If you are viewing this presentation in a PDF, please stop and open the slides in browser (slides in the browser look way better):
https://switowski.github.io/functional-css-talk/

Maintainable CSS

with visual regression testing and functional CSS

Sebastian Witowski

Problems of CSS

Cascading Style Sheets

Hard to maintain large files
(fighting specificity with IDs and !important)

Duplicated and unused code

Duplicated CSS

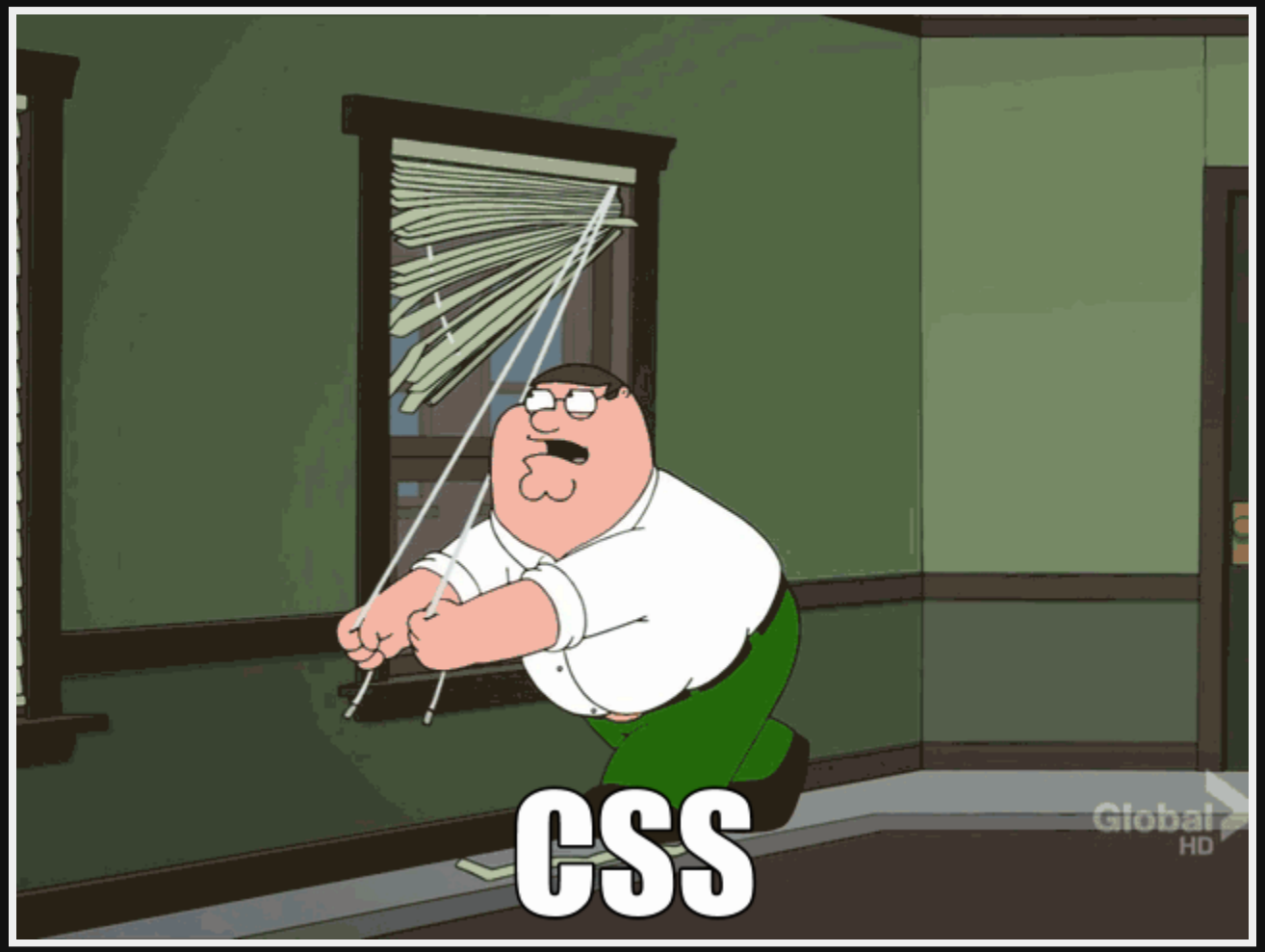
```
<div class="menu">...</div>
...
<div class="navigation">...</div>
```

```
.header2 {
    font-size: 24px;
}

...

h2 {
    font-size: 24px;
}
```

Unused CSS

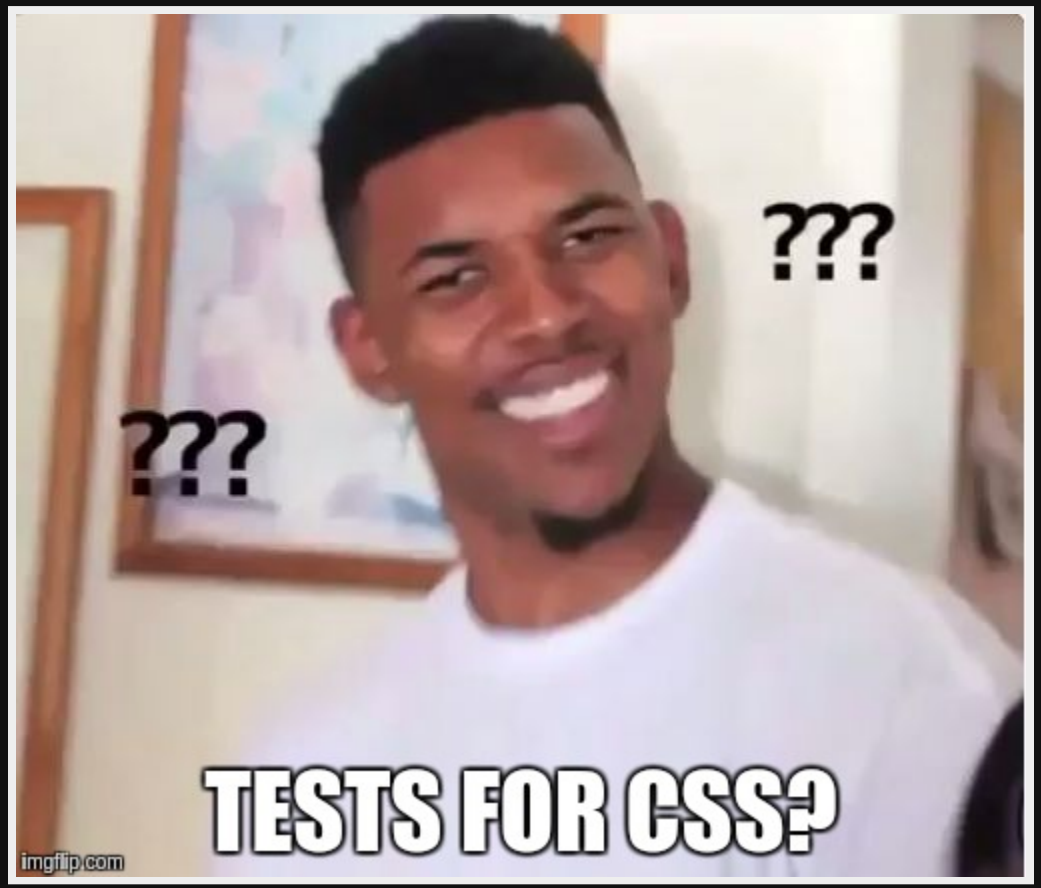
```
.menu {
    font-size: 16px;
}

...

# Chained selectors and cascading
.menu ul li.active {
    font-size: 2em;
}
```

CSS is hard to refactor



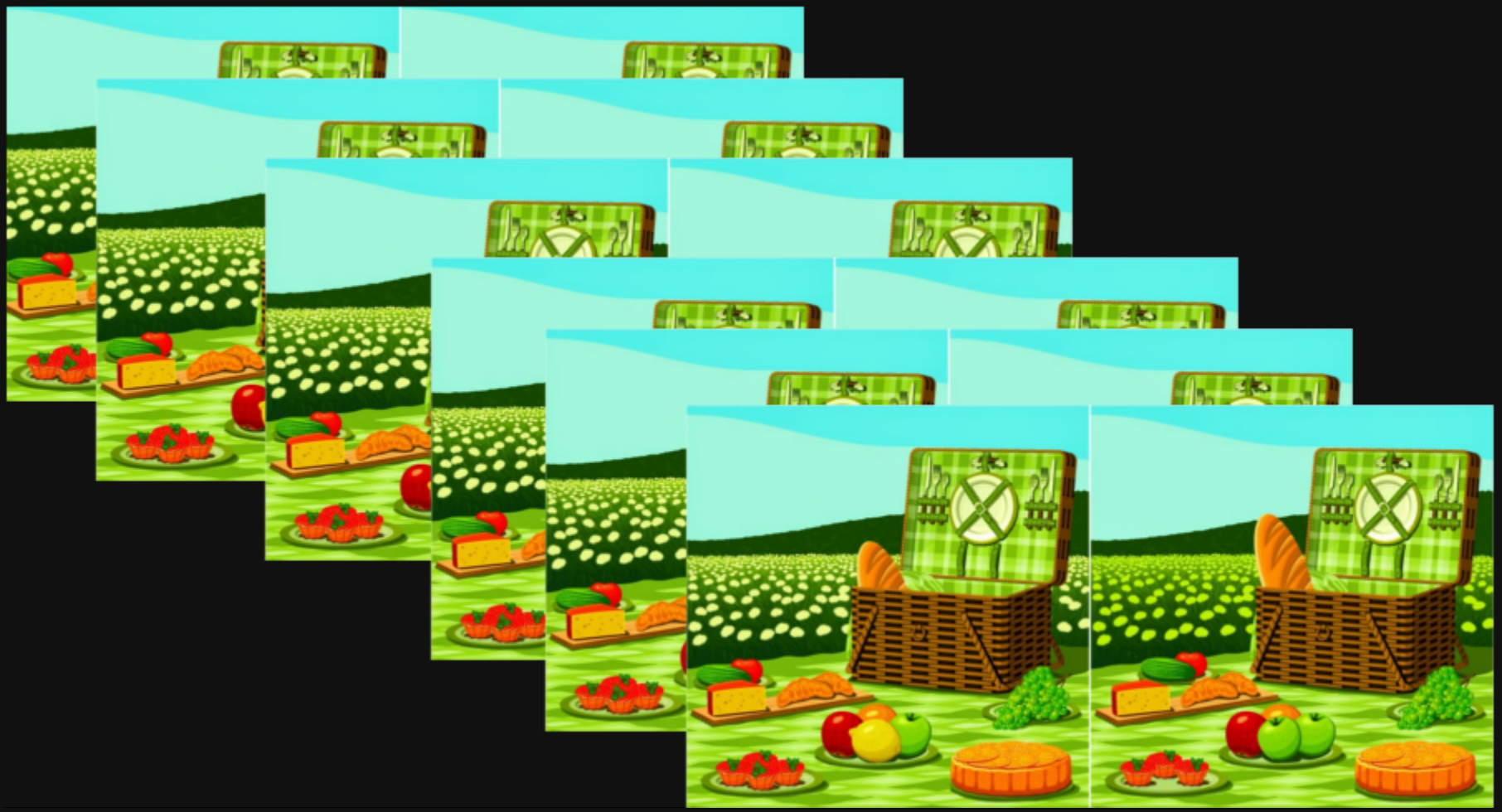
How can we solve this problem?

We can write tests for CSS

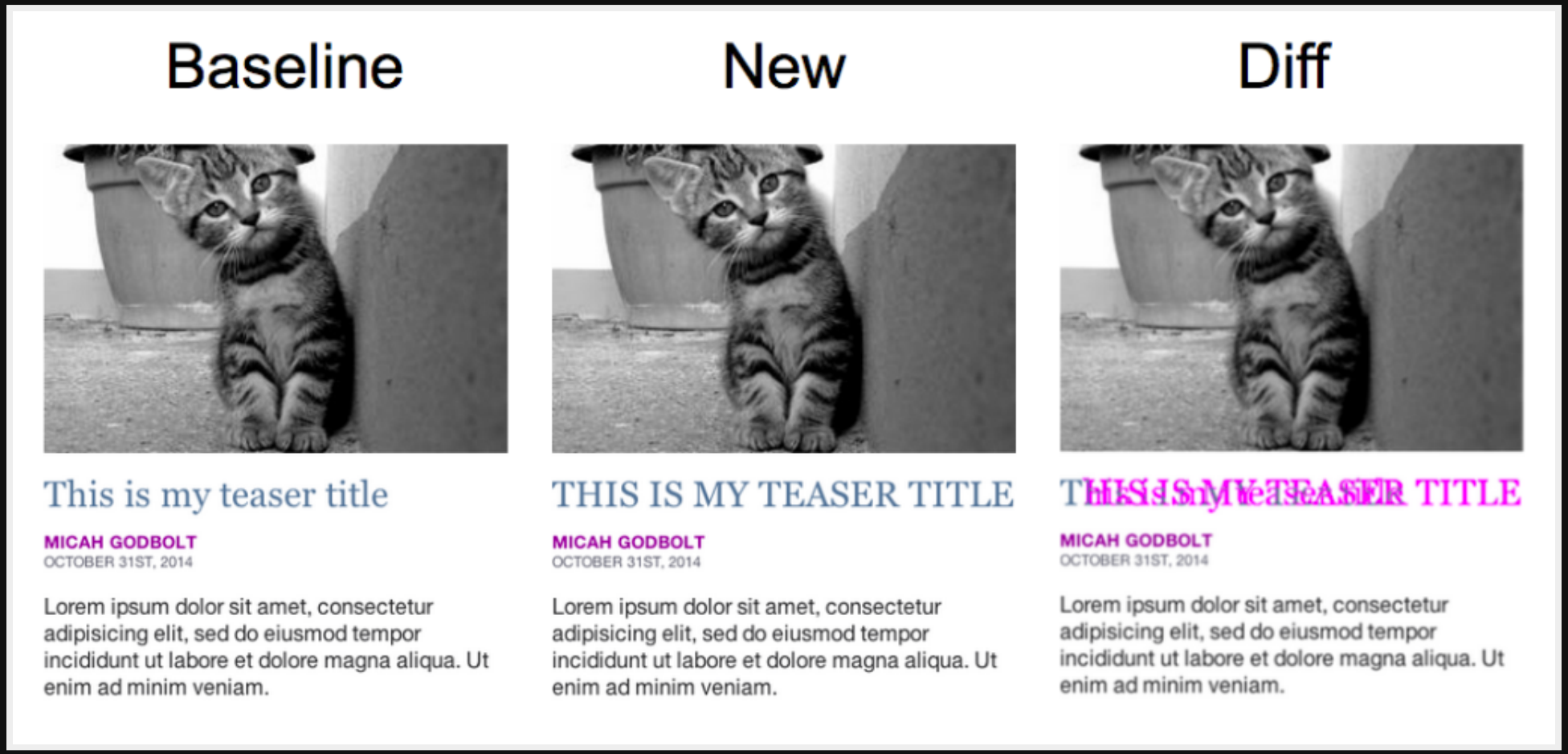


Find 5 differences...

...100 times



Visual regression test



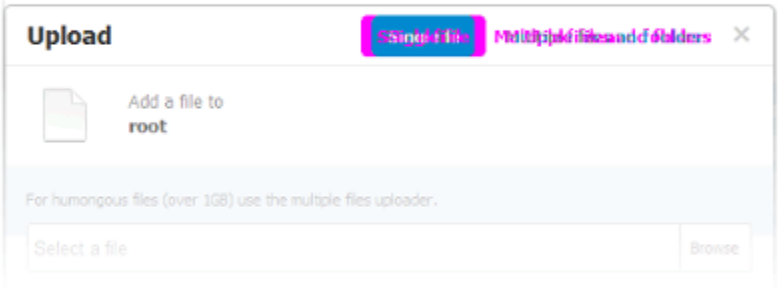
PhantomCSS

PhantomCSS

CSS regression testing. A CasperJS module for automating visual regression testing with PhantomJS 2 or SlimerJS and Resemble.js. For testing Web apps, live style guides and responsive layouts. Read more on Huddle's Engineering blog: CSS Regression Testing.

What?

PhantomCSS takes screenshots captured by CasperJS and compares them to baseline images using Resemble.js to test for rgb pixel differences. PhantomCSS then generates image diffs to help you find the cause.

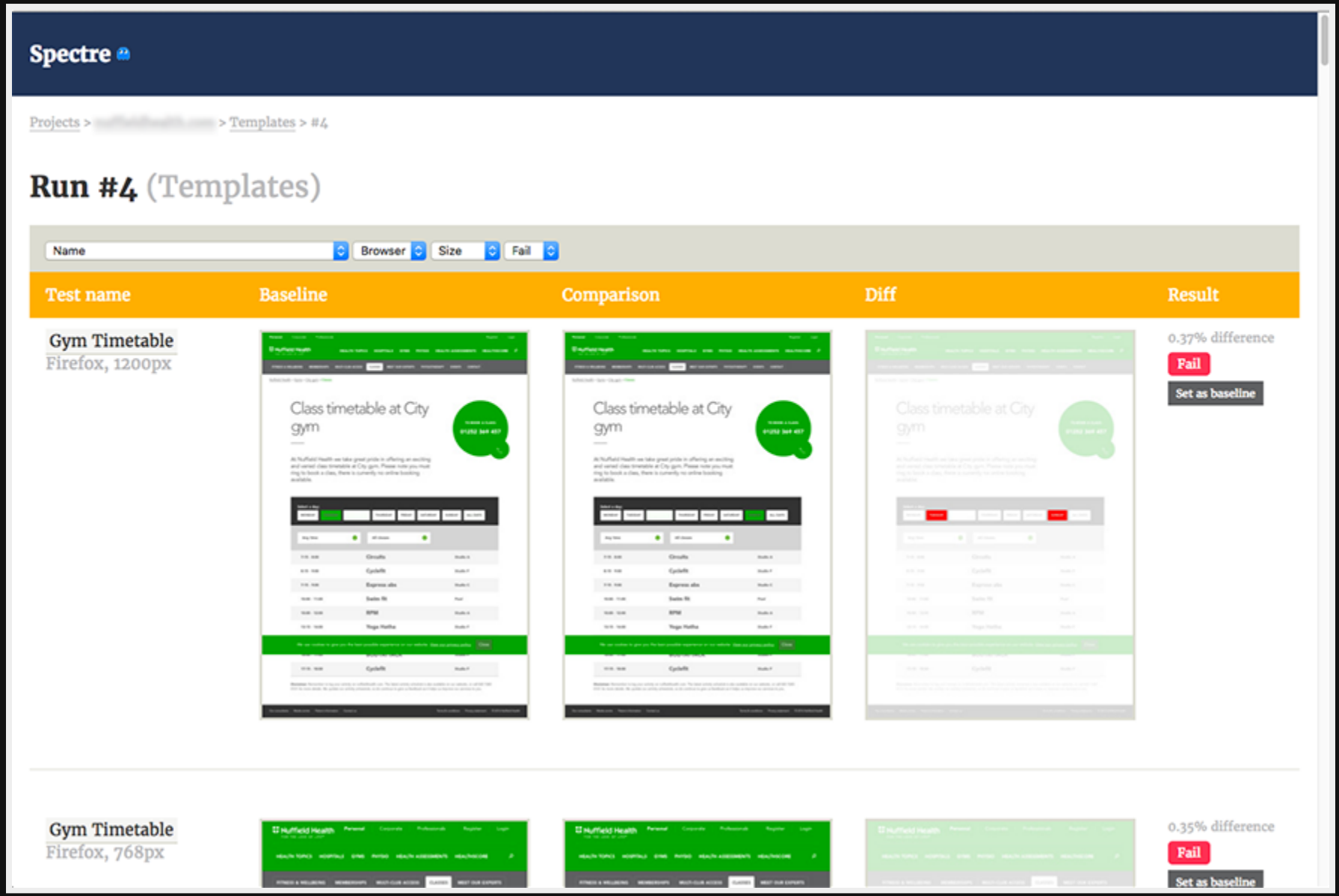


Screenshot based regression testing can only work when UI is predictable. It's possible to hide mutable UI components with PhantomCSS but it would be better to test static pages or drive the UI with faked data during test runs.

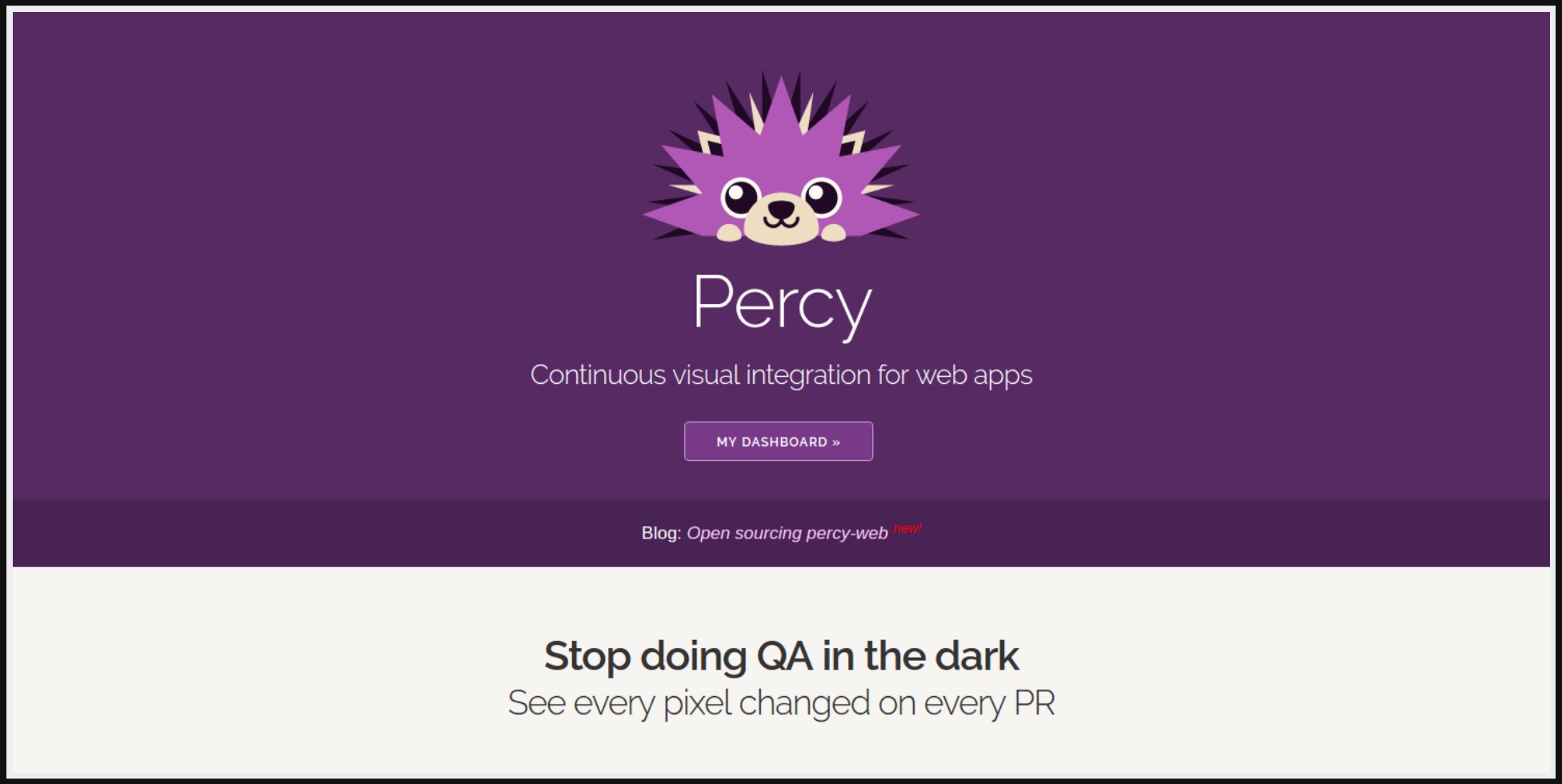
Example

```
casper.
    start( url ).
    then(function(){

        // do something
        casper.click('button#open-dialog');

        // Take a screenshot of the UI component
        phantomcss.screenshot('#the-dialog', 'a screenshot of my dialog');

    });
```

spectre


Spectre
Projects > [redacted] > Templates > #4
Run #4 (Templates)
Name
Browser
Size
Fail
Test name
Baseline
Comparison
Diff
Result
Gym Timetable
Firefox, 1200px
Class timetable at City gym
0.37% difference
Fail
Set as baseline
Gym Timetable
Firefox, 768px
0.35% difference
Fail
Set as baseline

percy.io


Percy
Continuous visual integration for web apps
MY DASHBOARD »
Blog: Open sourcing percy-web new!
Stop doing QA in the dark
See every pixel changed on every PR

Other tools

Needle (for Python)

dpxdt (for Python)

Challenges of visual regression testing:

Dynamic content
(changing number or random blog post)

Sharing tests with your team
(different OS, different graphic card, different configuration)

Writing maintainable CSS

OOCSS

BEM

SMACSS

Object Oriented CSS

1. Separate style from structure

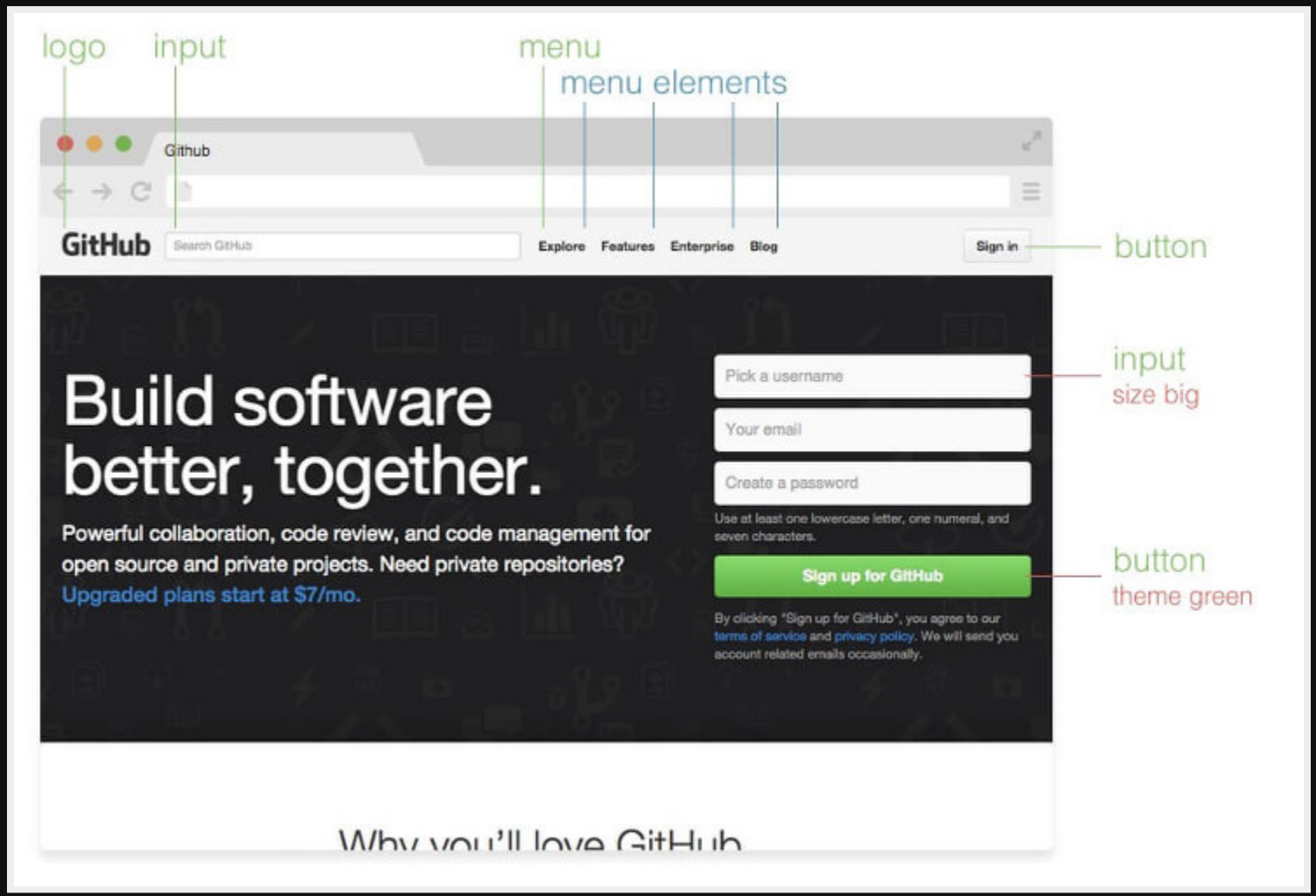
```
.button {
    width: ...;
    height: ...;
    overflow: ...;
}

...

.alert {
    border: ...;
    background: ...;
    color: ...;
}
```

Object Oriented CSS

2. Separate content from container

```
// BAD
.sidebar h3 {
    font-family: ...;
    font-size: ...;
}

// GOOD
.fancy-header {
    font-family: ...;
    font-size: ...;
}
```

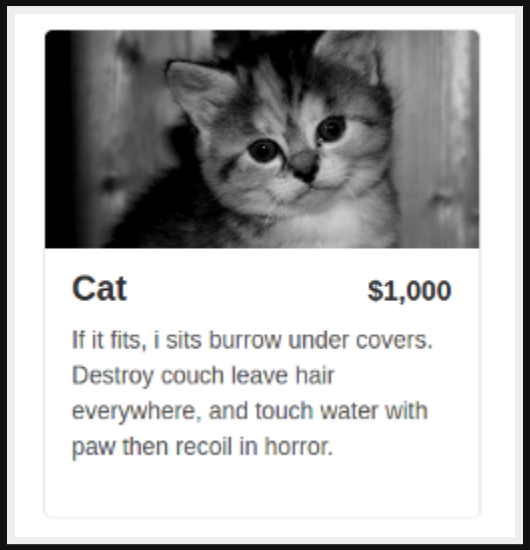
Block, Element, Modifier



Scalable and Modular Architecture for CSS (SMACSS)

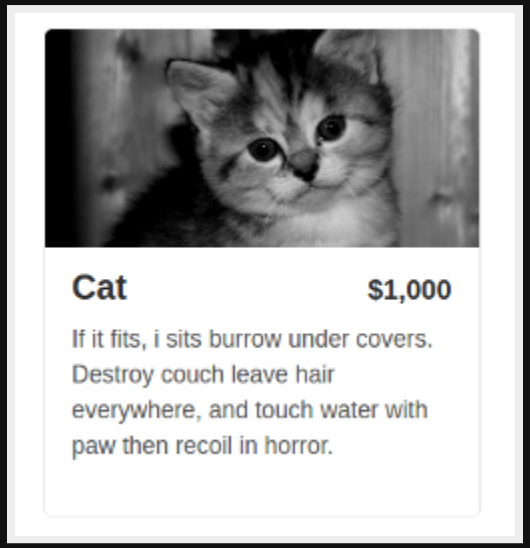
Base
Layout
Module
State
Theme

Functional CSS




```
<article class="br2 ba dark-gray b--black-10 mv4 w-100 w-50-m w-25-l mw5 center">
  <img src="http://placekitten.com/g/600/300" class="db w-100 br2 br--top">
  <div class="pa2 ph3-ns pb3-ns">
    <div class="dt w-100 mt1">
      <div class="dtc">
        <h1 class="f5 f4-ns mv0">Cat</h1>
      </div>
      <div class="dtc tr">
        <h2 class="f5 mv0">$1,000</h2>
      </div>
    </div>
    <p class="f6 lh-copy measure mt2 mid-gray">...</p>
  </div>
</article>
```

```
.ba { border-style: solid; border-width: 1px; }
.b--black-10 { border-color: rgba(0, 0, 0, .1); }
.br2 { border-radius: .25rem; }
.br--top { border-bottom-left-radius: 0; border-bottom-right-radius: 0; }
.db { display: block; }
.dt { display: table; }
.dtc { display: table-cell; }
.lh-copy { line-height: 1.6; }
.mw5 { max-width: 16rem; }
.w-100 { width: 100%; }
.dark-gray { color: #333; }
.mid-gray { color: #555; }
.pa2 { padding: .5rem; }
.mt1 { margin-top: .25rem; }
.mt2 { margin-top: .5rem; }
.mv0 { margin-top: 0; margin-bottom: 0; }
.mv4 { margin-top: 2rem; margin-bottom: 2rem; }
.tr { text-align: right; }
.f5 { font-size: 1rem; }
.f6 { font-size: .875rem; }
.measure { max-width: 30em; }
.center { margin-right: auto; margin-left: auto; }

@media screen and (min-width: 30em) {
  .pb3-ns { padding-bottom: 1rem; }
  .ph3-ns { padding-left: 1rem; padding-right: 1rem; }
  .f4-ns { font-size: 1.25rem; } }
@media screen and (min-width: 30em) and (max-width: 60em) {
  .w-50-m { width: 50%; } }
@media screen and (min-width: 60em) {
  .w-25-l { width: 25%; } }
```




```
<p class="f6">...</p>
```

↓

```
<p class="f5">...</p>
```

Functional CSS is:

Pure (no side effects)

Composable

Immutable

Transparent (rem vs em)

Frameworks

Tachyons

Basscss

The good parts

Modifying existing styling is easier

No more wrestling with the specificity

Very easy to add themes

Coding is faster

Encourages HTML components
(Tachyons components)

Encourages design consistency

Cache-friendly

Faster websites

The bad parts

"It's just replacing duplicated CSS rules with duplicated CSS classes!"

```
.table-list-triage {
    display: none;
}
.triage-mode .table-list-non-triage, .triage-mode .table-list-filters {
    display: none;
}
.boxed-group-list>li.approved .btn-sm, .boxed-group-list>li.rejected .btn-sm {
    display: none;
}
.repo-list .participation-graph.disabled {
    display: none;
}
.payment-methods .selected-payment-method {
    display: none;
}
.payment-methods.paypal-logged-in .paypal-sign-in {
    display: none;
}
.payment-methods.has-paypal-account .paypal-sign-in {
    display: none;
}
.currency-container .local-currency, .currency-container .local-currency-block {
    display: none;
}
.currency-container.open .default-currency {
    display: none;
}
.plan-chooser-repo-menu .price-label {
    display: none;
}
```

```
<p class="dn ...">...</p>
<div class="dn ...">...</div>
<li class="dn ...">...</li>
<div class="dn ...">...</div>
<span class="dn ...">...</span>
<article class="dn ...">...</article>
```

The bad parts

"It's just replacing duplicated CSS rules with duplicated CSS classes!"

Updating existing components can be a pain

Won't work with non-functional frameworks

No more writing CSS?

```
.pixel-perfect-button {
    font-size: 18px;
    margin: 47px;
    padding-top: 153px;
}
```

It's ok to write CSS

```
.modal {
    position: fixed;
    top: 50%;
    left: 50%;
    -webkit-transform: translate(-50%,-50%);
    -ms-transform: translate(-50%,-50%);
    transform: translate(-50%,-50%);
    *width: 600px;
    *margin-left: -300px;
    *top: 50px;
}
```

Functional CSS is not a silver bullet

CSS bloat vs HTML bloat

```
<div class="sidebar">...</div>
```

+

```
sidebar {
    float: left;
    margin-right: 10px;
    width: 25%;
}
```

vs.

```
<div class="float-left margin-right-1 width-25">...</div>
```

as

Thank you

Questions?