

# React components

What are components in React and how can we create them?

To create a `ReactComponent` we write a function that returns a `ReactElement`:

```
var Wrapper = function(props) {  
  return React.createElement('div', { className: 'wrapper' });  
}
```

We can then use these components like any DOM Node by passing it into a `createElement` call:

```
React.createElement(Wrapper);  
// -> <div class="wrapper"></div>
```

Our wrapper component isn't very useful so far, since it doesn't render its children:

```
React.createElement(Wrapper, {}, 'Hello World!');  
// -> <div class="wrapper"></div> (🙄 no "Hello World" visible!)
```

This is because our component function gets passed the properties (`props`). In the last example we didn't use the properties we got passed from the `createElement(Wrapper)` call at all!

Let's render the children that our `Wrapper` gets passed:

```
var Wrapper = function(props) {  
  // Render the children we get passed from the createElement(Wrapper) call  
  return React.createElement('div', { className: 'wrapper' }, props.children);  
}
```

Now the above example works perfectly:

```
React.createElement(Wrapper, {}, 'Hello World!');  
// -> <div class="wrapper">Hello World</div>
```

Let's try rendering our heading inside our **Wrapper** component:

HTML

CSS

JavaScript

```
1 var Wrapper = function(props) {  
2   return (  
3     React.createElement('div', { className: 'wrapper' }, props.children)  
4   );  
5 };  
6  
7 ReactDOM.render(  
8   React.createElement(Wrapper, null,  
9     React.createElement('h1', { className: 'heading' }, 'Hello World!')  
10  ),  
11  document.getElementById('container')  
12 );
```

output

# Hello World!

Amazing! 🎉 Now we have a reusable **Wrapper** component that could make sure every page across our application has consistent styling.