

React Component Approaches



Cory House

@housecor

reactjsconsulting.com



Here's the Plan

- **React component creation approaches**
- **Container vs Presentational Components**



Four+ Ways to Create React Components?!



Ways to Create Components

- **ES5 createClass**
- **ES6 class**
- **ES5 stateless function**
- **ES6 stateless function**
- **Many more...**



ES5 Class Component

```
var HelloWorld = React.createClass({  
  render: function () {  
    return (  
      <h1>Hello World</h1>  
    );  
  }  
});
```



ES6 Class Component

```
class HelloWorld extends React.Component {  
  constructor(props) {  
    super(props);  
  }  
  
  render() {  
    return (  
      <h1>Hello World</h1>  
    );  
  }  
}
```



React in ES6

A yellow square containing the text "ES6" in a bold, dark blue, sans-serif font.

ES6

No autobind

PropTypes declared separately

Default props declared separately

Set initial state in constructor



ES5 Stateless Functional Component

```
var HelloWorld = function(props) {  
  return (  
    <h1>Hello World</h1>  
  );  
});
```



ES6 Stateless Functional Component

```
const HelloWorld = (props) => {  
  return (  
    <h1>Hello World</h1>  
  );  
});
```



```
1 import React from 'react';
2
3 class HelloWorld extends React.Component {
4   constructor(props) {
5     super(props);
6   }
7
8   sayHi(event) {
9     alert(`Hi ${this.props.name}`);
10  }
11
12  render() {
13    return (
14      <div>
15        <a
16          href="#"
17          onClick={this.sayHi.bind(this)}>Say Hi</a>
18      </div>
19    );
20  }
21 }
22
23 HelloWorld.propTypes = {
24   name: React.PropTypes.string.isRequired
25 };
26
27 export default HelloWorld;
```

```
1 import React from 'react';
2
3 const HelloWorld = (props) => {
4   const sayHi = (event) => {
5     alert(`Hi ${props.name}`);
6   };
7
8   return (
9     <div>
10       <a
11         href="#"
12         onClick={sayHi}>Say Hi</a>
13     </div>
14   );
15 };
16
17 HelloWorld.propTypes = {
18   name: React.PropTypes.string.isRequired
19 };
20
21 export default HelloWorld;
```



Stateless Functional Components: 9 Benefits

```
1 import React from 'react';
2
3 class HelloWorld extends React.Component {
4   constructor(props) {
5     super(props);
6   }
7
8   sayHi(event) {
9     alert('Hi ${this.props.name}');
10  }
11
12  render() {
13    return (
14      <div>
15        <a
16          href="#"
17          onClick={this.sayHi.bind(this)}>Say Hi</a>
18      </div>
19    );
20  }
21 }
22
23 HelloWorld.propTypes = {
24   name: React.PropTypes.string.isRequired
25 };
26
27 export default HelloWorld;
```

```
1 import React from 'react';
2
3 const HelloWorld = (props) => {
4   const sayHi = (event) => {
5     alert('Hi ${props.name}');
6   };
7
8   return (
9     <div>
10       <a
11         href="#"
12         onClick={sayHi}>Say Hi</a>
13     </div>
14   );
15 };
16
17 HelloWorld.propTypes = {
18   name: React.PropTypes.string.isRequired
19 };
20
21 export default HelloWorld;
```

No class needed

Avoid `this` keyword

Enforced best practices

High signal-to-noise ratio

Enhanced code completion / intellisense

Bloated components are obvious

Easy to understand

Easy to test

Performance



Use stateless functional components when possible.



When Should I Use Each?

Class Component

State

Refs

Lifecycle methods

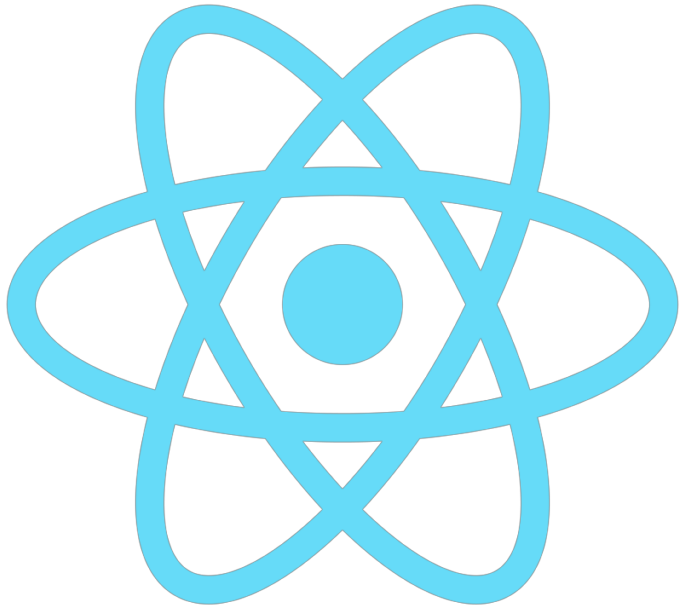
Child functions (for performance)

Stateless Components

Everywhere else 😊



Other Ways to Create Components



Object.create

Mixins

Parasitic Components

Stamplt

More info: bit.ly/react-define-component



Container vs Presentation Components



Container

Little to no markup

Pass data and actions down

Knows about Redux

Often stateful

Presentation

Most components



Nearly all markup

Receive data and actions via props

Doesn't know about Redux

Typically functional components



Alternative Jargon

Container	Presentational
Smart	Dumb
Stateful	Stateless
Controller View	View



”When you notice that some components don’t use props they receive but merely forward them down...it’s a good time to introduce some container components.”

Dan Abramov



Wrap up



ES5 `createClass`

ES6 Class

ES5 Stateless Function

ES6 Stateless Function

Many more!

Container vs Presentation Components

Next up: Let's start building!

