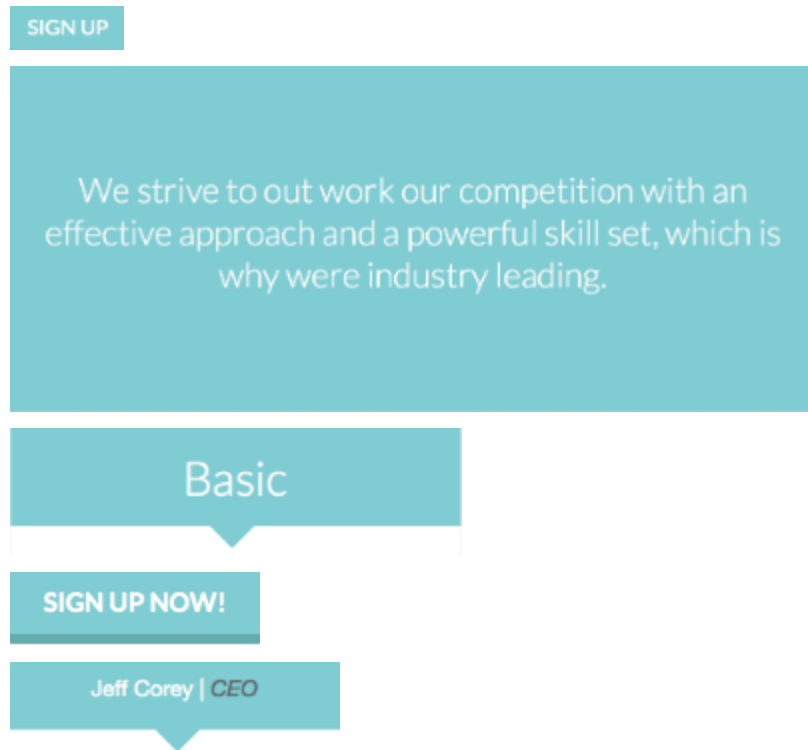


Patterns

Due: Next class

Grading: 20 points possible

This exercise focuses on the concept of “patterns” when translating visual design into HTML and CSS. The goal is to identify *common* visual patterns present in a design and then create CSS rules that can be reused throughout the site. From the Breakroom mockups, a simple example of identifying a visual pattern would be the following:



Though they differ in size and shape, these visual elements share the same background color and text color. This might be expressed in CSS using a functional class name like “contentbox-style” as the following:

```
.contentbox-style {  
  background-color: turquoise;  
  color: white;  
}
```

With Sass, these properties in common can be written using the “@mixin” directive and then inserted into CSS rules where needed.

Sass mixins

Sass mixins are “groups of CSS declarations that you want to reuse throughout your site.” The syntax for a mixin is “@mixin” followed by a *space* and then the *name* you give the mixin. So for the above example, a mixin called “contentbox-style” could be written like this:

```
@mixin contentbox-style() {
  background: $turquoise;
  color: $white;
}
```

You could then use this mixin by referencing it in your CSS using the “@include” directive:

```
.btn-primary {
  @include contentbox-style();
  &:hover, &:focus {
    background-color: darken($turquoise, 10%);
  }
}
```

For this exercise, your use of *mixins* feature will focus primarily on how sections of content are handled in Breakroom within the main body for each page, with the goal of keeping our CSS as DRY as possible!

As you follow this exercise, be sure to reference the breakroom screenshots for visual cues.

Exercise details:

- Create new mixins partial
- Add Sass mixins to SCSS files
- Add new classes to markup

Directions

Create mixins partial

1. Start off by launching Scout and pressing the play button.
2. Next, launch a web editor, create a new blank file and save it as “_mixins.scss” in your local scss folder.
3. In *_mixins.scss*, create a new mixin called “contentbox-style” with the following parameters and properties:

```
// Mixin for common background color and text color
@mixin contentbox-style($bgcolor, $color) {
  background-color: $bgcolor;
  color: $color;
  position: relative; // Reference point for absolutely positioning child element
}
```

Further along in the exercise, you will have access to this mixin using the following directive:

```
@include contentbox-style($bgcolor, $color);
```

Update master Sass file to include mixin partial

4. Next, open *style.scss* and add a reference to the mixins partial under “variables”. This placement is important as your mixins need to come first in the source code order before any Sass rules that reference them.

```
@import url(http://fonts.googleapis.com/css?family=Lato:100,300,400,700,900);  
@import "variables";  
@import "mixins";  
@import "typography";  
@import "patterns";
```

Pattern: contentbox

The first pattern you will create defines the markup and CSS for a basic “content box” in the main body area of Breakroom. There are a number of variations on content boxes but all are comprised of some variation of the following:

- Image
- Font Awesome icon
- Heading 2
- One or more paragraphs
- Links



Heading 2

Paragraph.lead - Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt.

Paragraph. Lorem ipsum dolor sit amet, **consectetur** adipisicing elit. Natus quis doloremque asperiores sunt non provident nostrum. Maiores ipsam ea non perspiciatis facilis tempore atque harum esse excepturi temporibus vitae dolorem.

[More](#)

Example content box pattern

- Open `_patterns.scss` and add a new class selector rule called “contentbox” with the following:

```
.contentbox {
  @include contentbox-style (transparent, $gray);

  h2 {
    margin-bottom: measure;
  }
  p {
    margin-bottom: measure;
  }
  a, .fa {
    color: $turquoise;
  }
}
```

Add .contentbox class to HTML partials

Home

- Open `home.html`. Within each of the 4 columns under the `.well` section, wrap the content with a new div and the `.contentbox` class.

```

<div class="col-sm-3">
  <div class="contentbox">
    <i class="fa fa-thumbs-o-up fa-4x"></i>
    <h2>Built w/ Bootstrap</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
      tempor incididunt.</p>
  </div>
</div>

```

About

- Open *about.html*. Within each of the 3 columns under the page header, wrap the content with a new div and the *.contentbox* class.

```

<div class="col-sm-6">
  <div class="contentbox">
    <h2>Our Skills</h2>
    <div>
      <span>Concepts</span>
      <span>60%</span>
    </div>
    ...
  </div>

```

Services

- Open *services.html*. Within each of the 4 columns under the page header section, wrap the content with a new div and the *.contentbox* class.
- Next, subdivide the content into 2 columns; place the FA icon into a column that spans 2 columns and the rest into a column that spans 10.

```

<div class="col-sm-6">
  <div class="contentbox">
    <div class="col-sm-2">
      <i class="fa fa-thumbs-o-up fa-4x"></i>
    </div>
    <div class="col-sm-10">
      <h2>Built w/ Bootstrap</h2>
      <p>Lorem ipsum dolor ...</p>
    </div>
  </div>

```

```

    </div>
  </div>

```

Blog

10. Open *blog.html*. Add the *.contentbox* class to each of the *article* tags.

```

<div class="col-sm-9">
  <article class="contentbox">
    ...

```

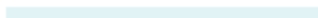
Pattern variation: contentbox-var1

The next pattern is just a variation on contentbox and an opportunity to use the Sass extend feature.



Heading 2

Lorem ipsum dolor sit
amet, consectetur
adipiscing elit, sed do
eiusmod tempor
incididunt.



11. Open *patterns.scss*, add the following CSS:

```

.contentbox-var1 {
  border-bottom: 5px solid lighten($turquoise, 20%);
  margin-top: measure;
  margin-bottom: measure;
}

```

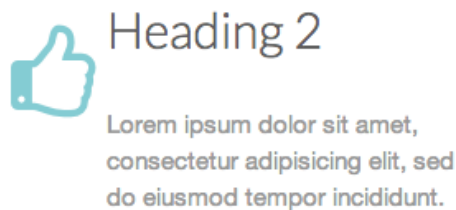
Note the use of the Sass [lighten function](#). A convenient way to introduce a value variation on an existing color.

12. Open *home.html*. Add the *.contentbox-var1* class after each instance of the *.contentbox* class.

```
<div class="col-sm-3">
  <div class="contentbox contentbox-var1">
    <i class="fa fa-thumbs-o-up fa-4x"></i>
    <h2>Built w/ Bootstrap</h2>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
      tempor incididunt.</p>
  </div>
</div>
```

Add new utility class

This example uses the *.contentbox* class but requires a few new classes to position the icon correctly. This is also an example of Bootstrap grid classes nested within grid classes.



13. In *patterns.scss*, add the the following:

```
.vertical-middle {
  position: absolute;
  left: 50%;
  top: .5em;
}

@media (max-width: 768px) {
  .vertical-middle {
    position: static;
  }
}
```

14. Open *services.html*. Add the *.vertical-middle* class after each instance of the *.fa* related classes.

```
<div class="col-sm-6">
  <div class="contentbox">
    <div class="col-sm-2">
```

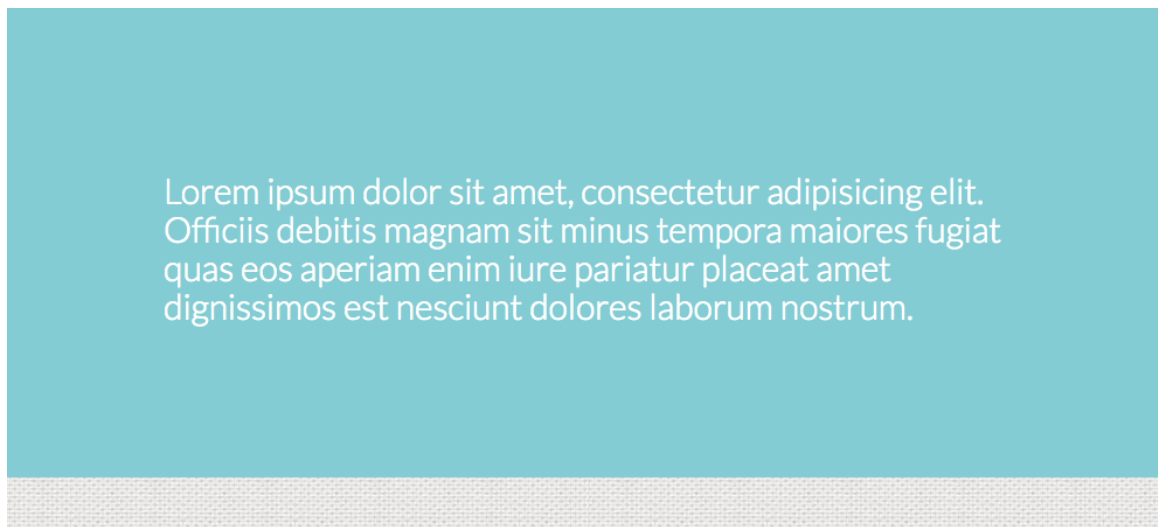
```

      <i class="fa fa-thumbs-o-up fa-4x vertical-middle"></i>
    </div>
    <div class="col-sm-10">
      <h2>Built w/ Bootstrap</h2>
      <p>Lorem ipsum dolor ...</p>
    </div>
  </div>
</div>

```

Pattern: message

Like the previous pattern, this last pattern is another one that stretches across the full width of the browser. Also note the image pattern at the bottom. This pattern is unique to Breakroom.

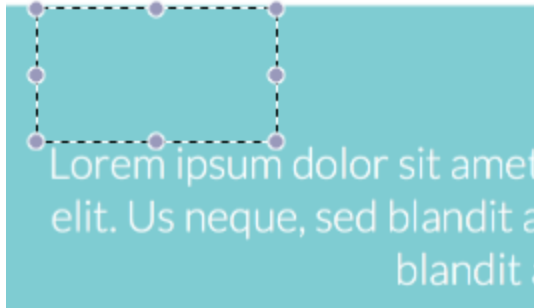


15. Open *patterns.scss*. add the following:

```

.message {
  @include contentbox-style($turquoise, $white);
  padding-bottom: measure;
  padding-bottom: measure;
}

```

```
.image-break {
  background: #fff url(../images/paper-texture.jpg) repeat-x;
  height: measure;
}
```

The background image that the *.image-break* class references should already be present in your images folder.



- Open *home.html*. Add a new empty div with the *.image-break* class directly after the *.message* section.

```
<section class="message">
  ...
</section>
<div class="image-break"></div>
```

- Add *.image-break* div to other HTML partials where *.message* section is present.

Pattern: titlebox

Like the previous pattern, this last pattern is another one that stretches across the full width of the

browser. Also note the image pattern at the bottom. This pattern is unique to Breakroom.



18. In *patterns.scss*, create a new class selector rule called “titlebox” with the following:

```
.titlebox {
  @include contentbox-style($turquoise, $white);
  padding: 10px;
  em {
    color: $black;
  }
}
```

19. In *patterns.scss*, create a new class selector rule called “titlebox-after” with the following:

```
// Creates the pointy bit
.titlebox:after {
  content: "";
  position: absolute;
  bottom: -15px;
  left: 44%;
  border-style: solid;
  border-width: 15px 15px 0;
  border-color: $turquoise transparent;
  display: block;
  width: 0;
  z-index: 1;
}
```

20. Open *about.html*. Add the *.titlebox* class to each *p* tag containing team member name.

```

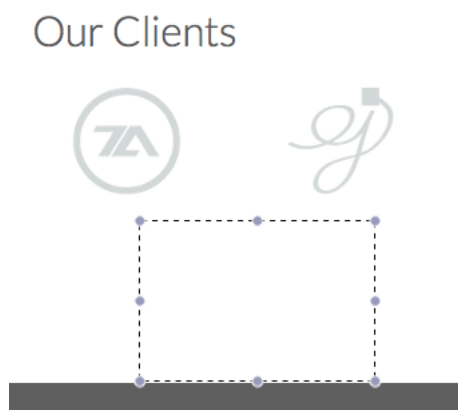
<p class="titlebox">Jeff Corey | <em>CEO</em></p>
<ul class="list-inline">
```

21. Open *services.html* and add the *.titlebox* class to each of the 3 *h2* tags in the services section.

```
<h2 class="titlebox">Basic</h2>
<ul class="list-unstyled">
<li>$99<span>/month</span>
```

22. In *patterns.scss*, create a new class selector rule called “clients” with the following:

```
.clients {
  padding-top: measure;
  padding-bottom: measure;
}
```



23. Open *home.html*. Add the *.clients* class to the *section* tag containing client logos.

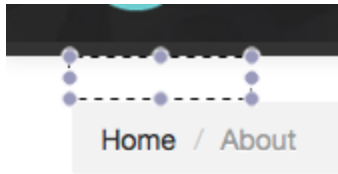
```
<section class="clients">
  <div class="container">
    <h2>Our Clients</h2>
    ...
```

24. Add *.clients* class to other HTML partials where relevant.

25. In *patterns.scss*, add the following update to the Bootstrap *.breadcrumb* class:

```
// Update to Bootstrap class
.breadcrumb {
```

```
margin-top: measure;
margin-bottom: measure;
}
```



Services page CSS

26. Add the following classes impacting the Services page:

```
// Services page
.pricing-table {
  border: 1px solid $lightGray;
  border-bottom: 7px solid $lightGray;
  margin-bottom: -10px;
  margin-bottom: 40px;

  li {
    color: $black;
    padding: 20px;
    border-bottom: 1px dotted $lightGray;
    font-weight: bold;
  }

  .price {
    font-size: 26px;
    border-bottom: 1px dotted $lightGray;
    margin-top: 0;
  }

  .info {
    font-weight: normal;
    font-size: 13px;
    color: $gray;
  }
}
```

```
.btn-price {
  @include contentbox-style($turquoise, $white);
  border-bottom: 5px solid darken($turquoise, 20%);
}
```

27. Add `.price-list` class to `ul` tag for each of the 3 unordered lists.

28. Add `.price` class to 1st list item for each of the 3 unordered lists.

29. Add “info” class to 2nd list item for each of three unordered lists.

30. Add “btn btn-price” classes to bottom for each of the 3 unordered lists.

```
<h2 class="titlebox">Basic</h2>
<ul class="list-unstyled price-list">
  <li class="price">$99/month</li>
  <li class="info">2 row section for ...</li>
  <li>Up To 25 Projects</li>
  <li>10GB Storage</li>
  <li>Other info</li>
  <li><a href="#" class="btn btn-price">SIGN UP NOW!</a></li>
</ul>
```

Blog page

31. In `patterns.scss`, create a new `.article-date` class with the following:

```
.article-date {
  @include contentbox-style($black, $turquoise);
  padding: 10px;

  h2 {
    font-size: measure;
    margin: 0;
  }

  h3 {
    color: $white;
  }
}
```

```

        margin: 0;
    }
}

```



32. Create a new *.sidebar-links* class with the following:

```

.sidebar-links {
  li {
    padding: 5px 0 5px 0;
  }
  a {
    color: $turquoise;
    text-decoration: none;
  }
}

```

33. Open *blog.html*. Wrap the date-related *h2* and *h3* with a new div and the *.article-date* class for each instance.

```

<div class="col-sm-2">
  <div class="article-date">
    <h2>15</h2>
    <h3>MAR</h3>
  </div>
</div>

```

34. Add *.sidebar-links* class to “Links” ul tag.

```

<h2>Links</h2>
<ul class="list-unstyled sidebar-links">

```

Contact page

35. In *patterns.scss*, add the following update to the Bootstrap *.form-control* class:

```
// Update to Bootstrap class
.form-control {
  background-color: $lightGray;
  margin-bottom: 10px;
}
```

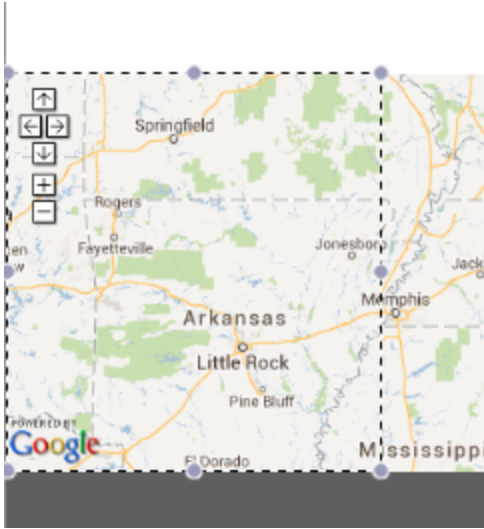
36. Add the following update to the Bootstrap *.btn-primary* button class:

```
// Update to Bootstrap class
.btn-primary {
  background-color: $black;
  border-color: $black;
  &:amp;:hover {
    background-color: $darkGray;
  }
}
```

37. Create a new *.map* class:

```
.map {
  height: measure;

  iframe {
    background-color: transparent;
    border: medium none;
    height: inherit;
    overflow: hidden;
    width: 100%;
  }
}
```



38. Open *contact.html*, and add a Bootstrap “form-control” class to each *input* and *textarea* tag.

```
<input type="text" placeholder="Your name" class="form-control" required>
```

39. Add Bootstrap “btn btn-primary pull-right” classes for submit button.

```
<button type="button" class="btn btn-primary pull-right">Submit</button>
```

40. Add “map” class to section tag wrapping Google map *iframe* tag.

```
<section class="map">
  <iframe src="https://maps.google.com/maps..."
```

Test using local server

41. Use Brackets preview to launch Chrome in a local server environment or by running an alternative local server.

Transfer project folder to web host

42. Use an FTP client like Fugu or Filezilla to transfer your project folder to your remote account.
You don’t need to upload your scss folder. That can stay on your local drive.

Test after uploading

43. Launch a web browser, go to your project URL, and make sure that your loads as expected.

Notify instructor when exercise is complete

44. Email me the URL to your remote *breakroom* project at chris.clark@mail.ccsf.edu.

For next class

- [Bootstrap Navbar](#) (familiarize yourself with it)
- [Bootstrap JavaScript](#) (we'll use the Carousel in particular)
- [CSS Animation](#) (Mozilla Developer Network)