**Cascading Style Sheets (CSS)**

-used to specify the presentation (e.g Layout, formatting, fonts, colors, etc.) of structurally marked up documents

-developed by Håkon Wium Lie (CHSS) and Bert Bos (SSP) in World Wide Web Consortium (W3C)

CHSS means Cascading HTML Style Sheets

SSP means Stream-based Style Sheet Proposal

Version History

* CSS Level 1 (CSS 1, W3C Recommendation, Dec 1996)
* CSS Level 2 (CSS 2, W3C Recommendation, May 1998)
* CSS Level 2 Revision 1 (CSS 2.1, W3C Recommendation, Jun 2011)
* CSS Level 3

-modular approach to CSS development (as opposed to the monolithic specification of CSS 2.1) which is the last version

HTML Style Sheets

-sources of styles for HTML documents

* author styles
  + external (linked) stylesheets (recommended)
  + embedded (internal) stylesheets
  + inline styles
* user styles
  + some browsers provide (non-standard) alternatives through plugins/extensions (i.e Stylish, Web Maker, JSbin, CSS-Tricks)
* user agent styles
  + sample default user agent stylesheet from CSS 2.1
  + the source of a style denotes its origin, which is used in determining its precedence in the cascade

Extensions

* + Override
    - DOM - LEVEL-2-STYLE
  + Animation
    - CSS
  + Transition
    - CSS

CSS Statements

* At-Rules (@)
  + @charset
  + @import
  + @namespace
  + @media
  + @supports
  + @page
  + @font-face
  + @keyframes
* CSS Rule Sets (a.k.a CSS Rules, Style Rules)

-consists of a selector, followed by a brace-enclosed declaration block which contains zero or more semi-colon ( ; ), separated property declarations, which in turn consists of a property name, followed by a colon ( : ), followed by a property value

/\* example of a CSS Rule… \*/

h1 + ul > a [target] {

background-color: #bada55;

color: yellow;

font-family: ‘Times New Roman’, serif;

margin: 10px 20px;

padding: auto 1em 0 20px;

}

CSS Selectors

* Selector (Level 3)

-structure used as a condition in CSS rule to determine which elements in the document tree and matched by the selector and are thus targeted

* + Selector Syntax

-chain of one or more sequences of simple selectors separated by combinators

img.brand [src\*=google]

type element

attribute element

class element

\* = indicates that it matches everything

article [date-time=latest] > ul: last-of-type + p.note

section#adverts: hover > header.info + \*[title]::after

pseudo element

* + - Sequences of Simple Selectors

-chain of simple selectors not separated by combinators

Combinators

-are used to improve additional matching constrains

* Selector Group

-comma-separated list of selectors representing the union of all elements selected by each of the selectors in the list

Types of Combinators

* Descendant Combinator - (whitespace)
* Child Combinator (>)
* Sibling Combinator
  + Adjacent sibling combinatory (+) – immediately after or the first
  + General sibling combinatory (~) – any sibling after

Pseudo Elements

* :first-letter, ::first-letter
* :first-line, ::first-line
* :before, ::before
* :after, ::after

(^) –pre/ first

($) – post/ last

Simple Selectors

* Universal Selector – matches everything
* Type Selector – different elements
* ID Selector – value of the id, case-sensitive (#)
* Class Selector – class variable
* Attribute Selector – value of attributes (i.e p[class=lead])
* Pseudo-Classes
  + Dynamic Pseudo-Classes
  + Target Pseudo-Classes
  + Language Pseudo-Classes
  + UI Element State Pseudo-Classes
  + Structural Pseudo-Classes
  + Negation Pseudo-Classes

CSS Rule Precedence

-an HTML element may be the subject of the selectors of multiple style rules

-when such rules target different properties

* Resolution
  + By origin and importance
    - Important
      * Important user-agent declarations
      * Important user declarations
      * Important author declarations
    - Normal
      * Normal author declarations
      * Normal user declarations
      * Normal user-agent declarations
    - Note: Highest precedence: **Transition declaration**
  + By specificity
    - Count the number of ID selectors in the selector
    - Count the number of class selectors, attribute selectors, and pseudo-classes in the selector
    - Ignore the universal selector
  + By order (last priority)
    - The later takes prioritize

CSS Declarations

Each declaration contains a property

* Properties
  + Shorthand properties
    - Allows authors to specify values of several properties with a single property
    - eg. background, font, margin, padding
    - Border-width-shorthand property (top, right, bottom, left)
  + Vendor specific extensions (vendor prefixes)
    - Used by browser vendors as a prefix for the name of experimental or non-standard CSS properties; lately, vendors are moving away from vendor prefixes in favor of the user controlled flags or preferences
    - eg.(-webkit-),(-moz-),(-o-),(-ms-)
  + custom properties a.k.aa CSS variables (experimental)
    - property names prefixed with “—“, representing a value that can be reused throughout the document using the var() function.
* Values
  + Value processing
    - Declared, cascaded, specified, computed, used, actual values
  + Value type
    - Keywords
      * CSS-wide keywords
        + Initial, Inherit, Unset
      * Property- specific keywords
    - Numbers
      * Integers (scientific) decimal notation
    - Dimensions
      * Length, angle, duration (or time), frequency, resolution
      * Length units
        + Font relative: em, ex, ch, rem
        + Viewport-percentage: vw, vh, vmin, vmax
        + Absolute: cm, mm, in, pt, pc, px
      * Angle units
        + Deg, grad, rad, turn
        + Used in some gradient and transform functions
      * Duration (or time) units
        + S, ms
        + Used in animation, transition, and relation properties
      * Frequency units
        + Hz, kHz
        + Initially introduced in CSS2 for the (obsoleted) aural media type; reintroduced in CSS3 but is currently unused
      * Resolution units
        + Number with a % suffix
        + Calculated as a percentage of some value( usually taken from the parent element)
      * URLs & URIs
        + url() function with an absolute or relative
        + denotes a pointer to a resonte such as an image or a font
      * colors
        + color keywords: red
        + RGB hexadecimal notation
        + RGB functions

rgb (255,0,0)

* + - * + HSL functions

Hsl(0,100%,0)

Hsl color wheel

* + - * + Currentcolor, transparent
      * Strings
        + Delimited by single quotes(‘) or double quotes (“).