TailwindCSS

— how to make CSS great again

Dariusz Czajkowski

dczajkowski.com

@DCzajkowski

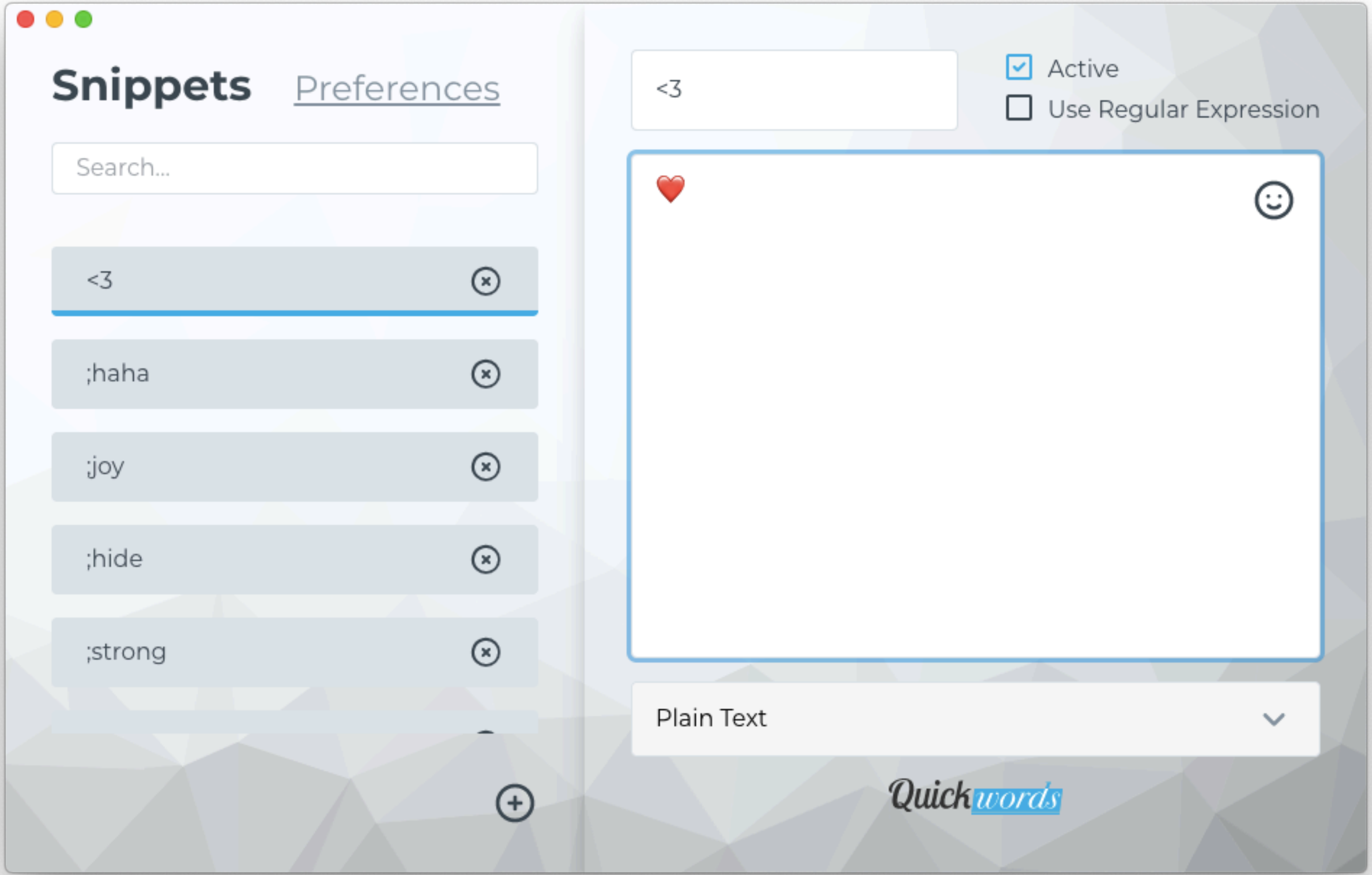
@CzajkowskiDarek

AGH

airly

mercari

Quickwords



<3

Active [✓]

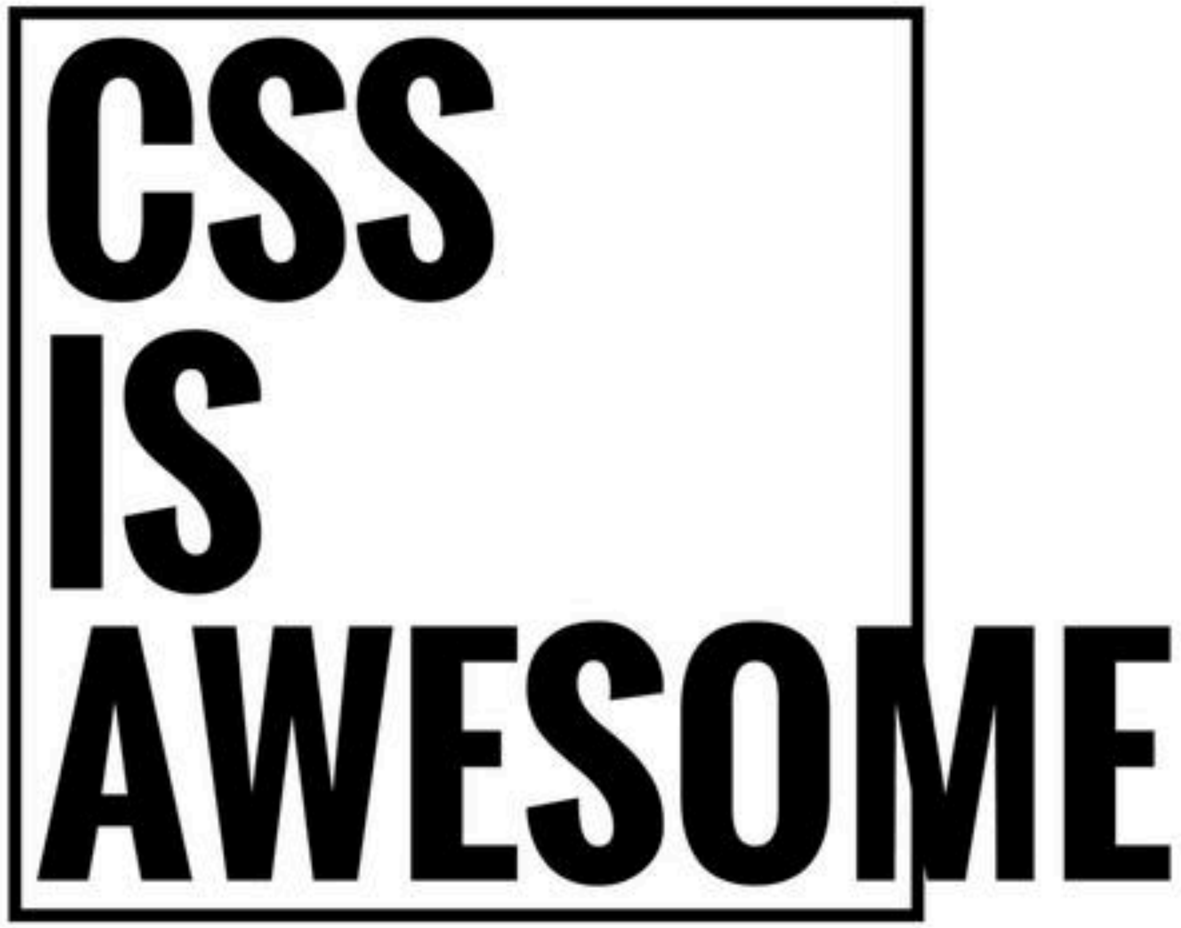
Use Regular Expression [ ]

❤️

Plain Text

Who has never written any CSS?

Who heard about atomic/utility css?


CSS
IS
AWESOME

CSS

IS

~~AWESOME~~

hard

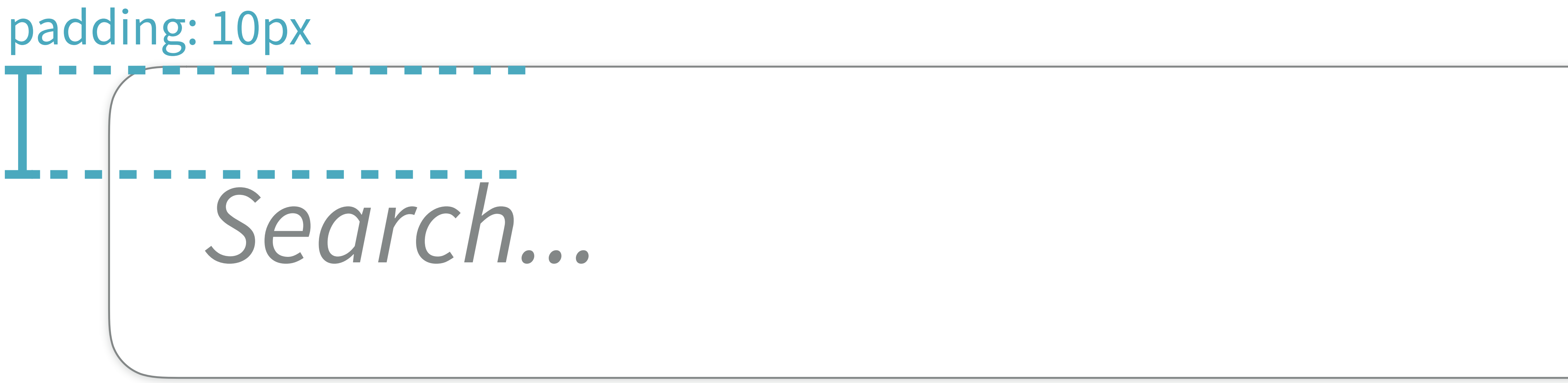
Naming stuff

”

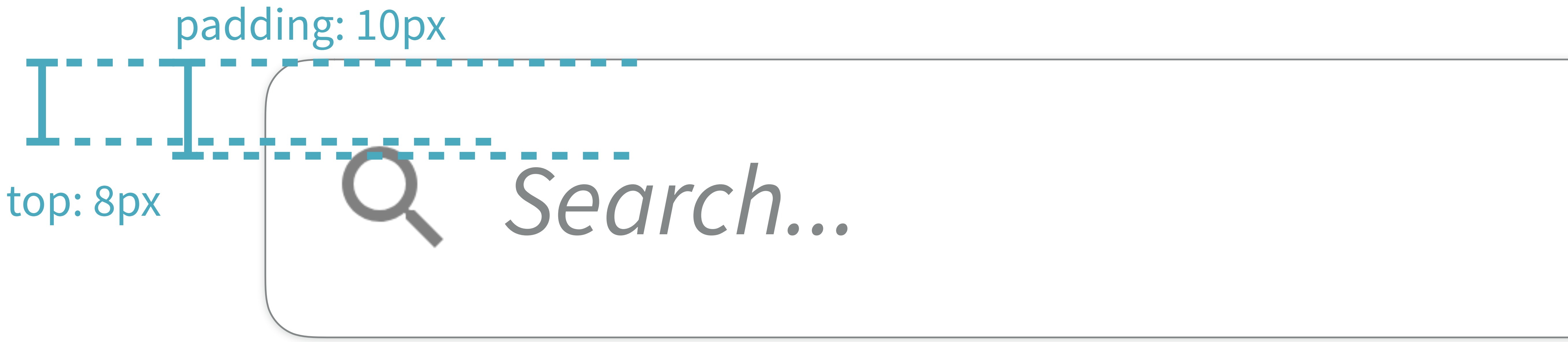
There are only two hard things in Computer Science: cache invalidation and naming things.

— Phil Karlton

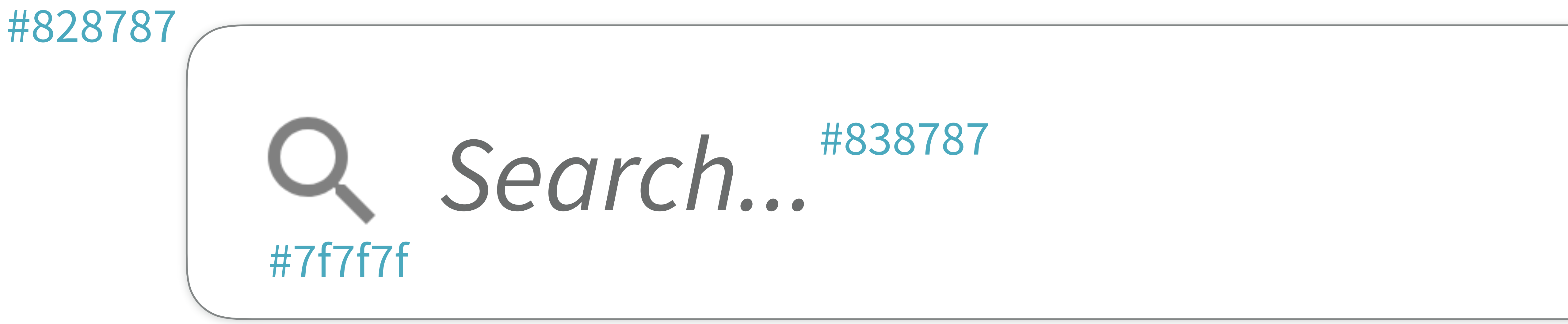
Random unit values



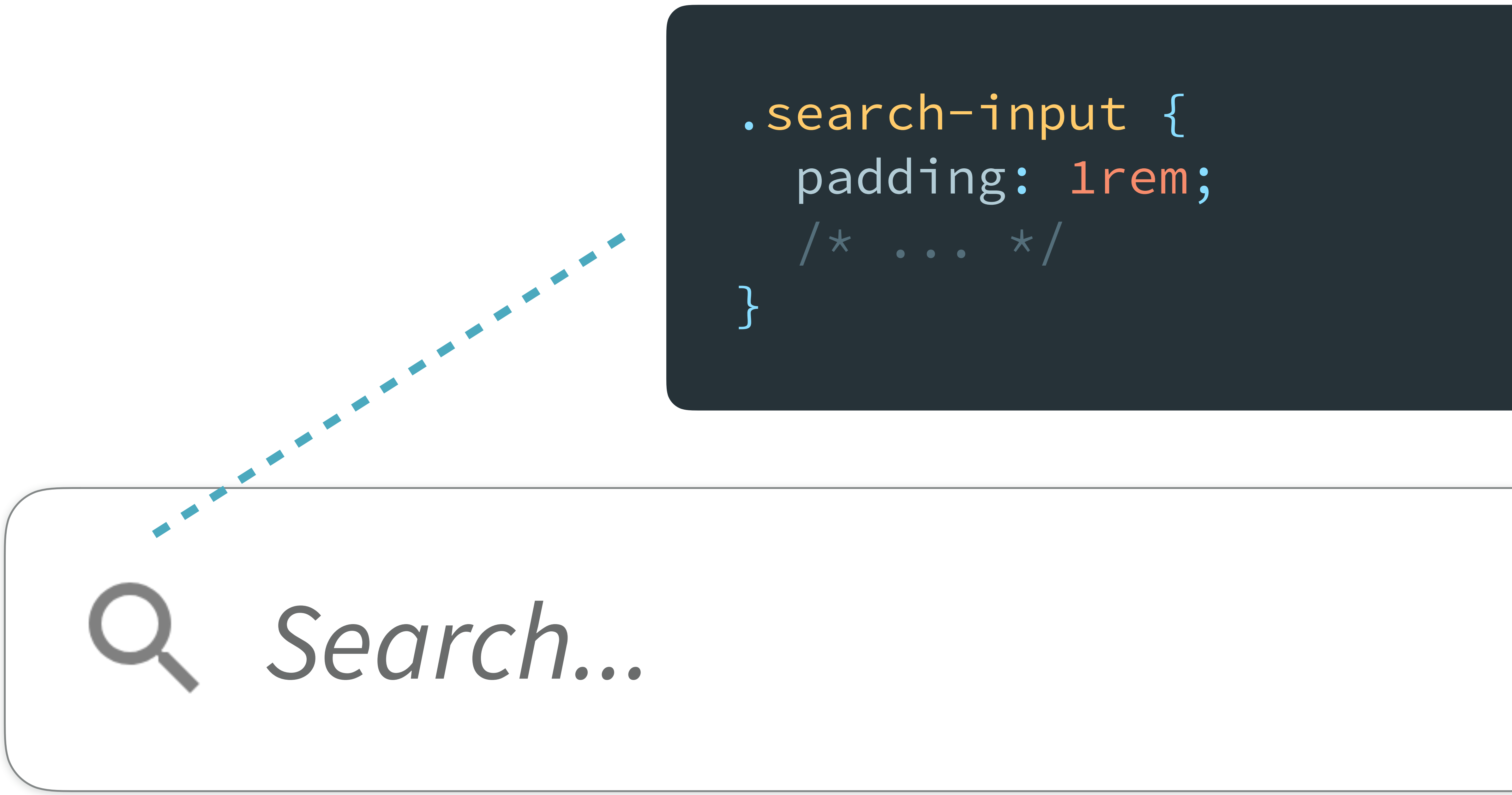
Random unit values



Random colors



Premature abstractions



```css
.search-input {
  padding: 1rem;
  /* ... */
}
```

Specificity is !important

```css
.list-item {
  padding: 1rem;
}
```

Specificity is !important

```
.list-item {
  padding: 1rem;
}

.header .list-item {
  padding: 1.5rem;
}
```

Specificity is !important

```css
.list-item {
  padding: 1rem;
}

.header .list-item {
  padding: 1.5rem;
}

.list-item--large-padding {
  padding: 3rem;
}
```

Specificity is !important

```css
.list-item {
  padding: 1rem;
}

.header .list-item {
  padding: 1.5rem;
}

.list-item--large-padding {
  padding: 3rem !important;
}
```

Specificity is !important

```css
.list-item {
  padding: 1rem;
}

.header .list-item {
  padding: 1.5rem;
}

.header .list-item--large-padding, .list-item--larg
  padding: 3rem;
}
```

Specificity is !important

```css
.list-item {
  padding: 1rem;
}

.header .list-item {
  padding: 1.5rem;
}

.list-item--large-padding.list-item--large-padding
  padding: 3rem;
}
```

Appending to global CSS

Naming collisions

CSS recompilation

Solution?

Use variables and modules

Use variables and modules

Ups

- Consistent unit values
- Predefined colors
- Divides code into smaller modules

Use variables and modules

Downs

- Does not solve premature abstractions problem
- Does not solve specificity problem
- Does not solve naming collisions
- Does not solve recompilation
- Does not enforce any style

Use CSS-in-JS

Use CSS-in-JS

```
import { css } from 'emotion';
import { gray500, gray700, teal600 } from '@/styles/colors';
import { s4, s8 } from '@/styles/scale';
import { large as fontLarge } from '@/styles/fonts';
import { radius } from '@/styles/misc';

const input = css`
  padding: ${s4} ${s8};
  background-color: ${gray500};
  border: 1px solid ${gray700};
  font-size: ${fontLarge};
  border-radius: ${radius};

  &:focus {
    border-color: ${teal600};
  }
`;

export default () => <input className={input} placeholder="Search..." />;
```

Use CSS-in-JS

Ups

- More-or-less solves random units
- More-or-less solves random colors
- Solves premature abstractions
- Solves specificity problem
- Solves appending to global styles
- Solves naming collisions
- Solves rebulding CSS

Use CSS-in-JS

Downs

- Does not enforce usage of variables
- CSS styles leak to the JS code
- Not easily reusable
- Forces a use of a build system

Use utility classes™

Use utility classes

```html
<input
  class="
    py-4 px-8 bg-gray-500 border border-gray-700 text-lg
    rounded focus:border-teal-700
  "
  placeholder="Search..."
/>
```

Use utility classes

Ups

- Solves unit problems
- Solves weird colors problem
- Solves premature abstraction
- Solves specificity problem
- Solves appending to global styles
- Solves naming collisions
- Solves rebuilding CSS
- CSS is still in your CSS file
- Gives you building blocks = easily reusable

Use utility classes

Downs

- Pollutes markup with the presentation logic
- Classes are reusable, but components are not

tailwindcss

How does it work?

Tailwind is just a PostCSS plugin

```
$ yarn add tailwindcss
```

```js
// postcss.config.js

module.exports = {
  plugins: [
    // ...
    require('tailwindcss'),
    require('autoprefixer'),
    // ...
  ],
};
```

```
$ npx tailwind init
```

```
// tailwind.config.js

module.exports = {
  theme: {},
  variants: {},
  plugins: [],
};
```

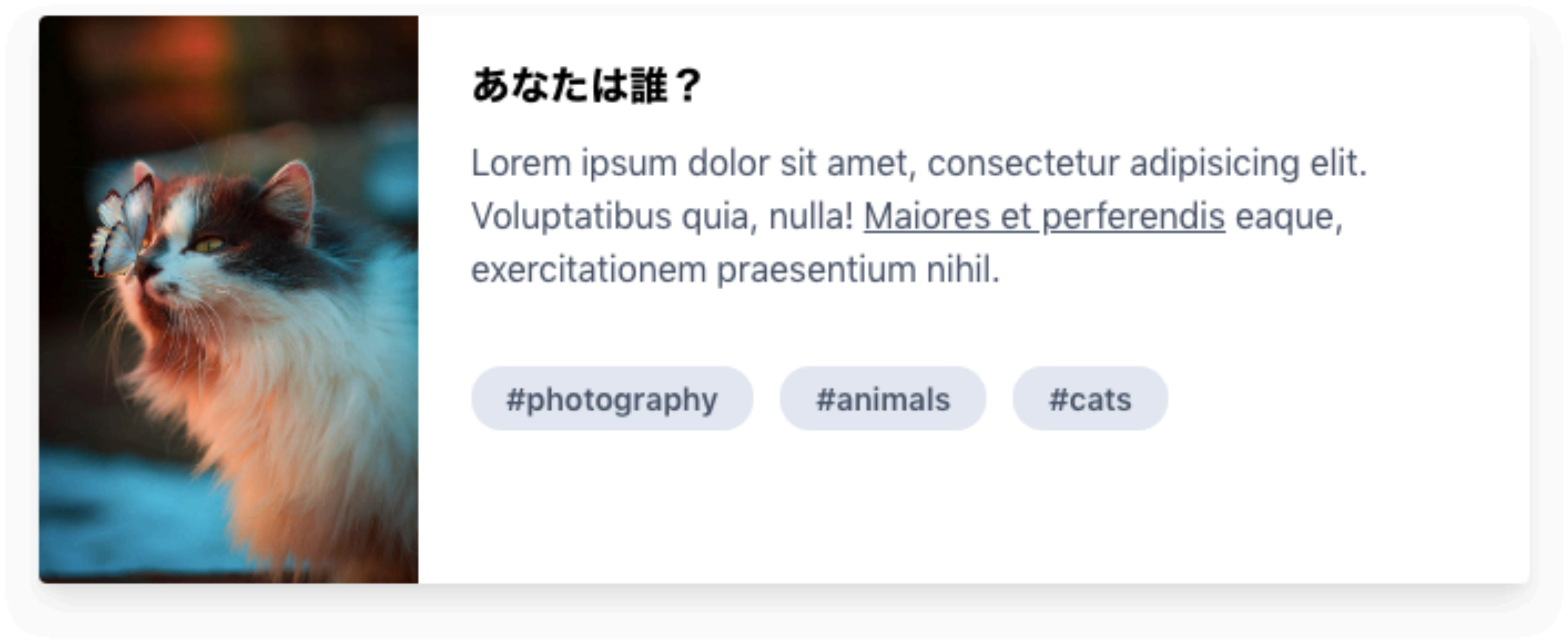
```css
/* app.css */

@tailwind base;

@tailwind components;

@tailwind utilities;
```

Demo time!



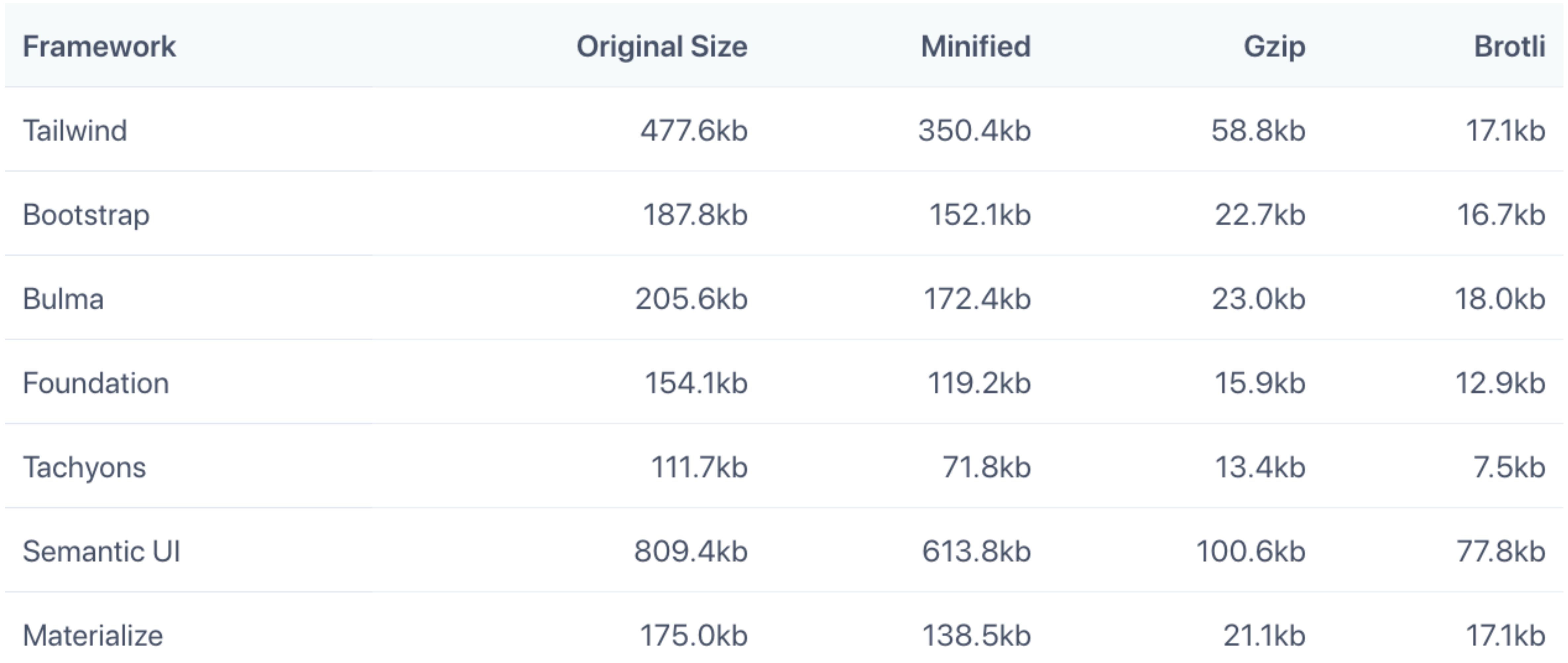
あなたは誰？

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Voluptatibus quia, nulla! Maiores et perferendis eaque, exercitationem praesentium nihil.

#photography #animals #cats

K, but what about the file size?

With a default config



Framework	Original Size	Minified	Gzip	Brotli
Tailwind	477.6kb	350.4kb	58.8kb	17.1kb
Bootstrap	187.8kb	152.1kb	22.7kb	16.7kb
Bulma	205.6kb	172.4kb	23.0kb	18.0kb
Foundation	154.1kb	119.2kb	15.9kb	12.9kb
Tachyons	111.7kb	71.8kb	13.4kb	7.5kb
Semantic UI	809.4kb	613.8kb	100.6kb	77.8kb
Materialize	175.0kb	138.5kb	21.1kb	17.1kb

Resources

Articles

- https://adamwathan.me/css-utility-classes-and-separation-of-concerns
- https://frontstuff.io/in-defense-of-utility-first-css

Tools

- https://tailwind.run
- https://github.com/aniftyco/awesome-tailwindcss

Thank you for listening!

Questions?

Slides and source code are available at:

darek.dev/tailwind