



Working with styles in React

Agenda

- Traditional styling & its shortcomings.
- Different ways of styling components in React
- React Bootstrap



Problems with traditional CSS

- Everything is global
- CSS grows over time (Dead code elimination)
- Hard to maintain (SMACSS, BEM, OOCSS)
- Minification
- Sharing constants
- Non-deterministic resolution
- Dependencies (import whole file even if you want a single style from it)



Ways to style a React Component

1. Inline styles
2. Styled-components(CSS in JS)
3. CSS Stylesheets



Inline Styles

- Unlike traditional inline styles, inline styles in react are passed as an object.
- **Key** is the **camelCased** version of the style name and **value** are passed as a string
- Style properties can be stored inside a variable can can be passed to style attribute.
- Styles object can also be passed directly inside style attribute



Styling in React

Example :- <https://codesandbox.io/s/n3v9l40my0>



Popular React Styling Libraries

- Radium
- Emotion
- Aphrodite



Advantages Inline Styles

- Component as a standalone entity
- No colliding styles, no dead code
- Close to react native



Problems with React Inline Styles

- Inline styles are at the top of the specificity spectrum
- Does not support pseudo-classes and pseudo-elements
- No support for Media Queries
- Makes debugging difficult
- You can't use any existing tooling
- CSS is cacheable
- Performance, not portable/reusable
- External library dependencies



Styled Components

- *“The basic idea of styled-components is to enforce best practices by removing the mapping between styles and components.”*
- Component-level styles that are written with a mixture of JavaScript and CSS
- Props over classes
- **npm install styled-components --save**
- `const Wrapper = styled.section`color: pink``
- `<Wrapper>Hello world !! </Wrapper>`



```
npm install --save styled-components
```

```
import styled from 'styled-components'
```

```
const Button = styled.button`
```

```
  background: transparent
```

```
  color: red
```

```
,
```

```
<Button>Button</Button>
```





```
import styled from 'styled-component';  
const Button = styled.button`  
  background: transparent;  
  color: ${props => props.primary ? 'blue' : 'purple'};  
  `;  
  
<Button primary>Primary Button</Button>
```



Advantages Styled Components

- All the CSS
- Independent and reusable
- Containers and Presentational Component
- Single class name
- Basic Sass support
- Native mobile support



Disadvantages styled component

- Yet another library
- New dependencies.
- Learning curve could be difficult for the designer



CSS Styles

- Syntax like traditional CSS
- Create CSS file and import it.
- CSS Loader, Style Loader, Sass Loader



React Bootstrap

- The most popular FrontEnd Framework, rebuild for React.
- **react-bootstrap** is a library of reusable front-end components. You'll get the look-and-feel of Twitter Bootstrap, but with much cleaner code, via Facebook's React.js framework.



Getting Started With React Bootstrap

- `$ npm install react-bootstrap --save`

- `<link rel="stylesheet"`

`href="https://maxcdn.bootstrapcdn.com/bootstrap/latest/css/b
ootstrap.min.css">`



Exercise

- Create a landing page of an ecommerce website. It should have a carousel with some text over it and some product's listing. Try to make it responsive. Example - <https://ibb.co/1n2y9s8>
- Create a simple Higher Order Components that will inject some new props in the inner component.



Thank You

