**Cascading Style Sheets (CSS)**

-used to specify the presentation 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

/\*CSS Comments \*/

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, their effects cascade (i.e. are combined)
* When the style involve the same property, they conflict, and must be resolved such that only one style is applied
* Resolution
  + By origin and importance
  + By specificity
  + By order (last priority)

Important

1. Important user agent declarations
2. Important user declarations
3. Important author declarations

Normal

1. Normal author declarations
2. Normal user declarations
3. Normal user agent declarations

* Count the number of ID selectors in the selector (=a)
* Count the number of class selectors, attribute selectors and pseudo classes in the selector (=b)

CSS Declarations

* Properties
  + Shorthand Properties
    - Allows authors to specify the values of the several properties with a single property

i.e. background, font, margin, padding, border, border-left, border-width, etc.

* + - Vendor Specific Extensions (a.k.a Vendor Prefixes)
      * Used by the browsers vendors as a prefix for the names of experimental or non-standard CSS properties; lately, vendors are moving away from vendor prefixes in favor of user-controlled flags on preferences

i.e. –webkit-, -moz-, -o-, -ms-, etc.

* + - * Custom properties (a.k.a 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 Types
    - Keywords
      * CSS-Wide Keywords
        + Initial, Inherit, Unset
      * Property-Specific Keywords
    - Numbers
      * Integers or reals in (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 related properties

* + - * + Frequency Units

Hz, kHz

Initially introduced in CSS2 for the (obsoleted) aural media type; reintroduced in CSS3, but is correctly unused

* + - * + Resolution Units

Dp, dpcm, dppx

Used in media queries

* + - Percentages
      * Number with a% suffix
      * Calculated as a percentage of some value (usually taken from the par)
    - URLs and URIs
      * url() function with an absolute or relative (to the stylesheet) URL parameter
      * denotes a pointer to a resource, such as an image or a font
    - Colors
      * Color keywords
      * RGB hexadecimal notation
      * RGB functions
      * HSL functions
      * Currentcolor, transparent
    - Strings
      * Delimited by single quotes (‘) or double quotes (“)
    - Functions

CSS Preprocessors. Frameworks and Polyfills

* CSS Preprocessors
  + Generate CSS a custom language syntax that typically includes features that don’t exist in pure CSS (e.g. Variables, Control Flow, Nesting, Mixins, Functions, etc.)

e.g SASS, LESS, Stylus, SCSS \*more popular

* CSS Frameworks
  + Provides predefined CSS design functionality that can be reused, extended and customized

e.g Bootstrap, Foundation, Maximizer, etc.

* CSS Polyfills
  + Provides future developers expect browsers to provide natively

e.g Modernizr, Selectivizr, etc.

Sites:

colour-affects.co.uk/psychological-properties-of-colours

quackit.com/CSS/functions/

develop.mozilla.org/en-US/docs/Web/CSS/CSS-Type

saas-lang.com/documentation/Saas/script/functions.html