

# 35-short-Ink video

## Button Styles

- lets start syling paragraphs first in our \_base.scss

```
*{  
  padding: 0;  
  margin: 0;  
}  
  
html {  
  font-size: 62.5%;  
}  
  
body {  
  font-family: Helvetica, Arial, sans-serif;  
  font-size: $base-font-size;  
}  
  
p{  
  margin-bottom: $space;  
}  
  
h1 {  
  font-weight: 300;  
  margin-bottom: $space;  
}
```

- lets now create a new component under styles/components, it'll be a partial \_button.scss
- lets first import this new file in our \_main.scss

```
@import './variables';  
@import './base';  
@import './components/boxed-view';  
@import './components/button';
```

- **now lets** start styling the \_button.scss

```
.button {
  background: #5b4681;
}
```

- now lets add our class to the code in Login.js

```

}
render(){
  return (
    <div className = "boxed-view">
      <div className = "boxed-view__box">
        <h1>Short Lnk</h1>

        {this.state.error ? <p>{this.state.error}</p>: undefined}

        <form onSubmit={this.onSubmit.bind(this)} noValidate className="boxed-view__
form">

          <input type="email" ref="email" name="email" placeholder="Email"/>
          <input type="password" ref="password" name="password"
placeholder="Password"/>

          <button className="button">Create Account</button>

```

- lets keep styling \_button.scss

```

button {
  background: #5b4681;
  border: none;
  color: white;
  cursor: pointer;
  font-size: $base-font-size;
  margin-bottom: $space;
  padding: 1rem;
  text-transform: uppercase;
}
```

- now lets go to \_variables.scss and make another section

```

//color
$grey: #dddddd;

//spacing
$space: 1.4rem;

// Font-size
$base-font-size: $space;

//form inputs
$input-border: $grey;

// Boxed view
$boxed-view-overlay-bg: $grey;
$boxed-view-bg: white;

//button
$button-bg: #5b4681;

```

- **now** lets implement that variable to the \_button.scss
- and lets also change the button color

```

button {
  background: $button-bg;
  border: none;
  color: $button-color;
  cursor: pointer;
  font-size: $base-font-size;
  margin-bottom: $space;
  padding: 1rem;
  text-transform: uppercase;
}

```

- and create that variable in \_variables

```

//color
$grey: #dddddd;

```

```
//spacing
$space: 1.4rem;

// Font-size
$base-font-size: $space;

//form inputs
$input-border: $grey;

// Boxed view
$boxed-view-overlay-bg: $grey;
$boxed-view-bg: white;

//button
$button-bg: #5b4681;
$button-color: white;
```

- **lets** create another variable

```
//color
$grey: #dddddd;

//spacing
$space: 1.4rem;

// Font-size
$base-font-size: $space;

//form inputs
$input-border: $grey;
$input-spacing: 1rem;

// Boxed view
$boxed-view-overlay-bg: $grey;
$boxed-view-bg: white;
```

```
//button  
  
$button-bg: #5b4681;  
  
$button-color: white;
```

- now lets implement that variable in \_button.scss

```
button {  
    background: $button-bg;  
    border: none;  
    color: $button-color;  
    cursor: pointer;  
    font-size: $base-font-size;  
    margin-bottom: $space;  
    padding: $input-spacing;  
    text-transform: uppercase;  
}
```

- and also in \_base.scss

```
*{  
    padding: 0;  
    margin: 0;  
}  
  
html {  
    font-size: 62.5%;  
}  
  
body {  
    font-family: Helvetica, Arial, sans-serif;  
    font-size: $base-font-size;  
}  
  
p{  
    margin-bottom: $space;  
}  
  
h1 {
```

```

    font-weight: 300;

    margin-bottom: $space;
}

a {
    color: black;
}

input[type=text], input[type=email], input[type=password] {
    border: 1px solid $input-border;
    font-size: $base-font-size;
    margin-bottom: $space;
    padding: $input-spacing;
}

```

- lets now add a modifier unto \_button.scss to style the link
- first lets add it to the button in NotFound.js

```

import React from 'react';
import { Link } from 'react-router';

export default ()=>{
    return (
        <div className = "boxed-view">
            <div className = "boxed-view__box">
                <h1>Page Not Found</h1>
                <p>Hmmm, we're unable to find that page.</p>
                <Link to="/" className="button button--link">HEAD HOME</Link>
            </div>
        </div>
    );
}

```

- now lets give some style to that link in the \_button.scss
- first by default links are inline, so we want to change it to inline-block

```

.button {
    background: $button-bg;
}

```

```

border: none;
color: $button-color;
cursor: pointer;
font-size: $base-font-size;
margin-bottom: $space;
padding: $input-spacing;
text-transform: uppercase;
}

.button--link {
display:inline-block;
text-decoration: none;
}

```

- **now lets** implement the .button class to the button in Signup.js

```

render(){
  return (
    <div className = "boxed-view">
      <div className = "boxed-view__box">
        <h1>Join Short Lnk</h1>

        {this.state.error ? <p>{this.state.error}</p>: undefined}

        <form onSubmit={this.onSubmit.bind(this)} noValidate className="boxed-view__
form">

          <input type="email" ref="email" name="email" placeholder="Email"/>
          <input type="password" ref="password" name="password" placeholder="Passw
ord"/>

          <button className="button">Create Account</button>

        </form>

        <Link to="/">Already have an account?</Link>

      </div>
    </div>
  )
}

```

```

    );
  }

```

- lets now login to see the buttons in our page
- lets first start by adding our button class to AddLink.js

```

render(){
  return(
    <div>
      <button className="button" onClick={() => this.setState({isOpen: true})}>+ Add Li
nk</button>

      <Modal isOpen={this.state.isOpen} contentLabel="Add link" onAfterOpen=
{()=>this.refs.url.focus()}
      onRequestClose={this.handleModalClose.bind(this)}>
        <h1>Add Link</h1>
        {this.state.error ? <p>{this.state.error}</p> : undefined }
        <form onSubmit={this.onSubmit.bind(this)}>
          <input ref="url" type="text" placeholder="URL" value={this.state.url} on
Change={this.onChange.bind(this)}>
          <button>Add Link</button>
        </form>
        <button onClick={this.handleModalClose.bind(this)}>Cancel</button>
      </Modal>
    </div>
  );
}

```

- now we are going to style the rest of the buttons, lets go to \_button first to create that class
- lets first implement that class to our components inside of LinksListItem.js

```

render(){
  return(
    <div>
      <p>{this.props.url}</p>
      <p>{this.props.shortUrl}</p>
      <p>{this.props.visible.toString()}</p>
      {this.renderStats()}
    </div>
  );
}

```



```

        <a className="button button--pill button--link" href={this.props.shortUrl} target=
t= "_blank">
            Visit
        </a>
        <button className="button button--pill" ref="copy" data-clipboard-text={this.pro
ps.shortUrl}>
            {this.state.justCopied ? 'Copied' : 'Copy'}
        </button>
        <button className="button button--pill" onClick={()=>{
            Meteor.call('links.setVisibility', this.props._id, !this.props.visible);
        }}>{this.props.visible ? 'Hide' : 'Unhide'}</button>
    </div>
)
}
}

```

- now lets start adding some style to button--link in \_button.scss

```

.button {
    background: $button-bg;
    border: none;
    color: $button-color;
    cursor: pointer;
    font-size: $base-font-size;
    margin-bottom: $space;
    padding: $input-spacing;
    text-transform: uppercase;
}

.button--link {
    display: inline-block;
    text-decoration: none;
}

.button--pill {
    background-color: transparent;
}

```

```
border: 1px solid $button-pill-color;
color: $button-pill-color;
}
```

- **now** lets create that variable in \_variables.scss

```
//color
$grey: #dddddd;

//spacing
$space: 1.4rem;

// Font-size
$base-font-size: $space;

//form inputs
$input-border: $grey;
$input-spacing: 1rem;

// Boxed view
$boxed-view-overlay-bg: $grey;
$boxed-view-bg: white;

//button
$button-bg: #5b4681;
$button-color: white;
$button-pill-color: #5b4681
```

- **now** lets take care of the padding in \_button.scss

```
.button {
  background: $button-bg;
  border: none;
  color: $button-color;
  cursor: pointer;
  font-size: $base-font-size;
  margin-bottom: $space;
  padding: $input-spacing;
```

```

    text-transform: uppercase;
}

.button--link {
    display: inline-block;
    text-decoration: none;
}

.button--pill {
    background-color: transparent;
    border: 1px solid $button-pill-color;
    color: $button-pill-color;
    padding: .3rem .8rem;
}

```

- now lets take care the button line height and give it a 1.2

```

.button {
    background: $button-bg;
    border: none;
    color: $button-color;
    cursor: pointer;
    font-size: $base-font-size;
    line-height: 1.2;
    margin-bottom: $space;
    padding: $input-spacing;
    text-transform: uppercase;
}

```

-