

LEARNING FACEBOOK'S



React.js

Styling Components

Lesson 04



Lesson Objectives

At the end of this module on React fundamentals you will be able to:

- Explain and demonstrate
 - CSS Styling
 - Scoping Styles using Inline Styles
 - Limitations of inline styles
 - Inline Styles with Radium
 - Using Psuedo classes/media queries with inline styles
 - CSS Modules, importing css classes
 - Adding Bootstrap, Semantic UI to React apps
 - Using react-bootstrap, reactstrap packages

LEARNING FACEBOOK'S



React.js



- In react there are 4 ways of styling
 - **inline styling**
style-component
 - **CSS Modules**
 - **Regular CSS stylesheets.**

1. CSS StyleSheet

import CSS file `import './DottedBox.css'` so you can have a separate CSS file for each component

2. Inline Styling

- We can create a variable that stores style properties and then pass it to the element like `style={nameOfvariable}`
- We can also pass the styling directly `style={{color: 'pink'}}`

3. CSS Modules

A CSS Module is a CSS file in all class names and animation names are scoped locally by default

4. Styled-components is a library for React and React Native that allows you to use component-level styles in your application that are written with a mixture of JavaScript and CSS



In React, inline styles are not specified as a string; instead they are specified with an object whose key is the camelCased version of the style name, and whose value is the style's value.

When specifying a pixel value for inline style prop, React automatically appends the string "px" after the number value.

```
var divStyle = {height: 10}; // rendered as "height:10px"  
ReactDOM.render(<div style={divStyle}>Hello World!</div>, mountNode);
```

Limitations of Inline styles

In React such as pseudo selectors like :hover or media queries will not work



Inline Styles with Radium

Radium is a collection of tools to manage styles in a ReactJS element. It was Created by Formidable labs.

Some of the key features of Radium are as follows:-

1. Conceptually simple extension of normal inline styles
2. supports pseudo class like :hover, :focus and :active etc.
3. Media queries for responsive design
4. Automatic vendor prefixing like -moz, -webkit, -o and -ms to support experimental or nonstandard CSS properties.
5. Supports CSS3 Keyframes animation.
6. ES2015 class and createClass support.

Radium can be installed using
npm install radium -S command.

Demo



Inline-Styles

create-react-styles



Using Psuedo classes/media queries with inline styles

There are many css-in-js frameworks like emotion,fela,react-jss etc,. out of which Radium is good

React supports following pseudo selectors [:hover](#), [:focus](#) and [:active](#) with less effort from developers.

```
import Radium from 'radium'

const myStyle = {
  color: '#000000'
  ':hover': {
    color: '#ff00ff'
  }
};

const AppComponent = () => {
  return (
    <div style={myStyle}>
      <button/>
    </div>
  );
};

const MyStyledComponent = Radium(AppComponent);
```

Using Psuedo classes/media queries with inline styles contd.



Add media queries to your style objects like how we add pseudo class selectors like hover, focus etc,.

The key must start with @media, and the [syntax](#) is identical to CSS:

Note that you must wrap your top-level component in the `<StyleRoot>` component to render the Radium stylesheet. Like `<AppContent/>` enclosed with `<StyleRoot>` below code snippet.

```
class App extends Component { render(){
  return (
    <StyleRoot>
    <AppContent /> </StyleRoot> )
  }
}

'@media screen and (orientation: landscape)': {
width: '20%',
textAlign: 'center',
paddingBottom: '200px',
}
```


CSS Modules, importing css classes



CSS has always been easy and flexible, but if the project grows ie if it is large projects

Some of the problems are:

1. Global Warming Namespaces.
2. Dead code elimination.
3. Dependencies.
4. Conditionals.

There are multiple ways to create a basic application

1. using create-react-app
2. webpack and babel

If we are using create-react-app to create our react application. Using cssModule is very simple.

Using CssModules with create-react-app



Using cssModules is simple like normal css .But one main thing is to give your CSS files the “module” prefix prior the extension: *.module.css*.

For eg: app.module.css.

After which we have to import the css file as shown below

import styles from './app.module.css';

Then we can use styles like below code snippet

```
class App extends Component {  
  render() {  
    return (  
      <div>  
        <div className={styles['one']}>  
          <h2>welcome</h2>  
        </div>  
        <h1>hello</h1>  
      </div>  
    );  
  }  
}
```

Demo



Styling

react-style-pseudoselector

react-radium-2019

react-create-app-radium-cssmodule



Add Bootstrap for React



Bootstrap is the world's most popular front-end development framework. Bootstrap is the leading HTML, CSS, and JavaScript framework for creating mobile-first, responsive websites and web application.

Bootstrap can be added to your React app in several ways.

1. Using CDN links - try avoid using it
2. Used as dependency & packages

Bootstrap As a Dependency & packages

```
npm install react-bootstrap bootstrap
```

After installation of bootstrap, import the below css into the file,

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

Import individual components like: react-bootstrap/Button rather than the entire library.

```
import Button from 'react-bootstrap/Button';
```

```
// or
```

```
import { Button } from 'react-bootstrap';
```

Using packages



we can also make use of another packages called Using reactstrap

We can install using

```
npm install bootstrap --save  
npm install --save reactstrap react react-dom
```

After this

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

Then we can import the component ie button say for eg like below in js file

```
import { Button } from 'reactstrap';
```

And shown below is an example code snippet

```
import React from 'react';  
import { Button } from 'reactstrap';  
export default (props) => {  
  return ( <Button color="danger">Danger!</Button> );  
};
```

Demo



React-bootstrap-button
react-bootstrap-reactstrap

(demo also includes form)



Semantic UI to React apps



Semantic UI is nothing but an alternate of Bootstrap or any other front-end or user interface.

Main advantage of Semantic UI is

jQuery Free

Declarative API

Augmentation

Shorthand Props

Sub Components

Auto Controlled State

We can use in our project as usual by create an application using webpack or create-react-app

Then we install semantic UI using below command

```
npm i semantic-ui-react --save
```

Now, on your package.json file there is a semantic-ui dependency. However, it does not provide CSS-style. Install Semantic UI CSS via:

adding the Semantic UI CDN link in your index.html file

```
<link rel="stylesheet" href="//cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.3.1/semantic.min.css"></link>
```

OR

Semantic UI to React apps contd.



installing Semantic UI package in your React app from [npm](https://www.npmjs.com/package/semantic-ui-react)

npm i semantic-ui –save

This installation will ask few questions

```
[07:25:32] Starting 'run setup'...  
? Set-up Semantic UI Automatic (Use default locations and all components)  
? We detected you are using NPM Nice! Is this your project folder? /Users/alberto/Desktop/my-semantic-ui-app Yes  
? Where should we put Semantic UI inside your project? semantic/  
[07:26:52] Finished 'run setup' after 1.32 min  
[07:26:52] Starting 'create install files'...
```

After this we can include below line into include the minified CSS file in your *index.js*

import 'semantic-ui/dist/semantic.min.css';

Advantages of using Semantic UI:

1. Incredible customization
2. Countless UI Components
3. Beautiful design
4. Official support for third-party apps

Demo



React-semantic-2019





Summary

- By now You should be clear with:
- CSS Styling
- Scoping Styles using Inline Styles
- Limitations of inline styles
- Inline Styles with Radium
- Using Psuedo classes/media queries with inline styles
- CSS Modules, importing css classes
- Adding Bootstrap, Semantic UI to React apps
- Using react-bootstrap, reactstrap packages





What are two ways the get values from <FORM> element?

1. Contolled components
2. Nested Components
3. input ref's
4. routing

The _____ operator is called as spread operator.

1. "..."
2. ".."
3. []
4. !..!