**CSS (Cascading Style Sheets)**

Cascading Style Sheets (CSS) is a language in defining the document design of your content written in markup language specifically of Hypertext Markup Language (HTML). CSS is  developed by Hakon Wium Lie(HTML Cascading Style Sheets / CHSS), Bert Bos(Stream-based Style Sheet Proposal / SSP) and W3C.

Versions:

CSS Level 1

CSS Level 2

CSS Level 2 Revision 1

CSS Level 3

**(X)HTML Style Sheets**

* Sources of styles for HTML documents
* Author Styles can be specify by the author himself

External (aka linked) style sheets (recommended)

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

Embedded (aka internal) styles

<style>

   p{

   font-size: 1.5em;

   }

</style>

Inline styles (not recommended)

<p style=”color:red”> Hello World! </p>

User Style - can be specify by the user

User Agent Style - a browser has default style sheet

CSS Statements

**At-Rules** (**@**) - are set of instructions in a statement for the CSS/style sheet. /\*ayusin ko pa ‘to later\*/

o   @charset - this is the first element in the style sheet and used for declaring the characters that will be use

o   @import - it is used to import the style from different style sheets.

o   @media - is used for applying the styles of media queries in each type of the devices.

o   @font-face – allows the author to download fonts

o   @keyframes – it is used for animation

o   @page – characteristics of a page

CSS Selector

* It is used to select element of interest or elements you want to style
* The matched elements are the subject

**Selector Group**

**Definition:** It is a comma separated list of selecting an elements

**Example:** h1, h2, h3 {

Font-family: Sans-Serif;

     }

**Explanation:** Instead of having the style for h1, h2 and h3 in a separated block, when elements share the same value of style we can group it into one.

**Simple Selector**

**Universal Selector**

**Definition:** It is being indicate with the use of the (\*) asterisk sign. This type of selector is being used for selecting any kinds of elements in your Hypertext Markup Language or HTML.

**Example:** \*

**Explanation:** \* will simply match all the elements in your document

**Type Selector**

**Definition:** It selects all elements that match the given node name

**Example:** input

**Explanation:** In html, input tag is use to let the user input a data. In the example above, you can create a style for all of the input tag that you have in your html file. Type selector simply means you are referencing all elements that have this certain type.

**Class Selector**

**Definition:** Selects all the elements that have the same class attribute.

**Example:** .heading

**Explanation:** In your html file, you can have a class given the attribute ‘heading’. This will allow you to create one style for all the elements who would be using the class.

**ID Selector**

**Definition:** It is being used to select an element based on the value of its **id** attribute.

**Example:** #main-content

**Explanation:** Assuming in your html file, you have the element p, <p id = “main-content”> This is the main content. </p> and want to put a style in p element, in your css file (e.g styles.css) you have to reference your style to your html file by calling the id attribute.

**Attribute Selector**

Definition: It matches elements based on the value of a given attribute

Syntax:

[attr] – any

[attr=value] – specific

[attr~=value]  - the value is one of the variant

[attr|=value] – the value should be the beginning

[attr^=value]css3 – starts

[attr$=value] css3 – ends

[attr\*=value] css3 – contains

Example: a [href$=”.org”]

Explanation: Assuming you used the element a for hyperlinks. In the example, the attribute selector selects the a element where the the hyperlink reference ends with .org

**Combinators**

o   Descendant Combinator  - whitespace (i.e. space, tab, line feed, carriage return, form feed)

o   Child Combinator – (>)

o   Sibling Combinators

Adjacent Sibling Combinator – (+) immediately follows

General Sibling Combinator – (~) anything that follows

**Pseudo Classes**

1. Structural pseudo class

root pseudo class (:root) css3

First child pseudo class (:first-child) css3

li:first-child{

}

:last-child css3

 :only-child css3

 nth-child()css3

 nth-last-child()css3

first-of-type css3

 last-of-type css3

 only-of-type css3

 nth-of-type()css3

 nth-of-last-type()css3

 :empty css3– element with no children

b.  Target pseudo class

 :target css3

c.  Language pseudo class

 :lang()

d.  UI element pseudo class

§  :enabled css3

§  :disabled css3

§  :checked css3

§  :intermediated css3

Ø  Link pseudo class (Dynamic pseudo class)

§  :link

§  :visited

Ø  User action pseudo class (Dynamic pseudo class)

§  :hover

§  :active

§  :focus

Ø  Negation pseudo class

§  :not()css3

o   Pseudo elements

::first-letter

::first-line

::before

::after

**CSS Rule of Precedence**

**A.    By origin and importance**

o   User Agent Important Declaration

o   User Important Declaration

o   Author Important Declaration

o   Author Normal Declaration

o   User Normal Declaration

o   User Agent Normal Declaration

**B.      By specificity**

o   Inline styles (1) or not (0)

o   Number of ID selectors

o   Number of class selectors, attribute selectors and pseudo-classes

o   Number of type selectors and pseudo-elements

**C.     By order**