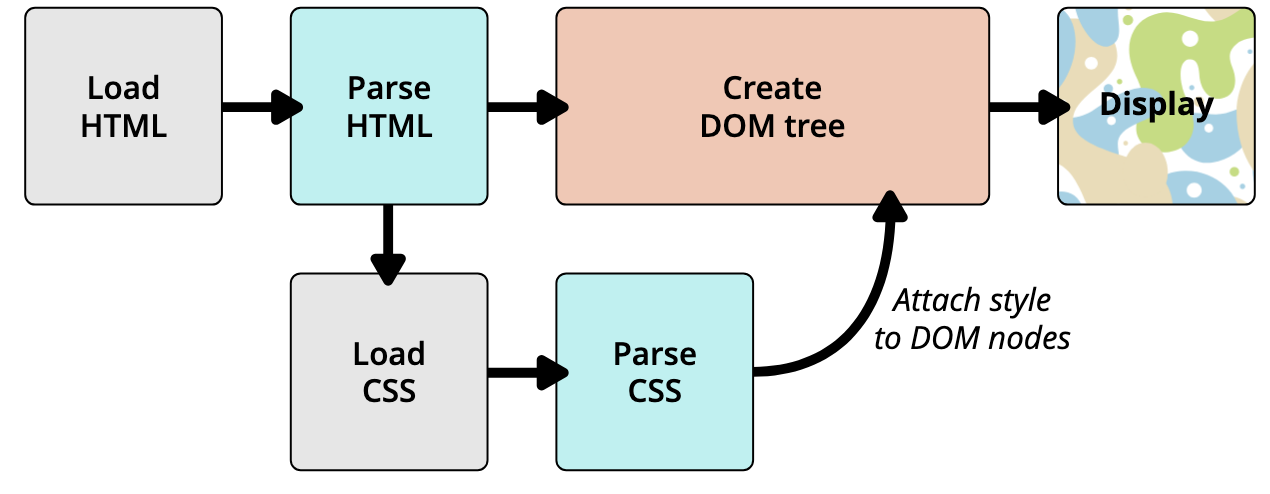
CSS

Overview

Cascading Style Sheets (CSS) are used to define where things are on a webpage and make it look pretty.

CSS process:

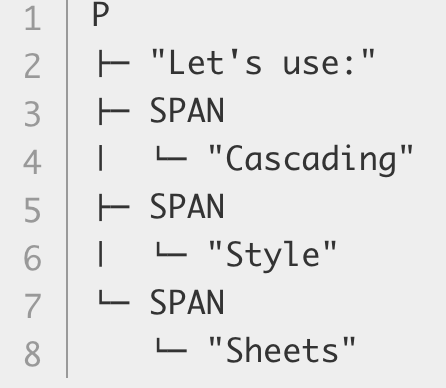
* Browser loads HTML
* Converts HTML into Document Object Model (DOM) - DOM represents the document in the computers memory
* Browser fetches resources that a linked to by the HTML document (images, videos, CSS)
* Browser parses CSS and sorts the different rules by their selector types into different ‘buckets’ (element, class, ID, etc), based on select it finds which rules should be applied to which nodes in the DOM and and attaches style as required
* Render tree is layed out after rules applied
* Visual display of page is shown on screen



DOM

Dom has a tree like structure with each element, attribute and bit of text becoming a node. Nodes are defined by their relationship to other nodes, some being parents and some children/siblings.

For example, the <p> element is a parent and its children would be a text node and three span elements, each span having their own text node children

Selectors

CSS can be applied by specifying the node which it affects using a selector. Nodes can be selected directly using element type or attributes added to an element:

* <element> - selected with element name
* type=”<type”> - selected with <element-name>[type=<type>]
* class=”<name” - selected with . before name
* id=”<anme>” - selected with # before name

The class attribute can be used to identify more than one element, where as id is unique and can only select one element.

*body {*

*font-family: Arial, Helvetica, sans-serif;*

*}*

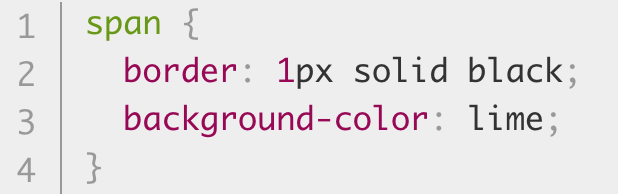
*.job-title {*

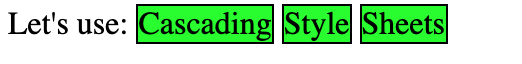
*color: #999999;*

*font-weight: bold;*

*}*

In the case below the selector picks all span elemetns and applies style to those in the document:





Selectors also have the abilty to chain attributes and values, such as a link can be selected then the attribute visited specified, to change the colour of the link once it has been visited.

*a:link, a:visited {*

*color: #fb6542;*

*}*

Incorrect CSS

If the browser encounters CSS it doesn’t understand, it will just fail silently and move on. Therefore, CSS can be used as an enhancement since any incorrect styling will be ignored.

Newer functions can be added after old ones which will then overwrite the old ones if supproted.

* class=”<name” - selected with . before name
* id=”<anme>” - selected with # before name

Normalize

Browsers have default CSS which makes developing applications for all browsers harder. It is common to use a normalise CSS sheet which resets the default CSS of a page. The normalise sheet is the first thing loaded in CSS, so everything else always has the same canvas to work on.

Fonts

Fonts can be added to the server then used in a webpage using the @font-face rule. For example:

Loading:

@font-face {

font-family: ‘<font-name>’;

src: url(<font-url>') format('truetype'); /\* Safari, Android, iOS \*/

}

Usage:

html {

font-family: '<font-name>', Fallback, sans-serif;

}

Variable Fonts

A variable font is a single font file, which behaves like mutliple styles, such as different weighting and italics. Since there is only a single font file for one family, variable fonts can dramatically increase webpage speed and it is supported on most modern browsers.

Download a variable font file, these are generally in ttf, a full system font, and therefore should be convert to WOFF2 to compatibility and compression. To convert to a WOFF2 subset of fonts, it is possible to use pyftsubset in they python library fontTools.

For webpages in english it is typical to include the following unicodes:

* U+0020-004F - standard space & numbers & letters & symbols
* U+00A0-00FF - latin supplement (copyright and fractions)

Therefore, to cut down a full fontset to these vital charaters use:

pyftsubset <font-name>.ttf --unicodes=’U+0020-004F,U+00A0-00FF’ --flavor="woff2" --output-file="<font-name>.woff2"

To use, include using the @font-face rule, set the source and backup for non-compatible browsers, and set a weight range for the font:

@font-face {

font-family: 'RobotoBody';

src: font-url('Roboto-Variable.woff2') format(woff2), font-url('Robot-Regular.woff2') format(woff2);

font-weight: 1 999;

}

@media

To create responsive webapp, there must be different CSS set for different sizes of device screen. The @media rule is used to include a block of CSS only if certain conditions are true.

Media breakpoints (pixel limits) should be based on the size of the content (images etc) included in the site.

Notes:

* Logos and icons should all be in SVG vectors so the scale correctly on any screen size.