudo-selector

* Select the last element inside of another element
* :last-child
* span:last-child : selects all last-child <span> elements.
* ul li:last-child selects the last <li> elements inside of <ul>

## Nth Child Pseudo-selector

* Select an element by its order in ~
* Nth(1st, 3rd, 12th)
* :nth-child(8) : selects every element that is the 8th child of ~
* div p:nth-child(2) : selects the second <p> in every div

## Nth Last Child Selector

* Select an element by its order in another element, counting from the back
* :nth-last-child(2) : selects all second-to-last child elements

## First of Type Selector

* Select the first element of a specific type
* :first-of-type
* span:first-of-type : selects the first <span> in any element

## Nth of Type Selector

* :nth-of-type(A)
* div:nth-of-type(2) selects the second instance of a div
* .example:nth-of-type(odd) : selects all odd instances of a the example class

## Nth-of-type Selector with Formula

* :nth-of-type(An+B)
* span:nth-of-type(6n+2): selects every 6th instance of a <span>, starting from the second instance (include)

## Only of the type selector

* select elements that are the only ones of their type
* :only-of-type
* P span:only-of-type : selects a <span> within any <p> if it is the only <span> in there

## Last of Type Selector

* Select the last element of a specific type
* :last-of-type
* div:last-of-type : selects the last <div> in every element
* p span:last-of-type : selects the last <span> in every <p>

## Empty Selector

* Select elements that don’t have children
* :empty
* Div:empty selects all empty <div> elements

## Negation Pseudo-class

* Select all elements that don’t match the negation selector
* :not(X)
* :not(#fancy) : selects all elements that do not have id=”fancy”
* Div:not(:first-child) : selects every div that is not a first child

# CSS/Pseudo Class

Selector:pseudo-class{}

* a:link : an universal link
* :visited : a link has been visited
* :hover : change style on mouseover (no IE<8)
* :active : change style when the moment clicked
* :focus : keyboard version of :hover for links

# CSS/BOX

Border-width: 1px;

Border-top-width: 1px;

Border-right-width: 1px;

Border-bottom-width: 1px;

Border-left-width: 1px;

Border-radius:25px;

Margin: 10px 5px 15px 30px;

Margin-top: 10px;

Margin-right: 5px;

Margin-bottom: 15px;

Margin-left: 30px;

Margin: auto

* Make the element at center

Border: 1px solid #333;

border-bottom: 2px solid #695968;

Border-width: 1px;

Border-style: solid;

* Dotted;

Border-color: #333333;

Border-radius: 20px;

* Rounder corner

# Measurements in CSS

## Fixed

Px: pixels on screen

Pt: points (text)

Mm, cm & in: for print media styling

Em: is relative to the font-size of the parent

Rem: is relative to the root or the html element. We can define a single font size on the html element and define all rem units to be a percentage of that.

html { font-size: 62.5%; }

body { font-size: 1.4rem; } /\* =14px \*/

h1 { font-size: 2.4rem; } /\* =24px \*/

# Colour in CSS

0: no light, 255: all light

* Hex: Each pair of digits maps to RGB declaration

Color: #BA55D3

* RGB: rgb(118, 126, 23)
* RGBA: plus alpha transparency -0 to 1
  + Rgba(186,85,211,1);
* Common color properties
  + Color
  + Background-color
  + Border-color

# Fonts in CSS

* Web safe fonts
  + Font must be installed on users computer to be displayed
* Font-stacks
  + Specify “fall-back” fonts to use in case a font is not installed
  + Should have minimum: desired + generic
  + Can have as many fallback as you like
  + Font-family: Desired, Fallback1, Generic;
  + font-family: "Trebuchet MS", "Lucida Grande", "Lucida Sans Unicode", "Lucida Sans", Tahoma, sans-serif;

# CSS/Text

* font-size: -px, em, pt
* font-weight: normal, bold
* font-style: normal, italic, olique
* text-align: center, right, left, justify
* text-decoration: underline, overline, line-through, none
* text-transform: uppercase, lowercase, capitalize
* text-indent:
* letter-spacing: space between characters
* line-height: space between lines

# Inheritance

* If no declaration is found for certain element properties in CSS, value is inherited from parent element.
* E.g: if the font-family is not set in CSS for h1, it will inherit it from the body element.

# CSS/Box Model

## Padding

* **padding: 25px 50px;**
  + top and bottom paddings are 25px
  + right and left paddings are 50px

## Border

* Border: stroke around the element
  + Border-width: 1px
    - Border-top-width: 1px
    - Border-right-width: 1px
    - Border-bottom-width: 1px
    - Border-left-width: 1px
  + Margin: 10px 5px 15px 30px
    - Margin-top: 10px;
    - Margin-right: 5px;
    - Margin-bottom: 15px;
    - Margin-left: 30px;
  + Border-style: dashed, dotted..
  + Border-color
  + Border: 1px solid #333;
    - Border-width: 1px;
    - Border-style: solid;
    - Border-color: #333333;
* Background Images
  + Background-color: “backup” in case image fails
  + Background-image: url("images/weather.png");
  + Background-repeat:
    - No-repeat
    - Repeat
    - Repeat-x
    - Repeat-y
  + Background-position
    - Top
    - Left
    - Right
    - Bottom
    - Center
* Dealing with collapse
  + Apply “clear” to an element directly following floats to prevent parent collapse
  + Clear: both;

# CSS/Navigation Bar

* List-style-type: none;
  + Removes the bullets
* Horizontal navigation bar

li{

display: inline;

}

# CSS/Arrangement

## CSS/word-wrap

Word-wrap: normal;

* Normal: break words only at allowed break points
* Break-word: allows unbreakable words to be broken

## CSS/Float

Float property can be used to wrap text around images

Image float and text wrap at the right side.

Img{

Float: right;

Margin: 0 0 10px 10px;

}

## Overflow

specifies what happens if content overflows an element's box.

div {  
    width: 150px;  
    height: 150px;  
    overflow: scroll;  
}

* Attributes:
  + Auto: If overflow is clipped, a scroll-bar should be added to see the rest of the content
  + Scroll: The overflow is clipped, but a scroll-bar is added to see the rest of the content

## Position

specifies the type of positioning method used for an element

h2 {  
    position: absolute;  
    left: 100px;  
    top: 150px;  
}

|  |  |  |
| --- | --- | --- |
| static | Default value. Elements render in order, as they appear in the document flow | [Play it »](http://www.w3schools.com/cssref/playit.asp?filename=playcss_position) |
| absolute | The element is positioned relative to its first positioned (not static) ancestor element |  |
| fixed | The element is positioned relative to the browser window |  |
| relative | The element is positioned relative to its normal position, so "left:20px" adds 20 pixels to the element's LEFT position |  |
| initial | Sets this property to its default value. |
| inherit | Inherits this property from its parent element. |  |

## Z-index

specifies the stack order of an element. An element with greater stack order is always in front of an element with a lower stack order.

z-index: -1;

|  |  |
| --- | --- |
| auto | Sets the stack order equal to its parents. This is default |
| *number* | Sets the stack order of the element. Negative numbers are allowed |
| initial | Sets this property to its default value. |
| inherit | Inherits this property from its parent element. |  |

# CSS/Gradients

Background: line-gradient(organe, red)

# CSS/Cursor

the type of cursor to be displayed when pointing on an element.

span.crosshair {  
    cursor: crosshair;  
}

* Attributes
  + pointer: The cursor is a pointer and indicates a link (hand)

# CSS/Layout

## Inline-block

create a grid of boxes

.floating-box {  
    display: inline-block;  
    width: 150px;  
    height: 75px;  
    margin: 10px;  
    border: 3px solid #73AD21;   
}

Display attributes

* table: Let the element behave like a <table> element

## Object in the center