EXTENDS & SILENT CLASSES IN Sass 3.2

it’s like ninjas in your code
EXTEND & SILENT CLASSES
it's like ninjas in your code
OOCSS FRAMEWORKS
create ‘objects’ to streamline application?
OOCSS FRAMEWORKS
attempts to automate OOCSS

CSS frameworks have been using OOCSS methods since inception.

Where these fail is the insane amount of CSS selectors that by default are added to the site’s CSS.

From grid systems to UI elements, CSS bloat is a real issue.
OOCSS FRAMEWORKS attempts to automate OOCSS

CSS frameworks have been using OOCSS methods since inception. Where these fail is the insane amount of CSS selectors that by default are added to the site's CSS. From grid systems to UI elements, CSS bloat is a real issue.

Twitter’s bootstrap default includes 4914 lines of CSS.
OOCSS FRAMEWORKS

attempts to automate OOCSS

CSS frameworks have been using OOCSS methods since inception. Where these fail is the insane amount of CSS selectors that by default are added to the site's CSS. From grid systems to UI elements, CSS bloat is a real issue.

Twitter's bootstrap default includes 4914 lines of CSS.
DRY CODING

find the pattern and Don’t Repeat Yourself

9 out of 10 developers agree that using a DRY method of coding will save your bacon in the long run.
9 out of 10 developers agree that using a DRY method of coding will save your bacon in the long run.
OOCSS Teaches Us ...

create ‘objects’

- Identify reusable objects
- Be semantic w/HTML
- Minimize selectors
- Extend your classes
- ‘Style’ separate from content
- ‘Content’ separate from container
OOCSS TEACHES US ...

create ‘objects’

- Identify reusable objects
- Be semantic w/HTML
- Minimize selectors
- Extend your classes
- ‘Style’ separate from content
- ‘Content’ separate from container

what does this mean?
TWITTER'S BUTTONS

let's look at some code
OOCSS TEACHES US ...

create ‘objects’

```html
<ul class="nav nav-pills">
  <li class="dropdown">
    <a class="dropdown-toggle" data-toggle="dropdown" href="#">
      Dropdown
    </a>
    <ul class="dropdown-menu">
      <!-- links -->
    </ul>
  </li>
</ul>
```
OOCSS TEACHES US ...

create ‘objects’

<ul class="nav nav-pills">
    <li class="dropdown">
        <a class="dropdown-toggle" data-toggle="dropdown" href="#">
            Dropdown
            <b class="caret"></b>
        </a>
        <ul class="dropdown-menu">
            <!-- links -->
        </ul>
    </li>
</ul>

Markup with presentational classes
OOCSS TEACHES US ...

create ‘objects’

```html
<ul class="nav nav-pills">
<li class="dropdown">
  <a class="dropdown-toggle" data-toggle="dropdown" href="#">Dropdown
    <b class="caret"></b>
  </a>
  <ul class="dropdown-menu">
    <!-- links -->
  </ul>
</li>
</ul>
```

**here**

Markup with presentational classes
OOCSS TEACHES US ...

create ‘objects’

Markup with presentational classes
OOCSS Teaches Us ...

create 'objects'

```html
<ul class="nav nav-pills">
    <li class="dropdown">
        <a class="dropdown-toggle" data-toggle="dropdown" href="#">Dropdown<br<b class="caret"></b></a>
        <ul class="dropdown-menu">
            <!-- links -->
        </ul>
    </li>
</ul>
```

Markup with presentational classes
OOCSS TEACHES US ...

create ‘objects’

Markup with presentational classes
OOCSS TEACHES US ...

create ‘objects’

```html
<ul class="nav nav-pills">
  <li class="dropdown">
    <a class="dropdown-toggle" data-toggle="dropdown" href="#">
      Dropdown
      <b class="caret"></b>
    </a>
    <ul class="dropdown-menu">
      <!-- links -->
    </ul>
  </li>
</ul>
```

Markup with presentational classes
OOCSS TEACHES US ...

create ‘objects’

Markup with presentational classes
OOCSS TEACHES US ...

create ‘objects’

Markup with presentational classes
OOCSS TEACHES US ...

create ‘objects’

How much CSS can it really be?

It’s just for .nav-tabs and .nav-pills
Dude, that’s a lot of code!
.nav-tabs .open .dropdown-toggle,  
.nav-pills .open .dropdown-toggle, 
.nav > li.dropdown.open.active > a:hover {  
  color: #ffffff;  
  background-color: #999999;  
  border-color: #999999;  
}

.nav li.dropdown.open .caret,  
.nav li.dropdown.open.active .caret,  
.nav li.dropdown.open a:hover .caret {  
  border-top-color: #ffffff;  
  border-bottom-color: #ffffff;  
  opacity: 1;  
  filter: alpha(opacity=100);  
}
This makes baby kittens cry.

Dude, that’s a lot of code!

That’s line 169 right there.
Face it, CSS is NOT DRY!

This makes baby kittens cry

Dude, that’s a lot of code!

That’s line 169 right there.
Face it, CSS is NOT DRY!

This makes baby kittens cry

Dude, that's a lot of code!

10 references of #ffffff in this example alone

That's line 169 right there.
Face it, CSS is NOT DRY!

This makes baby kittens cry

Presentational classes, nothing semantic at all.

Dude, that’s a lot of code!

10 references of #ffffff in this example alone

That’s line 169 right there.

```
  color: #ffffff;
  background-color: #ffffff;
  border-color: #ffffff;
}

  border-top-color: #ffffff;
  border-bottom-color: #ffffff;
  color: #ffffff;
}
```
OOCSS BUTTONS

twitter bootstrap classes for buttons

.btn {
  display: inline-block;
  padding: 4px 10px 4px;
  margin-bottom: 0;
  margin-left: .3em;
  font-size: 13px;
  line-height: 18px;
  line-height: 20px;
  color: #333333;
  text-align: center;
  text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75);
  vertical-align: middle;
  cursor: pointer;
  background-color: #f5f5f5;
  background-color: #e6e6e6;
  background-image: -ms-linear-gradient(top, #ffffff, #e6e6e6);
  background-image: -webkit-gradient(linear, 0 0, 0 100%,
    from(#ffffff), to(#e6e6e6));
  background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6);
  background-image: -o-linear-gradient(top, #ffffff, #e6e6e6);
  background-image: linear-gradient(top, #ffffff, #e6e6e6);
  background-image: -moz-linear-gradient(top, #ffffff, #e6e6e6);
}

.btn:hover {
  background-position: 0 -15px;
  -webkit-transition: background-position 0.1s linear;
  -moz-transition: background-position 0.1s linear;
  -ms-transition: background-position 0.1s linear;
  transition: background-position 0.1s linear;
}

.btn:focus {
  outline: thin dotted #333;
  outline: 5px auto -webkit-focus-ring-color;
  outline-offset: -2px;
}

.btn.active,
.btn:active {
  filter: progid:dximagetransform.microsoft
    soft.gradient
    (startColorstr='#62c462',
    endColorstr='#51a351',
    GradientType=0);
  filter: progid:dximagetransform.microsoft
    soft.gradient(enabled=false);
}

.btn-success:hover,
.btn-success:active,
.btn-success.active,
.btn-success.disabled,
.btn-success[disabled] {
  background-color: #51a351;
  background-color: #499249;
}

.btn-success.active,
.btn-success.active {
  background-color: #408140 \9;
}

// -------------------------------

.btn-large {
  padding: 9px 14px;
  font-size: 15px;
  line-height: normal;
  -webkit-border-radius: 5px;
  -moz-border-radius: 5px;
  background-color: #e6e6e6;
  text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75);
  text-align: center;
  vertical-align: middle;
}

.btn-primary,
.btn-primary:hover,
.btn-primary.active,
.btn-primary.active:hover,
.btn-primary.active:active,
.btn-primary.active:active:hover {
  color: #ffffff;
  background-color: #51a351;
  background-color: #499249;
  text-shadow: 0 -15px 0 rgba(0, 0, 0, 0.25);
}

.btn-primary.active:hover,
.btn-primary.active:active,
.btn-primary.active:active:hover,
.btn-primary.active:active:active {
  color: #ffffff;
  text-shadow: 0 -15px 0 rgba(0, 0, 0, 0.25);
}

.btn-info:hover,
.btn-info:active,
.btn-info.active,
.btn-info.active:hover,
.btn-info.active:active,
.btn-info.active:active:hover {
  color: #ffffff;
  background-color: #5bb75b;
  background-color: #51a351;
  text-shadow: 0 -15px 0 rgba(0, 0, 0, 0.25);
}

.btn-info.active:hover,
.btn-info.active:active,
.btn-info.active:active:hover,
.btn-info.active:active:active {
  color: #ffffff;
  text-shadow: 0 -15px 0 rgba(0, 0, 0, 0.25);
}

.btn-warning:hover,
.btn-warning:active,
.btn-warning.active,
.btn-warning.active:hover,
.btn-warning.active:active,
.btn-warning.active:active:hover {
  color: #ffffff;
  background-color: #d9d9d9;
  background-color: #9e9e9e;
  text-shadow: 0 -15px 0 rgba(0, 0, 0, 0.25);
}

.btn-warning.active:hover,
.btn-warning.active:active,
.btn-warning.active:active:hover,
.btn-warning.active:active:active {
  color: #ffffff;
  text-shadow: 0 -15px 0 rgba(0, 0, 0, 0.25);
}

.btn-danger:hover,
.btn-danger:active,
.btn-danger.active,
.btn-danger.active:hover,
.btn-danger.active:active,
.btn-danger.active:active:hover {
  color: #ffffff;
  background-color: #b3b3b3;
  background-color: #9e9e9e;
  text-shadow: 0 -15px 0 rgba(0, 0, 0, 0.25);
}

.btn-danger.active:hover,
.btn-danger.active:active,
.btn-danger.active:active:hover,
.btn-danger.active:active:active {
  color: #ffffff;
  text-shadow: 0 -15px 0 rgba(0, 0, 0, 0.25);
}

.btn-inverse:hover,
.btn-inverse:active,
.btn-inverse.active,
.btn-inverse.active:hover,
.btn-inverse.active:active,
.btn-inverse.active:active:hover {
  color: #ffffff;
  background-color: #e6e6e6;
  background-color: #cccccc;
  text-shadow: 0 -15px 0 rgba(0, 0, 0, 0.25);
}

.btn-inverse.active:hover,
.btn-inverse.active:active,
.btn-inverse.active:active:hover,
.btn-inverse.active:active:active {
  color: #ffffff;
  text-shadow: 0 -15px 0 rgba(0, 0, 0, 0.25);
}
I’ve decided to update all the buttons.
WE WANT OUR CAKE
and eat it too
PRESENTATIONAL v SEMANTIC

looks good and have meaning too
PRESENTATIONAL vs SEMANTIC

CSS needs to convey design!
CSS needs to convey design!

CSS needs to have semantic meaning!
PRESENTATIONAL v SEMANTIC

looks good and have meaning too

Managing presentational and semantic classes with vanilla CSS is as about as hard as Ice Man getting a top 10 hit.

In other words ... IMPOSSIBLE!
Managing presentational and semantic classes with vanilla CSS is as about as hard as Ice Man getting a top 10 hit.

In other words ... IMPOSSIBLE!

That’s why I remodel homes YO!
PRESENTATIONAL v SEMANTIC

Managing presentational and semantic classes with vanilla CSS is about as hard as Ice Man getting a top 10 hit.

In other words ... IMPOSSIBLE!

That’s why I remodel homes YO!

No, really. Google it!
let’s look at a real ‘object’ solution
let's look at a real 'object' solution
let's look at a real ‘object’ solution
Sass 3.2 Silent Classes

take everything you know and set it on fire!
Article header

Pariatur Austin adipiscing, biodiesel voluptate beard four loko vegan sapiente retro readymade labore wes anderson irony. PBR cillum sriracha dolor, beard irony shoreditch laborum. Wayfarers gastropub nihil stumptown kogi. Artisan wolf aute cred. Readymade esse in four loko swag. Thundercats miksik dolor sriracha, semiotics gentrify authentic sustainable bicycle rights delectus typewriter esse tofu deserunt ethnic. Tattooed VHS sunt adipiscing qui, viral typewriter artisan pitchfork ad bicycle rights gentrify pork belly aliquip.

Read more
OOCSS
that's a lot of classes :0

<div class="main_content">
  <article class="post_summary post_border">
    <h1 class="post_header ">Article header</h1>
    <p>Pariatur Austin adipiscing, biodiesel voluptate beard four loko vegan sapiente retro readymade labore wes anderson irony. PBR cillum sriracha dolor, beard irony shoreditch laborum. Wayfarers gastropub nihil stumptown kogi. Artisan wolf aute cred. Readymade esse in four loko swag. Thundercats mlkshk dolor sriracha, semiotics gentrify authentic sustainable bicycle rights delectus typewriter esse tofu deserunt ethnic. Tattooed VHS sunt adipiscing qui, viral typewriter artisan pitchfork ad bicycle rights gentrify pork belly aliquip.</p>
    <a href="#" class="post_button primary_post_button">Read more</a>
  </article>
</div>
OOCSS

that's a lot of presentational CSS

.post_summary {
  width: 480 / 12 * 1em;
  box-sizing: border-box;
  padding: 2em;
  margin: 1em 10 / 16 * 1em 1em 0;
  box-shadow: 0 0 30px transparentize(#000, 0.5);
}

.post_border {
  border: 2px solid transparentize(#000, 0.7);
  border-radius: 15 / 16 * 1em;
}

.post_header {
  font-family: "Arial Narrow";
  font-weight: lighter;
}

.post_button {
  font-size: 1.5em;
  display: inline-block;
  font-weight: bold;
  color: white;
  text-shadow: 0 0 3px transparentize(#000, 0.5);
  padding: 10px;
  border-radius: 5px;
  border: 1px solid darken(#cdeb8e, 50%);
  text-decoration: none;
}

.primary_post_button {
  background: -webkit-linear-gradient(top, #cdeb8e 0%,#a5c956 100%);
  &:hover {
    background: -webkit-linear-gradient(top, lighten(#cdeb8e, 10%) 0%,lighten(#a5c956, 10%) 100%);
  }
  &:active {
    background: -webkit-linear-gradient(top, #a5c956 0%,#cdeb8e 100%);
  }
}
OOCSS

that’s a lot of presentational CSS

Re-using this gets complicated real fast.
OOCSS

that’s a lot of presentational CSS

Re-using this gets complicated real fast.

Extending classes like this in the DOM sucks!
<div class="main_content">
  <article class="post_summary">
    <h1>Article header</h1>
    <p>Pariatur Austin adipisicing, biodiesel voluptate beard four loko vegan sapiente retro readymade labore wes anderson irony. PBR cillum sriracha dolor, beard irony shoreditch laborum. Wayfarers gastropub nihil stumptown kogi. Artisan wolf aute cred. Readymade esse in four loko swag. Thundercats mlkshk dolor sriracha, semiotics gentrify authentic sustainable bicycle rights delectus typewriter esse tofu deserunt ethnic. Tattooed VHS sunt adipiscing qui, viral typewriter artisan pitchfork ad bicycle rights gentrify pork belly aliquip.</p>
    <a href="#">Read more</a>
  </article>
</div>
<div class="main_content">
<article class="post_summary">
  <h1>Article header</h1>
  <p>Pariatur Austin adipiscing, biodiesel voluptate beard four loko vegan sapiente retro readymade labore wes anderson irony. PBR cillum sriracha dolor, beard irony shoreditch laborum. Wayfarers gastropub nihil stumptown kogi. Artisan wolf aute cred. Readymade esse in four loko swag. Thundercats mlkshk dolor sriracha, semiotics gentrify authentic sustainable bicycle rights delectus typewriter esse tofu deserunt ethnic. Tattooed VHS sunt adipiscing qui, viral typewriter artisan pitchfork ad bicycle rights gentrify pork belly aliquip.</p>
  <a href="#">Read more</a>
</article>
</div>
<div class="main_content">
  <article class="post_summary">
    <h1>Article header</h1>
    <p>Pariatur Austin adipisicing, biodiesel voluptate beard four loko vegan sapiente retro readymade labore wes anderson irony. PBR cillum sriracha dolor, beard irony shoreditch laborum. Wayfarers gentrify authentic sustainable bicycle rights delectus typewriter esse tofu deserunt ethnic. Tattooed VHS sunt adipisicing qui, viral typewriter artisan pitchfork ad bicycle rights gentrify pork belly aliquip.</p>
    <a href="#">Read more</a>
  </article>
</div>
Conveys simple meaning about the block.

Any redesign hinges off of editing one class in the CSS

No need to edit the markup. That’s cool!

```html
<div class="main_content">
  <article class="post_summary">
    <h1>Article header</h1>
    <p>Pariatur Austin adipisicing, biodiesel voluptate beard four loko vegan sapiente retro readymade labore wes andersson color, beard irony shoreditch laborum. Wayfarers goose food craft beer Artisan wolf aute cred. Readymade esse in four loko sriracha, nose kogi, authentic sustainable bicycle rights sunt adipisicing qui, viral typewriter essent VHS sunt adipisicing qui, viral typewriter artisan pitchfork ad bicycle rights gentrify pork belly aliquip.</p>
    <a href="#">Read more</a>
  </article>
</div>
```
oo’s’css

@mixin is a cool tool ...

@mixin default_block {
  width: 480 / 12 * 1em;
  box-sizing: border-box;
  padding: 2em;
  margin: 1em 10 / 16 * 1em 1em 0;
  box-shadow: 0 0 30px transparentize(#000, 0.5);
  @extend %default_border;
}

@mixin default_border {
  border: 2px solid transparentize(#000, 0.7);
  border-radius: 15 / 16 * 1em;
}

@mixin post_header {
  font-family: "Arial Narrow";
  font-weight: lighter;
}

@mixin button {
  font-size: 1.5em;
  display: inline-block;
  font-weight: bold;
  color: white;
  text-shadow: 0 0 3px transparentize(#000, 0.5);
  padding: 10px;
  border-radius: 5px;
  border: 1px solid darken(#cdeb8e, 50%);
  text-decoration: none;
}

@mixin primary_button {
  background: -webkit-linear-gradient(top, #cdeb8e 0%, #a5c956 100%);
  &:hover {
    background: -webkit-linear-gradient(top, lighten(#cdeb8e, 10%) 0%, lighten(#a5c956, 10%) 100%);
  }
  &:active {
    background: -webkit-linear-gradient(top, #a5c956 0%, #cdeb8e 100%);
  }
}

.post_summary {
  @include .default_block;
  h1 {
    @include .post_header;
  }
  a {
    @include .button;
    @include .primary_button;
  }
}
We have combined all these ‘object’ mixins into a nice module.

@extend @mixin default_block;
width: 480 / 12 * 1em;
box-sizing: border-box;
padding: 2em;
margin: 1em 10 / 16 * 1em 1em 0;
box-shadow: 0 0 30px transparentize(#000, 0.5);

@extend @mixin default_border;
border: 2px solid transparentize(#000, 0.7);
border-radius: 15 / 16 * 1em;

@extend @mixin post_header;
font-family: "Arial Narrow";
font-weight: lighter;

@extend @mixin button;
font-size: 1.5em;
display: inline-block;
font-weight: bold;
color: white;
text-shadow: 0 0 3px transparentize(#000, 0.5);
padding: 10px;
border-radius: 5px;
border: 1px solid darken(#cdeb8e, 50%);
text-decoration: none;

@extend @mixin primary_button;
background: -webkit-linear-gradient(top, #cdeb8e 0%, #a5c956 100%);
&:hover {
  background: -webkit-linear-gradient(top, lighten(#cdeb8e, 10%) 0%, lighten(#a5c956, 10%) 100%);
}
&:active {
  background: -webkit-linear-gradient(top, #a5c956 0%, #cdeb8e 100%);
}

.post_summary {
  @include .default_block;
  h1 {
    @include .post_header;
  }
  a {
    @include .button;
    @include .primary_button;
  }
}
Mixins are never processed into CSS until used.

We have combined all these ‘object’ mixins into a nice module.
OO'SCSS
the robot will copy and paste

.post_summary {
    width: 40em;
    box-sizing: border-box;
    padding: 2em;
    margin: 1em 0.625em 1em 0;
    box-shadow: 0 0 30px rgba(0, 0, 0, 0.5);
    border: 2px solid rgba(0, 0, 0, 0.3);
    border-radius: 0.938em;
}
.post_summary h1 {
    font-family: "Arial Narrow";
    font-weight: lighter;
}
.post_summary a {
    font-size: 1.5em;
    display: inline-block;
    font-weight: bold;
    color: white;
    text-shadow: 0 0 3px rgba(0, 0, 0, 0.5);
    padding: 10px;
    border-radius: 5px;
    border: 1px solid #4c6812;
    text-decoration: none;
    background: -webkit-linear-gradient(top, #cdeb8e 0%, #a5c956 100%);
}
.post_summary a:hover {
    background: -webkit-linear-gradient(top, #e0f3b9 0%, #bad57d 100%);
}
.post_summary a:active {
    background: -webkit-linear-gradient(top, #a5c956 0%, #cdeb8e 100%);
}
This is really clean and looks awesome right?
This is really clean and looks awesome right?

But for reuse, this method will 100% duplicate all the CSS under a new selector

```css
.post_summary {
  width: 40em;
  box-sizing: border-box;
  padding: 10px;
  border-radius: 5px;
  border: 1px solid #4c6812;
  text-decoration: none;
  background: -webkit-linear-gradient(top, #cdeb8e 0%, #a5c956 100%);
}
.post_summary a:hover {
  background: -webkit-linear-gradient(top, #e0f3b9 0%, #bad57d 100%);
}
.post_summary a:active {
  background: -webkit-linear-gradient(top, #a5c956 0%, #cdeb8e 100%);
}
.post_summary a {
  font-size: 1.5em;
  display: inline-block;
  font-weight: bold;
  color: white;
}```
This is really clean and looks awesome.

But for reuse, this method will 100% duplicate all the CSS under a new selector.
@extend is pretty awesome ...

```css
.oo's.css

.default_block {
  width: 480 / 12 * 1em;
  box-sizing: border-box;
  padding: 2em;
  margin: 1em 10 / 16 * 1em 1em 0;
  box-shadow: 0 0 30px transparentize(#000, 0.5);
  @extend %default_border;
}

.default_border {
  border: 2px solid transparentize(#000, 0.7);
  border-radius: 15 / 16 * 1em;
}

.post_header {
  font-family: "Arial Narrow";
  font-weight: lighter;
}

.button {
  font-size: 1.5em;
  display: inline-block;
  font-weight: bold;
  color: white;
  text-shadow: 0 0 3px transparentize(#000, 0.5);
  padding: 10px;
  border-radius: 5px;
  border: 1px solid darken(#cdeb8e, 50%);
  text-decoration: none;
}

.primary_button {
  background: -webkit-linear-gradient(top, #cdeb8e 0%, #a5c956 100%);
  &:hover {
    background: -webkit-linear-gradient(top, lighten(#cdeb8e, 10%) 0%, lighten(#a5c956, 10%) 100%);
    &:active {
      background: -webkit-linear-gradient(top, #a5c956 0%, #cdeb8e 100%);
    }
  }
}

.post_summary {
  @extend .default_block;
  h1 {
    @extend .post_header;
  }
  a {
    @extend .button;
    @extend .primary_button;
  }
}
```
@extend is pretty awesome...

We have combined all these 'object' classes into a nice module...
@extend is pretty awesome ...

... but all these classes will get processed into the CSS

We have combined all these ‘object’ classes into a nice module ...
OO’S’CSS

@extend is pretty awesome ...

... but all these classes will get processed into the CSS

We have combined all these ‘object’ classes into a nice module ...

so we have presentational classes for no reason but to be extended by semantic ones?
our processed CSS

```
.default_block, .post_summary {
    width: 40em;
    box-sizing: border-box;
    padding: 2em;
    margin: 1em 0.625em 1em 0;
    box-shadow: 0 0 30px rgba(0, 0, 0, 0.5);
}

.default_border, .default_block, .post_summary {
    border: 2px solid rgba(0, 0, 0, 0.3);
    border-radius: 0.938em;
}

.post_header, .post_summary h1 {
    font-family: "Arial Narrow";
    font-weight: lighter;
}

.button, .post_summary a {
    font-size: 1.5em;
    display: inline-block;
    font-weight: bold;
    color: white;
    text-shadow: 0 0 3px rgba(0, 0, 0, 0.5);
    padding: 10px;
    border-radius: 5px;
    border: 1px solid #4c6812;
    text-decoration: none;
}

.primary_button, .post_summary a {
    background: -webkit-linear-gradient(top, #cdeb8e 0%, #a5c956 100%);
}

.primary_button:hover, .post_summary a:hover {
    background: -webkit-linear-gradient(top, #e0f3b9 0%, #bad57d 100%);
}

.primary_button:active, .post_summary a:active {
    background: -webkit-linear-gradient(top, #a5c956 0%, #cdeb8e 100%);
}
```
.default_block, .post_summary {
  width: 40em;
  box-sizing: border-box;
  padding: 2em;
  margin: 1em 0.625em 1em 0;
  box-shadow: 0 0 30px rgba(0, 0, 0, 0.5);
}

.default_border, .default_block, .post_summary {
  border: 2px solid rgba(0, 0, 0, 0.3);
  border-radius: 0.938em;
}

.post_header, .post_summary h1 {
  font-family: "Arial Narrow";
  font-weight: lighter;
}

.button, .post_summary a {
  font-size: 1.5em;
  display: inline-block;
  font-weight: bold;
  color: white;
  text-shadow: 0 0 3px rgba(0, 0, 0, 0.5);
  padding: 10px;
  border-radius: 5px;
  border: 1px solid #4c6812;
  text-decoration: none;
}

.primary_button, .post_summary a {
  background: -webkit-linear-gradient(top, #cdeb8e 0%, #a5c956 100%);
}

.primary_button:hover, .post_summary a:hover {
  background: -webkit-linear-gradient(top, #e0f3b9 0%, #bad57d 100%);
}

.primary_button:active, .post_summary a:active {
  background: -webkit-linear-gradient(top, #a5c956 0%, #cdeb8e 100%);
OO’CSS class

our processed CSS

Semantic class

```css
.default_block, .post_summary {
  width: 40em;
  box-sizing: border-box;
  padding: 2em;
  margin: 1em 0.625em 1em 0;
  box-shadow: 0 0 30px rgba(0, 0, 0, 0.5);
}

.default_border, .default_block, .post_summary {
  border: 2px solid rgba(0, 0, 0, 0.3);
  border-radius: 0.938em;
}

.post_header, .post_summary h1 {
  font-family: "Arial Narrow";
  font-weight: lighter;
}

.button, .post_summary a {
  font-size: 1.5em;
  display: inline-block;
  font-weight: bold;
  color: white;
  text-shadow: 0 0 3px rgba(0, 0, 0, 0.5);
  padding: 10px;
  border-radius: 5px;
  border: 1px solid #4c6812;
  text-decoration: none;
}

.primary_button, .post_summary a {
  background: -webkit-linear-gradient(top, #cdeb8e 0%, #a5c956 100%);
}

.primary_button:hover, .post_summary a:hover {
  background: -webkit-linear-gradient(top, #e0f3b9 0%, #bad57d 100%);
}

.primary_button:active, .post_summary a:active {
  background: -webkit-linear-gradient(top, #a5c956 0%, #cdeb8e 100%);
}
```
OO'CSS class

our processed CSS

```
.default_block, .post_summary {
  width: 40em;
  box-sizing: border-box;
  padding: 2em;
  margin: 1em 0.625em 1em 0;
  box-shadow: 0 0 30px rgba(0, 0, 0, 0.5);
}

.button, .post_summary a {
  font-size: 1.5em;
  display: inline-block;
}
```
OO'S'CSS

our processed CSS

.OOCS class

Semantic class

.OOCS class

Semantic class
# 1 COMPLAINT

all these classes, they exist even if I don’t use them ;( 
w/SILENT CLASSES
they don’t exist until you want them to
OO'S'CSS
the way of the future!

%default_block {
  width: 480 / 12 * 1em;
  box-sizing: border-box;
  padding: 2em;
  margin: 1em 10 / 16 * 1em 1em 0;
  box-shadow: 0 0 30px transparentize(#000, 0.5);
  @extend %default_border;
}

%default_border {
  border: 2px solid transparentize(#000, 0.7);
  border-radius: 15 / 16 * 1em;
}

%post_header {
  font-family: "Arial Narrow";
  font-weight: lighter;
}

%button {
  font-size: 1.5em;
  display: inline-block;
  font-weight: bold;
  color: white;
  text-shadow: 0 0 3px transparentize(#000, 0.5);
  padding: 10px;
  border-radius: 5px;
  border: 1px solid darken(#cdeb8e, 50%);
  text-decoration: none;
}

%primary_button {
  background: -webkit-linear-gradient(top, #cdeb8e 0%, #a5c956 100%);
  &:hover {
    background: -webkit-linear-gradient(top, lighten(#cdeb8e, 10%) 0%, lighten(#a5c956, 10%) 100%);
  }
  &:active {
    background: -webkit-linear-gradient(top, #a5c956 0%, #cdeb8e 100%);
  }
}

.post_summary {
  @extend %default_block;
  h1 {
    @extend %post_header;
  }
  a {
    @extend %button;
    @extend %primary_button;
  }
}
% is all the magic
OO's 'CSS
the way of the future!

% is all the magic

None of these 'silent classes' will be in the CSS
% is all the magic

None of these ‘silent classes’ will be in the CSS

We have combined all these ‘object’ classes into a nice module.
OO'S'CSS
silent classes processed

.post_summary {
  width: 40em;
  box-sizing: border-box;
  padding: 2em;
  margin: 1em 0.625em 1em 0;
  box-shadow: 0 0 30px rgba(0, 0, 0, 0.5);
}

.post_summary {
  border: 2px solid rgba(0, 0, 0, 0.3);
  border-radius: 0.938em;
}

.post_summary h1 {
  font-family: "Arial Narrow";
  font-weight: lighter;
}

.post_summary a {
  font-size: 1.5em;
  display: inline-block;
  font-weight: bold;
  color: white;
  text-shadow: 0 0 3px rgba(0, 0, 0, 0.5);
  padding: 10px;
  border-radius: 5px;
  border: 1px solid #4c6812;
  text-decoration: none;
}

.post_summary a {
  background: -webkit-linear-gradient(top, #cdeb8e 0%, #a5c956 100%);
}

.post_summary a:hover {
  background: -webkit-linear-gradient(top, #e0f3b9 0%, #bad57d 100%);
}

.post_summary a:active {
  background: -webkit-linear-gradient(top, #a5c956 0%, #cdeb8e 100%);
}
looks much like the Mixin solution, but no styles will be duplicated
looks much like
the Mixin solution,
but no styles will
be duplicated

as styles are
reused, they will be
properly extended
in the CSS
looks much like the Mixin solution, but no styles will be duplicated

as styles are reused, they will be properly extended in the CSS

no unnecessary presentational classes created
looks much like the Mixin solution, but no styles will be duplicated

as styles are reused, they will be properly extended in the CSS

easily manageable and elegant solution from the CSS, not in the markup!

no unnecessary presentational classes created
looks much like the Mixin solution, but no styles will be duplicated as styles are reused, they will be properly extended in the CSS, no unnecessary presentational classes created easily manageable and elegant solution from the CSS, not in the markup!
SUMMARY

what have we learned?
Summary
what have we learned?

OOCSS UI frameworks will add some serious CSS bloat
what have we learned?

**SUMMARY**

- OOCSS UI frameworks will add some serious CSS bloat.
- Managing presentational CSS extending in the DOM has its issues.
OOCSS UI frameworks will add some serious CSS bloat

Managing presentational CSS extending in the DOM has its issues

Extending classes with 4914 lines of presentational CSS makes my brain hurt A LOT!
OOCSS UI frameworks will add some serious CSS bloat.

Managing presentational CSS extending in the DOM has its issues.

Extending classes with 4914 lines of presentational CSS makes my brain hurt a lot.

OO’S’CSS using a CSS processor will change your life!
OOCSS UI frameworks will add some serious CSS bloat.

Managing presentational CSS extending in the DOM has its issues.

Extending classes with 4914 lines of presentational CSS makes my brain hurt a lot!

OO’S’CSS using a CSS processor will change your life!

@mixin was a great start, but we can do it better.
OOCSS UI frameworks will add some serious CSS bloat

Managing presentational CSS extending in the DOM has its issues

Extending classes with 4914 lines of presentational CSS makes my brain hurt a lot!

OO’S’CSS using a CSS processor will change your life!

@extend is awesome, but we can do it better.

@mixin was a great start, but we can do it better.
SUMMARY
what have we learned?

OOCSS UI frameworks will add some serious CSS bloat.

Managing presentational CSS extending in the DOM has its issues.

Extending classes with 4914 lines of presentational CSS makes my brain hurt a lot!

OO’S’CSS using a CSS processor will change your life!

@mixin was a great start, but we can do it better.

@extend is awesome, but we can do it better.

Take everything you know AND SET IT ON FIRE!
Take everything you know AND SET IT ON FIRE!

Extending classes with 4914 lines of presentation CSS makes my brain hurt a lot!

@.mixin was a great start, but we can do it better.

@extend is awesome, but we can do it better.

OO'CSS using a CSS processor will change your life!

%foo

OOCSS UI frameworks will add some serious CSS bloat.

Managing presentational CSS extending in the DOM has its issues.
QUESTIONS?

you, in the front ...
http://speakerrate.com/speakers/15438

You don’t win anything, but it helps me to get better at this ;)}
EXTENDS & SILENT CLASSES IN Sass 3.2
it's like ninjas in your code

https://speakerdeck.com/u/anotheruiguy/p/sass-32-silent-classes