4. Four ways to style react components

Agata Krzywda   Follow
Mar 22, 2017 · 3 min read

This is the fourth part of my React tutorials. See the Intro to react.js here

There are four different options to style React components. All depend on your personal preferences and the specific complexity of your application.

If you want to add just a few style properties, then **inline styling** is the best option.

When you want to reuse your style properties in the same file then **style-componen**t are perfect.

When your application is more complex I recommend **CSS Modules** or regular **CSS stylesheets.**

1. CSS Stylesheet

```
1   import React from 'react';
2   import './DottedBox.css';
3
4   const DottedBox = () => (
5     <div className="DottedBox">
6       <p className="DottedBox_content">Get started with CSS styling</p>
7     </div>
8   );
9
10  export default DottedBox;
```
DottedBox.js hosted with ❤ by GitHub   view raw

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

```
1   .DottedBox {
2     margin: 40px;
3     border: 5px dotted pink;
4   }
5
6   .DottedBox_content {
7     font-size: 15px;
8     text-align: center;
9   }
```
DottedBox.css hosted with ❤ by GitHub   view raw

2. Inline styling

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, usually a string.

```
1   import React from 'react';
2
3   const divStyle = {
4     margin: '40px',
5     border: '5px solid pink'
6   };
7   const pStyle = {
8     fontSize: '15px'
```

```
 8      fontSize: 15px;
 9      textAlign: 'center'
10    };
11
12    const Box = () => (
13      <div style={divStyle}>
14        <p style={pStyle}>Get started with inline style</p>
15      </div>
16    );
17
18    export default Box;
```

- 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 which all class names and animation names are scoped locally by default. Great article about css modules here.

```
 1    import React from 'react';
 2    import styles from './DashedBox.css';
 3
 4    const DashedBox = () => (
 5      <div className={styles.container}>
 6        <p className={styles.content}>Get started with CSS Modules style</p>
 7      </div>
 8    );
 9
10    export default DashedBox;
```

- Similar to css we import css file `import styles './DashedBox.css'`
- then we access to className as we access to object

```
 1    :local(.container) {
 2      margin: 40px;
 3      border: 5px dashed pink;
 4    }
 5    :local(.content) {
 6      font-size: 15px;
 7      text-align: center;
 8    }
```

`:local(.className)` -this when you use create-react-app because of webpack configurations

`.className` -this if you use your own react boilerplate.

To make CSS modules work with Webpack you only have to include the modules mentioned above and add the following loader to your `webpack.config.js` file:

```
. . .
{
  test: /\.css$/,
  loader: 'style!css-loader?
modules&importLoaders=1&localIdentName=
[name]__[local]___[hash:base64:5]'
}
. . .
```

4. Styled-components 💅

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

```
11    `;
12
13    const Paragraph = styled.p`
14      font-size: 15px;
15      text-align: center;
16    `;
17
18    const OutsetBox = () => (
19      <Div>
20        <Paragraph>Get started with styled-components 💅</Paragraph>
21      </Div>
22    );
```

```
23  };
24  export default OutsetBox;
```
OutsetBox.js hosted with ❤ by GitHub   view raw

- First we need to install `styled-components` library

- `npm install styled-components --save`

- Now we can create a variable by selecting a particular html element where we store our style keys `const Div = styled.htmlElemnet` color: pink`

- Then we use the name of our variable as a wrapper `<Div></Div>` kind of react component:)

- Tips to use emoji icons key shortcut `CTRL+CMD+SPACE` 💡

All these ways of styling react components have pros and cons.

It all boils down to both your own personal preferences and the specific complexity of your application.
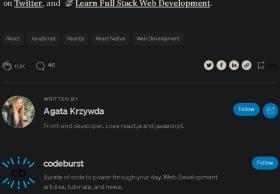
I suggest you to make 4 projects, each using different way of style.

Enjoy 😃

Next tutorial The Component Lifecycle

codeburst.io

📰 Subscribe to *Codeburst's* once-weekly **Email Blast,** 🔔 Follow *Codeburst* on **Twitter**, and 🎓 **Learn Full Stack Web Development**.

React   JavaScript   Reactjs   React Native   Web Development

👏 11.1K   💬 46   🐦 📘 in 🔗 🔖 ...

WRITTEN BY

Agata Krzywda   Follow ✉

Front-end developer. Love react.js and javascript.

codeburst   Follow

Bursts of code to power through your day. Web Development articles, tutorials, and news.

More From Medium

Technologies that makes JavaScript a beautiful programming language!

Gabriel

A finite-state machine (FSM) is a mathematical model of computation used to model plenty of...

Emin Aliyev

Solving Common Vue Problems — List Filtering, Global Libraries, Vuex, and More

John Au-Yeung in JavaScript in Plain English

JavaScript Data Structures: Trees!

Bret Gibson

Developing Vue Apps with the Quasar Library — Rating Control

John Au-Yeung

Complete Confluence Compliant Avro Deserializer that can do schema evolution

Wu Jiaqiao

Tech Stack 2019: Frontend Bundle Splitting

John Tucker in codeburst

AnyChart JS Charts 8.11.0 Released with Awesome New Data Visualization Features

AnyChart in JavaScript in Plain English