{ CSS Training }

*#1*

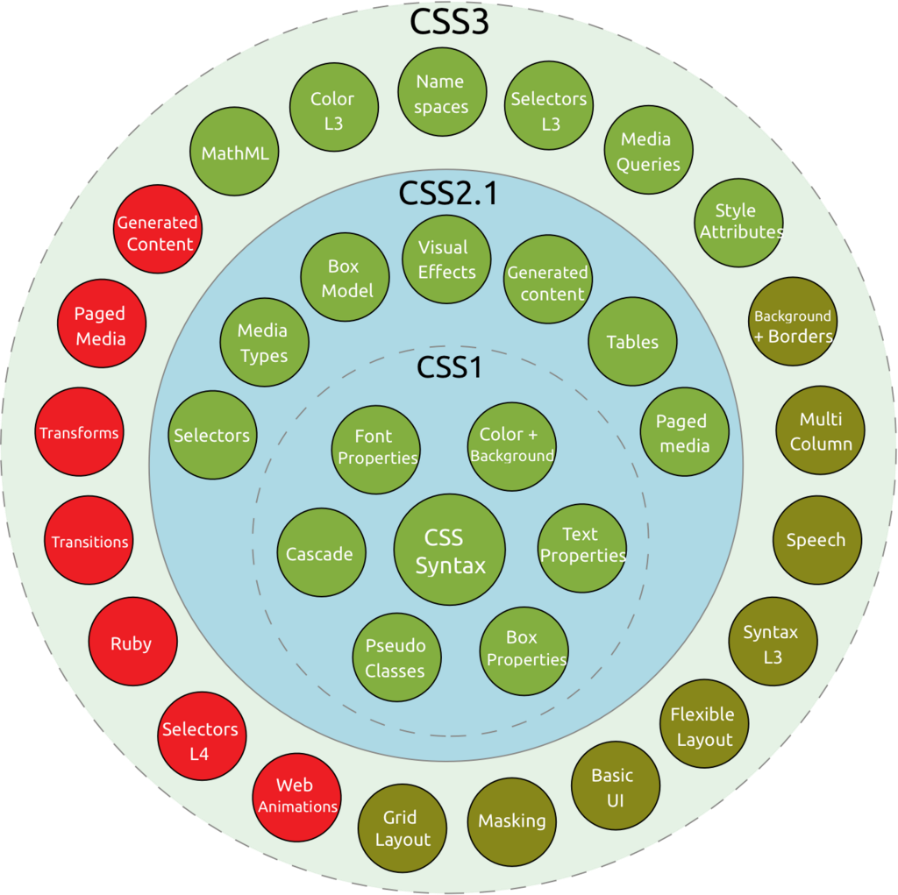
*History*

CSS Level 1 – 1996

CSS Level 2 – 1998

CSS 2.1 – 2011 removes poorly supported or not fully interoperable features

CSS 3 – 2011-2012 divided into several separate documents called "modules"



CSS 4 – 2017 (snapshots)

browser support: <https://css4-selectors.com/browser-selector-test/>

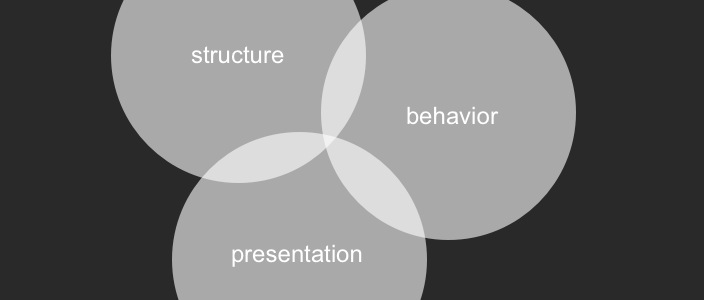
Separation of Concerns

* Separation of concerns is a guiding principal in software development centered on the idea

that programs should have distinct sections, with each section being responsible for its own concern.

* Separating HTML + CSS + JS

(Browsers also have HTML parser, CSS parser and JS engine)



*CSS meaning*

* **CSS** stands for Cascading Style Sheets (egymásba ágyazott stíluslapok)
* CSS describes **how** HTML elements are to be displayed on screen, paper, or in other media
* It can control the layout of multiple web pages
* External stylesheets are stored in CSS files

<link rel="stylesheet" type="text/css" href="style.css">

<style>  
 ... internal stylesheet...

</style>

*Syntax*



p {  
    text-align: center;  
    color: red;

/\* This is a comment \*/  
}

<p class="center large">This paragraph refers to two classes.</p>

* Selectors select html elements on a specific criteria.
* Selectors can be written in groups:

h1, h2, p {  
    text-align: center;  
    color: red;  
}

*Selectors*

Basic selectors:

* Type selector

h1 { ... }

* Class selector

.classname { ... }

* ID selector

#id { ... }

* Attribute selector

[attr] { ... }

[class="parent"] or [data-sso="1"]

* Universal selector = matches all the elements

\*

*Combinators*

Combinators:

* Child combinator = direct children of the first element

A > B div > p

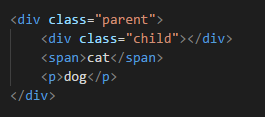
* Descendant combinator = match all „p” elements that are inside a „div” element

A B div p

* Adjacent sibling combinator = second element directly follows the first and has the same parent

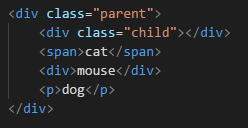
A + B span + p

will match all „p” (!) elements that follow a span



* General sibling combinator = match all „p” elements that that follow a span

A ~ B span ~ p



*Pseudo classes and pseudo elements*

Pseudo classes:

* [Pseudo-classes](https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes) allow the selection of elements based on state information that is not contained in the document tree.

a:visited will match all [<a>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a) elements that have been visited by the user

[Pseudo elements](https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements):

* represent entities that are not included in HTML (new fake element)



* Double colon :: syntax

<https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Pseudo-classes_and_pseudo-elements>

CSS cascade

What selector win out in the cascade depends on:

* Target

media type

* Importance

for example: ID is unique so targetting that wins over class selectors

inline style > !important > ID > class > elements

* Specificity

measure how specific a selector is

override something with the same specificity later in the source order, or one with a higher specificity

* Source order

later rules win over earlier rules

* Inheritance

browser default style used when no other styling is set

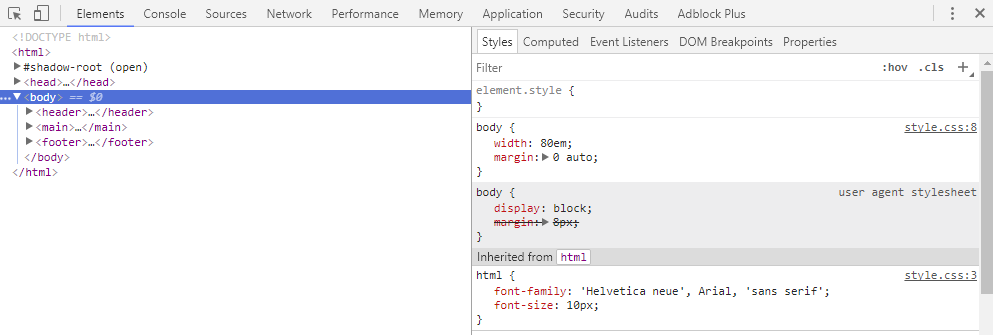
!important

* always win over all others
* override a style that can’t be overridden in any way
* don’t use it unless you absolutely have to

Debugging CSS

CSS is permissive. (Page still displays even if there are syntax errors. You will get something even if it is not what you expect.)

Developer tools: page inspector, CSS editor



HTML validator: <https://validator.w3.org/>

CSS validator: <https://jigsaw.w3.org/css-validator/>

CSS tools

CSS preprocessors:

less <http://lesscss.org/> .less

sass <http://sass-lang.com/> .sass / .scss

stylus <http://stylus-lang.com/> .styl

post css <http://postcss.org/>

myth <http://www.myth.io/>

...

DRY principle = don’t repeat yourself

( JS equivalent:

Transpiler = source-to-source compiler, produces the equivalent source code in another programming language.)

CSS minifier: Removing unnecessary characters without changing the functionality

20% smaller -> faster download time

<https://csscompressor.com/>

...

Not obfuscator – but part from facebook:

<div class="\_cy6 \_2s24">

<div class="\_4kny">

<div class="uiToggle \_8-a \_1kj2 \_4d1i \_-57 \_5-sk"...

Well if your class name is *facebook-header-panel-main-text-button* and you shorten it to \_af243 you've saved quite a bit of bandwidth, right?