

Co warto wiedzieć o CSSie w 2020. roku

Paweł Zmarzły
ZOSIA 2020

Agenda

PLAN NA DZIŚ

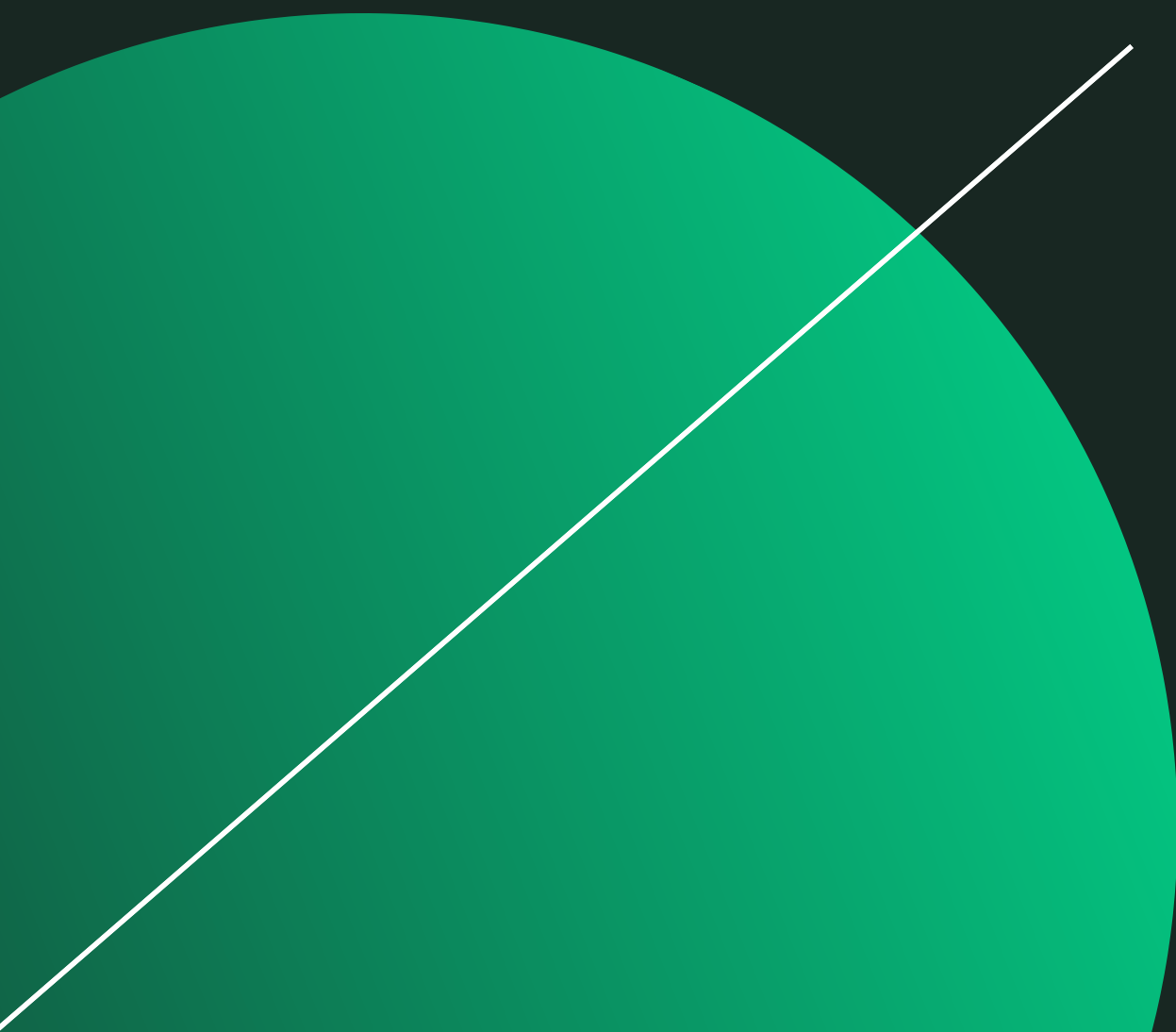
CSS – jak było, a jak jest

Beyond CSS – frameworki,
metodologie, narzędzia

Może więcej? Może mniej?
Slajdy na pzmarzly.pl

DISCLAIMER:

(ಽ) ze mnie a nie ekspert



Specificity

Web technology for developers ▸ CSS: Cascading Style Sheets ▸ Specificity

English ▾

On this Page

[How is specificity calculated?](#)

[Specifications](#)

[See also](#)

Specificity is the means by which browsers decide which CSS property values are the most relevant to an element and, therefore, will be applied. Specificity is based on the matching rules which are composed of different sorts of [CSS selectors](#).


Related Topics

[CSS](#)

[CSS Reference](#)

How is specificity calculated?

Specificity is a weight that is applied to a given CSS declaration, determined by the number of each [selector type](#) in the matching selector. When multiple declarations have equal specificity, the last declaration found in the CSS is applied to the element. Specificity only applies when the same element is targeted by multiple declarations. As per CSS rules, [directly targeted elements](#) will always take precedence over rules which an element inherits from its ancestor.

 **Note:** Proximity of elements in the document tree has no effect on the specificity.

Selector Types

The following list of selector types increases by specificity:

1. [Type selectors](#) (e.g., `h1`) and pseudo-elements (e.g., `::before`).
2. [Class selectors](#) (e.g., `.example`), [attributes selectors](#) (e.g., `[type="radio"]`) and



Fitting Text to a Container



Author

Chris Coyier

Published

Jun 29, 2018

9 Comments

[Go to Comments](#) →

FIT ME

FLUID TYPE

TEXT EFFECT

TEXT-JUSTIFY

Learn Development at [Frontend Masters](#)

There are a number of ways to go about putting some text in a container and having it size itself to fill that container. There are different technologies we can use and different considerations to think about. Let us count the ways.

TABLE OF CONTENTS

1	Introduction
1.1	Module Interactions
1.2	Values
2	The Viewport
3	Containing Blocks
3.1	Definition of containing block
4	Normal Flow
5	Floats
6	Positioning schemes
6.1	Relative positioning
6.2	Sticky positioning
6.3	Absolute positioning
6.4	Fixed positioning
6.5	Choosing a positioning scheme: 'position' property
6.6	Box offsets: 'top', 'right', 'bottom', 'left'
6.7	Logical box offsets: 'offset-before', 'offset-end', 'offset-after' and 'offset-start'
7	Choosing a different containing block for positioning
7.1	Attachment properties: 'attachment-type', 'attachment-top', 'attachment-right', 'attachment-bottom', 'attachment-left', and 'attachment'
8	Sizing and positioning details
8.1	The width of absolute or fixed positioned, non-replaced elements

§ 6. Positioning schemes

In CSS, a box may be laid out according to three positioning schemes:

1. Normal flow

In CSS, [normal flow](#) includes [block formatting](#) of block-level boxes, [inline formatting](#) of inline-level boxes, and [relative](#) and [sticky](#) positioning of block-level and inline-level boxes.

2. Floats

In the [float](#) model, a box is first laid out according to the [normal flow](#), then taken out of the flow and positioned, typically to the left or right. Content may flow along the side of a [float](#).

3. Absolute positioning

In the absolute positioning model, a box is removed from the normal flow entirely (it has no impact on later siblings) and assigned a position with respect to a [containing block](#).

An element is called *out-of-flow* if it is [floated](#), absolutely positioned, or is the root element. An element is called *in-flow* if it is not out-of-flow. The *flow of an element A* is the set consisting of A and all in-flow elements whose nearest out-of-flow ancestor is A.

§ 6.1. Relative positioning

Once a box has been laid out according to the [normal flow](#) or [floated](#), it may be offset relative to this position. This is called **relative positioning**. Offsetting a box (B1) in this way has no effect on the box (B2) that follows: B2 is given a position as if B1 were not offset and B2 is not re-positioned after B1's offset is applied. This implies that relative positioning may cause boxes to overlap. However, if relative positioning causes an ['overflow: auto'](#) or ['overflow: scroll'](#) box to have overflow, the user agent must allow the user to access this content (at its offset position), which, through the creation of a scrolling mechanism, may affect layout.

A relatively positioned box keeps its [normal flow](#) size, including line breaks and the space originally reserved for it.

A relatively positioned box establishes a new a new [containing block](#) for absolutely positioned descendants. (This is a common use of relatively positioned boxes.) The section on [containing blocks](#) explains when a relatively positioned box establishes a new [containing block](#).

For relatively positioned elements, ['left'](#) and ['right'](#) move the box(es) horizontally, without changing their size. ['Left'](#) moves the boxes to the right, and ['right'](#) moves them to the left. Since

Jak dodać CSS do strony

- `<style>/* ... */</style>`
- `<link rel="stylesheet" href="...">`
- `<div style="/* ... */">...</div>`

Jak dodać CSS do strony

- `<style>/* ... */</style>`
- `<link rel="stylesheet" href="...">`
- `<div style="/* ... */">...</div>`
- `<!DOCTYPE html>`

<https://eev.ee/blog/2020/02/01/old-css-new-css/>

https://developer.mozilla.org/en-US/docs/Mozilla/Mozilla_quirks_mode_behavior

```
<link rel="stylesheet" href="font.css" media="none"
      onload="if(media!='all')media='all'">
```

FOUC - Flash Of Unstyled Content

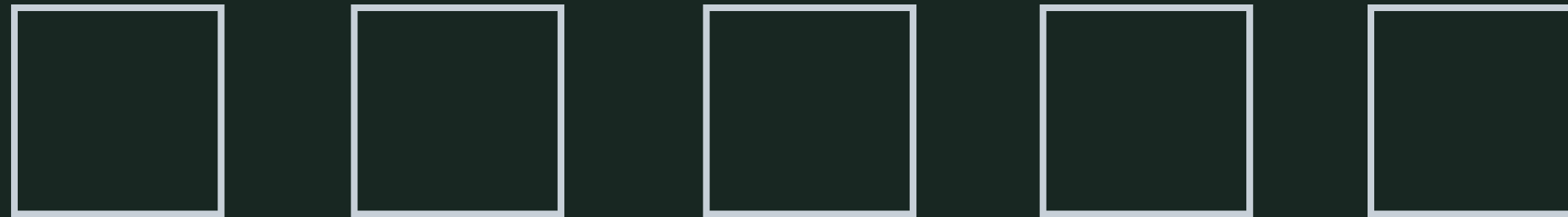
splash screen? what's the point

WSOD - White Screen Of Death

preload, inline

CSS Specificity - szczegółowość reguł

important, inline, id, class, tag



<https://github.com/gregoriohc/laravel-nova-theme-responsive/pull/23>

Layout

position: absolute, relative, static, fixed, sticky

<https://codepen.io/Shawo/pen/oNXwgpg>

float + clear: both (clearfix)

uwaga na Safari

Component is broken only in Safari

Asked 4 years, 1 month ago Active 4 years, 1 month ago Viewed 67 times

Layout

flexbox

grid (+ ciekawostka)

CSS variables

reset.css

:invalid

linear-gradient

Opanować chaos

BEM (Block Element Modifier)

.site-search__field--selected {}

<https://news.ycombinator.com/item?id=22430878>

SCSS / Sass

SCSS **Sass**

```
@use "sass:color";

.button {
  $primary-color: #6b717f;
  color: $primary-color;
  border: 1px solid color.scale($primary-color, $lightness: 20%);
}
```

CSS

```
.button {
  color: #6b717f;
  border: 1px solid #878d9a;
}
```

SCSS **Sass**

```
@use "sass:color"

.button
  $primary-color: #6b717f
  color: $primary-color
  border: 1px solid color.scale($primary-color, $lightness: 20%)
```

CSS

```
.button {
  color: #6b717f;
  border: 1px solid #878d9a;
}
```

```
1  .nav {
2    display: flex;
3
4    li {
5      flex: 1;
6    }
7
8    &:hover {
9      background: rgba(■ black, 0.5);
10   }
11 }
```

SCSS Sass

```
@mixin avatar($size, $circle: false) {  
  width: $size;  
  height: $size;  
  
  @if $circle {  
    border-radius: $size / 2;  
  }  
}  
  
.square-av { @include avatar(100px, $circle:  
false); }  
.circle-av { @include avatar(100px, $circle: true);  
}
```

CSS

```
.square-av {  
  width: 100px;  
  height: 100px;  
}  
  
.circle-av {  
  width: 100px;  
  height: 100px;  
  border-radius: 50px;  
}
```

CLI (PostCSS), IDE

grunt

gulp

webpack

Symfony Webpack Encore

laravel-mix


```
1  const mix = require('laravel-mix')
2  const tailwindcss = require('tailwindcss')
3
4  mix
5      .setPublicPath('dist')
6      .js('src/app.js', '')
7      .sass('src/app.scss', '')
8  ✓ .options({
9      processCssUrls: false,
10     postCss: [tailwindcss('./tailwind.config.js')],
11 })
12 ✓ .autoload({
13     jquery: ['$', 'window.jQuery', 'jQuery'],
14     moment: ['moment'],
15 })
16 .version()
17 .disableNotifications()
```

Bootstrap

Bulma

Semantic UI

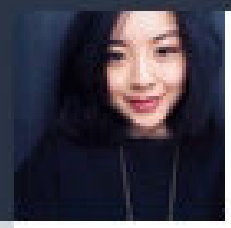
Tailwind CSS

"Utility first framework"

Adam Wathan - Full Stack Radio, [Cruddy by Design](#)



```
1 <div class="bg-white rounded-lg">
2   
3   <div>
4     <h2>Erin Lindford</h2>
5     <div>Customer Support</div>
6     <div>erinlindford@example.com</div>
7     <div>(555) 765-4321</div>
8   </div>
9 </div>
```



Erin Lindford
Customer Support
erinlindford@example.com
(555) 765-4321

linki [1] [2] [3] [4]

Tinkerwell

Figma

Sizzy, Polypane

BONUS #1

about:profiles

BONUS #2

masonry.