

Style Conference Part V

We have already placed the background color on the `<body>` tag even though our website will have a few different rows of background colors. We have done this because the most frequent background color will be the dark purple-blueish color.

So, let's go ahead and within the `<header>` and `<footer>` elements let's make all our link colors start as white then, when we hover over them, they turn the same color as our heading. The first step in doing this is to add a class of `primary-header` to the `<header>` element (in addition to the existing container and group classes). Remember that this needs to be done across all of your HTML pages.

```
<header class="primary-header container group">
  ...
</header>
```

Switch over to the `main.css` file and under the `primary-footer` rule set add:

```
.primary-footer a {
  color: #fff;
}
.primary-footer a:hover {
  color: #a9b2b9;
}
```

```
.primary-footer a {
  color: #fff;
}
.primary-footer a:hover {
  color: #a9b2b9;
}
```

Under the primary header rule set add the following:

```
.primary-header a {
```

```
  color: #fff;
```

```
}
```

```
.primary-header a:hover {
```

```
  color: #a9b2b9;
```

```
}
```

```
1 .primary-header a {  
2   color: #fff;  
3 }  
4  
5 .primary-header a:hover {  
6   color: #a9b2b9;  
7 }
```

Next, let's make some of our text white. Within the hero section of our home page we will do this. Within the hero class selector these are the following styles we should have:

```
.hero {  
  color: #fff;  
  line-height: 44px;  
  padding: 22px 80px 66px 80px;  
  text-align: center;  
}
```

We're going to continue by cleaning up the button styles within the hero section of our home page. Start by setting the text color to be white, make sure the cursor is always pointer, increase the font-weight, add a small amount of letter-spacing, and change the text-transform to uppercase. The btn class rule set will now look like this.

```
.btn {  
  border-radius: 5px;  
  color: #fff;  
  cursor: pointer;  
  display: inline-block;  
  font-weight: 400;  
  letter-spacing: .5px;  
  margin: 0;  
  text-transform: uppercase;  
}
```

```
.btn {  
  border-radius: 5px;  
  color: #fff;  
  cursor: pointer;  
  display: inline-block;  
  font-weight: 400;  
  letter-spacing: .5px;  
  margin: 0;  
  text-transform: uppercase;  
}
```

Continue by updating the btn-alt and btn-alt: hover selectors as followed:


```
.btn-alt {  
  border: 1px solid #fff;  
  padding: 10px 30px;  
}  
  
.btn-alt: hover {  
  background: #fff;  
  color: #647188;  
}
```

```
.btn-alt {  
  border: 1px solid #fff;  
  padding: 10px 30px;  
}  
  
.btn-alt: hover {  
  background: #fff;  
  color: #648880;  
}
```

Let's create a new section within our main.css file for rows, just below the clearfix section. Within this new rows section, let's create a new class selector named row.

Within our new row class rule set, let's add a white background, a minimum width of 960 pixels (to make sure our row elements are always larger than the width of our container or grid elements), and some vertical padding. Altogether our new row section within our main.css file should look like this:

```
.row {  
    background: #fff;  
    min-width: 960px;  
    padding: 66px 0 44px 0;  
}
```



```
/*  
=====   
Rows  
=====   
*/  
  
.row {  
    background: #fff;  
    min-width: 960px;  
    padding: 66px 0 44px 0;  
}
```

With our row class styles in place, let's add a row with a white background to our home page. We'll do this on our teasers section. Currently this area has a `<section>` element with the class of grid wrapping three additional `<section>` elements with the classes of `teaser` and `col-1-3`.

To add a white background to this section, we're going to wrap all of these elements in an element with the class of `row`.

Because we'll want the entire teasers section wrapped in a `<section>` element, we're going to add a new `<section>` element with the class of `row` that surrounds the existing `<section>` element with the class of `grid`.

Having two `<section>` elements wrapping the exact same content diminishes semantic value. To correct this we'll change the second `<section>` element, the one with the class of `grid`, to a `<div>` element. After all, at this point this element is only adding styles, not semantic meaning, and is appropriate as a `<div>` element.

The structure of our new teasers element should look like this:

```
<section class="row">
  <div class="grid">

    <!-- Speakers -->

    <section class="teaser col-1-3">
      ...
    </section><!--

    Schedule

    --><section class="teaser col-1-3">
      ...
    </section><!--

    Venue

    --><section class="teaser col-1-3">
      ...
    </section>

  </div>
</section>
```

Next in our CSS we'll create a new row with a gradient background by using the class of row-alt. Because the new row will share the same min-width property and value as the row class selector, we'll combine these two selectors.

```
.row,
.row-alt {
  min-width: 960px;
}
```

Using the `linear-gradient()` function with the appropriate values and vendor prefixes, we'll add the gradient background to the `row-alt` class rule set. We'll also include a single background color before the gradient background as a fallback, just in case a browser doesn't support gradient backgrounds.

Lastly, we'll also add in some vertical padding. Our updated row section now looks like this:

```
.row,  
.row-alt{  
    min-width: 960px;  
}  
  
.row {  
    background: #fff;  
    padding: 66px 0 44px 0;  
}  
  
.row-alt {  
    background: #cbe2c1;  
    background: -webkit-linear-gradient(to right, #a1d3b0, #f6f1d3);  
    background: -moz-linear-gradient(to right, #a1d3b0, #f6f1d3);  
    background: linear-gradient(to right, #a1d3b0, #f6f1d3);  
    padding: 44px 0 22px 0;  
}
```

```
.row,  
.row-alt{  
    min-width: 960px;  
}  
  
.row {  
    background: #fff;  
    padding: 66px 0 44px 0;  
}  
  
.row-alt {  
    background: #cbe2c1;  
    background: -webkit-linear-gradient(to right, #a1d3b0, #f6f1d3);  
    background: -moz-linear-gradient(to right, #a1d3b0, #f6f1d3);  
    background: linear-gradient(to right, #a1d3b0, #f6f1d3);  
    padding: 44px 0 22px 0;  
}
```

With our row-alt styles in place, let's put them to use on all of our interior pages. Currently, all of our interior **(every page except for the home page)** pages have a <section> element with a class of container. Then, inside each <section> element is an <h1> element containing the heading of the page.

We're going to alter these <section> elements much like we did the teaser <section> element on our home page. We'll wrap each <section> element with a class of container in a <section> element with the class of row-alt. We'll then change each <section> element with a class of container to a <div> element for better semantic alignment.

Each of our interior pages should now include the following:

```
<section class="row-alt">
  <div class="container">

    <h1>...</h1>

  </div>
</section>
```

Because we are updating our interior pages, let's make their introductions, or leads, a little more appealing. We'll begin by adding a paragraph introducing each page just below the <h1> element in each <section> element with a class of row-alt. Our speakers.html page, for example, may now include the following lead section:

```
<section class="row-alt">
```

```
  <div class="container">
```

```
    <h1>Speakers</h1>
```

```
    <p>We're happy to welcome over twenty speakers to present on the industry's latest technologies. Prepare for an inspiration extravaganza.</p>
```

```
  </div>
```

```
</section>
```

```

<section class="row-alt">
  <div class="container">

    <h1>Speakers</h1>

    <p>We're happy to welcome over twenty speakers to
    present on the industry's latest technologies. Prepare for
    an inspiration extravaganza.</p>

  </div>
</section>

```

In addition to inserting the paragraph, let's also change some of the styles within the lead section. To do this, we'll add a class of lead to the `<div>` element that already has a class of container; this can be found nested directly inside the `<section>` element with a class of row-alt. Our lead section for each interior page will now look like this:

```

<section class="row-alt">
  <div class="lead container">

    ...

  </div>
</section>

```

Once the lead class is in place, we'll center all of the text within these `<div>` elements. We'll also increase the font-size and line-height of any paragraphs within these `<div>` elements.

We'll create a new section for leads within our main.css file, just below the typography section, and add the following styles:

```

1  /*
2    =====
3    Leads
4    =====
5  */
6
7  .lead {
8    text-align: center;
9  }
10 .lead p {
11   font-size: 21px;
12   line-height: 33px;
13 }
14

```