

DATEBOX

Tips for building a design system

Tips for building a
design system
or pattern library
or styleguide

A design system
is dynamic

The first thing you
should build is a
process for how to
make changes

**Prioritize it as a
product or it will die**

Control your colors

\$variable hell

```
$brand-primary: #69bec0;  
$brand-primary-alt: #87e5e7;  
$brand-primary-alt-light: #69bec0;  
$brand-secondary: #69bec0;  
$brand-secondary-alt: #87e5e7;  
$brand-secondary-alt-light: #87e5e7;
```



```
$colors: (  
  failure: (  
    lighter: #fbeae5,  
    light: #fead9a,  
    base: #ed6347,  
    dark: #bf0711,  
    darker: #330101,  
    text: #583c35  
  ),  
  success: (  
    lighter: #e3f1df,  
    light: #bbe5b3,  
    base: #50b83c,  
    dark: #108043,  
    darker: #173630,  
    text: #414f3e  
  )  
);
```

```
@function color($hue, $value: base) {  
  return map-get( ... );  
}  
  
.my-class {  
  color: color( success );  
}  
  
@each $color in $colors {  
  .btn--#{ $color } {  
    background-color: color( success, light );  
    color: color( success, text );  
  }  
}
```

Separate your
spacing

This is a nice heading



margin-bottom: 1.5rem



This is a nice heading

HOW ABOUT A SUBHEADING



```
$spacing_scale: (  
  '0': 0rem,  
  '1': .5rem,  
  '2': 1rem,  
  '3': 2rem,  
  '4': 4rem,  
  '5': 8rem,  
  '6': 16rem,  
  '7': 32rem  
);
```

```
@for $i from 0 through 7 {  
  // margin-bottom  
  .mb#{$i} {  
    margin-bottom: map-get($spacing_scale, #{$i});  
  }  
  
  // margin-top  
  .mt#{$i} {  
    margin-bottom: map-get($spacing_scale, #{$i});  
  }  
}
```

```
<h1 class="mb1">Heading</h1>  
<h2 class="mb2">Subheading</h2>
```

**Give your teammates
the best tools to
do their job**

For Developers:
DOCUMENT EVERYTHING

For Designers:
PROVIDE WORKING CODE

Thanks!

@tywayne