

React

Lecture 3





Stateless Functional Component

- Made using a Function instead of a Class - that's why functional
- Does not have any state - that's why stateless
- Still a React Component - that's why Component
- Both anonymous and arrow functions can be used to declare a SFC
- Takes the props as an argument



Stateless Functional Component

```
const User = () => {  
  return (  
    <div>  
      <p>Name: </p>  
      <p>Age: </p>  
    </div>  
  );  
};  
  
ReactDOM.render(<User />, document.getElementById('app'));
```

```
const User = (props) => {  
  return (  
    <div>  
      <p>Name: {props.name}</p>  
      <p>Age: {props.age}</p>  
    </div>  
  );  
};  
  
ReactDOM.render(<User name="Andrew" age={26} />, document.getE
```



Default Props

If you are interested, Have a look at this:

<https://reactjs.org/docs/typechecking-with-proptypes.html#default-prop-values>



React Dev Tool

- Google: “React Dev Tool For Chrome (or Your favorite browser)”
- Add it to your browser
- Restart your browser
- Reload the page



React Component Lifecycle

- Various situations of a component

<https://reactjs.org/docs/react-component.html#the-component-lifecycle>

<http://projects.wojtekmaj.pl/react-lifecycle-methods-diagram/>

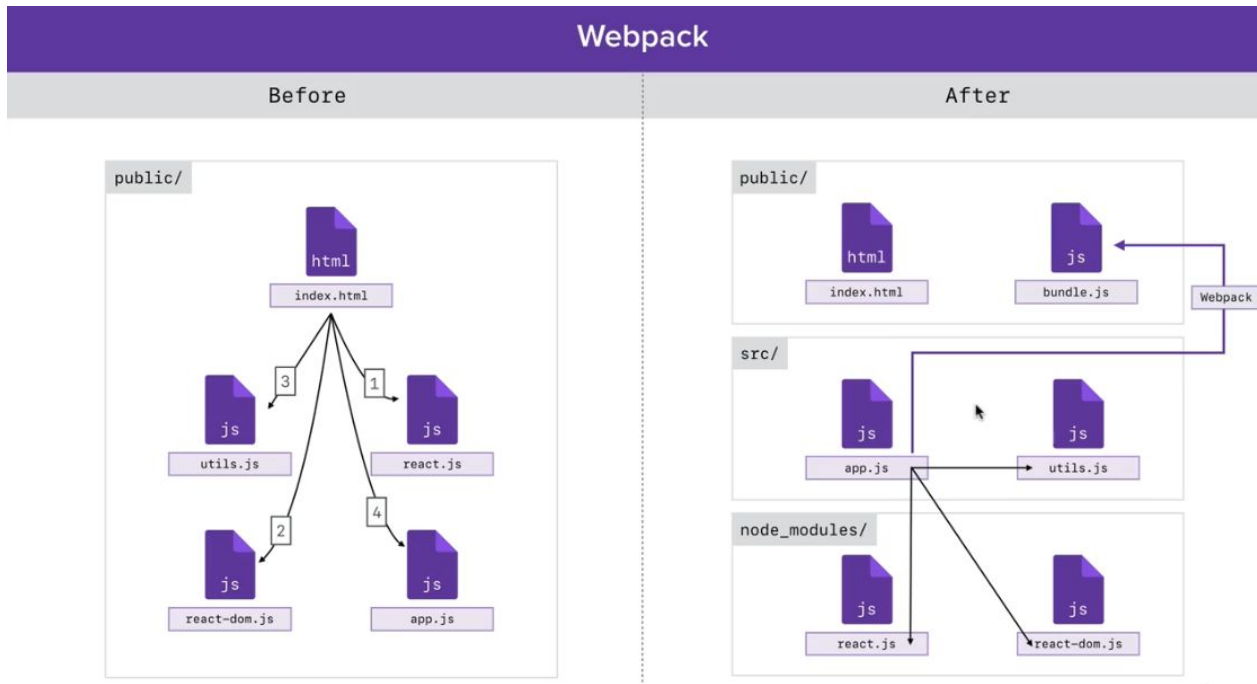


Life Cycle Example (+ Local Storage)

```
12  componentDidMount() {  
13    const json = localStorage.getItem('options');  
14    const options = JSON.parse(json);  
15  
16    if (options) {  
17      this.setState(() => ({ options }));  
18    }  
19  }  
20  componentDidUpdate(prevProps, prevState) {  
21    if (prevState.options.length !== this.state.options.length) {  
22      const json = JSON.stringify(this.state.options);  
23      localStorage.setItem('options', json);  
24    }  
25  }  
26  componentWillUnmount() {  
27    console.log('componentWillUnmount');  
28  }
```



What/Why Webpack





Webpack Config

Install webpack using npm (v3.1.0)

```
const path = require('path');

module.exports = {
  entry: './src/app.js',
  output: {
    path: path.join(__dirname, 'public'),
    filename: 'bundle.js'
  }
};
```

Define Entry and Output Using Path Module inside webpack.config.js



ES6 Import/Export

```
1 console.log('utils.js is running');
2
3 export const square = (x) => x * x;
4
5 export const add = (a, b) => a + b;
6
7 const subtract = (a, b) => a - b;
8
9 export default subtract;
10
11 // export { square, add, subtract as default };
12
13 // exports - default export - named exports
```

```
1 // import './utils.js';
2 import anythingIWant, { square, add } from './utils.js';
3
4 console.log('app.js is running');
5 console.log(square(4));
6 console.log(add(100, 23));
7 console.log(anythingIWant(100, 20));
```



Using Babel With Webpack

- Install ***babel-core@6.25.0*** and ***babel-loader@7.1.1***
- Add Rules using babel loader inside webpack config

```
module: {  
  rules: [{  
    loader: 'babel-loader',  
    test: /\.js$/,  
    exclude: /node_modules/  
  }]  
}
```

- Create a .babelrc file and add all the presets

```
1 {  
2   "presets": [  
3     "env",  
4     "react"  
5   ]  
6 }
```