March 3

Cascading Style Sheets (CSS)

* Language used to specify the presentation (e.q. Layout formatting, fonts, colords, etc ) of structurally marked up documents (e.q. HTML, XML, SVG, MathML, etc.)
* Developed by Hakon Wium Lie (CHSS - Cascading HTML Style Sheets v) and Bert Bos SSP
* Version history
* CSS Level 1 (CSS 1, W3C Recommendation, Dec 1996)
* CSS Level 2 (CSS 2 W3C Recommendation, May 1999)
* CSS Level 2 Revision (CSS 2.1, W3C Recommendation, June 2011)
* CSS Level 3
  + Modular approach to CSS development (as opposed to the monolithic specification of CSS 2.1)

HTML Stylesheets

* Sources of styles for HTML documents
  + Author styles
    - External (a.k.a linked) stylesheets (recommended)
    - Embedded (a.k.a. internal) stylesheets
    - Inline stylesheets
  + User styles
    - Some browsers provide (non-standard) alternatives through plugins/extensions
  + User Agent Styles
    - Sample default usre agent stylesheet from CSS 2.1
  + The source of a style denotes its *origin,* which is used in determining its precedence in the cascade

March 7

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 properly declaration which in turn consists of a property name followed by a colon (:) followed by a property value
  + //// insert example here…

CSS Selectors

* Selector
  + Structure used as a condition in a CSS rule to determine which elements in the document tree are matched by the selector and are thus targeted by the formatting specified in the CSS rule declaration block
  + The matched elements are called the subjects of the selector
* Selector syntax:
  + Chain of one or more sequences of simple selectors separated by combinators with one pseudo-element possible appended to the last sequence
    - Sequence of simple selectors
      * Chain of simple selectors not separated by combinators
      * Always starts with a type selector or a universal selector (may be implied)
      * Cannot contain other type selectors or universal selectors
  + Combinators are used to impose additional matching constraints, applied by prepending another sequence of selectors and the combinator to a sequence, with the subjects of the entire selector being some subset of the elements matched by the last sequence.
    - ///insert example here...
  + Selector group
    - Comma-separated list of selectors representing the union of all elements selected by each of the selectors in the list
* Simple selectors
  + Class selectors
  + //example
  + Attribute selectors
  + //example
  + Pseudo classes
  + //example

March 14

* Combinators
  + Descendants combinator (whitespace, space, tab, line feed, carriage return, from feed)
  + Child combinator (>)
  + Sibling combinator
    - Adjacent sibling cimbinator (+)
    - General sibling combinator (~)
* Pseudo-elements

March 17

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 styles 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

Origin and Importance

The origin of a declaration is based on where it comes from and its importance is whether or not is declared “important” (see below). The precedence of the various origins is in descending order.

1. Transition declarations [CSS3-TRANSITIONS]
2. Important user agent declarations
3. Important user declarations
4. Important override declarations [DOM-LEVEL-2-STYLE]
5. Important author declarations
6. Animation declarations [CSS3-ANIMATIONS]
7. Normal override declarations [DOM-LEVEL-2-STYLE]
8. Normal author declarations
9. Normal user declarations
10. Normal user agent declarations

9. Calculating a selector’s specificity

A selector’s specificity is calculated as follows:

· Count the number of ID selectors in the selector (= a)

· Count the number of class selectors, attributes selectors, and pseudo-classes in the selector (= b=)

· Count the number of type selectors and pseudo-elements in the selector ( = c)

· Ignore the universal selector

Selectors inside the negation pseudo-class are counted like any other but the

CSS Declarations

* Properties
  + Shorthand properties
    - Allows authors to specify the values of several properties with a single property.
* E.g., background, font, margin, padding, border, border – left, border – width, etc.
  + Vendor specific extensions (a.k.a., vendor prefixes)
    - Used by browser 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 or preferences
* E.g., -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, corrupted, used