

# **magic of** **CSS**

<http://bit.ly/cn-css>

# **CSS basics** **& HTML5**

# HTML

# HTML5

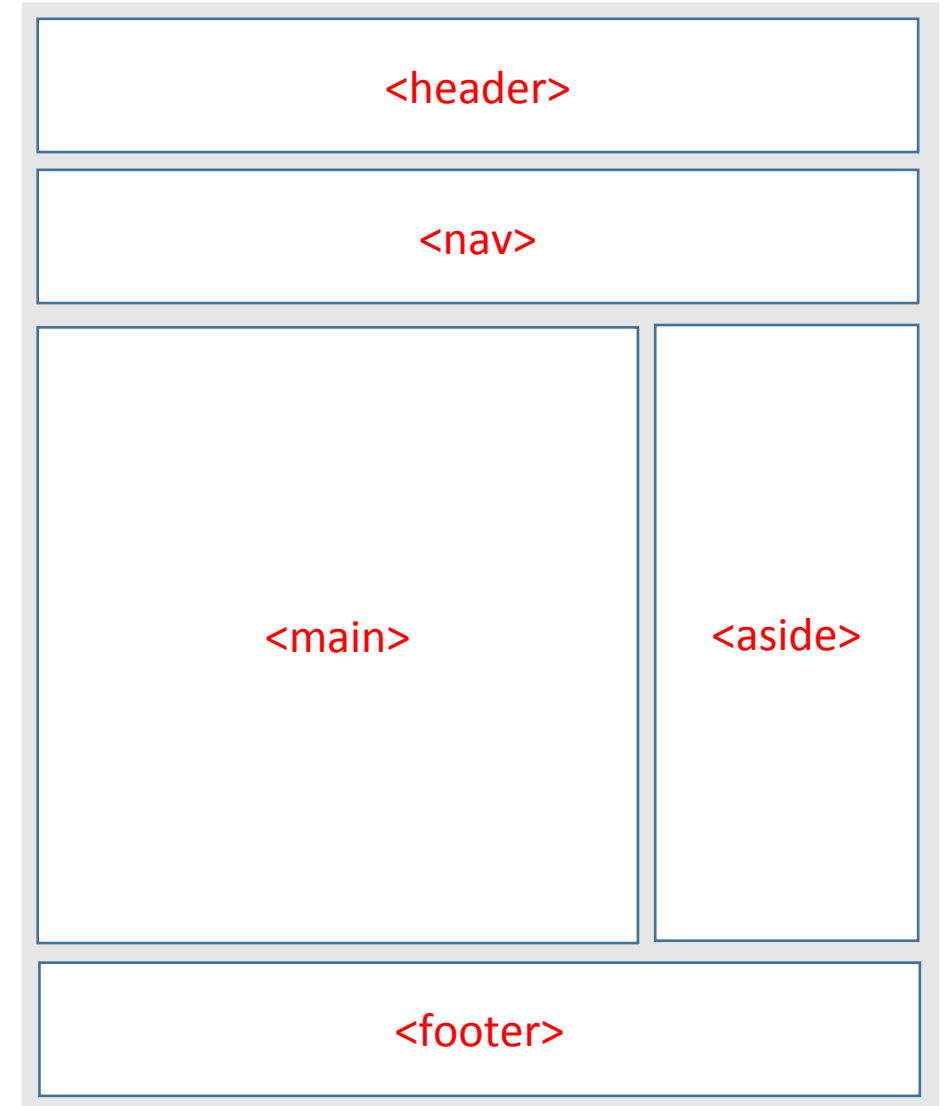
`<header>`  
introductory content, typically a group of introductory or navigational aids  
`</header>`

`<nav>`  
provide navigation links, either within the current document or to other documents  
`</nav>`

`<main>`  
content that is directly related to or expands upon the central topic of a document  
`</main>`

`<aside>`  
content is only indirectly related to the document's main content  
`</aside>`

`<footer>`  
footer for its nearest sectioning content or sectioning root element  
`</footer>`



# HTML HTML5

`<section>`

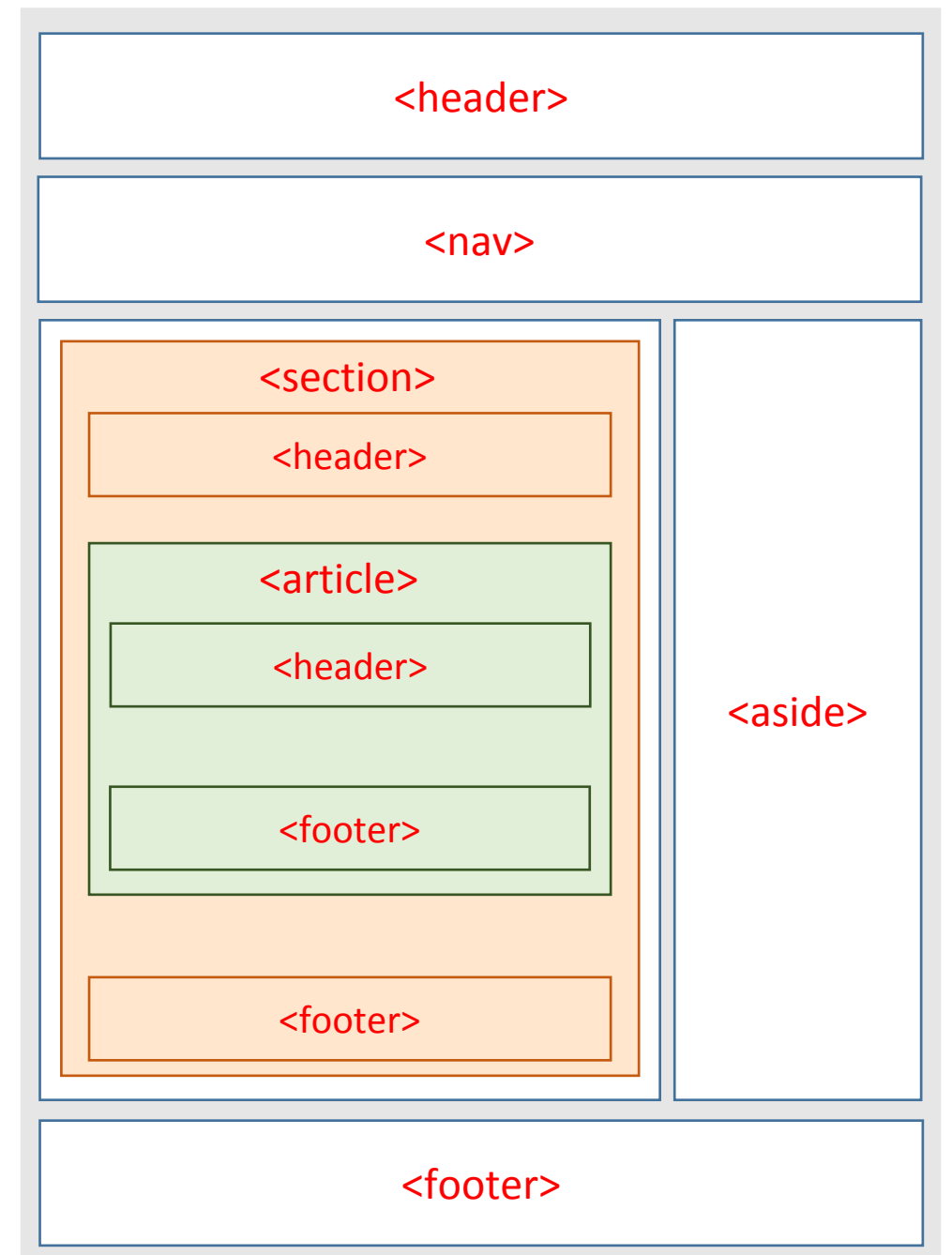
standalone section — which doesn't have a more specific semantic element to represent it

`</section>`

`<article>`

self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable

`</article>`



HTML

# HTML5 - Forms

```
<button><em>Search</em></button>
```

```
<input type="search">
```

```
<input type="email">
```

```
<input type="number">
```

```
<input type="number" min="24" max="45" step="0.5">
```

```
<input type="range" min="1" max="100" value>
```

```
<input type="date" min="2018-01-01" max="2019-01-01">
```

```
<input type="..." required>
```

```
<input placeholder="Search something...">
```

```
<label for="city">City:</label>
```

```
<input type="text" id="city" name="city-name">
```

CSS

# Cascading Style Sheets

What is the **cascade**?

CSS

# Cascade - specificity

```
p.text-red {  
  color: red;  
}
```

```
.text-blue {  
  color: blue;  
}
```

```
<p class="text-red text-blue">...</p>
```

<https://specificity.keegan.st/>

CSS

## Cascade - order

```
.text-red {  
  color: red;  
}
```

```
.text-blue {  
  color: blue;  
}
```

```
<p class="text-red text-blue">...</p>
```



CSS

# Cascade – importance

```
.text-red {  
  color: red !important;  
}
```

```
.text-blue {  
  color: blue;  
}
```

```
<p class="text-red text-blue">...</p>
```

CSS

## Inline styles

```
.text-red {  
  color: red !important;  
}
```

```
.text-blue {  
  color: blue;  
}
```

```
<p class="text-red text-blue" style="color: green;">...</p>
```

CSS

# Display

**display: inline;**

*elements are in one line, you can't use padding, margin, width  
mostly text elements `<a>`, `<strong>`, `<span>`*

**display: block;**

*you can use width, margin, ..., elements: `<div>`, `<p>`, `<ul>`, `<li>`*

**display: inline-block;**

*elements are in one line, you can use width, margin, ..., for example `<img>`*

**display: none;**

*hidden element (visible in DOMu, no affect)*

**display: flex;**

*block element with flexible content*

# CSS Visibility

## 1) Display

```
.d-none {  
    display: none;  
}
```

## 2) Visibility

```
.invisible {  
    visibility: hidden;  
}
```

## 3) Screen readers only

```
.sr-only {  
    position: absolute;  
    width: 1px;  
    height: 1px;  
    padding: 0;  
    overflow: hidden;  
    clip: rect(0, 0, 0, 0);  
    white-space: nowrap;  
    border: 0;  
}
```

CSS

# Position

position: static;

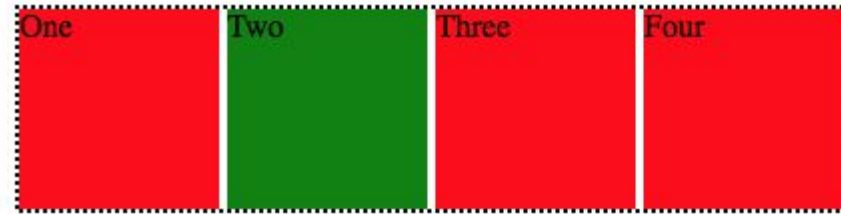
position: relative;

position: absolute;

position: fixed;

position: sticky;

position: static



position: relative



position: absolute



# CSS Spacing

padding: 25px;

padding-left: 25px;

padding: 25px 20px 5px 15px;

padding: 25px 15px;

padding: 25px 15px 5px;

margin: 10%;

border: 5px solid #177095;

border-width: 5px;

border-style: dashed;

border-color: #177095;

CSS

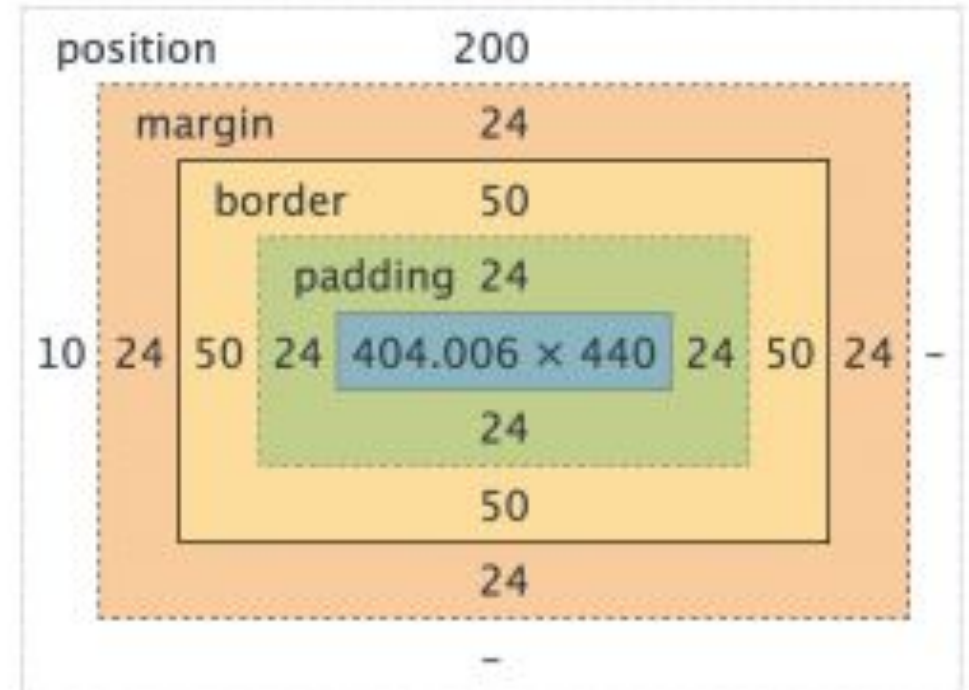
# Box model

`box-sizing: border-box;`

Element dimension = width or height of content + padding + border.

`box-sizing: content-box; /* default */`

Element dimension = width or height of content.



# CSS Colors

color: #F3E6D2;

color: white;

color: rgb(123, 255, 13);

color: rgba(0, 0, 0, 0.3);

color: hsla(245, 0, 0, 0.3); // [hslpicker.com/#73e92f](https://hslpicker.com/#73e92f)

opacity: 0.5;



CSS

# Background

`background-color: blue;`

`background-image: url(..../img/photo.jpg);`

`background-position: center top;`

`background-repeat: no-repeat;`

`background-size: cover; /* contain, 24px, ... */`

`background-attachment: fixed;`

shorthand syntax

`background: #F5F5F5 url(..../img/photo.jpg) no-repeat left top;`

# CSS Shadow

```
box-shadow: 0 2px 4px 0 rgba(0, 0, 0, 0.5);
```

```
box-shadow: inset 2px 4px 0 #000; /* inner shadow */
```

```
box-shadow: 0 2px 4px 0 rgba(0, 0, 0, 0.5) , inset 2px 4px 0 #000; /* more shadows */
```

```
text-shadow: 0 2px 4px rgba(0, 0, 0, 0.5);
```

<https://codepen.io/haibnu/pen/FxGsl>

**Link:**

<https://developer.mozilla.org/en-US/docs/Web/CSS/box-shadow>

CSS

# Transformations

```
transform: rotate(90deg);
```

```
transform: translate(10rem, 25rem);
```

```
transform: translateY(-50%);
```

```
transform: scale(0.5);
```

```
transform: rotate(90deg) scale(2) ;
```

<https://codepen.io/svobodalukas/pen/zmdPNX>

**Link:**

<https://developer.mozilla.org/en-US/docs/Web/CSS/transform>

CSS

# Animations – transitions

```
transition: opacity .5s ease-in-out;
```

```
.btn {  
  color: #222;  
  background: #e5e5e5;  
  transition: all .5s ease-in-out;  
}
```

```
.btn:hover {  
  color: #fff;  
  background: red;  
}
```

# CSS

## Animations

```
.box {  
  animation: my-animation 6s ease infinite alternate;  
}
```

```
@keyframes my-animation {  
  0% {  
    background: blue;  
  }  
  50% {  
    background: yellow;  
  }  
  100% {  
    background: red;  
  }  
}
```

```
animation-name: my-animation;  
animation-duration: 6s;  
animation-timing-function: ease;  
animation-delay: 0;  
animation-iteration-count: infinite;  
animation-direction: alternate;  
  
animation-play-state: paused;
```

<https://codepen.io/svobodalukas/pen/KxEmZX>

# CSS

## Z-Index

z-index: 100;

- The z-index CSS property sets the z-order of a **positioned** element and its **descendants**

Link:

<https://developer.mozilla.org/en-US/docs/Web/CSS/z-index>

CSS

# Media content – images

- Select right image
  - jpg – photos
  - png, svg – line drawings, text, iconic graphics
  - gif – animations
  - webP – best size, animations, alpha, only for Chrome and Edge
- Optimisation tools (for example <https://kraken.io/web-interface>)

CSS

# Media content – SVG

- vector
- can be styled by CSS
- optimisation tool: svgo (<https://github.com/svg/svgo>)
- needs fallback for IE8 or Android Browser 2.3

```
<svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24">  
  <path d="M19 6.41L17.59 5 12 10.59 6.41 5 5 6.41 10.59 12 5 17.59 6.41 19 12 13.41  
17.59 19 19 17.59 13.41 12z"/>  
  <path d="M0 0h24v24H0z" fill="none"/>  
</svg>
```



# CSS SVG

## 1) img tag (no way to style it)

```

```

## 2) directly in a code (styles by CSS)

```
<svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" viewBox="0 0 24 24">  
  <path d="M19 6.41L17.59 5 12 10.59 6.41 5 5 6.41 10.59 12 5 17.59 6.41 19 12 13.41 17.59 19 19 17.59  
13.41 12z"/>  
</svg>
```

# CSS SVG

## 3) use tag (like "image sprite", styles by CSS)

```
<svg style="display: none;">
  <symbol id="ic_search" viewBox="0 0 24 24">
    <path d="M19 6.41L17.59 5 12 10.59 6.41 5 5 6.41 10.59 12 5 17.59 6.41 19 12 13.41 17.59 19 19 17.59 13.41
12z"/>
  </symbol>
  <symbol id="ic_edit" viewBox="0 0 24 24">
    <path d="M12 21.35l-1.45-1.32C5.4 15.36 2 12.28 2 8.5 2 5.42 4.42 3 7.5 3c1.74 0 3.41.81 4.5 2.09C13.09 3.81
14.76 3 16.5 3 19.58 3 22 5.42 22 8.5c0 3.78-3.4 6.86-8.55 11.54L12 21.35z"/>
  </symbol>
</svg>
```

....

```
<button>
  <svg width="16" height="16">
    <use xlink:href="#ic_arrow" />
  </svg>
</button>
```

# CSS

## Webfonts

### 1) link

```
<link href="https://fonts.googleapis.com/css?family=Lora:400,400i,700,700i" rel="stylesheet">
```

### 2) @import

```
<style>  
  @import url('https://fonts.googleapis.com/css?family=Lora:400,400i,700,700i');  
</style>
```

### 3) @font-face

```
@font-face {  
  font-family: 'Raleway';  
  src: url('../fonts/lora.woff2') format('woff2'), url('../fonts/lora.woff') format('woff');  
  font-weight: 700;  
  font-style: normal;  
}
```

# CSS

## Flexbox

- **flex layout gives the container the ability to alter its items' width/height (and order) to best fill the available space**

```
<div style="display: flex;">  
  <div> ... </div>  
  <div> ... </div>  
  <div> ... </div>  
</div>
```

CSS

# Flexbox – flex container

display: flex;

flex-direction: row;

*(row, row-reverse, column, column-reverse)*

flex-wrap: wrap;

*(nowrap)*

justify-content: space-between;

align-items: stretch;

align-content: center;

CSS

# Flexbox – items properties

order: 2;

flex-grow: 1;

flex-shrink: 3;

flex-basis: 100px;

*default auto (size by content)*

flex: 1 1 50%;

*shorthand flex-grow flex-shrink flex-basis*

align-self: center;

# CSS **Flexbox**

<https://codepen.io/svobodalukas/pen/EdjxMW>

# CSS

# Flexbox

- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- <https://flexboxfroggy.com/#cs>
- <http://www.flexboxdefense.com/>



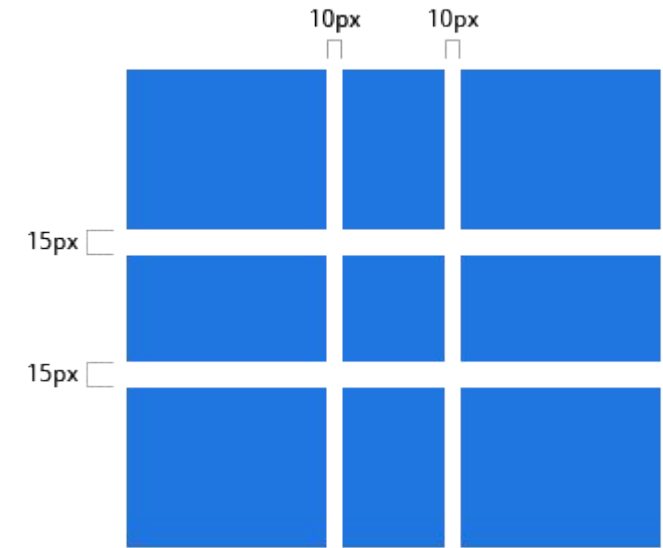
# CSS Grid

- set of properties for creating two dimensions layout
- some properties are the same as for flexbox

```
<div class="our-grid">  
  <div> ... </div>  
  <div> ... </div>  
  <div> ... </div>  
  <div> ... </div>  
</div>
```

<https://codepen.io/svobodalukas/pen/XPGKYJ>

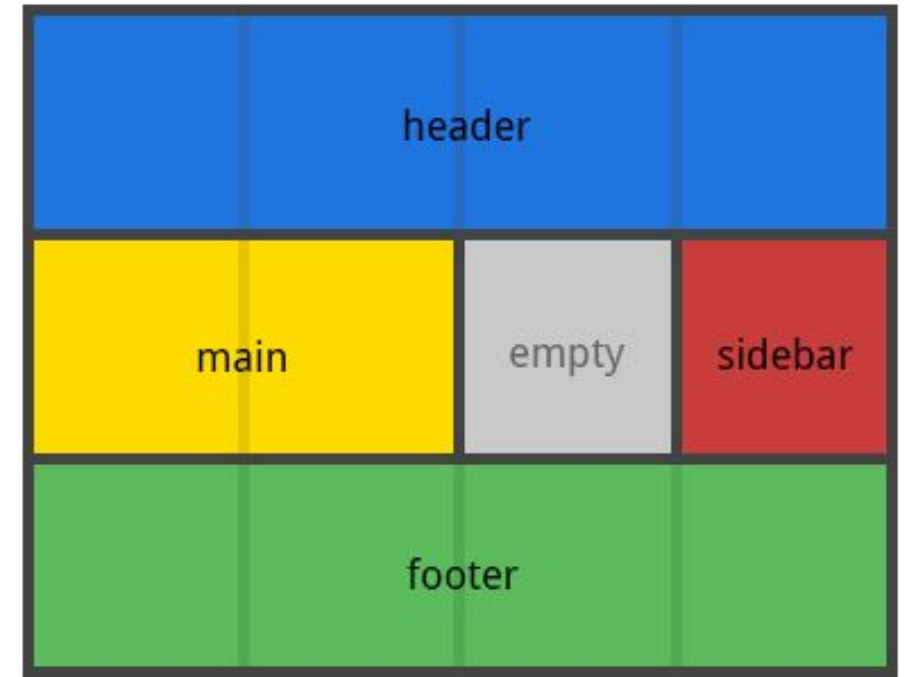
<https://css-tricks.com/snippets/css/complete-guide-grid/>



```
.our-grid {  
  display: grid;  
  grid-template-columns: 10rem 3fr;  
  grid-template-rows: 5rem 1fr 5rem;  
  grid-row-gap:  
  
}
```

# CSS Grid

```
.container {  
  grid-template-columns: 50px 50px 50px 50px;  
  grid-template-rows: auto;  
  grid-template-areas:  
    "header header header header"  
    "main main . sidebar"  
    "footer footer footer footer";  
}  
.item-a {  
  grid-area: header;  
}  
.item-b {  
  grid-area: main;  
}  
.item-c {  
  grid-area: sidebar;  
}  
.item-d {  
  grid-area: footer;  
}
```



```
<div class="container">  
  <div class="item-a"> ... </div>  
  <div class="item-b"> ... </div>  
  <div class="item-d"> ... </div>  
  <div class="item-c"> ... </div>  
</div>
```

CSS

# supports

```
@supports (display: grid) {  
  .wrapper {  
    display: grid;  
    grid-gap: 10px;  
    grid-auto-flow: dense;  
    ...  
  }  
}
```

<https://developer.mozilla.org/en-US/docs/Web/CSS/@supports>

CSS

# Grid x Flexbox: let's fight!

- Grid: two dimensions, Flex: one dimension
- Grid page layout, Flex: components layout
- Grid has bad support in IE and older Safari

CSS

# New units

## rem

- 1rem equals the font size of the html (root) element (mostly 1rem = 16px)

## vw, vh

- 1vw = relative to 1% of the width of the viewport (viewport width)
- 1vh = relative to 1% of the height of the viewport (viewport height)

Links:

[https://www.w3schools.com/cssref/css\\_units.asp](https://www.w3schools.com/cssref/css_units.asp)

# CSS Selectors

<code>ul li { }</code>	<code>ul &gt; li { }</code>
------------------------	-----------------------------

```
<ul>  
  <li>List item one</li>  
  <li>List item two  
    <ol>  
      <li>Nested item one</li>  
      <li>Nested item two</li>  
    </ol></li>  
  <li>List item three</li>  
</ul>
```

<code>p + p { }</code>	<code>div + p { }</code>
------------------------	--------------------------

```
<div>  
  <p>Line One</p>  
  <p>Line Two</p>  
  <div>Box</div>  
  <p>Line Three</p>  
</div>
```

<code>p ~ p { }</code>	<code>div ~ p { }</code>
------------------------	--------------------------

```
<div>  
  <p>Line One</p>  
  <p>Line Two</p>  
  <div>Box</div>  
  <p>Line Three</p>  
</div>
```

CSS

# Pseudoclasses & pseudoselectors

```
a:hover {}
```

```
input:focus {}
```

```
button:disabled {}
```

```
::placeholder {}
```

```
li:first-child {}
```

```
li:last-child {}
```

```
tr:nth-child(even) {}
```

```
li:nth-of-type(3n + 2) {}
```

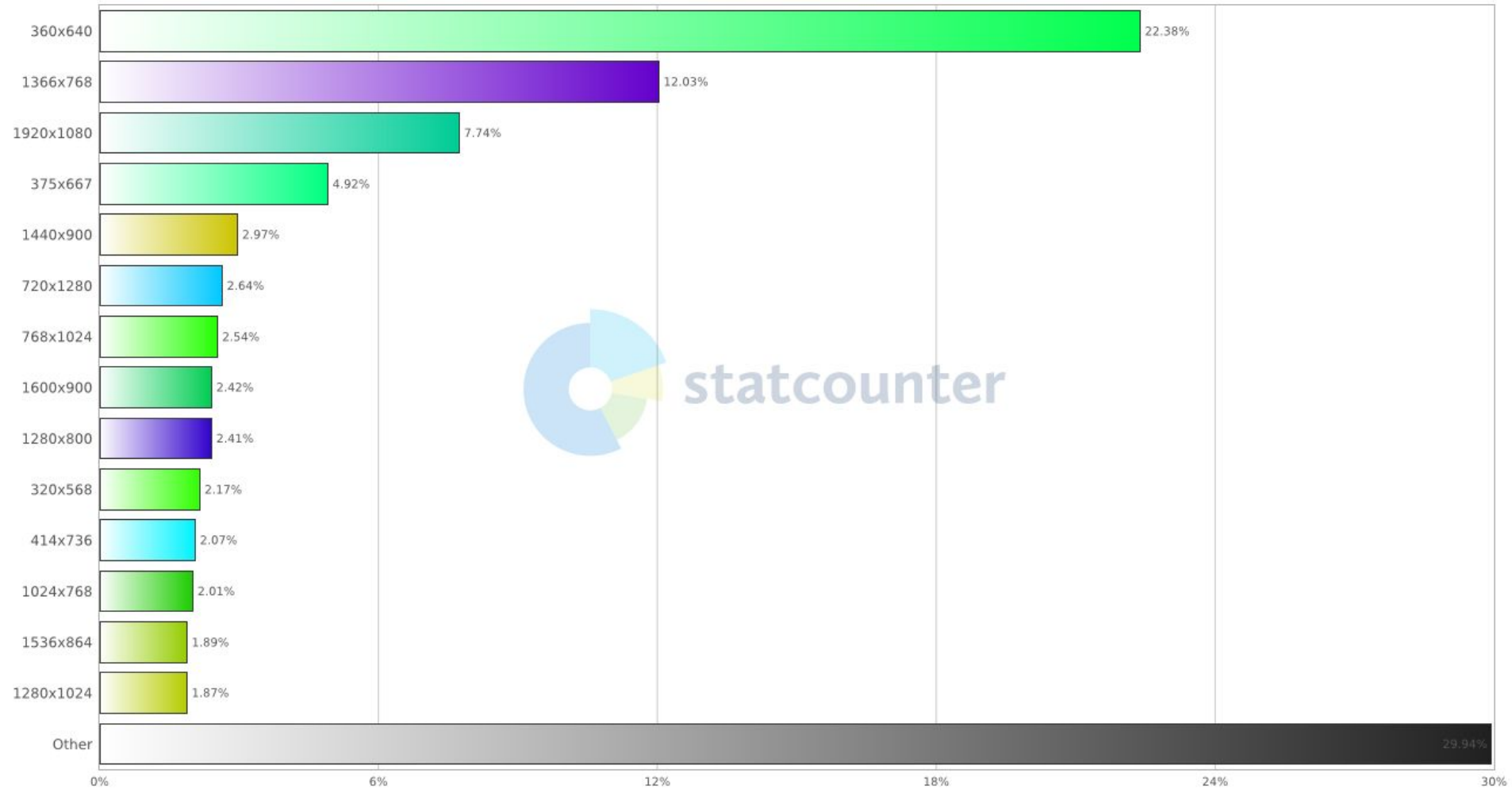
<https://codepen.io/svobodalukas/pen/EeMRKL>

**Responsivity**



# Responsivity Why?

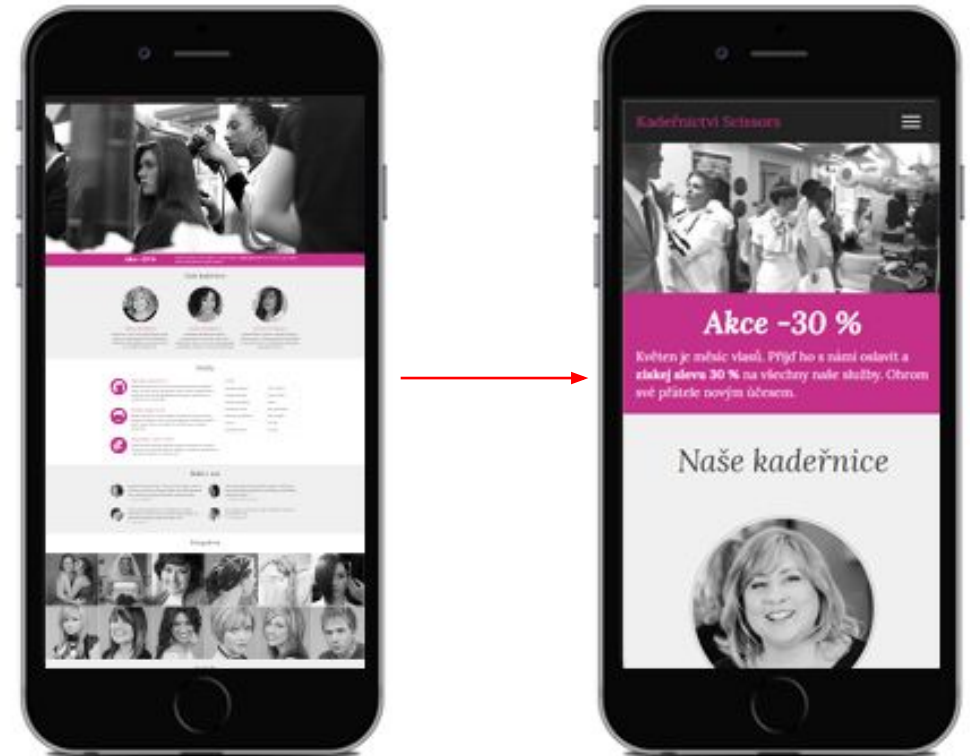
StatCounter Global Stats  
Screen Resolution Stats Worldwide from Aug 2017 - Aug 2018



# Responsivity **Viewport**

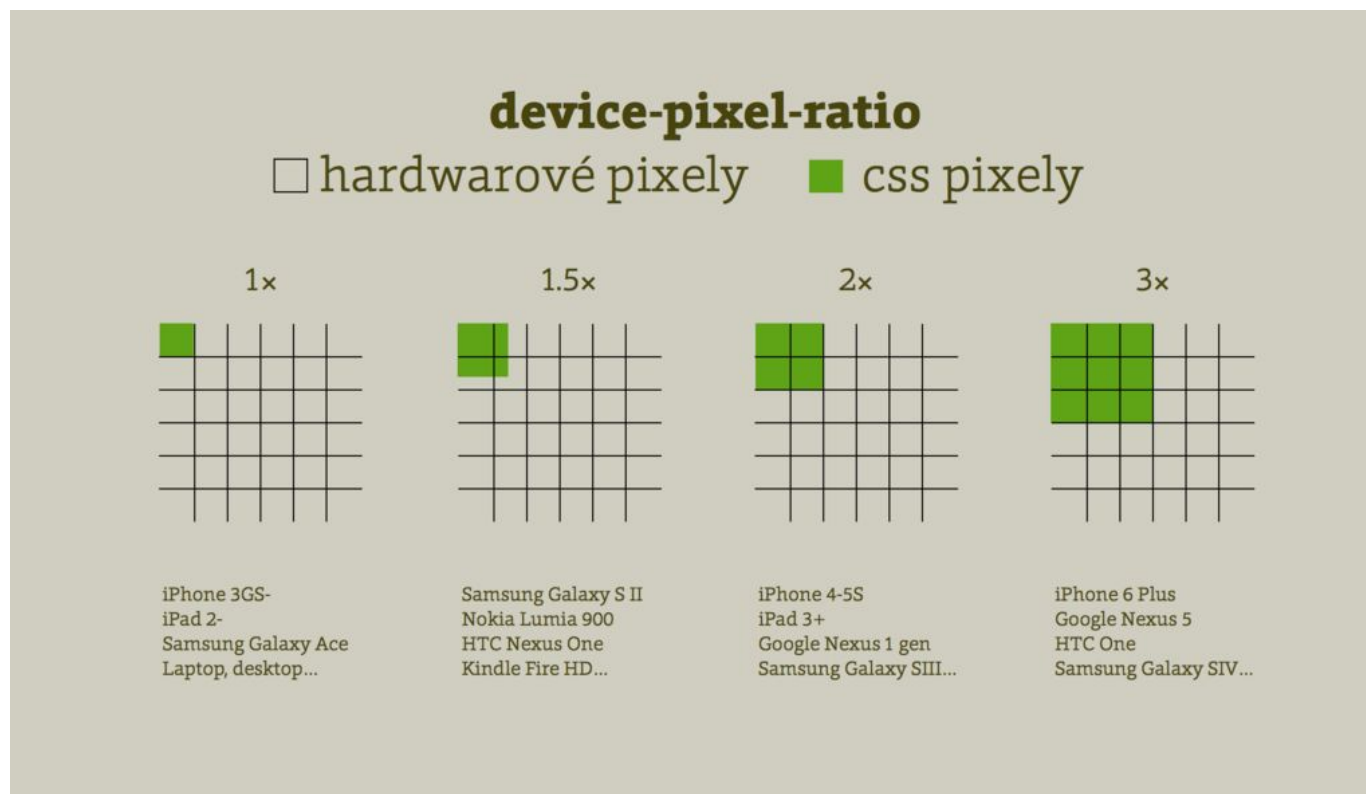
```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

- Unification of layout, visual and ideal viewport



# Media Queries

## Pixel



source: Martin Michálek, <https://www.vzhurudolu.cz/prirucka/css-pixel>

# Media Queries

## Pixel

	HW Resolution	Layout viewport	Ideal Viewport	DPR
iPhone 4	640 x 960	980 × 1091	320 x 480	2
iPad	1536 x 2048	980 × 1225	768 x 1024	2
Galaxy S5	1080 x 1920	980 × 1532	360 x 640	3
Nexus 6	1440 x 2560	980 × 1402	360 x 592	4

# Responsivity

## Flexible content

```
<div>
  <div class="blok">
    <h2>První sekce</h2>
    <p>Tady je text o mně</p>
  </div>
  <div class="blok">
    <h2>Druhá sekce</h2>
    <p>Tady bude taky text o mně</p>
  </div>
</div>
```

```
.blok {
  width: 320px;
  height: 150px;
  float: left;
  background: blue;
  color: white;
}

.blok {
  width: 50%;
  float: left;
  background: blue;
  color: white;
}
```



# Media Queries

## Queries...

@media only screen **and** (min-width: 800px) **and** (max-width: 1200px) { ...

@media only screen **and** (**min-height**: 800px) { ...

@media only **print** { ...

@media only screen **and** (**orientation**: landscape) { ...

@media only screen **and** (**min-aspect-ratio**: 16/9) { ...

Links:

- [https://developer.mozilla.org/en-US/docs/Web/CSS/Media\\_Queries/Using\\_media\\_queries](https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries)

# Media Queries

## Mobile First!

### mobile first

```
h1 {  
  font-size: 18px;  
  color: black;  
}  
  
@media only screen and (min-width: 768px) {  
  h1 {  
    font-size: 20px;  
  }  
}  
  
@media only screen and (min-width: 1200px)  
{  
  h1 {  
    font-size: 26px;  
  }  
}
```

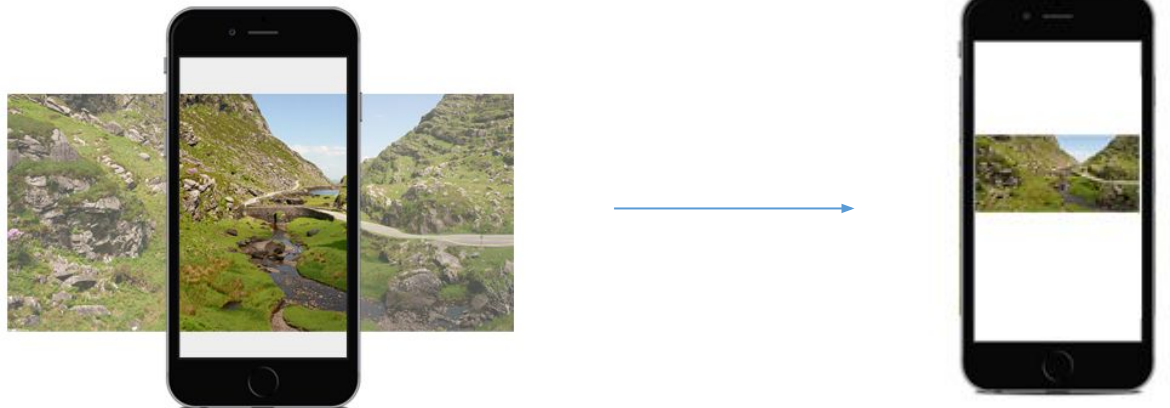
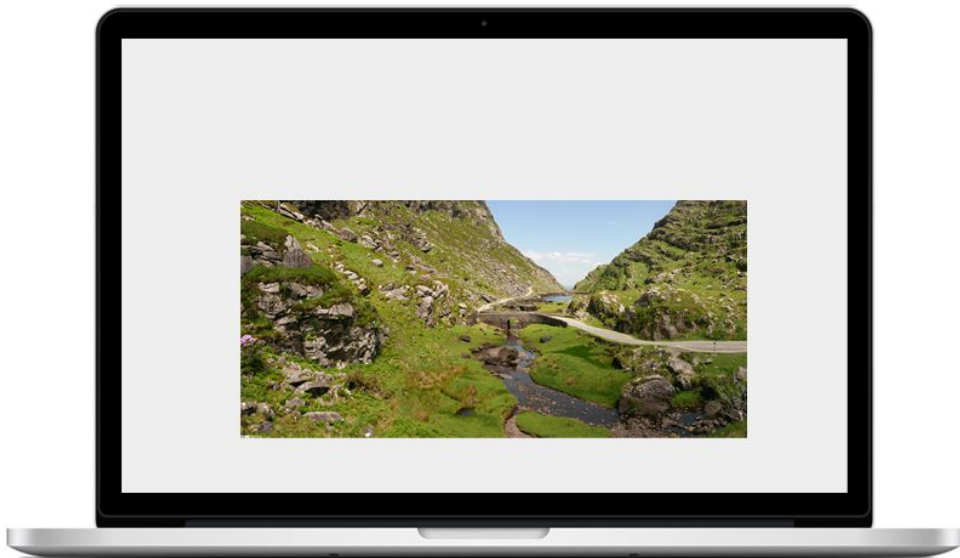
### desktop first

```
h1 {  
  font-size: 26px;  
  color: black;  
}  
  
@media only screen and (max-width: 1199px) {  
  h1 {  
    font-size: 20px;  
  }  
}  
  
@media only screen and (max-width: 767px) {  
  h1 {  
    font-size: 18px;  
  }  
}
```

# Responsivity

## Flexible images

```
.img-fluid {  
  max-width: 100%;  
  height: auto;  
}
```





# Responsivity

## Responsive images

```

```

- <https://www.vzhurudolu.cz/prirucka/srcset-sizes>
- <https://responsiveimages.org/>

# Responsivity

## Responsive images

```
<picture>
```

```
  <source srcset="images/teaser-992_560.jpg" media="(max-width: 992px)">
```

```
  <source srcset="images/teaser-1857_333.jpg" media="(min-aspect-ratio: 16/9)">
```

```
  
```

```
</picture>
```

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/picture>

# Responsivity

# **Responsive images**

<https://codepen.io/svobodalukas/pen/bmNPoO>

# CSS

## Responsive typography

```
h1 {  
  font-size: 22px;  
}  
@media (min-width:576px) {  
  h1 {  
    font-size: 22px;  
  }  
}  
@media (min-width:768px) {  
  h1 {  
    font-size: 24px;  
  }  
}  
@media (min-width:992px) {  
  h1 {  
    font-size: 34px;  
  }  
}
```



```
h1 {  
  font-size: 22px;  
}  
@media (min-width: 576px) {  
  h1 {  
    font-size: calc(1.04166667vw + 16px);  
  }  
}  
@media (min-width: 768px) {  
  h1 {  
    font-size: calc(4.46428571vw - 10.28571429px);  
  }  
}  
@media (min-width: 992px) {  
  h1 {  
    font-size: 34px;  
  }  
}
```

CSS

# Responsive iframe

```
.video-wrap {  
  position: relative;  
  padding-bottom: 56.25%; /* 16:9 */  
  height: 0;  
}
```

```
.video-wrap iframe {  
  position: absolute;  
  top: 0;  
  left: 0;  
  width: 100%;  
  height: 100%;  
}
```

<https://codepen.io/svobodalukas/pen/jvJazQ>

# CSS

## Responsive tables

First Name	Last Name	Points	Points	Points	Points	Points	Points
Jill	Smith	50	50	50	50	50	50
Eve	Jackson	94	94	94	94	94	94
Adam	Johnson	67	67	67	67	67	67

Options:

1. unnecessary columns are hidden on smaller screen
2. table scrolls inside container
3. columns into rows

Statement Summary

ACCOUNT	DUE DATE	AMOUNT	PERIOD
Visa - 3412	04/01/2016	\$1,190	03/01/2016 - 03/31/2016
Visa - 6076	03/01/2016	\$2,443	02/01/2016 - 02/29/2016

Statement Summary

ACCOUNT	Visa - 3412
DUE DATE	04/01/2016
AMOUNT	\$1,190
PERIOD	03/01/2016 - 03/31/2016
ACCOUNT	Visa - 6076
DUE DATE	03/01/2016
AMOUNT	\$2,443

# **CSS Projects** **& when code smells**

CSS

# Editor configuration

- **Editor Config**
  - <http://EditorConfig.org>
- **Prettier**
  - <https://prettier.io/>
- **Stylelint**
  - <https://stylelint.io/>

<https://css-tricks.com/prettier-stylelint-writing-clean-css-keeping-clean-code-two-tool-game/>



# CSS **Stylelint**

## **CSS warnings**

- high specificity
- BEM errors
- list of units
- colors only by hex code
- ...

# CSS Stylelint

```
"extends": "stylelint-config-standard",  
  
"rules": {  
  
  "at-rule-empty-line-before": null,  
  
  "at-rule-name-space-after": null,  
  
  "at-rule-no-unknown": null,  
  
  "color-hex-case": "lower",  
  
  "color-named": "never",  
  
  ...
```

Config:

<https://maximgatilin.github.io/stylelint-config/>

# CSS

# Atomic webdesign

The diagram illustrates the Atomic Design process through five vertical panels, each representing a different level of design abstraction:

- ATOMS:** Shows individual UI components such as navigation icons (home, search, camera, heart, profile), a photo caption, and a placeholder image labeled "1080 x 1080".
- MOLECULES:** Shows a collection of atoms assembled into a functional unit, such as a complete Instagram post from "thisistheusersinstagramhandle".
- ORGANISMS:** Shows multiple molecules combined into a larger structure, representing a grid of posts on a social media feed.
- TEMPLATES:** Shows a dashed outline representing a layout structure or grid that defines the overall page architecture.
- PAGES:** Shows a final, concrete page layout where all elements are rendered with real content, including a photo of a bulldog and a specific user profile.

CSS

# CSS Methodologies

- **SMACSS, ITCSS**

- <https://smacss.com/>

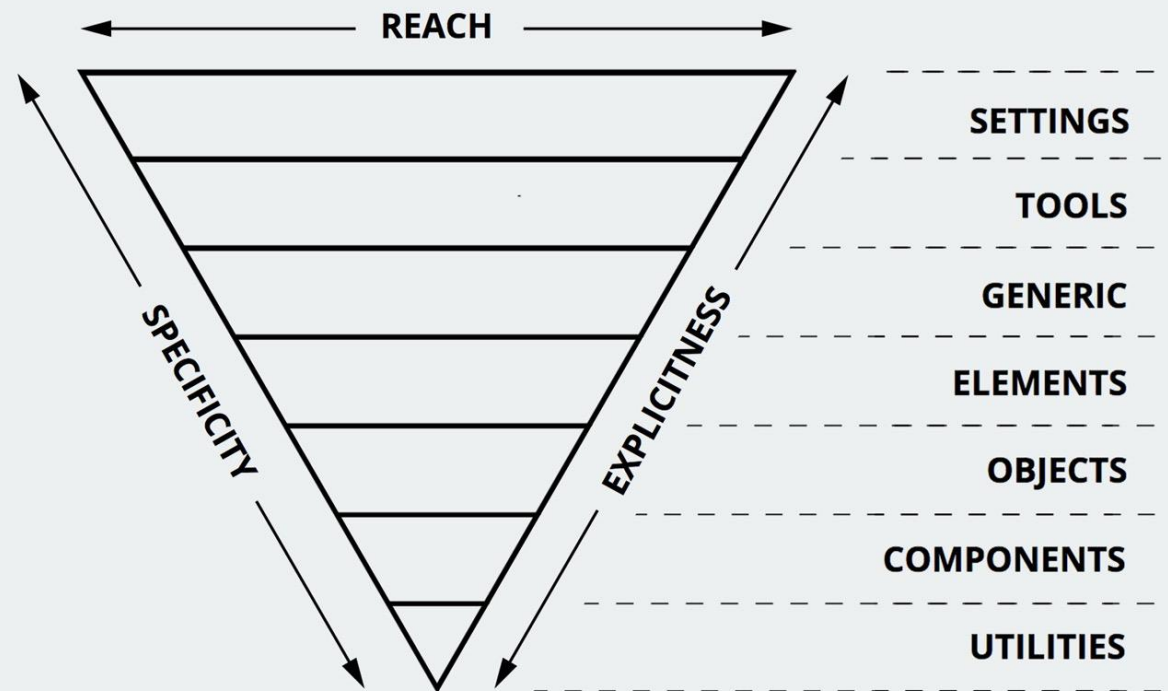
- **OOCSS**

- <http://EditorConfig.org>

- **BEM**

- <http://getbem.com/>

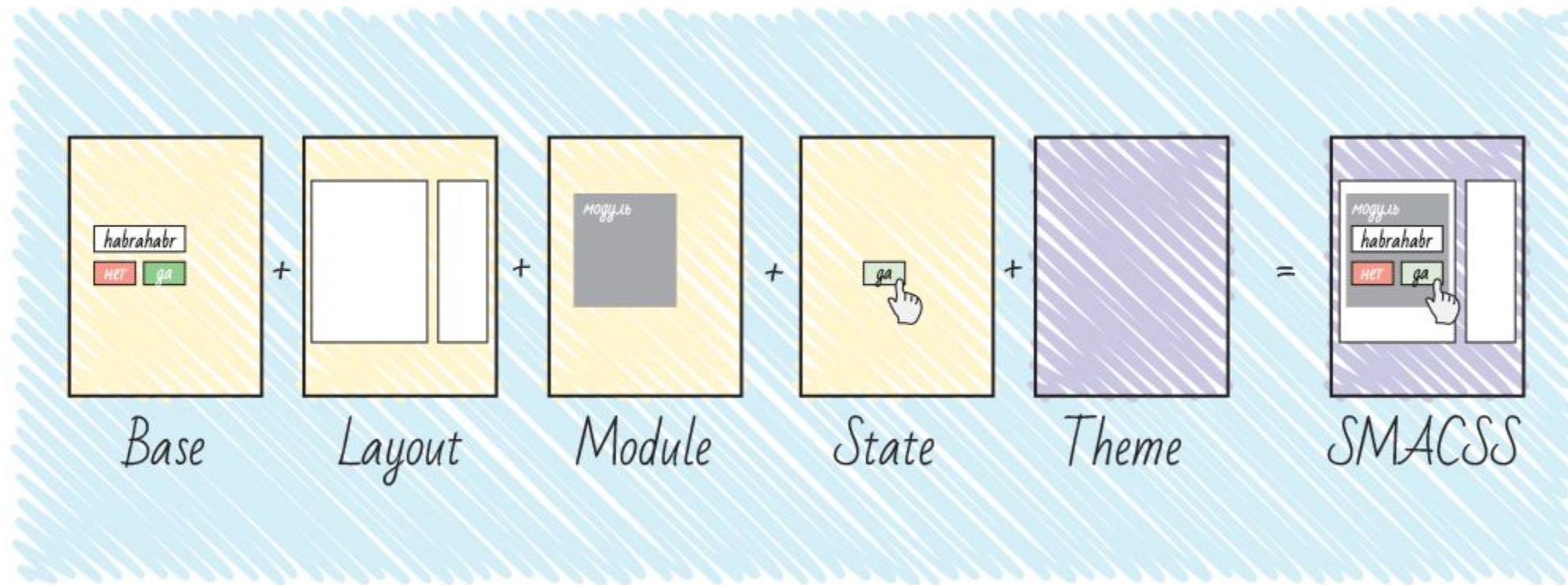
## Define an Architecture (ITCSS)



ITCSS	what kind of styles	css classes
<b>Settings</b>	<i>variables</i>	\$bg-color, \$font-base, ...
<b>Tools</b>	<i>function, mixins</i>	@function calc-spacing ...
<b>Generic</b>	<i>reset (Reboot, Normalize)</i>	body, input, ...
<b>Elements</b>	<i>layoutu base setting</i>	body, input, h1, p, ...
<b>Objects</b>	<i>base components</i>	.media, .button,
<b>Components</b>	<i>other components</i>	.article-item, .list, .footer-nav, ...
<b>Utilities</b>	<i>utility classes</i>	.img-fluid, .text-center, ...

# CSS SMACSS

## Scalable and Modular Architecture for CSS



# CSS

## SMACSS

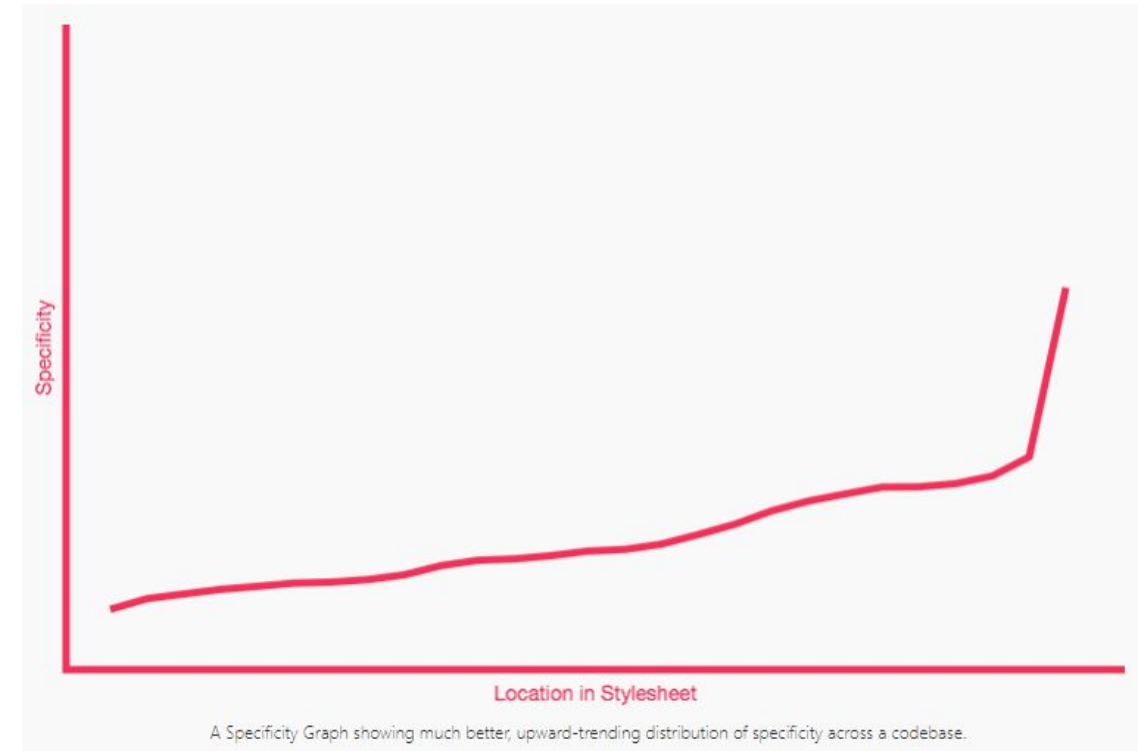
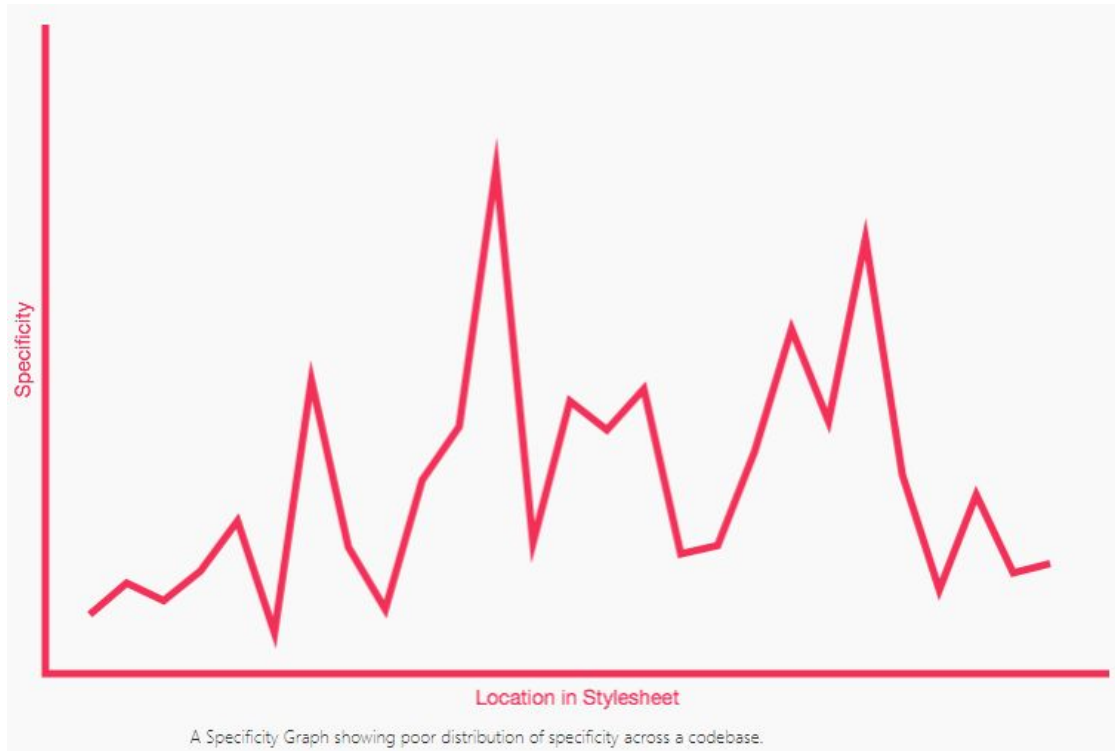
SMACSS	what kinds of styles	css classes
<b>Base</b>	<i>reset, base styles</i>	body, input, h1, p, ...
<b>Layout</b>	<i>grid, responsive fw, wrappers</i>	.row, .container, .header, ...
<b>Modules</b>	<i>components</i>	.article-list, .header-nav, ...
<b>State</b>	<i>states</i>	.active, .focus, ...



CSS

# Specificity graph

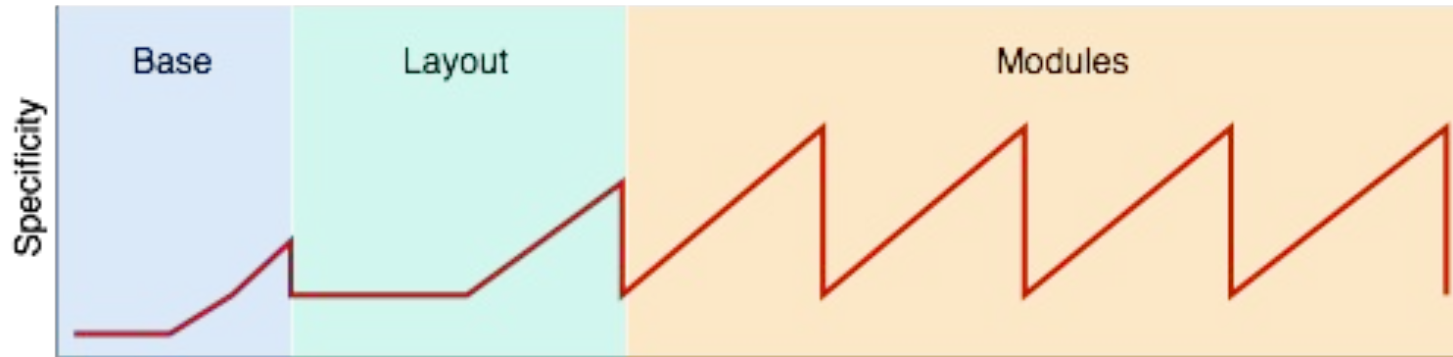
<https://cssstats.com/>



<https://csswizardry.com/2014/10/the-specificity-graph/>

CSS

# Specificity graph



H1, p, a, ... | .header, .grid, ... | .media-article, .pagination, ...

[https://snook.ca/archives/html\\_and\\_css/specificity-graphs](https://snook.ca/archives/html_and_css/specificity-graphs)

# CSS

## OOCSS

1. **Separation of Structure and Skin**
  - no HTML tags as selectors (except base layout settings)
2. **Separation of Container and Content**

`.footer .nav { ... }` → `.nav-footer { ... }`
3. **Keep specificity low**
  - no ID selectors, !important or selector combinations
  - `.nav.nav-footer { ... }` → `.nav-footer { ... }`

CSS

# Object oriented CSS

## Pros

1. Speed
2. Scalability
3. Efficiency
4. Maintainability
5. Readability
6. Relatability to Other Concepts

## Cons

1. Increases the Number of Classes Added to an Element
2. May Be Overkill for Small Projects
3. Requires a Learning Curve

# CSS

## OOCSS

```
<a href="#" class="button-blue-small">Click me!</a>
<style>
  .button-blue-small {
    display: inline-block;
    text-align: center;
    margin: 10px 5px;
    text-decoration: none;
    color: #FFF;
    border: 1px solid #0082BE;
    background: #00A4EF;
    font-size: 13px;
    padding: 5px 20px;
  }
</style>
```

CSS

# OOCSS – base

```
.button {  
  display: inline-block;  
  text-align: center;  
  margin: 10px 5px;  
  text-decoration: none;  
}
```

CSS

## OOCSS – size

```
.button-small {  
  font-size: 13px;  
  padding: 6px 20px;  
}
```

CSS

# OOCSS – styles

```
.button-blue {  
  color: #FFF;  
  border: 1px solid #0082BE;  
  background: #00A4EF;  
}
```



# CSS

## OOCSS

```
<a href="#" class="button button-small button-blue">Click me!</a>
```

```
<style>  
  .button { ... }  
  .button-small { ... }  
  .button-blue { ... }  
</style>
```

CSS

# OOCSS - problems

```
<div class="article article-list">  
  <div class="article-header">  
    ...  
  </div>  
</div>
```

CSS

# **components versus context**

<https://codepen.io/machal/pen/JmdRaa>

CSS

# Block – Element – Modifier

<http://getbem.com/>

**Block** `.nav-main {}`

- Standalone entity that is meaningful on its own.

**Element** `.nav-main__list {}`

- A part of a block that has no standalone meaning and is semantically tied to its block.

**Modifier** `.nav-main--dark {}`

- A flag on a block or element. Use them to change appearance or behavior.

CSS

# Block –Element – Modifier

## Pros

1. Fix OOCSS cons
2. Only naming methodology
3. Easy to understand

## Cons

1. Strange `__` and `--`
2. Sometimes very long names.

# CSS

## BEM

```
<!-- Don't do this -->
<figure class="photo">
  
  <figcaption>Look at me!</figcaption>
</figure>

<style>
  .photo { }
  .photo img { }
  .photo figcaption { }
</style>
```

# CSS

## BEM - elements

```
<!-- Takhle ano -->
```

```
<figure class="photo">
```

```
  
```

```
  <figcaption class="photo__caption">Look at me!</figcaption>
```

```
</figure>
```

```
<style>
```

```
  .photo { }
```

```
  .photo__img { }
```

```
  .photo__caption { }
```

```
</style>
```

CSS

## BEM - modifiers

```
<a href="#" class="button button--small button--blue">  
  Click me!  
</a>
```

```
<style>  
  .button { ... }  
  .button--small { ... }  
  .button--blue { ... }  
</style>
```



# CSS BEM

```
<!-- Don't do this -->
```

```
<figure class="photo">  
    
  <figcaption class="photo__caption">  
    <blockquote  
      class="photo__caption__quote">  
      Look at me!  
    </blockquote>  
  </figcaption>  
</figure>
```

```
<style>  
.photo { }  
.photo__img { }  
.photo__caption { }  
.photo__caption__quote { }  
</style>  
</body>
```

# CSS BEM

```
<!-- OK -->
```

```
<figure class="photo">  
    
  <figcaption class="photo__caption">  
    <blockquote  
      class="photo__quote">  
      Look at me!  
    </blockquote>  
  </figcaption>  
</figure>
```

```
<style>  
.photo { }  
.photo__img { }  
.photo__caption { }  
.photo__quote { }  
</style>  
</body>
```

# CSS BEM

```
<!-- Don't do this -->
```

```
<figure class="photo">
```

```
  
```

```
  <figcaption class="photo__caption photo__caption--highlighted">
```

```
    Look at me!
```

```
  </figcaption>
```

```
</figure>
```

```
<style>
```

```
  .photo__img--highlighted { }
```

```
  .photo__caption--highlighted { }
```

```
</style>
```

# CSS

## BEM

```
<!-- OK -->
```

```
<figure class="photo photo--highlighted">  
    
  <figcaption class="photo__caption">  
    Look at me!  
  </figcaption>  
</figure>
```

```
<style>  
  .photo--highlighted .photo__img { }  
  .photo--highlighted .photo__caption { }  
</style>
```

# CSS

## BEM

```
<!-- Don't do this -->
```

```
<div class="somethesis somethesis--fastread">  
  <div class="somethesis__someelement"></div>  
</div>
```

```
<style>  
  .somethesis { }  
  .somethesis--fastread { }  
  .somethesis__someelement { }  
</style>
```

# CSS

## BEM

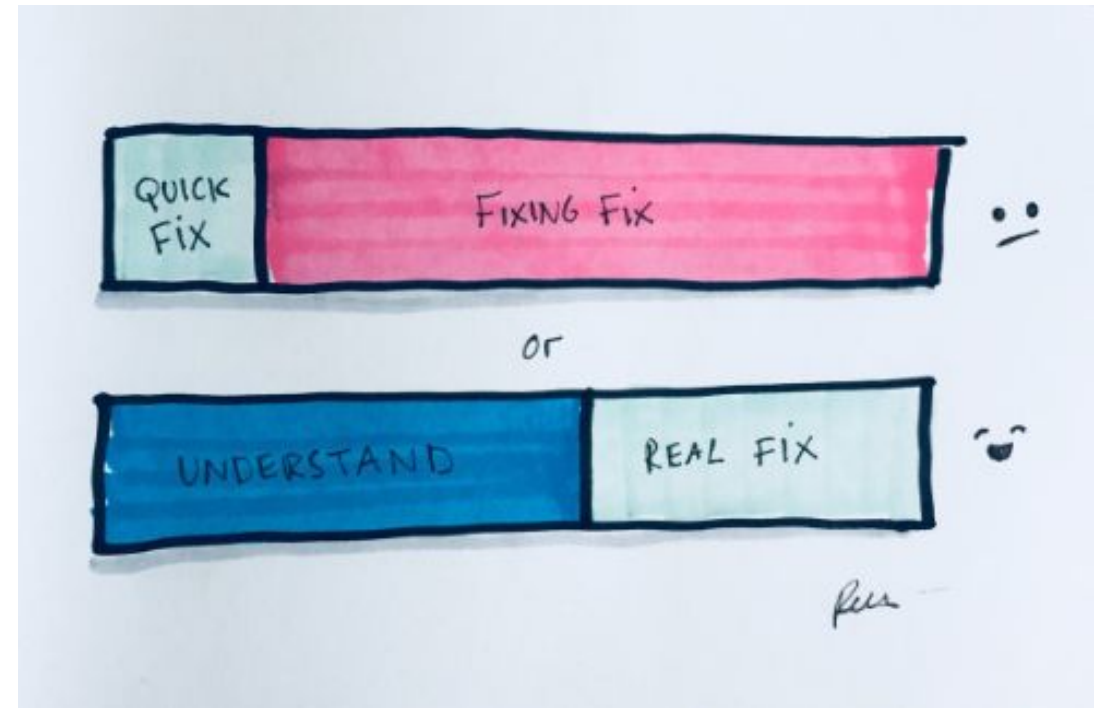
```
<!-- OK -->
```

```
<div class="some-thesis some-thesis--fast-read">  
  <div class="some-thesis__some-element"></div>  
</div>
```

```
<style>  
  .some-thesis { }  
  .some-thesis--fast-read { }  
  .some-thesis__some-element { }  
</style>
```

# CSS Refactoring

- Small and often used parts of code
- Atomization of component  
`.article .article-main h3` → `.article-heading`
- Isolation of new code
- Bad code on one place (*shame.css*)



# CSS Patterns

```
.heading {  
  font-size: 24px;  
}
```

```
#header .nav h3 {  
  font-size: 16px;  
}
```



```
.heading {  
  font-size: 24px;  
}
```

```
.heading--sm {  
  font-size: 16px;  
}
```



# CSS Patterns

```
h1.heading {  
  font-size: 24px;  
}
```

```
nav > ul > li {  
  font-size: 16px;  
}
```



```
.heading {  
  font-size: 24px;  
}
```

```
.list-item {  
  font-size: 16px;  
}
```

# CSS Patterns

don't over-engineer

```
.heading {  
  @include my-font-b(24px, 30px, #d5d5d5);  
}
```



```
.heading {  
  font-size: 24px;  
  margin-bottom: 30px;  
  border-bottom: 1px solid #d5d5d5;  
}
```

# CSS

## Patterns

use variables where values repeat

```
.heading {  
  font-size: 24px;  
  color: #d5d5d5;
```

```
@media (min-width: 768px) {  
  ...  
}  
}
```



```
.heading {  
  font-size: $text-xl;  
  color: $color-secondary;
```

```
@media (min-width: $screen-md) {  
  ...  
}  
}
```

# CSS Patterns

```
.heading .nav .nav-footer ul > li a {  
  ...  
}
```



```
.heading-link {  
  ...  
}
```

# CSS Patterns

always prefer code that is understandable at first sight


```
.header {  
  background: #f5f5f5;  
  &__nav {  
    font-size: 16px;  
    &--large {  
      font-size: 18px;  
    }  
  }  
}
```



```
.header {  
  background: #f5f5f5;  
}  
  
.header__nav {  
  font-size: 16px;  
}  
  
.header__nav--large {  
  font-size: 18px;  
}
```

# CSS Patterns

```
.content {  
  border-top: 76px; /* magic number */  
}
```



```
$header-height: 60px;  
$spacing-base: 16px;
```

```
.content {  
  border-top: calc($header-height + $spacing-base);  
}
```

# CSS Patterns

```
.content {  
  position: relative;  
}
```



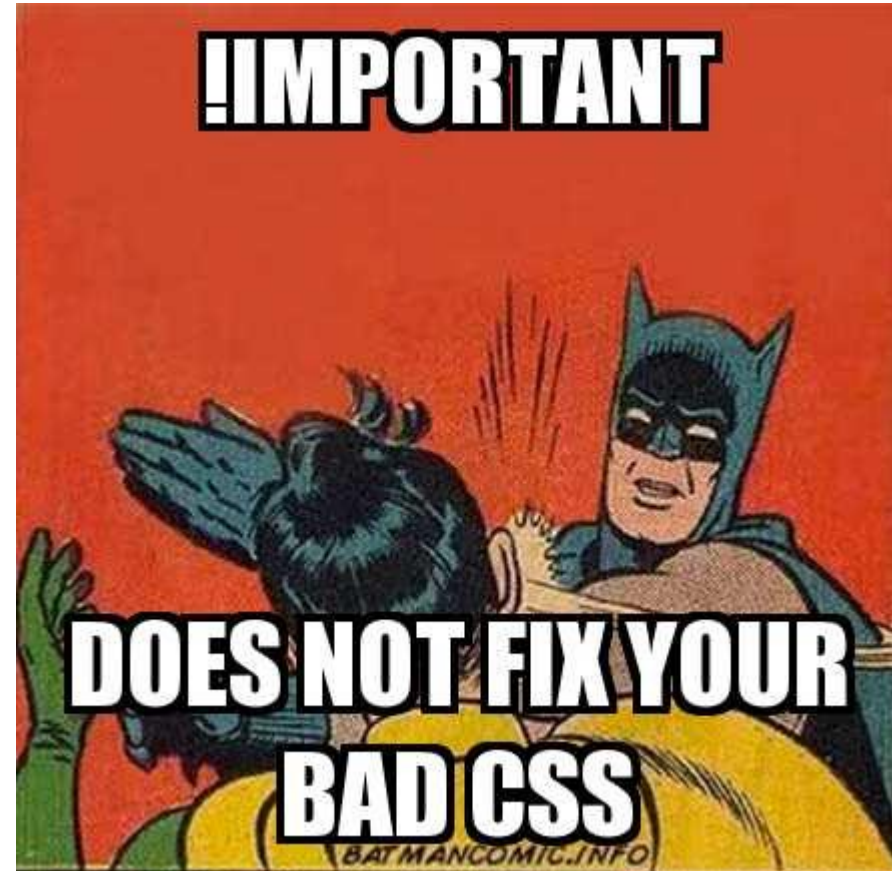
```
.content {  
  // inside is .ad-banner with absolute position  
  position: relative;  
}
```

# CSS Patterns

```
.text-red {  
  color: red !important;  
}
```

```
.text-red.text-more-red {  
  color: #e30604 !important;  
}
```

```
#main.text-blue {  
  color: blue !important;  
}
```





# CSS Patterns

## Inline styles kills kittens

```
.text-red {  
  color: red;  
}
```

```
<li class="text-red" style="color: blue;" >  
  ...  
</li>
```

# CSS Patterns

```
h1 {  
  font-size: 24px;  
}
```

```
.heading {  
  font-size: 3em;  
}
```

```
.heading-2 {  
  font-size: 2rem;  
}
```



```
h1 {  
  font-size: 1.5rem;  
}
```

```
.heading {  
  font-size: 3rem;  
}
```

```
.heading-2 {  
  font-size: 2rem;  
}
```

# CSS Patterns

```
.block {  
  -ms-transform: rotate(90deg);  
  -webkit-transform: rotate(90deg);  
  transform: rotate(90deg);  
}
```



```
.block {  
  transform: rotate(90deg);  
}  
  
/* autoprefixer for build */
```

# CSS Patterns

## when CSS smells

```
h2 {  
  font-size: 2em;  
  margin-bottom: 0.5em;  
  padding-bottom: 0.5em;  
  border-bottom: 1px solid #ccc;  
}  
  
.no-border {  
  padding-bottom: 0;  
  border-bottom: none;  
}
```



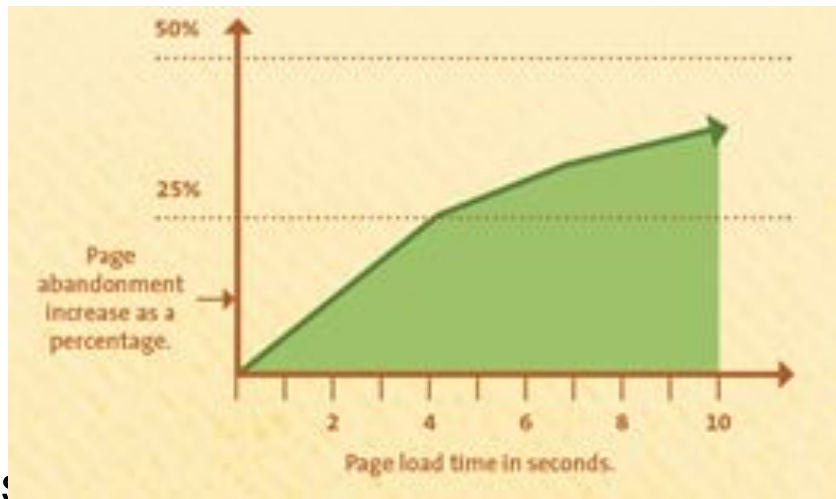
```
h2 {  
  font-size: 2em;  
  margin-bottom: 0.5em;  
}  
  
.headline {  
  padding-bottom: 0.5em;  
  border-bottom: 1px solid #ccc;  
}
```

# Page load & CSS

page loading

# Why to think about it

- 47% of consumers expect a web page to load in 2 seconds or less
- 40% of people abandon a website that takes more than 3 seconds to load
- A 1 second delay in page response can result in a 7% reduction in conversions
- Every second of increased page speed, increase two percent in conversion

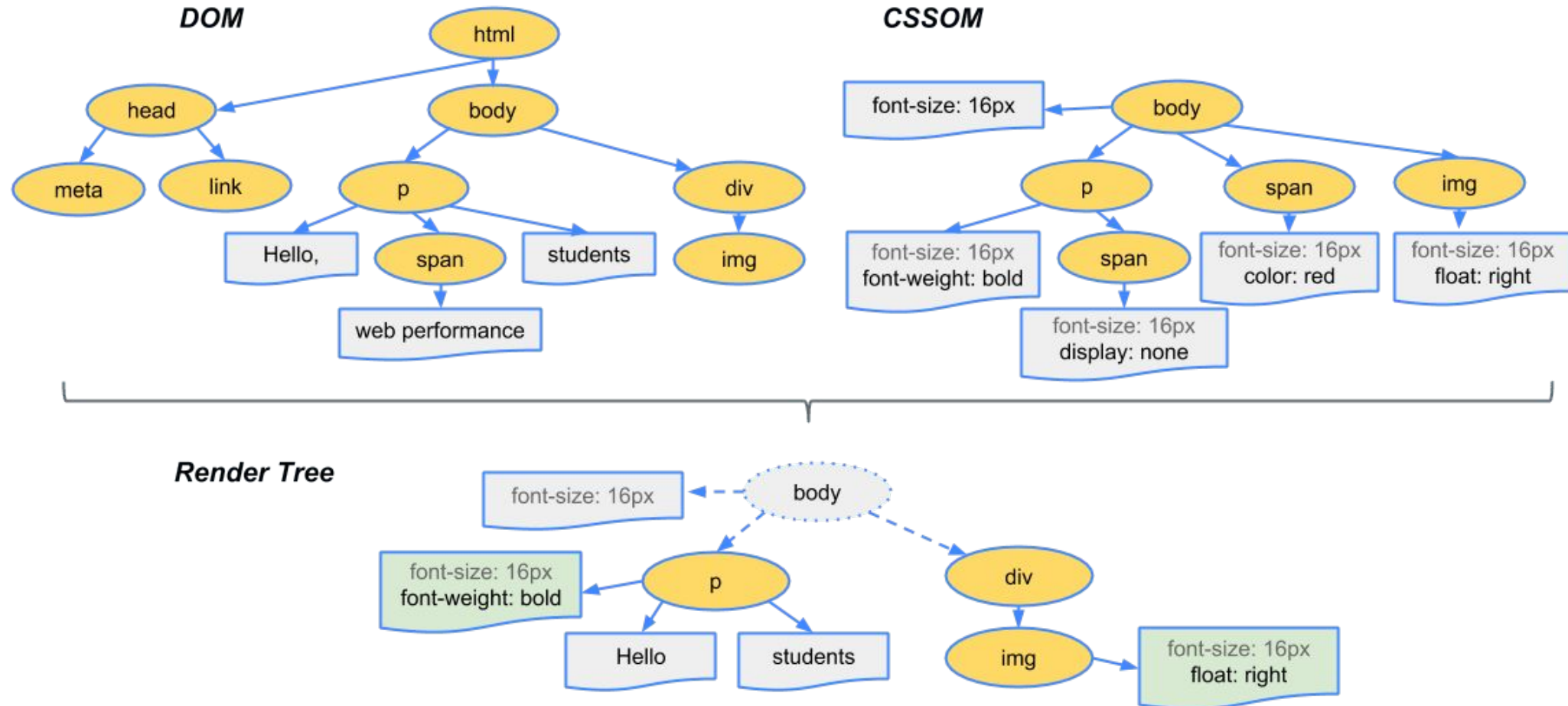


<https://headspin.io/resources/marketing/reports/5136-RR-performance-web-application.pdf>

<https://neilpatel.com/blog/loading-time/>

# page loading

## Render-tree Construction



# page loading **Metrics**

- **Time To First Byte (TTFB)**
- **DOM Content Loaded (DCL)**
- **First Paint (FP)**
- **First Contentful Paint (FCP)**
- **Time to Interactive (TTI)**
- **Speed Index**
- **Load**



# page loading Tools

- PageSpeed Insights

<https://developers.google.com/speed/pagespeed/insights/>

- Chrome DevTools

- Google Lighthouse

- WebPageTest

<https://www.webpagetest.org>

PageSpeed Tools > Insights

DOMOVSKÁ STRÁNKA PŘÍRUČKY REFERENCE PODPORA

PageSpeed Insights

http://www.aspectworks.cz/

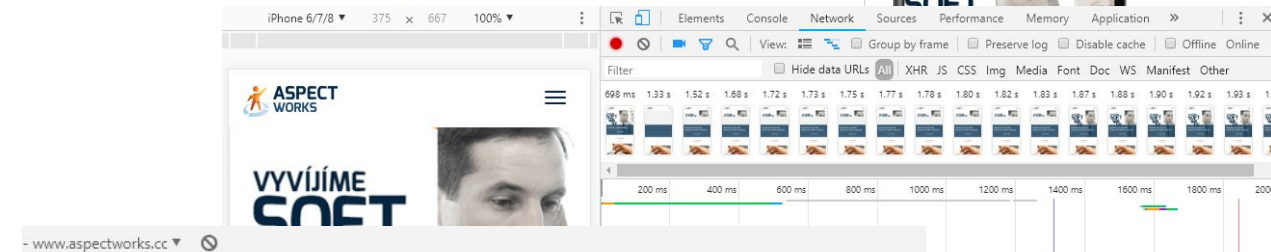
ANALYZOVAT

Mobilní zařízení

Počítač

Rychlost stránek  
Unavailable

Optimalizace  
Medium  
67 / 100



Web Page Performance Test for [www.aspectworks.com/](http://www.aspectworks.com/)

From: Prague, Czech Republic - Chrome - Cable  
24. 9. 2018 10:23:45

Summary Details Performance Review Content Breakdown Domains Processing Breakdown Screen Shot Image Analysis Request Map

Tester: WPTORGPRG-194.212.90.66  
First View only  
Test runs: 3  
Re-run the test

Need help improving?

A	A	A	F	B	X
First Byte Time	Keep-alive Enabled	Compress Transfer	Compress Images	Cache static content	Effective use of CDN

Raw page data - Raw object data  
Export HTTP Archive (.har)  
View Test Log

# page loading **CSS?**

- **Critical CSS**

- <https://jonassebastianohlsson.com/criticalpathcssgenerator/>
- <https://github.com/addyosmani/critical>

- **Async CSS Loading**

- in combination with Critical CSS
- <https://www.filamentgroup.com/lab/async-css.html>

# page loading

## FOIT versus FOUT

- Flash of Invisible Text – empty space until font is loaded
- Flash of Unstyled Text – system font until font is loaded
- <https://www.zachleat.com/foitfout/#4000,4000,4000,4000>



### Links:

<https://www.filmgroup.com/lab/font-events.html>

<https://www.zachleat.com/web/comprehensive-webfonts/>

# page loading

## FOIT versus FOUT

```
<style>
body {
  font-family: Helvetica, Arial, sans-serif;
}

.fonts-loaded body {
  font-family: "Roboto", Helvetica, Arial, sans-serif;
}
</style>
```

```
<script>
var roboto = new FontFaceObserver("Roboto", {
  weight: 400
});

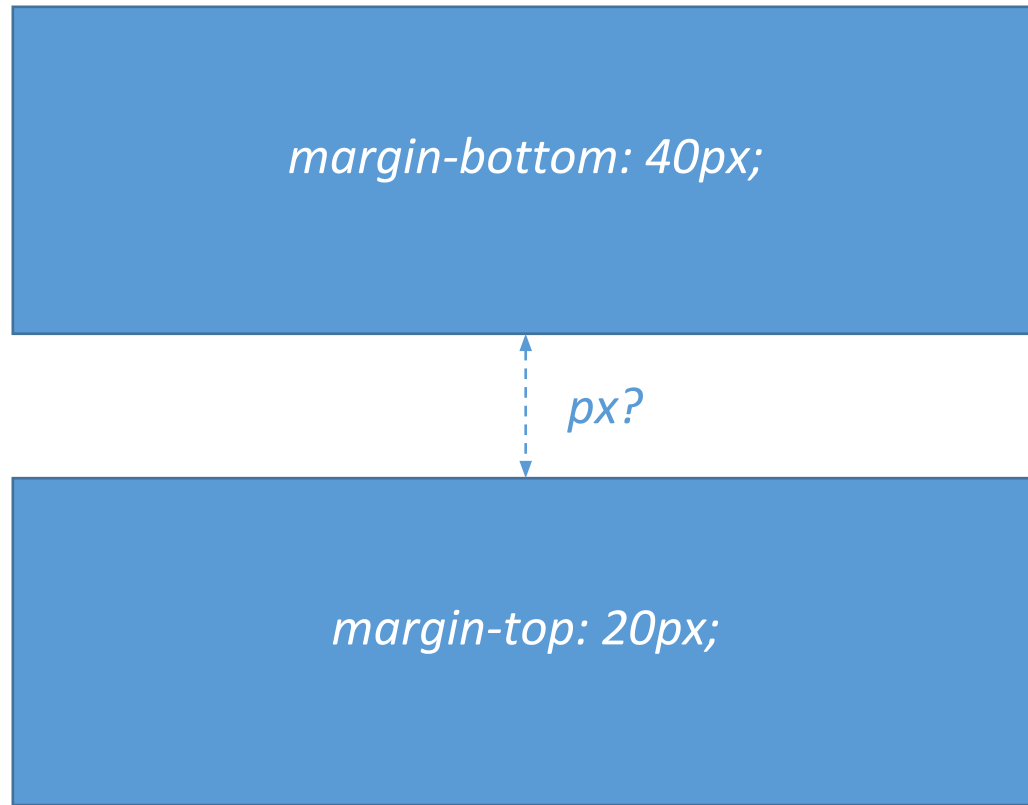
roboto.check().then(function() {
  document.documentElement.className += "fonts-loaded";
});
</script>
```

<https://jonsuh.com/blog/font-loading-with-font-events/>

# CSS use cases

CSS

# Margin-bottom flow



# CSS

## How to center

```
.centered {  
  position: absolute;  
  left: 50%;  
  top: 50%;  
  transform: translate(-50%, -50%);  
}
```

- <https://codepen.io/svobodalukas/pen/qMggRv>
- <https://developer.mozilla.org/en-US/docs/Web/CSS/transform>

# CSS Styled list

```
ul {  
  list-style: none;  
}  
  
ul li {  
  position: relative;  
  padding-left: 1.25rem;  
}  
  
ul li:before {  
  content: "";  
  position: absolute;  
  display: block;  
  ...  
}
```

<https://codepen.io/svobodalukas/pen/gdEKzP>



# CSS

## Styled checkbox

```
input[type="checkbox"] {  
  display: none;  
}
```

```
input[type="checkbox"] + label {  
  position: relative;  
  padding-left: 1.25rem;  
}
```

```
input[type="checkbox"] + label:before {  
  content: "";  
  ...  
}
```

```
input[type="checkbox"]:checked + label:before {  
  background: red;  
  ...  
}
```

<https://codepen.io/svobodalukas/pen/dqrKxP>

CSS

# Text truncate

```
white-space: nowrap;  
overflow: hidden;  
text-overflow: ellipsis;
```

<https://codepen.io/svobodalukas/pen/JaxVxZ>

# CSS Final

```
.workshop {  
  position: end;  
  patience: appreciated;  
}
```

<http://bit.ly/cn-css>