

Effective styling on Web and best practices

1) History

2) What was before CSS?

3) CSS arrival

4) CSS complications

5) Methodologies and architectures

6) What about styling with JavaScript?

How to create web design effectively

Problem



Specific
solution

Dr. Håkon Wium Lie



E-mail communication

<http://1997.webhistory.org/www.lists/www-talk.1994q1/0648.html>

> Now when viewing this under Lynx I get what I think is the correct
> response:

```
>
> =====LYNX DISPLAY=====
> GB
> Brunel University
> Counsellors
>
> [1]Dr S Smith
>
> Economics
>
> [2]Dr J Smith
> =====LYNX DISPLAY=====
```

> ***BUT*** when I view the same HTML with Mosaic I get:

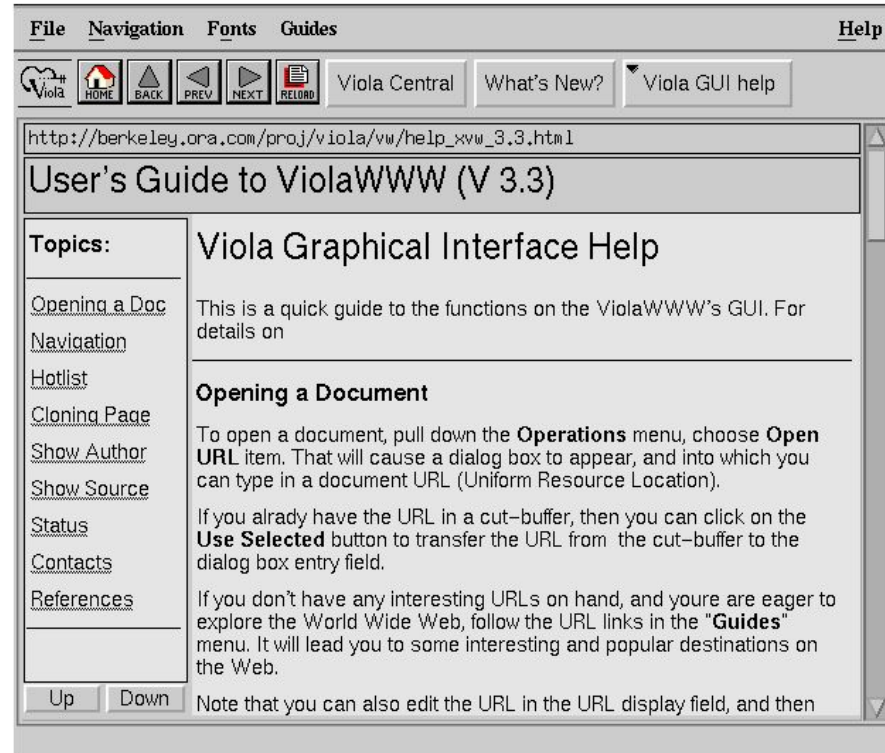
```
> =====MOSAIC DISPLAY=====
> GB
>
> Brunel University
> Counsellors
>
> Dr S Smith
>
> Economics
>
> Dr J Smith
> =====MOSAIC DISPLAY=====
>
> As you can see Mosaic has added in an extra blank line. Am I right in
> assuming that this is infact a bug in the mosaic rendering an that the
> LYNX output is correct.
>
> If not, how would I go about doing it?
```

It's not a bug -- it's a rendering choice made by the browser. As we kneel before the altar of the "ha ha, you can't control what your documents look like in HTML" philosophy, you, dear sinner (nay, blasphemer), can but take solace in the fact that Mosaic is putting a blank line at the start of all toplevel lists, including yours, entirely on purpose. (Why? Primarily because we thought that looked the best for most documents on the Web at the time, as I remember.)

What can you do about it? Probably nothing. Isn't that cheery news? I think so. In fact, it has been a constant source of delight for me over the past year to get to continually tell hordes (literally) of people who want to -- strap yourselves in, here it comes -- control what their documents look like in ways that would be trivial in TeX, Microsoft Word, and every other common text processing environment: "Sorry, you're screwed."

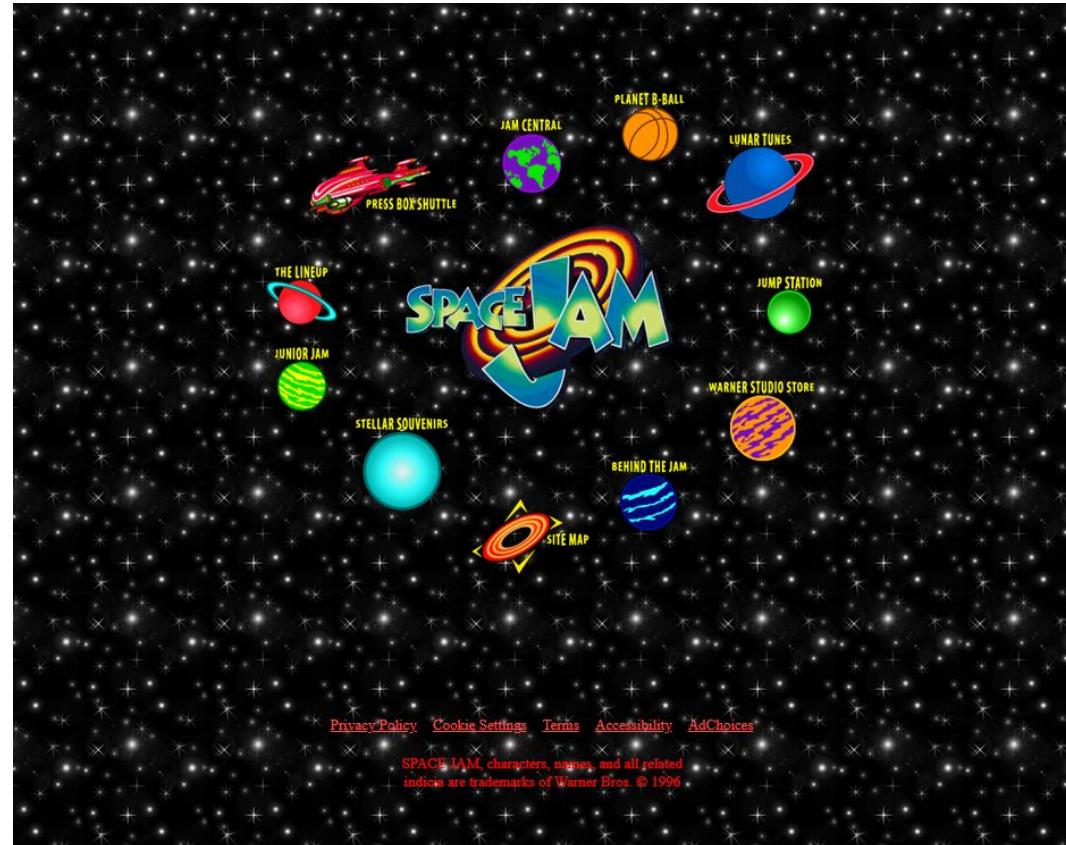
**What was before CSS
came along?**

ViolaWWW



[ViolaWWW browser \(source\)](#)

Space Jam



Space Jam

Colors

Layout


Space Jam - colors

[The Filmmakers](#) · [The Trailer](#) · [Photos From The Film](#)

PRODUCTION NOTES

- [About the Story](#)
- [About the Production](#)
- [The Jump Ball](#)
- [Halftime](#)
- [Voices, Please](#)

"Ehhh, what's up, Doc?"




Warner Bros.' Looney Tunes, led by the creative impetus of filmmaker IVAN REITMAN, blast into the future in their first feature film, "Space Jam," an animated comedy starring basketball star MICHAEL JORDAN and classic wisecracking animated hero BUGS BUNNY.

As audiences visually zoom between Earth and outer space, between basketball court, baseball field and Looney Tunes Land, from live action with some of the most talented athletes in the world to animation, they'll get a laugh-loaded glimpse of the future as it's never looked before -- in the company of basketball's most watchable genius and the hip-hoppin'-est hero in animation.

Ivan Reitman produces and JOE PYTKA directs "Space Jam," a fast-breaking comic fantasy. JOE MEDJUCK and DANIEL GOLDBERG also produce with Reitman, from a script by TIMOTHY HARRIS & HERSCHEL WEINGROD. DAVID FALK and KEN ROSS executive produce. GORDON WEBB, SHELDON KAHN and CURTIS POLK co-produce.

"Space Jam" mixes live action with eye-popping animation created by WARNER BROS. FEATURE ANIMATION under animation producer RON TIPPE, animation co-directors BILL PERKINS and director BILL PERKINS. Special live action/animation visual effects are created by CINESITE under the direction of ED JONES. Warner Bros. will distribute "Space Jam" worldwide.

About the Story...



Bugs Bunny has gotten himself and his Looney Tunes cohorts into a jam by facing off against the Nerdluks, a grouchy gang of tiny space creatures who land on Earth. Swackhammer (voice by DANNY DeVITO), intend to kidnap and export the Looney Tunes to Moron Mountain, Swackhammer's failing theme park on the moon.

Bugs has challenged the small, weak aliens to a fateful basketball tournament: if the Looney Tunes win, they'll remain on Earth. But if the aliens win, Bugs and his cohorts will be sent to a single cell...

The aliens quickly take up Bugs' challenge -- too quickly, it turns out. Bugs hasn't found out yet about the Nerdluks' ability to instantly "absorb" new skills from their opponents. In a matter of minutes, the Nerdluks have absorbed the likes of NBA stars CHARLES BARKLEY, PATRICK EWING, MUGGSEY BOGUES, LARRY JOHNSON and SHAWN BRADLEY, turning the Nerdluks into the powerful, speed-demon basketball team that Bugs and his cohorts will have to face.

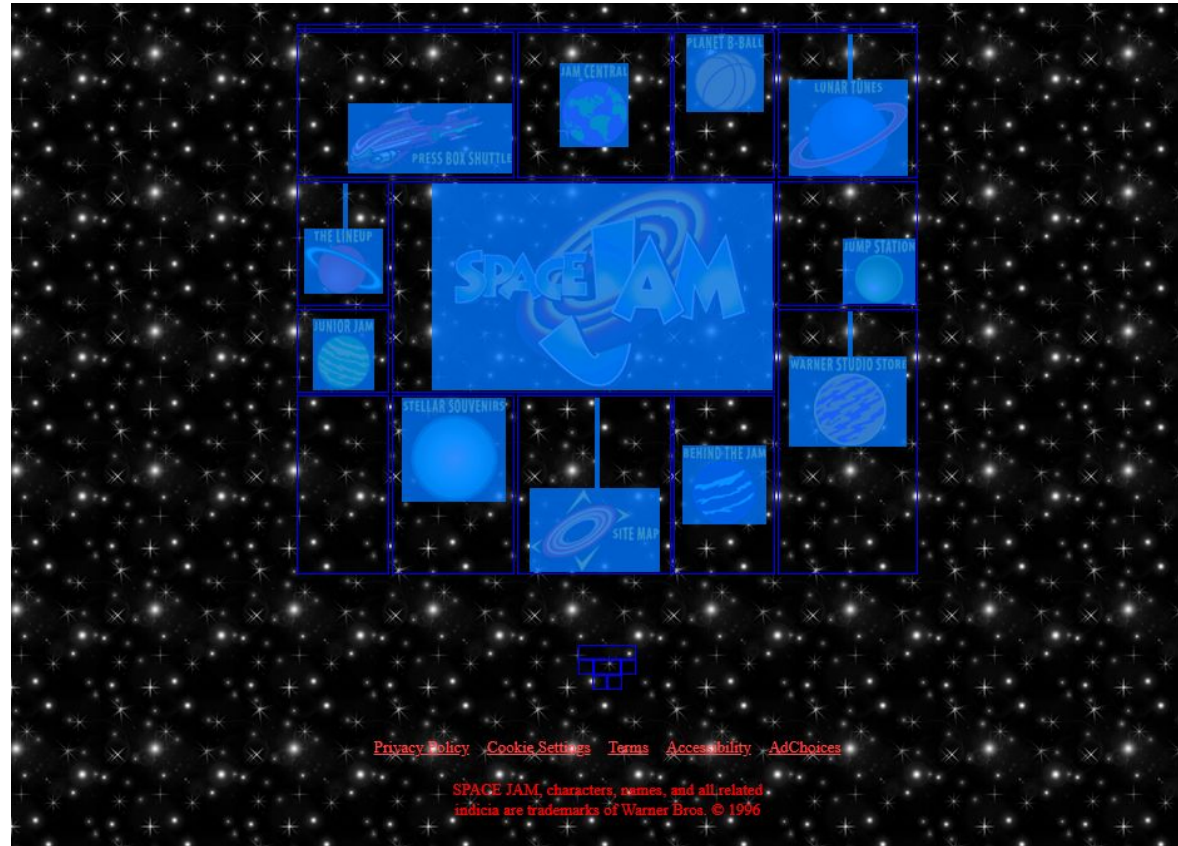
When Bugs does discover what he's up against, he knows he needs some serious help. But where can he go? Basketball's best players are all having their skills "absorbed" by the alien team.

Of course, there's a baseball player he happens to know...

Space Jam – colors in code

```
<body bgcolor="#000000" text="#ff0000" link="#ff4c4c" vlink="#ff4c4c" alink="#ff4c4c">
```

Space Jam – layout



Space Jam – layout

```
<table width=500 border=0>
  <tr>
    <td colspan=5 align=right valign=top>
    </td>
  </tr>
  <tr>
    <td colspan=2 align=right valign=middle>
      <br>
      <br>
      <br>
      <a href="cmp/pressbox/pressboxframes.html"></a>
    </td>
    ...
  </tr>
  ...
</table>
```

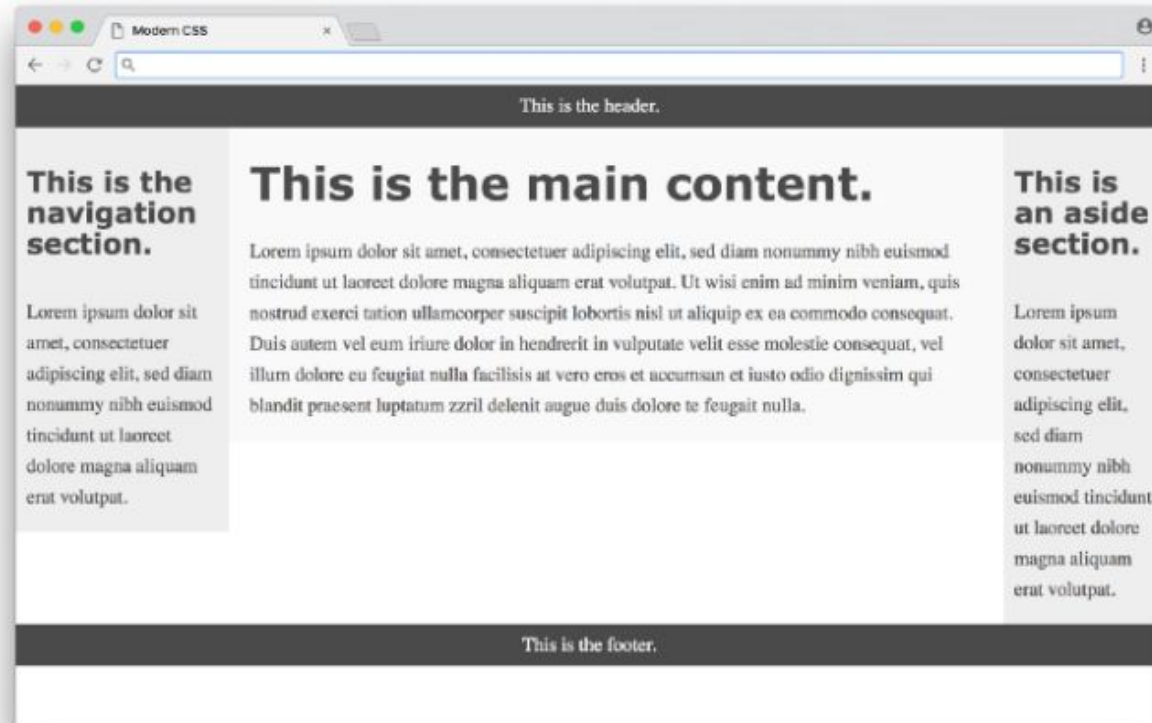
CSS came up with another layout solution

Float-based layout

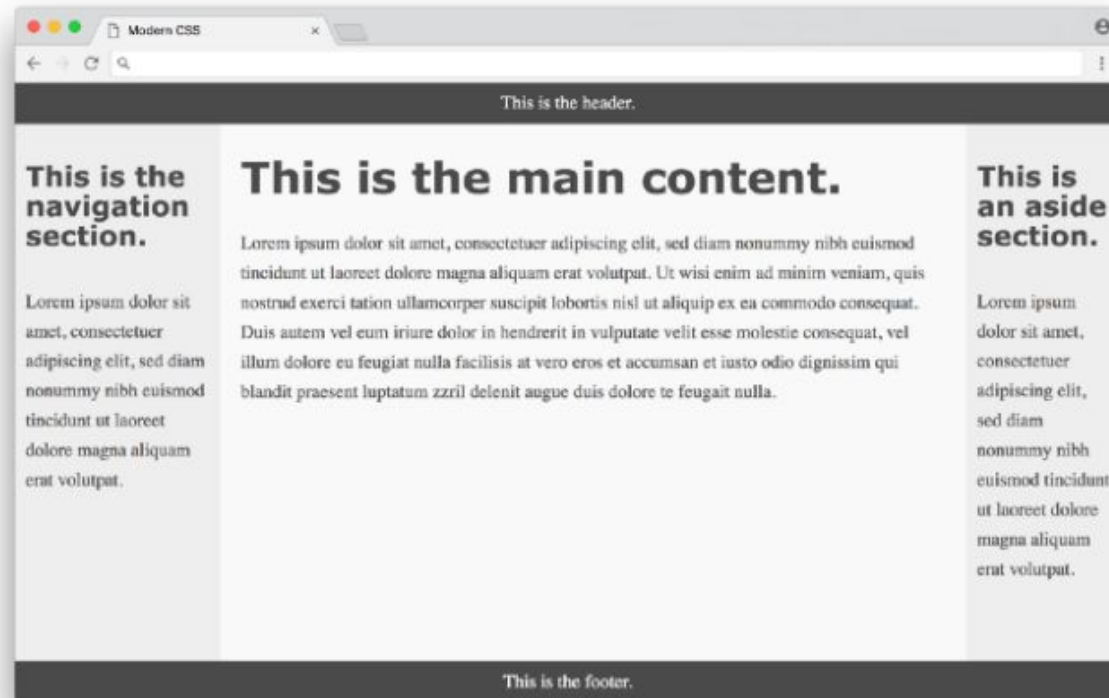
Flexbox-based layout

Grid layout

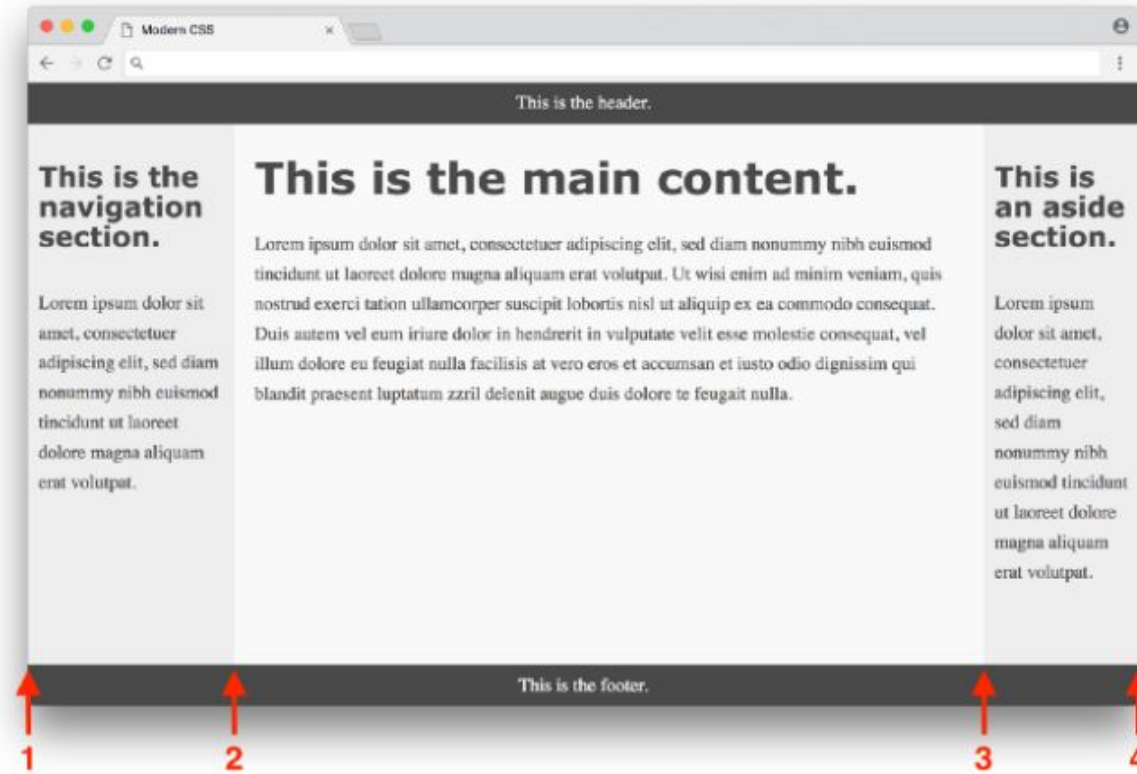
Float-base layout



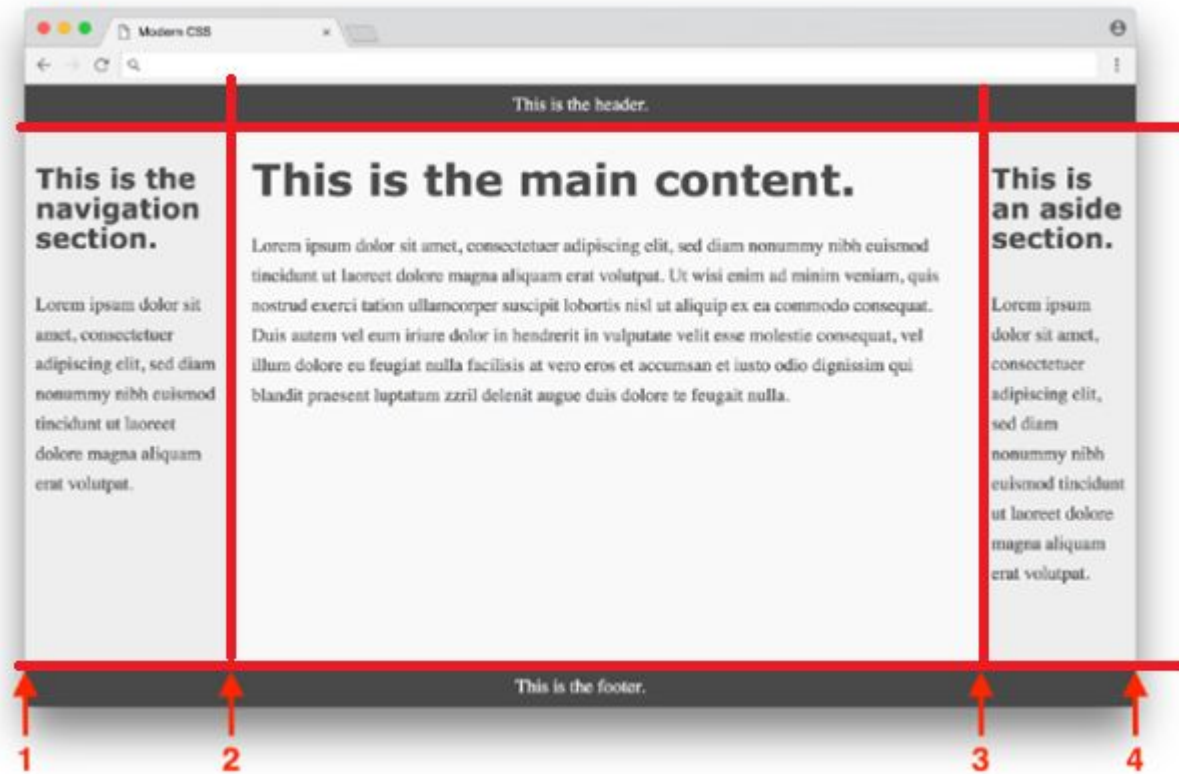
Flexbox base layout



Grid layout



Grid layout



CSS and related problems

CSS and related problems

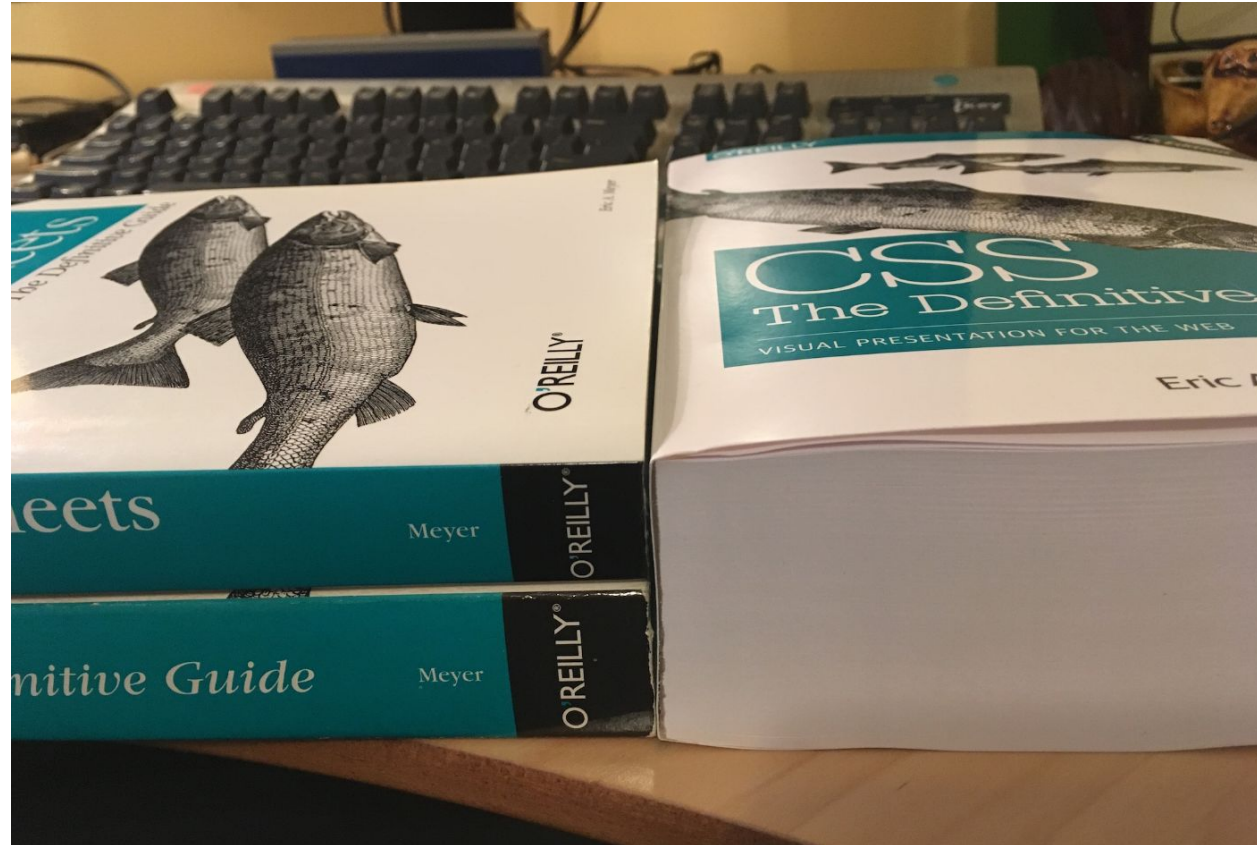


Revolution

Eric A. Meyer



CSS: The Definitive Guide



First CSS

```
<STYLE type="text/css">
BODY {
  font-family: serif;
  background-color: silver;
}
H1 {
  font: x-large Verdana, sans-serif;
  color: olive;
  border-bottom: thin black solid;
}
TABLE {
  margin: 0;
}
.sidebar {
  background-color: olive;
  padding: 0.5em;
}
.sidebar UL LI {
  list-style-type: none;
  margin-left: 0;
  margin-right: 0.5em;
}
.sidebar UL LI A {
  color: #ffcccc;
}
.body {
  background: white no-repeat center url(http://www.mysite.org/pix/logo.gif);
}
</STYLE>
```


One change will affect multiple places

```
p {  
  padding-bottom: 10px;  
  color: blue;  
}
```



Homepage paragraph



Category paragraph



Product detail paragraph

Problems with large projects

Complexity



Complications

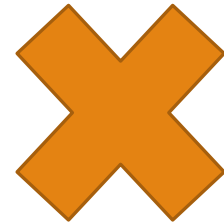
Two CSS properties walk into a bar.

A barstool in a completely different bar falls over.

...

Complications with sustainability

Straightforwardness
of language



Sustainability



The most common mistakes

Deep selectors and high specificity

Graphic



Implementation

```
<ul class="seznam">
  <li><a href="prvni.html">Odkaz 1</a></li>
  <li><a href="druhy.html">Odkaz 2</a></li>
  <li><a href="treti.html">Odkaz 3</a></li>
</ul>
```

```
|.seznam li a {  
  font-size: 20px;  
  color: blue;  
}
```

It doesn't work



```
body #content .page ul li a {  
  font-size: 16px;  
  color: red;  
  font-weight: bold;  
}
```

What are the modification options now?

Increase
specificity

Edit blocking
selector

!important

The only suitable option to solve the problem

```
body #content .page ul.seznam li a {  
    font-size: 16px;  
    color: red;  
    font-weight: bold;  
}
```

What have we achieved?

New selector that can no longer
be used

Nesting hell

preprocessors and
capability

```
.Checkbox--toggle {
  padding: $checkbox-toggle-diameter / 10 0;
  .Checkbox {
    &-input {
      &:checked {
        & + .Checkbox-label {
          @extend .Checkbox-toggle - active;
        }
      }
      &:not(:checked) {
        &:focus {
          & + .Checkbox-label {
            &::before {
              background-color: $checkbox-toggle-active-handle-bg;
            }
          }
        }
        & + .Checkbox-label {
          background-color: rgba($checkbox-toggle-bg, 0.46);
        }
        &[disabled], &[readonly] {
          & + .Checkbox-label {
            @extend .Checkbox-toggle - disabled;
          }
        }
      }
    }
  }
  &-label {
    @extend .Checkbox-toggle;
  }
}
```


What will the compiled file look like?

```
.Checkbox-toggle--active,  
.Checkbox--image .Checkbox-input:checked + .Checkbox-image .Checkbox-toggle,  
.Checkbox--image  
  .Checkbox-input:checked  
  + .Checkbox-image  
  .Checkbox--toggle  
  .Checkbox-Label,  
.Checkbox--toggle  
  .Checkbox--image  
  .Checkbox-input:checked  
  + .Checkbox-image  
  .Checkbox-Label,  
.Checkbox--toggle .Checkbox-input:checked + .Checkbox-Label {  
  background-color: #71c5e8;  
}  
.Checkbox-toggle--active::before,  
.Checkbox--image  
  .Checkbox-input:checked  
  + .Checkbox-image  
  .Checkbox-toggle::before,  
.Checkbox--image  
  .Checkbox-input:checked  
  + .Checkbox-image  
  .Checkbox--toggle  
  .Checkbox-Label::before,  
.Checkbox--toggle  
  .Checkbox--image  
  .Checkbox-input:checked  
  + .Checkbox-image  
  .Checkbox-Label::before,  
.Checkbox--toggle .Checkbox-input:checked + .Checkbox-Label::before {  
  left: calc(100% - 20px);  
  background-color: #0284ff;  
}  
  
.Checkbox-toggle--disabled,  
.Checkbox--image .Checkbox-input[disabled] + .Checkbox-image .Checkbox-toggle,  
.Checkbox--image  
  .Checkbox-input[disabled]  
  + .Checkbox-image  
  .Checkbox--toggle  
  .Checkbox-Label,  
.Checkbox--toggle  
  .Checkbox--image  
  .Checkbox-input[disabled]  
  + .Checkbox-image  
  .Checkbox-Label,  
.Checkbox--image .Checkbox-input[readOnly] + .Checkbox-image .Checkbox-toggle,  
.Checkbox--image  
  .Checkbox-input[readOnly]  
  + .Checkbox-image  
  .Checkbox--toggle  
  .Checkbox-Label,
```

How should we nest selectors?

```
.button {  
  padding: 10px;  
  
  @include breakpoint(tablet) {  
    padding: 8px;  
  }  
  
  &:hover {  
    background: blue;  
  }  
  
  &.is-active {  
    color: red;  
  }  
  
  &-icon {  
    max-width: 16px;  
  }  
  
  &-text {  
    font-size: 0.875rem;  
  }  
}
```

Cascade - order of writing rules and file structure

Order matters

Example:

```
.red {  
  color: ■red;  
}  
  
.blue {  
  color: ■blue;  
}
```

```
<div class="red blue">Hello</div>  
<div class="blue red">World</div>
```

```
.red {  
  color: ■ red;  
}  
  
.blue {  
  color: ■ blue;  
}
```

```
<div class="red blue">Hello</div>
```

```
<div class="blue red">World</div>
```

```
.red {  
  color: ■ red;  
}  
  
.blue {  
  color: ■ blue;  
}
```

```
<div class="red blue">Hello</div>
```

```
<div class="blue red">World</div>
```



CSS methodologies and architectures

In 2008, year of first experiments

OOCSS

SMACSS

SUITCSS

BEM

ITCSS

CUBE CSS

Making sense of methodologies and architectures

Sustainability

Extensibility

CSS problems

OOCSS

OOCSS defines an object as a visual pattern that is reusable across applications

Separation of
appearance from
structure

Content and container
separation



Separation of appearance from structure

```
.product-detail h2
```

```
.product-detail-heading {  
  ...  
}
```

Content and container separation

```
.hero .heading-secondary {}  
  
.page-detail h2 {}  
// v tomto případě porušujeme rovnou i první pravidlo
```

SMACSS

The main idea of SMACSS is to categorize the CSS rule system

Base

1. Base

Do této kategorie patří základní nastavení pravidel HTML elementů. Patří se normalizace a základní nastavení.

```
body, form {  
  margin: 0;  
  padding: 0;  
}  
  
a {  
  color: #039;  
}  
  
a: hover {  
  color: #03F;  
}
```

Layout

2. Layout

Do této vrstvy patří dimenzionální deklarace.

```
#header { ... }  
  
#primarynav { ... }  
  
#maincontent { ... }
```

Module

3. Module

Vrstva pro to, čemu říkáme znovupoužitelné komponenty. Zde by měli být všechny deklarace tlačítek, sidebarů, menu apod.

```
.module > h2 {  
  padding: 5px;  
}  
  
.module span {  
  padding: 5px;  
}
```

State

4. State

Sem patří deklarace námi předem definovaných pravidel rozšířených o stavy.

Jedná se o závislosti na JavaScriptu.

```
.is-tab-active {  
  background-color: white;  
  color: black;  
}
```

BEM

Typ třídy	Způsob pojmenování
Block	.Block
Element	.Block_element
Modifikátor	.Block--modifier
Modifikátor elementu	.Block_element--modifier

The biggest advantages

Creates selectors
of the lowest
specificity

Uniformity for
larger
development
teams

SOME ENGAGING TITLE

There are many variations of passages of Lorem Ipsum available.

The majority have suffered alteration in some form, by injected humour, or randomised words which don't look even slightly believable.

READ MORE

```
<article class="card">
```

```
<h1 class="card__title">Some engaging title</h1>
```

```
<p class="card__text"> There are many variations....</p>
```

```
<p class="card__text card__text--secondary"> The majority....</p>
```

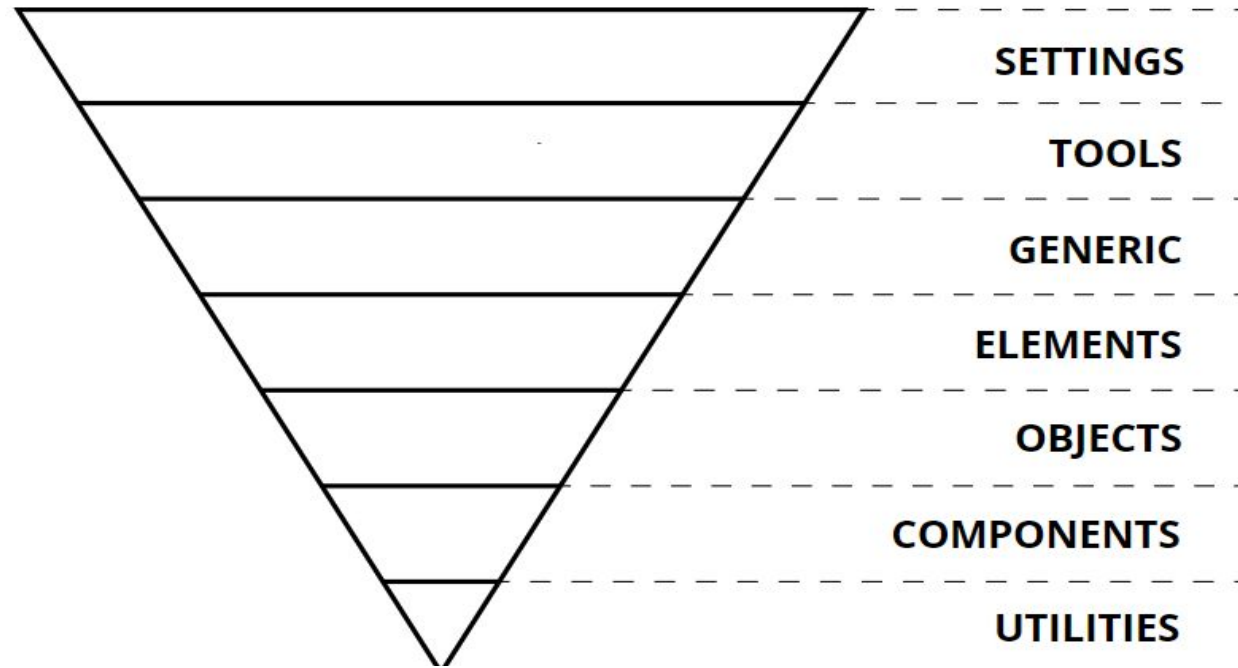
```
<a class="card__button button" href="#">Read more</a>
```

```
</article>
```

Cons?

The need to name all
elements

ITCSS



What ITCSS solve?

File
organization

Problems with
specificity

Settings

- settings.scss
- colors.scss
- typography.scss
- animation.scss

Tools

- functions.scss
- placeholders.scss
- mixins.scss
- media-queries.scss

Generic

- reset.scss
- normalize.scss
- bootstrap.scss
- fancybox.scss

Elements

- h1
- a
- article

Objects

- grid.scss
- content.scss
- .container
- .grid
- .row
- Spacers, Pushes/Pullers

Components

- Accordions
- Search-form
- Button
- Breadcrumbs
- Tooltip

Utilities

- helpers.scss
- utilities.scss
- overrides.scss

What about JavaScript and styling?

CSS Modules

Styled
components

Older than CSS



What is CSS-in-JS trying to solve?

Global scope

Write
everything in
JavaScript

CSS Modules

classic CSS



Don't think
about class
names

Cons?

Dependence
on JavaScript

Styled Components

classic CSS