

152-Styling the Editor

video

- lets make a new variable in the _variables.scss partial

```
//color

$brand-primary: #397ab1;

$grey: #dddddd;

$light-grey: #f9f9f9;

$dark-grey: #777777;


//Body

$body-background-color: #fafafa;

$body-default-color: #333333;


//spacing

$space: 1.4rem;

$large-space: 2.8rem;

$site-max-width: 100rem;


// Font-size

$base-font-size: $space;

$large-font-size : 1.8rem;
```

- now go to _space-content.scss

```
.page-content {

    display: flex;

    height: $page-content-height;

    max-width: $site-max-width;

    margin: 0 auto;

    padding: $large-space $space;

}
```

```
.page-content__sidebar{  
    display: flex;  
    width: $page-content-sidebar-width;  
    padding-right: $large-space;  
}  
  
.page-content__main{  
    display: flex;  
    width: $page-content-main-width;  
}
```

- now we are going to focus on the `_editor.scss` partial, we are going to enable flexbox and align our elements vertically

```
.editor {  
    background-color: white;  
    border: 1px solid $grey;  
    display: flex;  
    flex-direction: column;  
    padding: $large-space;  
    width: 100%;  
}
```

- lets now create new element selector in `_editor.scss`

```
.editor {  
    background-color: white;  
    border: 1px solid $grey;  
    display: flex;  
    flex-direction: column;  
    padding: $large-space;  
    width: 100%;  
}
```

```
.editor__title {

}

.editor__body {

}
```

- now lets go setup these classes on the components - go to Editor.js

```
render(){
  if(this.props.note){
    return (
      <div className="editor">
        <input className="editor__title" value={this.state.title} placeholder="Untitled" onChange={this.handleTitleChange.bind(this)}>
        <textarea className="editor__body" value={this.state.body} placeholder="Your note here" onChange={this.handleBodyChange.bind(this)}></textarea>
        <button onClick={this.handleRemoval.bind(this)}>Delete note</button>
      </div>
    );
  }else{
    return (
      <div className="editor">
        <p>
          {this.props.selectedNoteId ? 'Note not found' : 'Pick or create a note to get started'}
        </p>
      </div>
    );
  }
};
```

- and for our button we are going to wrap it in a div, and then style the div

```
render(){
  if(this.props.note){
    return (
      <div className="editor">
        <input className="editor__title" value={this.state.title} placeholder="Untitled" onChange={this.handleTitleChange.bind(this)} />
        <textarea className="editor__body" value={this.state.body} placeholder="Your note here" onChange={this.handleBodyChange.bind(this)} />
        <div>
          <button onClick={this.handleRemoval.bind(this)}>Delete note</button>
        </div>
      </div>
    );
  }else{
    return (
      <div className="editor">
        <p>
          {this.props.selectedNoteId ? 'Note not found' : 'Pick or create a note to get started'}
        </p>
      </div>
    );
  }
};
```

- now go back to _editor.scss partial to start styling our title and body

```
.editor {
  background-color: white;
  border: 1px solid $grey;
  display: flex;
```

```

    flex-direction: column;

    padding: $large-space;

    width: 100%;
}

.editor__title {

    border: none;

    border-bottom: 2px solid $grey;
}

.editor__body {

}

```

- we next want to increase the font size of the title, so lets go to the _variables.scss partial and increase our \$large-font-size and you use this variable

```

//spacing

$space: 1.4rem;

$large-space: 2.8rem;

$site-max-width: 100rem;


// Font-size

$base-font-size: $space;

$large-font-size : 2.2rem;


//form inputs

$input-border: $grey;

$input-spacing: 1rem;

```

- now go back to _editor.scss and add this variable and keep styling our title

```

.editor {

    background-color: white;

```

```

border: 1px solid $grey;

display: flex;

flex-direction: column;

padding: $large-space;

width: 100%;
}

.editor__title {

border: none;

border-bottom: 2px solid $grey;

font-size: $large-font-size;

margin-bottom: $large-space;

outline: none;

padding: $input-spacing;
}

.editor__body {

}

```

- now lets style the body of the editor, we are going to be using flex grow so it can be adjustable - it will take all leftover space

```

.editor {

background-color: white;

border: 1px solid $grey;

display: flex;

flex-direction: column;

padding: $large-space;

width: 100%;
}

.editor__title {

```

```

border: none;

border-bottom: 2px solid $grey;

font-size: $large-font-size;

margin-bottom: $large-space;

outline: none;

padding: $input-spacing;
}

.editor__body {

  flex-grow: 1;
}

```

- lets continue styling the editors body, the only thing here that is new for me is resize, which is the little hand grabbing to resize a textarea, we are going to set it to none

```

.editor {

  background-color: white;

  border: 1px solid $grey;

  display: flex;

  flex-direction: column;

  padding: $large-space;

  width: 100%;
}

.editor__title {

  border: none;

  border-bottom: 2px solid $grey;

  font-size: $large-font-size;

  margin-bottom: $large-space;

  outline: none;

  padding: $input-spacing;
}

```

```
.editor__body {
  flex-grow: 1;

  font-weight: 300;

  margin-bottom: $large-space;

  outline: none;

  padding: $input-spacing;

  resize: none;
}
```

- now go to _base.css to include the textarea in the input block rule

```
h2 {
  font-size: $large-font-size;
  margin-bottom: $space;
}

a {
  color: $body-default-color;
}

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

- we are going to be solving an issue with our fonts, in chrome this works fine but in firefox or other browsers it might not
- so go _variables.scss to create a new variable

```
//color
$brand-primary: #397ab1;
$grey: #dddddd;
$light-grey: #f9f9f9;
```



```
$dark-grey: #777777;

//Body

$body-background-color: #fafafa;

$body-default-color: #333333;

$body-default-font-family: Helvetica, Arial, sans-serif;


//spacing

$space: 1.4rem;

$large-space: 2.8rem;

$site-max-width: 100rem;
```

- now go to `_base.scss` to use this variable

```
*{

  box-sizing: border-box;

  padding: 0;

  margin: 0;

}

html {

  font-size: 62.5%;

}

body {

  background-color: $body-background-color;

  color: $body-default-color;

  font-family: $body-default-font-family;

  font-size: $base-font-size;

}

p{

  margin-bottom: $space;
```

```

}

h1 {
  font-weight: 300;
  margin-bottom: $space;
}

h2 {
  font-size: $large-font-size;
  margin-bottom: $space;
}

a {
  color: $body-default-color;
}

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

```

- we also need to add our color to our inputs because of browser issues, so we will reference a color variable that already exists

```

input[type=text], input[type=email], input[type=password], textarea {
  border: 1px solid $input-border;
  color: $body-default-color;
  font-family: $body-default-font-family;
  font-size: $base-font-size;
}

```

```
margin-bottom: $space;

padding: $input-spacing;

}
```

- and lastly for our button we are just going to use a class that we have used before and a modifier that we have used before, so go Editor.js

```
render(){
  if(this.props.note){
    return (
      <div className="editor">
        <input className="editor__title" value={this.state.title} placeholder="Untitled" onChange={this.handleTitleChange.bind(this)} />
        <textarea className="editor__body" value={this.state.body} placeholder="Your note here" onChange={this.handleBodyChange.bind(this)} />
        <div>
          <button className="button button--secondary" onClick={this.handleRemoval.bind(this)}>Delete note</button>
        </div>
      </div>
    );
  }else{
```

- now we need to style the text we get when there is no note so in _editor.scss let create another selector

```
.editor__body {
  flex-grow: 1;
  font-weight: 300;
  margin-bottom: $large-space;
  outline: none;
  padding: $input-spacing;
  resize: none;
}

.editor__message {
```

```

text-align: center;

margin: $large-space * 2;

font-style: italic;
}

```

- now lets go to Editor.js to apply this new class

```

render(){
  if(this.props.note){
    return (
      <div className="editor">
        <input className="editor__title" value={this.state.title} placeholder="Untitled" onChange={this.handleTitleChange.bind(this)}>
        <textarea className="editor__body" value={this.state.body} placeholder="Your note here" onChange={this.handleBodyChange.bind(this)}></textarea>
        <div>
          <button className="button button--secondary" onClick={this.handleRemoval.bind(this)}>Delete note</button>
        </div>
      </div>
    );
  }else{
    return (
      <div className="editor">
        <p className="editor__message">
          {this.props.selectedNoteId ? 'Note not found' : 'Pick or create a note to get started'}
        </p>
      </div>
    );
  }
}
};

```

- lets now commit and push our code (we are using the -a flag because we are not adding new files we are just modifying files that already exist in gitgub)

```
joses-MacBook-Pro:notes mendoza$ git commit -a -m "Style editor component"
```

```
joses-MacBook-Pro:notes mendoza$ git push
```

-