

Slightly more complicated React application, with the addition of props and state. Still built directly onto HTML document. Data can either be passed into a component by whatever renders it, or can be controlled within the component. Props are attributes specified at the time of calling a component. Within the component, these properties are accessed using the `this.props` keywords. These properties are passed to the Component and cannot be written to within the component, and thus are immutable. State, in contrast to props, is internal to and controlled by the component itself.

Also add event handlers and their associated functions to modify state of the React component.

1. Create file called `counter.html` :

```
>> touch counter.html
```

2. Add boilerplate HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>Second React app</title>
</head>
<body>
</body>
</html>
```

3. Add React library scripts before closing `<body>` tags, opening and closing `<script>` tags in which we'll code the React component, a `<div>` with id of `app` within `<body>` tags to specify location rendering the component:

```
<!DOCTYPE html>
<html>
<head>
  <title>Second React app</title>
</head>
<body>
  <div id='app'></div>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/react/16.1.0/umd/react.development.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/react-dom/16.1.0/umd/react-dom.development.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-core/5.8.34/browser.min.js"></script>
  <script type='text/babel'>
  </script>
</body>
</html>
```

4. Within the `<script>` tags, create a new React component called `Counter`. Within the component, reference `image`, `label`, and `date` props which will be passed into the component later on:

```
<!DOCTYPE html>

<html>

<head>

  <title>Second React app</title>

</head>

<body>

  <div id='app'></div>

  <script src="https://cdnjs.cloudflare.com/ajax/libs/r
eact/16.1.0/umd/react.development.js"></script>

  <script src="https://cdnjs.cloudflare.com/ajax/libs/r
eact-dom/16.1.0/umd/react-dom.development.js"></script>

  <script src="https://cdnjs.cloudflare.com/ajax/libs/b
abel-core/5.8.34/browser.min.js"></script>

  <script type='text/babel'>

    class Counter extends React Component{

      render(){

        return(

          <div>

            <img src={this.props.image} alt="Bloc
Logo"/>

            <h2>Current program: {this.props.label
}</h2>

            <p>Current Date: {this.props.date.toLo
caleDateString()}</p>

            <div>

              );

            }

  }
```

```
    }  
  </script>  
</body>  
</html>
```

5. add a constant called `info` which declares values for `date` and `image` properties:

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Second React app</title>  
  </head>  
  <body>  
    <div id='app'></div>  
    <script src="https://cdnjs.cloudflare.com/ajax/libs/react/16.1.0/umd/react.development.js"></script>  
    <script src="https://cdnjs.cloudflare.com/ajax/libs/react-dom/16.1.0/umd/react-dom.development.js"></script>  
    <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-core/5.8.34/browser.min.js"></script>  
    <script type='text/babel'>  
      class Counter extends React Component{  
        render(){  
          return(  
            <div>  
              <img src={this.props.image} alt="Bloc  
Logo"/>
```

```

        <h2>Current program: {this.props.label
    }</h2>

    <p>Current Date: {this.props.date.toLo
caleDateString()}</p>

    <div>

        );

    }

    }

    const info = {
    date: new Date(),
    image: 'https://avatars0.githubusercontent.com/u/1
441485?v=4&s=280',
    }

    </script>

</body>
</html>

```

- add a default property value for `label` for the `Counter` component using the `.defaultProps()` method:

```

<!DOCTYPE html>
<html>
<head>
    <title>Second React app</title>
</head>
<body>
    <div id='app'></div>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/r

```

```
react/16.1.0/umd/react.development.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/r
react-dom/16.1.0/umd/react-dom.development.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/b
abel-core/5.8.34/browser.min.js"></script>
    <script type='text/babel'>
        class Counter extends React Component{
            render(){
                return(
                    <div>
                        <img src={this.props.image} alt="Bloc
Logo"/>
                        <h2>Current program: {this.props.label
}</h2>
                        <p>Current Date: {this.props.date.toLo
caleDateString()}</p>
                        <div>
                            );
                        }
                    }
                const info = {
                    date: new Date(),
                    image: 'https://avatars0.githubusercontent.com/u/1
441485?v=4&s=280',
                }
                Counter.defaultProps = {
                    label: 'Web Developer Track'
                }
            }
        }
    </script>
```

```
    </script>
</body>
</html>
```

7. call the `Counter` component using `ReactDOM.render()` method, passing it props for date and image. Since label is already specified as a default prop, there is no need to add label as an attribute unless you want to change its default value. Don't forget to specify location of rendering the component, specifically at the `<div>` with id value of `app`:

```
<!DOCTYPE html>
<html>
<head>
  <title>Second React app</title>
</head>
<body>
  <div id='app'></div>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/react/16.1.0/umd/react.development.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/react-dom/16.1.0/umd/react-dom.development.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-core/5.8.34/browser.min.js"></script>
  <script type='text/babel'>
    class Counter extends React Component{
      render(){
        return(
```

```

        <div>
          <img src={this.props.image} alt="Bloc
Logo" />
          <h2>Current program: {this.props.label
}</h2>
          <p>Current Date: {this.props.date.toLo
caleDateString()}</p>
          <div>
            );
          }
        }
        const info = {
          date: new Date(),
          image: 'https://avatars0.githubusercontent.com/u/1
441485?v=4&s=280',
        }
        Counter.defaultProps = {
          label: 'Web Developer Track'
        }
        ReactDOM.render(
          <Counter
            image={info.image}
            date={info.date}
          />,
          document.getElementById('app')
        );
      </script>
</body>

```



```
</html>
```

8. **Counter** will hold state of a **number** with a value of 1. Add this state to the component, and access its value with an **<h3>** heading corresponding to the current week. To add state, create a **constructor** method within the component, and use the **this.state** keyword:

```
<!DOCTYPE html>
<html>
<head>
  <title>Second React app</title>
</head>
<body>
  <div id='app'></div>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/react/16.1.0/umd/react.development.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/react-dom/16.1.0/umd/react-dom.development.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-core/5.8.34/browser.min.js"></script>
  <script type='text/babel'>
    class Counter extends React.Component{
      constructor(props){
        super(props);
        this.state={number: 1};
      }
      render(){
```

```
        return(  
            <div>  
                <img src={this.props.image} alt="Bloc  
Logo"/>  
                <h2>Current program: {this.props.label  
}</h2>  
                <h3>Current week: {this.state.number}<  
/h3>  
                <p>Current Date: {this.props.date.toLo  
caleDateString()}</p>  
                <div>  
                    );  
                }  
            }  
            const info = {  
                date: new Date(),  
                image: 'https://avatars0.githubusercontent.com/u/1  
441485?v=4&s=280',  
            }  
            Counter.defaultProps = {  
                label: 'Web Developer Track'  
            }  
            ReactDOM.render(  
                <Counter  
                    image={info.image}  
                    date={info.date}  
                />,  
                document.getElementById('app')
```

```
);  
</script>  
</body>  
</html>
```

9. Create 2 new buttons to be rendered by `Counter` component. One button increments the `number` value held in state, while the second button decrements the value. Add the event handler `onClick` as attribute to each button, which takes an `increment` and `decrement` function as argument. Remember with JSX the arguments are enclosed by curly braces rather than quotation marks. Also remember the functions need to be manually bound using the `bind()` method; otherwise when passed to the functions the keyword `this` will be undefined:

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Second React app</title>  
</head>  
<body>  
  <div id='app'></div>  
    <script src="https://cdnjs.cloudflare.com/ajax/libs/react/16.1.0/umd/react.development.js"></script>  
    <script src="https://cdnjs.cloudflare.com/ajax/libs/react-dom/16.1.0/umd/react-dom.development.js"></script>  
    <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-core/5.8.34/browser.min.js"></script>
```

```
<script type='text/babel'>
  class Counter extends React Component{
    constructor(props){
      super(props);
      this.state={number: 1};
    }
    render(){
      return(
        <div>
          <img src={this.props.image} alt="Bloc
Logo"/>
          <h2>Current program: {this.props.label
}</h2>
          <h3>Current week: {this.state.number}<
/h3>
          <button onClick={this.increment.bind(t
his)}>Increment</button>
          <button onClick={this.decrement.bind(t
his)}>Decrement</button>
          <p>Current Date: {this.props.date.toLo
caleDateString()}</p>
          <div>
            );
          }
        }
      )
    }
    const info = {
      date: new Date(),
      image: 'https://avatars0.githubusercontent.com/u/1
```

```

441485?v=4&s=280' ,
    }
    Counter.defaultProps = {
      label: 'Web Developer Track'
    }

ReactDOM.render(
  <Counter
    image={info.image}
    date={info.date}
  />,
  document.getElementById( 'app' )
);
</script>
</body>
</html>

```

- add the `increment()` and `decrement()` functions to the `Counter` component. Since these functions are being triggered by an event handler (`onClick`), each function will take `event` as its argument. Within each function, the state of `number` should be updated using the `setState()` method. Do not let the value of `number` be lower than 1:

```

<!DOCTYPE html>
<html>
<head>
  <title>Second React app</title>

```

```
</head>
```

```
<body>
```

```
  <div id='app'></div>
```

```
    <script src="https://cdnjs.cloudflare.com/ajax/libs/react/16.1.0/umd/react.development.js"></script>
```

```
    <script src="https://cdnjs.cloudflare.com/ajax/libs/react-dom/16.1.0/umd/react-dom.development.js"></script>
```

```
    <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-core/5.8.34/browser.min.js"></script>
```

```
    <script type='text/babel'>
```

```
      class Counter extends React.Component{
```

```
        constructor(props){
```

```
          super(props);
```

```
          this.state={number: 1};
```

```
        }
```

```
        increment(event){
```

```
          this.setState({number: this.state.number +  
1})
```

```
        }
```

```
        decrement(event){
```

```
          this.setState({number: Math.max(1,this.state.number - 1)})
```

```
        }
```

```
        render(){
```

```
        return(  
            <div>  
                <img src={this.props.image} alt="Bloc  
Logo"/>  
                <h2>Current program: {this.props.label  
}</h2>  
                <h3>Current week: {this.state.number}<  
/h3>  
                <button onClick={this.increment.bind(t  
his)}>Increment</button>  
                <button onClick={this.decrement.bind(t  
his)}>Decrement</button>  
                <p>Current Date: {this.props.date.toLo  
caleDateString()}</p>  
                <div>  
                    );  
                }  
            }  
            const info = {  
                date: new Date(),  
                image: 'https://avatars0.githubusercontent.com/u/1  
441485?v=4&s=280',  
            }  
            Counter.defaultProps = {  
                label: 'Web Developer Track'  
            }  
            ReactDOM.render(  

```

```
<Counter
  image={info.image}
  date={info.date}
/>,
document.getElementById( 'app' )
);
</script>
</body>
</html>
```