Scott Browns HTML & CSS notes

<!DOCTYPE HTLM>

<HTML>

<Head>

</Head>

<Body>  
 </Body>

</HTML>

**Use firefox for testing** - test the site through a single browser, Firefox. After that, they check the same site in other browsers and continue to make the necessary changes from one browser to another.

**Thoroughly comment -** <!—THIS IS HOW TO COMMENT --> (one way)

**CSS reset register -** You can do all resetting at the beginning of the CSS file to avoid customizing each CSS rule that you write. You can rest assured that once you've added these rules to the style sheet (or to a separate file), it will work from the same point for all browsers.

* Use well-formed HTML.
* Pick good names and ID values.
* Indent consistently.
* Limit line length.
* Standardize character case.
* Use comments judiciously.

<a href=”reference link”>DisplayedText</a>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

Center element inside a container

<div id="container">

<div id="center-me">

</div>

</div>

CSS

 We will only use external stylesheets from the HTML document using the <head> tag. (EX; Below!)

A stylesheet is a list of ***rules***. Each rule (or "rule-set") consists of one or more ***selectors*** and

a ***declaration block***.

**Selectors**

* all elements with a specific HTML tag (e.g., p, h1)
* **id**(these are preceded by**#** in CSS) (Must be unique, can only be used once per page)
* **class** (these are preceded by **.**in CSS) (Used multiple times to share repeating CSS code)

**Declaration block**

Each individual declaration consists of a **property**, a colon (":"), and a **value**. If a block has multiple declarations, they must be separated by a semi-colon(";").

* Must be contained within braces
* Must have a a **property**, a colon (":"), and a **value**
* If a block has multiple declarations, they must be separated by a semi-colon(";

<link rel="stylesheet" type="text/css" href="styles.CSS">

Is one example of how to reference a CSS stylesheet and implement it into a webpage

#1. TAGS. #(tag name) { } (Use ID=”tag name:” in HTML)

#2. CLASSES. class="info"

#3 FONT SIZE. font-size:(number)px; (pt, px, em, %. pt and px)

#4. BACKGROUND COLOR. background-color:

#5 BORDER border: 1 px solid black; (Thickness/Type/Color)

#6. USE CLASSES. class="info"

#7. WIDTH width=

#8. BACKGROUND IMAGE background-image: url("cherries.png");

#9.BACKGROUND POS(Manual) background-position: center;

#10. BACKGROUND POS(Auto) background-size: auto;

#11. BACKGROUND REPEAT background-repeat: no-repeat;

#12. OVERFLOW TXT overflow: scroll; (Scroll, hidden) (height:auto)

#13. BORDER ROUNDED border-radius: 5px; (px or %) Thickness/Type/Color

#14. TEXT ALIGN text-align: center;

#15. TEXT DECORATION text-decoration: none;

#16. SAFE FONTS Verdana, Arial, Trebuchet MS, Times New Roman, Georgia, Andale Mono, Courier New, Comic Sans, and Impact.

#17. FONT STYLE font-style:

#18. FONT FAMILY font-family: (Used to have fallback fonts in case one doesn’t load)

#19. FONT WEIGHT font-weight: (normal, bold, numeric, semantic)

#20. DISPLAY display: (inline (Only whats needed) block (100%))

#21. MAGRIN margin: (Gap outside of border)(1st= top/bottom,2nd= left/right)

#22. PADDING padding: (creates a gap inside of border)

#23. FLOATING float: (left, right, up down)(floating all causes layout collapse)

#24 POSITION position: (absolute (Takes out of document flow, applies and directional rules that apply to it.) gets pushed however much you tell it to away from the nearest parent that has a position other than static, if Ex: position: absolute; top 40px)

(POSITION RELATIVE – In relative is not removed from document flow, document flow remains the same and movement rules apply normally) (POSITION FIXED – Will always be relative to the window no matter what parent has what position value, as you scroll the window is still in the same part of the screen)

WEIGHTS LIST

li {...} /\* a=0 b=0 c=1 -> specificity = 1 \*/

ul li {...} /\* a=0 b=0 c=2 -> specificity = 2 \*/

ul ol li {...} /\* a=0 b=0 c=3 -> specificity = 3 \*/

li.red {...} /\* a=0 b=1 c=1 -> specificity = 11 \*/

ul ol li.red {...} /\* a=0 b=1 c=3 -> specificity = 13 \*/

#list {...} /\* a=1 b=0 c=0 -> specificity = 100 \*/

* **a** represents the number of #id attributes in the selector
* **b** represents the number of class attributes
* **c** represents the number of tag names

inline cannot have a width or a height

inline block can, inline block

The clear Property

The clear property specifies what elements can float beside the cleared element and on which side.

The clear property can have one of the following values:

* none - Allows floating elements on both sides. This is default
* left - No floating elements allowed on the left side
* right- No floating elements allowed on the right side
* both - No floating elements allowed on either the left or the right side
* inherit - The element inherits the clear value of its parent

The most common way to use the clear property is after you have used a float property on an element

Center element inside a container

#container{

width:200px;

height:200px;

background: blue;

position:relative;

}

#center-me{

background: red;

height: 100px;

width: 100px;

margin: auto;

position: absolute;

left: 0;

bottom: 0;

top: 0;

right: 0;

}

OR

#container{

width:200px;

height:200px;

background: blue;

position:relative;

}

#center-me{

background: red;

height: 100px;

width: 100px;

margin: 0 auto;

}

**'console', 'elements'** **'sources'**

NOTES TO SELF!!!!!!!!!!!!!!!

Mark lots and lots of notes! <!-- -->

<div class="clearfix">

<div class="column menu">

<ul>

<li>The Flight</li> (Li has #1 priority on weight scale!)

<li>The City</li>

<li>The Island</li>

<li>The Food</li>

</ul>

</div>

Wrapper

Header

Navigation

main\_content

subcontents

advertisemen