Most of the web pages and web applications that we can accessed via Internet are made from HTML. Yes, we are able to decorate web pages with colors, font styles, background image, hyperlinks and much more BUT it is totally limited, also it makes HTML documents difficult to read and maintain. So, to make it easier to give layout to web pages, style sheets are used. The widely used style sheet is **Cascading Style Sheets**.

**CSS (Cascading Style Sheet)** is a stylesheet language that describes the presentation of HTML or XHTML documents. Along with the HTML and JavaScript, CSS is used to websites to create layout and style for web pages, and user interfaces for web applications as well as for mobile applications.

For a diminutive outlook about the history of CSS, the first draft of CSS was proposed by **Hakon Wium Lie**, who worked with Tim Berners-Lee at CERN, in October 1994. **Bert Bos**, who responded to the first draft of CSS and was building Argo web browser at that time, decided to join forces with Lie.

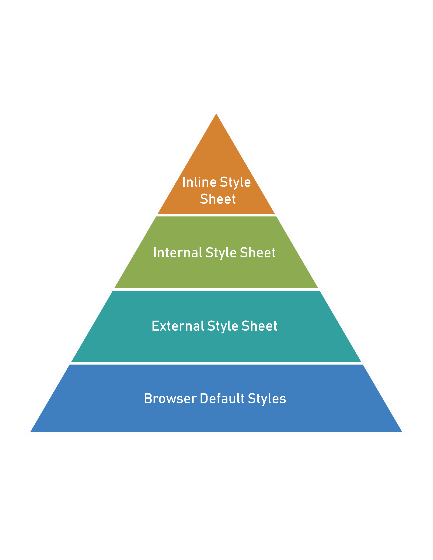
HTML and CSS are the core basis for building web pages and web applications. HTML provides the structure of the pages, while CSS provides the presentation or layout of the pages. The primary advantage of CSS is that it allows the separation of the content of HTML documents from the presentational aspect of the document. The separation of the stylesheet from the HTML makes it easier to maintain web pages and applications, and cascades by means of enabling to share and use one stylesheet across multiple web documents.

**Three Types of CSS**

1. External style sheets – The CSS instructions in this type of style sheet are within a separated file with .css file extension. It is the most used type of CSS, since it is accessible and can be applied to multiple web documents. External style sheets are defined using the **<link>** tag and places within the head section of a web document.
2. Internal style sheets – These style sheets are placed on top of each web document. If you want to have a unique style for a particular web document, it is the recommended type of CSS. It is defined using the **<style>** tag, where the CSS instructions located, and goes within the head section.
3. Inline style sheets – If you want to apply a single style of an element, inline style sheet is used. It is declared with the ***style*** attribute within a particular element and affects that element only.

**Style sheets by priority**

As stated before, it is possible to have a use multiple style sheets on a web document. For example, an external stylesheet and inline style sheet are used on one page. The question is, if a property that has been set for the same tag in different style sheet on a same page, what definition of property will be used? Of course, the style sheet that has the highest priority. Let’s based the priority level on the given diagram below:



**CSS Statements**

Statements are building blocks that begin with non-space characters and end with a semi-colon or first enclosing curly brace. There are two types of statements:

1. **CSS At-rules**

An at-rule statement starts with an at sign (@), and then followed by an identifier. Each type of at-rules may have its own internal syntax and semantics. These statements are used to convey conditional information, descriptive information and meta-data information. Here are some of examples of at-rules used in CSS:

* @charset
* @import
* @namespace
* @document
* @font-face
* @keyframes
* @media
* @page
* @supports

1. **CSS Rule-sets**

CSS Rule-sets consists of selector(s) followed by an enclosed declaration block, which contains one or more declarations. Each declaration must end with semi-colon, otherwise the declarations will not be applied except the first declaration.

**CSS Selectors**

Selectors outline the elements to which a set of CSS rules will be applied. A selector is part of the rule set that is being identified before the opening curly brace. If there are elements that have sample style definition, it is advisable to group the selectors so that the code is minimized. Each selector in a group selector is separated using a comma (,)

//Insert examples

**Simple Selectors**

Here are the brief descriptions of the different type of selectors:

* Universal Selector

The universal selector simply allows in selecting all elements in a page. It is declared using the asterisk (\*). Also, universal selector is often used in combination with other elements.

//Insert examples

* Element Type Selector

The element type selector, also referred to as “type selector”, must match HTML elements of the same name. Therefore, a selector of section should match all HTML **section** elements.

//Insert examples

* ID Selector

The ID selector matches any ID attribute providing a same value as of the selector excluding the hashtag symbol. The hashtag or pound symbol (#) is used to declare an ID selector. Note that the value of each ID attribute must be unique.

//Insert examples

* Class Selector

The class selector is declared using a dot preceding a string of one or more characters. It matches all elements on the page that have class attribute with a same value with the class selector excluding the dot. This selector is the most useful of all the CSS selectors.

//Insert examples

* Attribute Selector

The attribute selector matches any elements on a web document based on their attributes and attribute values. The syntax for this selector is consists of square brackets ([]) containing an attribute name followed by an optional condition in matching an attribute value. Note that there should be no space before the opening bracket unless the attribute selector is used along with a descendant combinator.

//Insert examples

**CSS Combinators**

Combinators are used to combine a selector to another selector. There are several combinators used in CSS that relies on the relationship between the selectors.

* Descendant Combinator

The descendant combinator targets child elements that are descendant of the parent element. It also includes elements that are not only direct descendants, these elements may be a child, grandchild, grandchild and beyond. Descendant combinator is technically one or more white space.

* Child Combinator

The child combinator is quite similar to descendant selector, but it only targets the child element of an element. Elements that are matched in the second selector must be the immediate children of the elements matched in the first selector. Technically, the child combinator is the greater sign (>)

//Insert example

* Sibling Combinators

There are two types of sibling combinators, namely adjacent sibling combinator and general sibling combinator.

1. Adjacent Sibling Combinator

The adjacent sibling combinator allows you to select an element that is directly after another specific element. The adjacent sibling combinator is the plus character (+).

//Insert example

1. General Sibling Combinator

The general sibling combinator is almost the same to the adjacent sibling combinator, since it also targets a sibling of a specific element. The only difference is that it may matches all the siblings as long as the hierarchy matches up. The combinator used here is the tilde character (~).

//Insert example

**CSS Pseudo Classes**

A pseudo-class is a keyword that included in a selector that defines a special state of the selected element(s). With the help of pseudo-classes, you do not need to use JavaScript to add effects to some selectors. For example, if you want to change a navigation tab’s color when user hovers over it, use the :hover pseudo-class.

//Insert example

Syntax

Here is a simple valid syntax of pseudo-classes:

selector: pseudo-class {

property: value;

}

If a CSS class is used with pseudo-classes, the syntax is as follows:

selector.classname:pseudo-class {

property: value

}

List of Pseudo Classes

Here is the list of most used pseudo-classes, and are categorized by its functionality:

1. Dynamic Pseudo Classes
   * Link pseudo-classes

:link

used to add style to unvisited links

:visited

used to add style to visited links

* + User Action

:active

used to add style to an active element

:hover

used to add style to an element when a user mouse over it

:focus

used to add style to an element while the element has focus

1. Target Pseudo Classes

:target

used to add style the current active target element

1. Language Pseudo Classes

:lang()

used to specify a language to use in a specified element

1. UI Element Pseudo Classes

:enabled

used to add style to any enabled element. It is only associated with form elements such as <input>, <textarea>, <select> and the like. An element is enabled if it can be activated, means it can be selected, clicked on, typed into, etc.

:disabled

used to add style to any disabled element. This class is vice versa of :enabled pseudo-class, stated above.

:checked

used to add style to any radio, checkbox, or option element that is checked or toggled to an **on** state.

:indeterminate

used to add style to any form element whose state is indeterminate

1. Structural Pseudo Classes

:root

This class matches the root element of a web document, for example, in HTML, the :root represents the <html> element. The only difference is that this the :root has higher specificity then the <html> element. So, it is used to add style to the root element of a document.

:first-child, :last-child, :only-child

used to add style to an element that is the first child of some other element. Same as on the situation of the last child and only child of an element.

:nth-child(), :nth-last-child()

used to add style to at least one element based on their position among a group of sibling elements. The :nth-child and “nth-last-child take an argument that can be a single integer, keywords (“odd” or “even”), or a formula.

The :nth-child() targets through elements starting from the top to bottom of a group of siblings. While, the :nth-last-child() starts from the bottom to top.

:first-of-type, :last-of-type, :only-of-type

used to add style the first element of its type among a group of sibling elements. It is quite similar to :nth-child() class, but this class is more specific.

:nth-of-type, :nth-last-of-type

Used to add style to one or more elements of a given type based on their position among a group of sibling elements. This class is quite similar to :nth-last-child() class, but it is more specific.

:empty

Used to add style to any element that has no children.

1. Negation Pseudo Classes

:not()

Used to prevent specific item or element from being selected. For example, you want to change the color of your page to red excluding those values in <p> element, use these rule set:

:not(p) {

color: red;

}