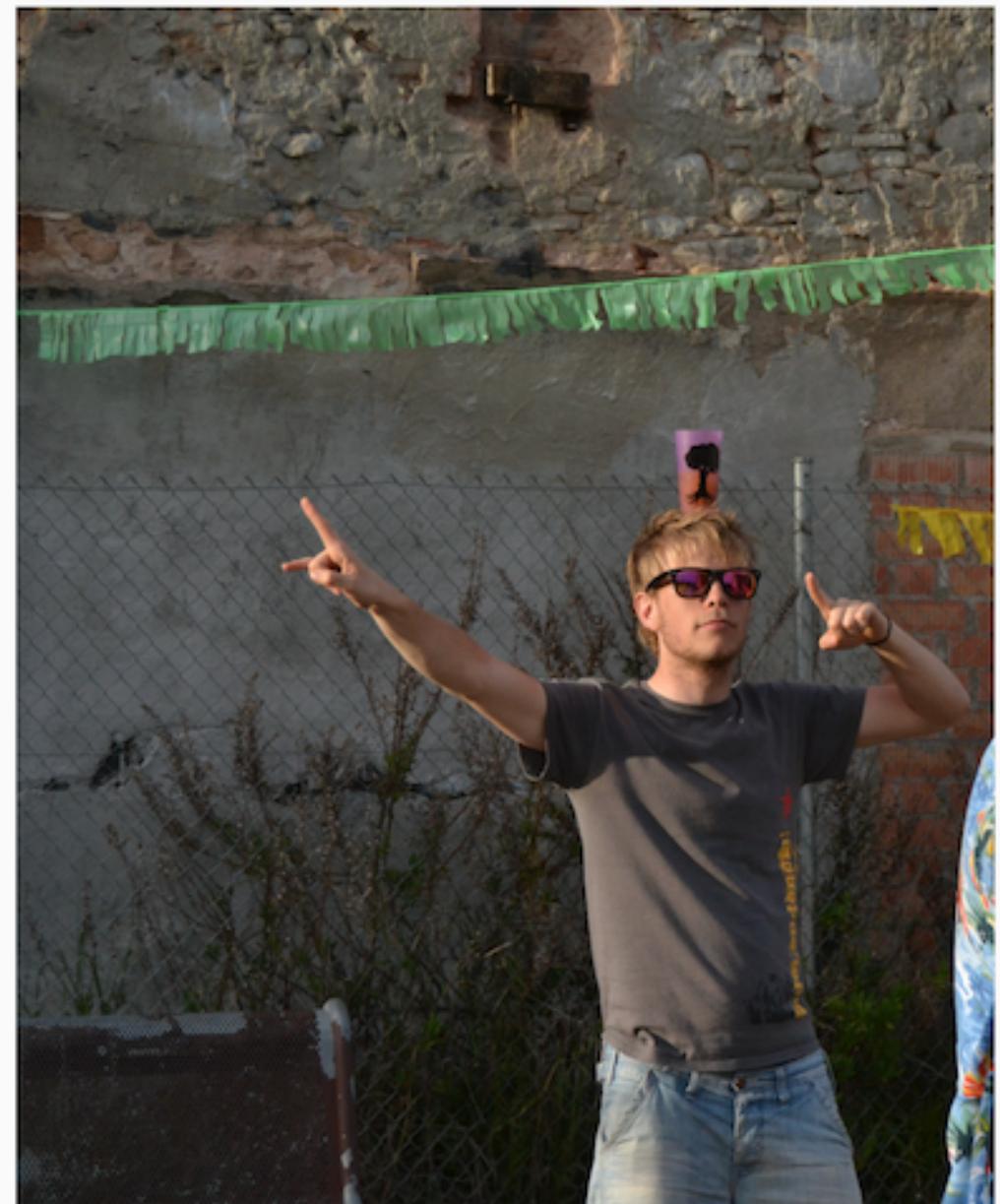


according to most neuroscience studies, 82% of the population agrees that their number one fear is **public speaking**.

number two is **death**.



Adrià Fontcuberta

@afontcu_

Software Engineer at Calidae

Adrià Fontcuberta

@afontcu_

Frontend Dev at Calidae

Adrià Fontcuberta

@afontcu_

UI Engineer at Calidae

Adrià Fontcuberta

@afontcu_

Webmaster at Calidae

Software Crafters Barcelona - VI Edition

Apiumhub

eDreams ODIGEO

plain concepts

CodelyTV



[...] netmind

SC
bcn

#SCBCN18



Agile-Spain

La Intercultural.com



**Max Stoiber**

@mxstbr

[Seguint](#)

How well do you know CSS?



Given these classes:

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

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

Which color would these divs be?

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

9% First red, second blue

44% First blue, second red

43% Both blue 

4% Both red

14,517 votes • Results finals



**Max Stoiber**

@mxstbr

[Seguint](#)

How well do you know CSS?



Given these classes:

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

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

Which color would these divs be?

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



Chapter 1

"wat CSS"

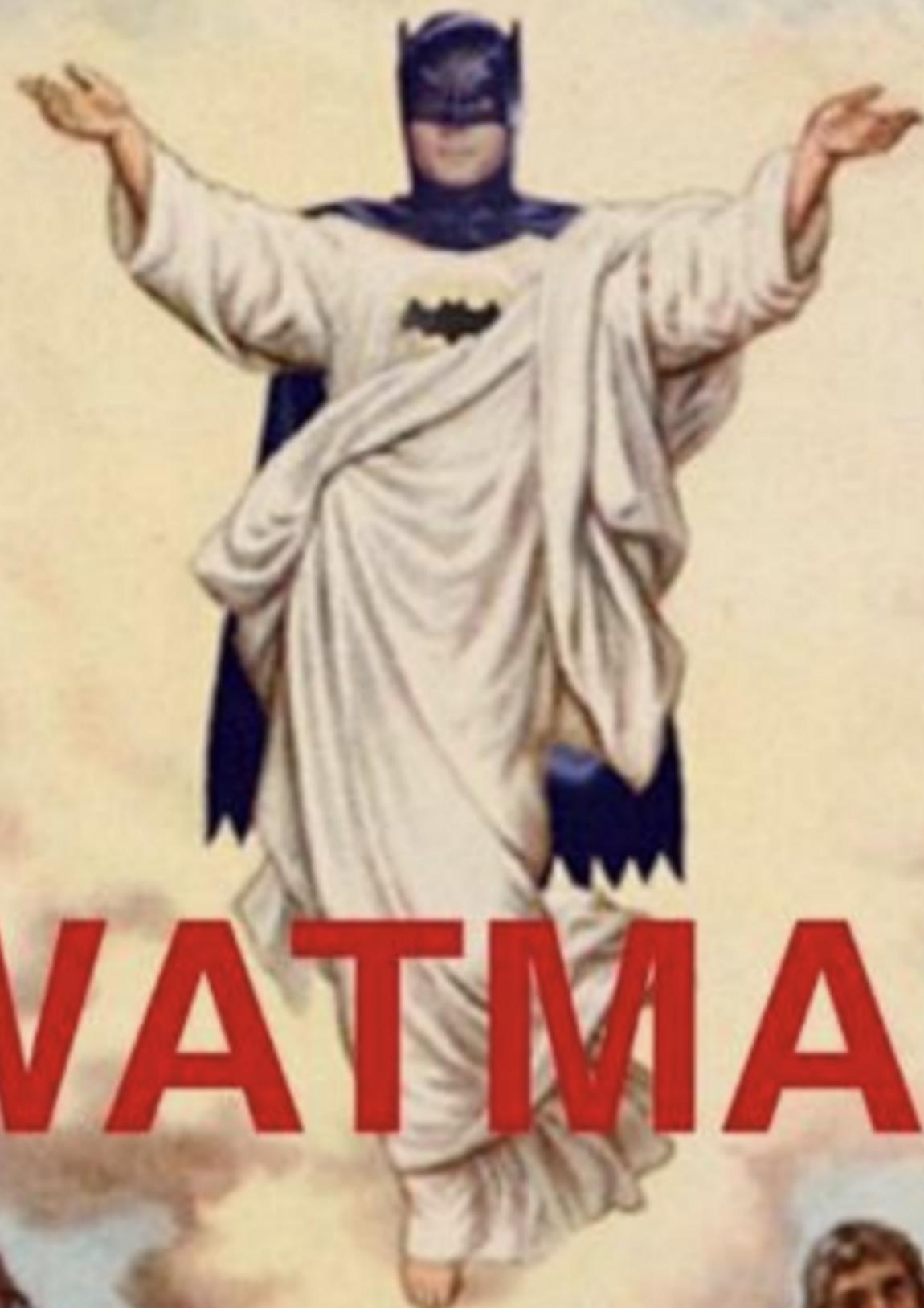
```
.qbutton.green {  
    border-color: #4c71a8;  
    background-color: #4c71a8;  
}
```



```
.home.column a.product.class300{  
    width: 300px;  
    float:left;  
}  
  
.home.column a.product.class400{  
    width: 450px;  
    float:right;  
}
```



```
.barra_blanca .menu .menu2 ul.sub {  
    content:"";  
    width: 0;  
    height: 0;  
    padding: 0;  
    margin: 0;  
    position:absolute;  
    top:-9999px;  
    left:-9999px;  
    line-height:0;  
    visibility:hidden;  
    display:none;  
}
```



WATMAN

```
        input {
            display: inline-block;
        }
    }
}
}
}

.map {
    width: 100%;
    height: 300px;
}
}
}
}
}
}
```

```
body.product .content-body-wrapper.tabs-block .switch  
.offer section.accommodation .content-width .accommodation  
.select-accommodation-box .list .map { width: 100%; height: 300px; }  
  
body.product .content-body-wrapper.tabs-block .switch .offer  
section.accommodation .content-width .accommodation  
.select-accommodation-box .list .list-content .items  
.accommodation-info #formAccomodation .accommodation-form  
.mealplan .select-price input { display: inline-block }
```

wat happened, CSS?

CSS Zen Garden happened

The screenshot shows a web browser window displaying the CSS Zen Garden homepage. The header features a large, stylized white 'Z' logo followed by the text 'CSS ZEN GARDEN' and 'The Beauty of CSS Design'. A prominent green banner at the top right contains the text 'VIEW ALL DESIGNS' with a right-pointing arrow. The main content area on the left includes a paragraph about CSS-based design, download links for example files, and sections titled 'THE ROAD TO ENLIGHTENMENT' and 'COMMUNICATING ABOUT?'. The right sidebar lists several design styles: 'MID CENTURY MODERN' by Andrew Lohman, 'GARMENTS' by Dan Mall, 'STEEL' by Steffen Knoeller, 'APOTHECARY' by Trent Walton, and 'SCREEN FILLER' by Elliot Jay Stocks.

CSS Zen Garden: The Beauty

← → C ⓘ No segur | www.csszengarden.com

VIEW ALL DESIGNS >

A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page.

Download the example [HTML FILE](#) and [CSS FILE](#)

THE ROAD TO ENLIGHTENMENT

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.

We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the [W3C](#), [WASP](#), and the major browser creators.

The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.

COMMUNICATING ABOUT?

MID CENTURY MODERN
by Andrew Lohman

GARMENTS
by Dan Mall

STEEL
by Steffen Knoeller

APOTHECARY
by Trent Walton

SCREEN FILLER
by Elliot Jay Stocks

The collage consists of four separate browser windows, each displaying a different design from the CSS Zen Garden:

- Top Left:** A design featuring a large white circle containing text about walking on the road to enlightenment.
- Top Right:** The "Select a Design" interface, showing a list of available designs: MID CENTURY MODERN by Andrew Lohman, GARMENTS by Dan Mall, STEEL by Stefan Kraslak, APOTHECARY by Trent Walton, and SCREEN FILLER by Elliot Jay Stocks.
- Middle Left:** A design with a minimalist, geometric aesthetic featuring a large 'CZG' logo and a sidebar titled "WASHES & STYLES".
- Middle Right:** A design titled "MADE LOCALLY" and "IMPECCABLE QUALITY" featuring a large "CSS ZEN GARMENTS" logo.
- Bottom Right:** A design featuring a pixelated landscape with a building labeled "CLOUDS" and a path leading to a house labeled "ZEN GARDEN". It includes sections for "SELECT DESIGN" and "WHAT IS THIS ABOUT?".

```
.sidebar .design-selection nav ul li a {  
    display: inline;  
}
```

"we are using separation of concerns"

separation of wat?



EXPLAIN
THIS IMAGE.COM

Remove the product



This action cannot be undone. Please proceed with caution.

Cancel

Yes, remove it

The "P" in CSS stands for "Principles"

The worst part about front-end development was that the other people solving your problems were front-end developers.

Adam Morse

1ssues

- 1) HTML and CSS are too coupled.
- 2) CSS is not reusable.

```
<div class="remove-product">  
    ...  
</div>
```

content-semantic classes is not a good idea.

Chapter 2

Goals

- 1) decouple CSS from HTML.
- 2) from semantic CSS to visual CSS.

```
<div class="remove-product">
  <h1>...
  <div>
    <img>...
    <p>...
  </div>
  <div>
    <button class="remove">...
    <button>...
  </div>
</div>
```

```
<div class="remove-product">
  <h1 class="remove-product-title">
    <div class="remove-product-body">
      <img class="remove-product-image">
      <p class="remove-product-message">
        </div>
      <div class="remove-product-buttons">
        <button class="remove">
          <button>
        </div>
      </div>
    </div>
```

```
<div class="modal">
  <h1 class="modal-title">
    <div class="modal-body">
      <img class="modal-icon">
      <p class="modal-message">
        </div>
      <div class="modal-buttons">
        <button class="btn primary">
        <button class="btn">
        </div>
    </div>
```

what if...

Remove the product



This action cannot be undone. Please proceed with caution.

Cancel

Yes, remove it



Changes saved successfully

```
.modal-messages {  
    .modal-icon { ... }  
    .modal-message { ... }  
}
```

```
.alert-content {  
    .alert-img { ... }  
    .alert-text { ... }  
}
```

```
.text-with-icon {}
```

```
.icon-text {}
```

The **media** object is an image to the left, with descriptive content to the right.

Nicole Sullivan

Remove the product



This action cannot be undone. Please proceed with caution.

Cancel

Yes, remove it



Changes saved successfully

```
<div class="modal">
  <h1 class="modal-title">
    <div class="modal-body">
      <img class="modal-icon">
      <p class="modal-message">
        </div>
      <div class="modal-buttons">
        <button class="btn primary">
        <button class="btn">
        </div>
    </div>
```

```
<div class="modal">
  <h1 class="modal-title">
    <div class="media">
      <img class="media-img">
      <p class="media-body">
    </div>
    <div class="modal-buttons">
      <button class="btn primary">
      <button class="btn">
    </div>
  </div>
```

```
<div class="media alert">
  <img class="media-img">
  <p class="media-body alert-content">
</div>
```

our CSS classes were now reusable across projects.

a component made of components.

modal = media + button + card + ...

we were not building pages anymore.

we were building components of a system.

from styling pages to creating design systems.

SENIOR DESIGN SYSTEM ENGINEER

Design systems

Principles

Rules

Constraints

Principles: Spacing should help establish relationship between elements.

Rules: Spacing values must be proportional.

Constraints: Only 5 spacing values available.

```
$baseline: 6px;
```

```
$unit-factor-tiny:    1;  
$unit-factor-small:   2;  
$unit-factor:         4;  
$unit-factor-large:  8;  
$unit-factor-huge: 16;
```

```
$spacing-unit-tiny  = $baseline * $unit-factor-tiny; // 6px
$spacing-unit-small = $baseline * $unit-factor-small; // 12px
$spacing-unit       = $baseline * $unit-factor;        // 24px
$spacing-unit-large = $baseline * $unit-factor-large; // 48px
$spacing-unit-huge  = $baseline * $unit-factor-huge;  // 96px
```

❤️ backends

what about **file structure**?

what about **naming**?

ITCSS

Inverted Triangle CSS

(by Harry Roberts)

Settings

Tools

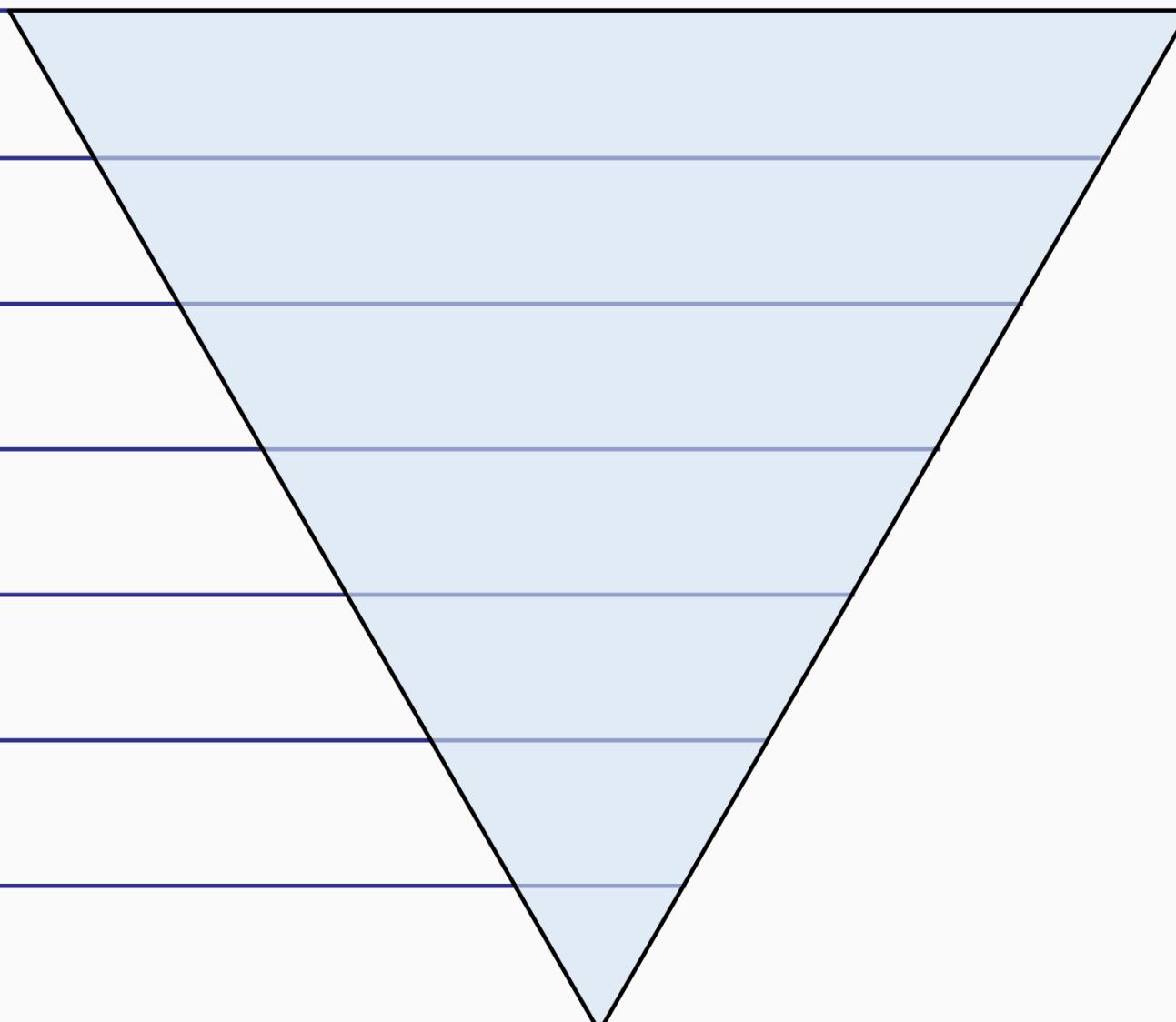
Generic

Base

Objects

Components

Utilities



```
1-Settings/  
    _colors.scss  
2-Tools/  
    _mixins.scss  
3-Base/  
    _normalize.scss  
4-Generic/  
    _image.scss  
5-Objects/  
    _media.scss  
6-Components/  
    _modal.scss  
7-Utilities/  
    _hidden.scss
```

/scss/1-Settings/_core.scss

```
$baseline: 6px;  
  
$unit-factor-tiny:    1;  
$unit-factor-small:   2;  
$unit-factor:         4;  
$unit-factor-large:  8;  
$unit-factor-huge: 16;
```

/scss/5-Objects/_media.scss

```
.media { ... }

.media-img { ... }

.media-body { ... }
```

/scss/6-Components/_modal.scss

```
.modal { ... }

.modal-buttons { ... }
```

you get the idea.

BEM

(Block – Element – Modifier)

```
.block {}
```

```
.block__element {}
```

```
.block--modifier {}
```

```
.block__element--modifier {}
```

```
.person {}
```

```
.person__eyes {}
```

```
.person--tall {}
```

```
.person__eyes--blue {}
```

last question:

how do we differentiate **objects**, **components** and
utilities?

Namespaces

Objects (.o-)

Components (.c-)

Utilities (.u-)

/scss/5-Objects/_objects.media.scss

```
.o-media { display: flex; align-items: flex-start; }

.o-media__body { flex: 1; }

.o-media__image { margin-right: $spacing-unit; }

.o-media--large .o-media__image { margin-right: $spacing-unit-large; }
```

Remove the product

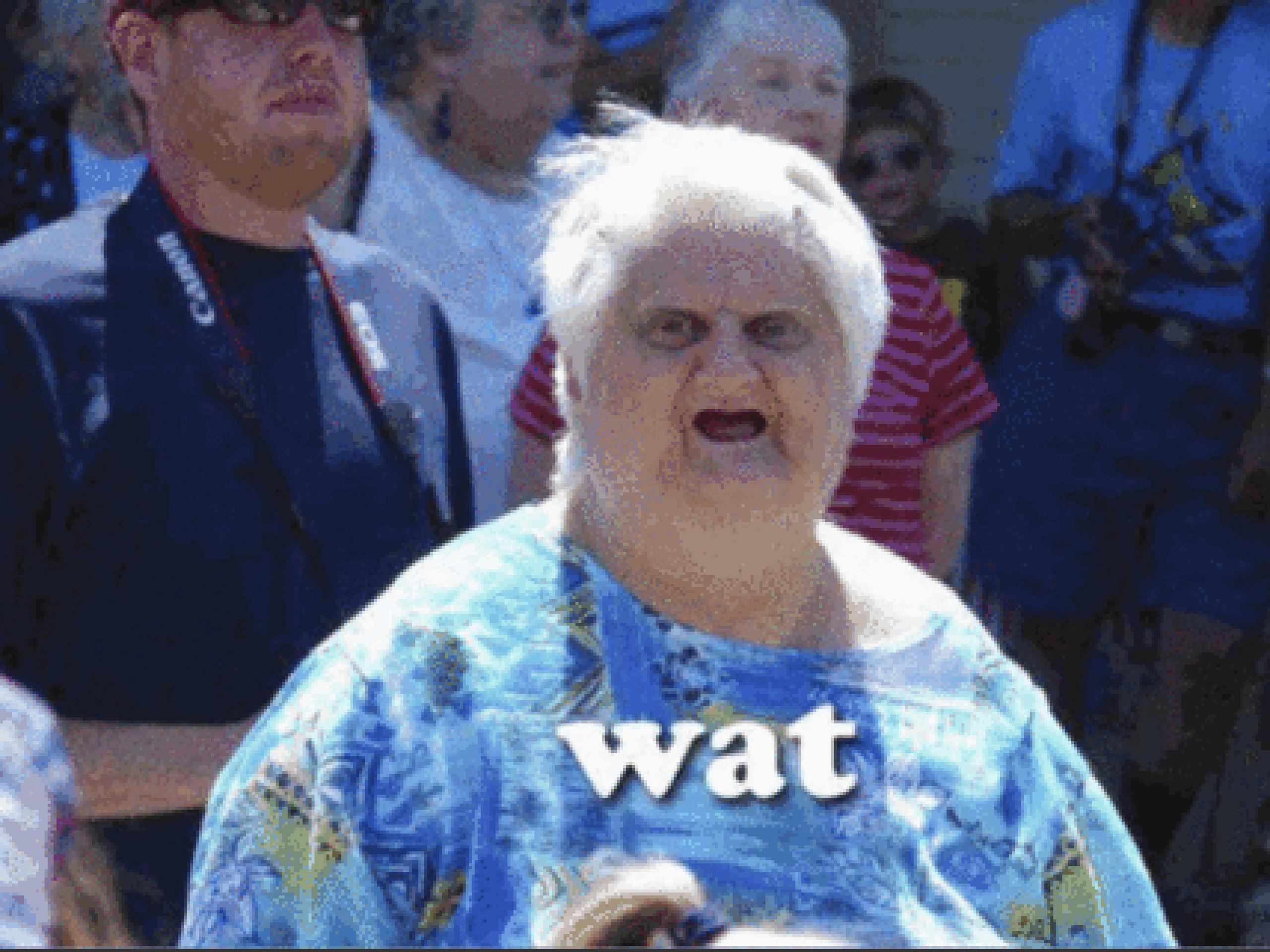


This action cannot be undone. Please proceed with caution.

Cancel

Yes, remove it

```
<div class="c-card c-card--rounded">
  <h1 class="c-card__header u-text-hairline">
    <div class="c-card__body">
      <div class="o-media o-media--center u-padding">
        <img class="o-media__img">
        <p class="o-media__body u-text-h5 u-text-hairline">
      </div>
      <div class="c-card__footer c-card__footer--between">
        <button class="c-btn c-btn--ghost">
        <button class="c-btn c-btn--primary">
      </div>
    </div>
  </div>
```



wat

```
<div class="c-card c-card--rounded">
  <h1 class="c-card__header u-text-hairline">
    <div class="c-card__body">
      <div class="o-media o-media--center u-padding">
        <img class="o-media__img">
        <p class="o-media__body u-text-h5 u-text-hairline">
      </div>
      <div class="c-card__footer c-card__footer--between">
        <button class="c-btn c-btn--ghost">
        <button class="c-btn c-btn--primary">
      </div>
    </div>
  </div>
```

it looks weird.



I WILL BE
YOUR HEART



death by repetition

```
.o-media--large .o-media__image {  
  margin-right: $spacing-unit-large; }
```

```
.c-list_item--large {  
  margin-right: $spacing-unit-large; }
```

```
.c-actions-list__item {  
  margin-right: $spacing-unit-large; }
```

```
.u-margin-right-large {  
  margin-right: $spacing-unit-large !important;  
}
```

Flexbox utilities

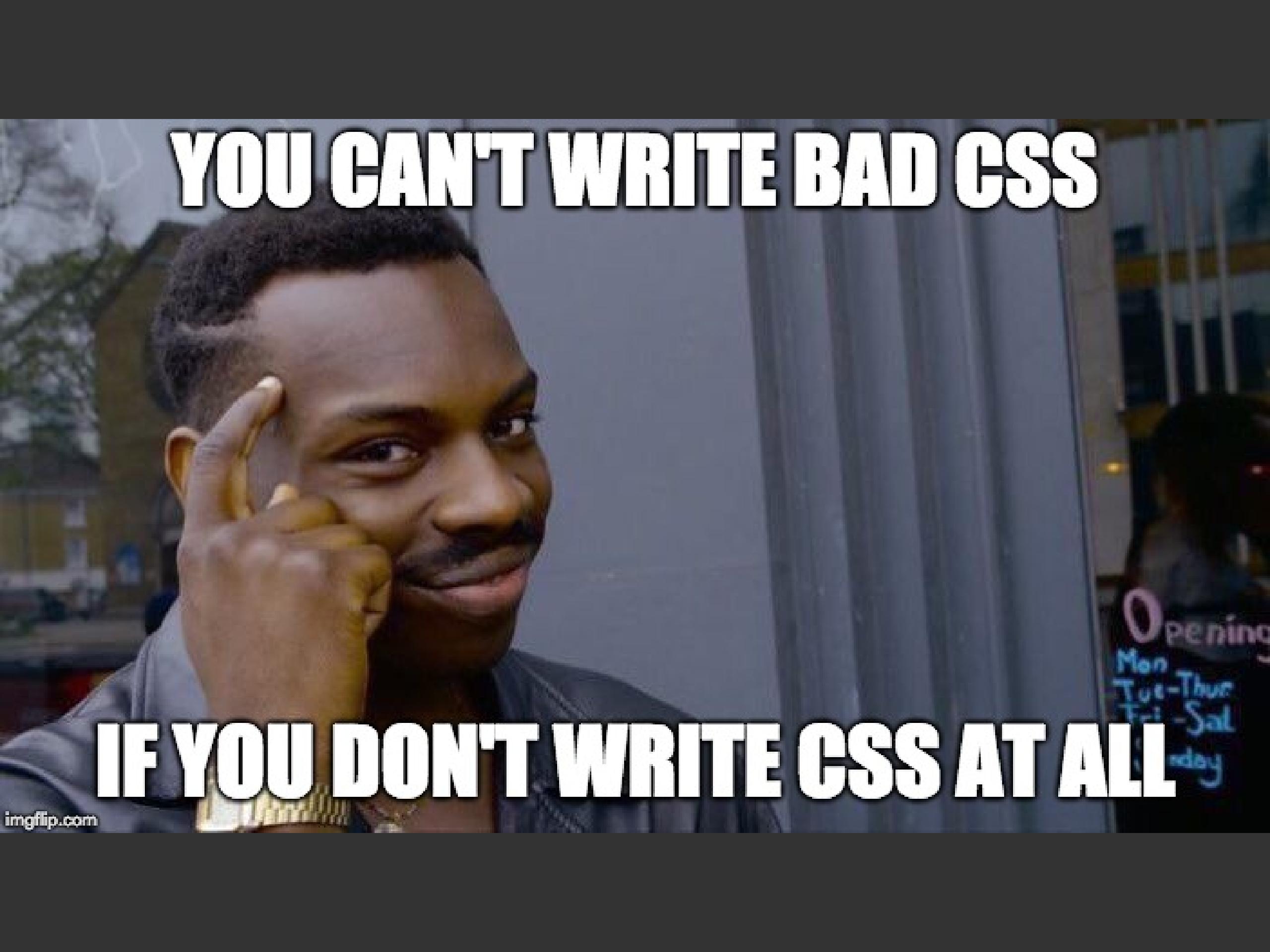
Padding and margin utilities

Width and height utilities

Text sizes, colors, styles and weights

Chapter 3

utility-first CSS.



YOU CAN'T WRITE BAD CSS

IF YOU DON'T WRITE CSS AT ALL

create a lot of small, reusable, focused CSS classes, and use them to build your UI.

```
<div class="u-padding u-border-radius u-background-white u-shadow-large">
  <h1 class="u-padding u-text-hairline">
    <div class="u-padding-vertical-small">
      <div class="o-media u-flex-items-center u-padding">
        <img class="o-media__img">
        <p class="o-media__body u-text-h5 u-text-hairline">
        </div>
      <div class="u-padding-small u-background-dark-gray
                u-flex u-flex-justify-between">
        <button class="c-btn c-btn--ghost">
        <button class="c-btn u-background-primary">
        </div>
      </div>
    </div>
  </div>
```

wat?

inline styles?



EXPLAIN
THIS IMAGE.COM

it is possible to build a whole new component without writing a single line of CSS.

Wrapping up

"a lot of things happened in 40 minutes"

Normandy CSS

Scss + ITCSS + BEM + Namespaces

[Github](#)

Next steps

JS + CSS = ❤?

Shadow DOM (Scoped CSS)

(Our) Learnings

- Use software principles.
- Use visual-based class naming.
- Think in terms of Design Systems.
- Utility-first. Abstract out components.
- Agree on structure, naming, principles...

Thank you !

What questions do you have?

(ask me anything! @afontcu_ on Twitter)

<https://maintainable-scalable-css.now.sh/>

References

wat – Gary Bernhardt

CSS and scalability – Adam Morse

Functional Programming, CSS, and your sanity – Jon Gold

Object Oriented CSS – Nicole Sullivan

The media object saves hundreds of lines of code – Nicole Sullivan

[Immutable CSS – Harry Roberts](#)

[CSS Utility Classes and "Separation of Concerns" – Adam Wathan](#)

[In Defense of Utility-First CSS – Sarah Dayan](#)

[Design Systems & React – Diana Mounter](#)

[Front-end Style Guides – Anna Debenham](#)

[Cooking with Design Systems – Dan Mall](#)

[About HTML semantics and front-end architecture – Nicholas Gallager](#)

[Atomic Design – Brad Frost](#)

[Shadow DOM: Self-Contained Web Components – Eric Bidelman](#)

[CSS Modules \(Github\)](#)

[Normandy CSS \(Github\)](#)
