

ES6 Autobinding:

```
class SayHello extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = {message: 'Hello!'};  
    // This line is important!  
    this.handleClick = this.handleClick.bind(this);  
  }  
  
  handleClick() {  
    alert(this.state.message);  
  }  
  
  render() {  
    // Because `this.handleClick` is bound, we can use it as an event handler.  
    return (  
      <button onClick={this.handleClick}>  
        Say hello  
      </button>  
    );  
  }  
}
```

Es6 without autobinding:

```
class SayHello extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = {message: 'Hello!'};  
  }  
  // WARNING: this syntax is experimental!  
  // Using an arrow here binds the method:  
  handleClick = () => {  
    alert(this.state.message);  
  }  
  
  render() {  
    return (  
      <button onClick={this.handleClick}>  
        Say hello  
      </button>  
    );  
  }  
}
```

All concepts:

Index.js

```
import CartItem from './App';

import * as serviceWorker from './serviceWorker';

const order = {

  title: 'Fresh fruits package',

  image: 'http://images.all-free-
download.com/images/graphiclarge/citrus_fruit_184416.jpg',

  initialQty: 3,

  price: 8

};
```

```
ReactDOM.render(

  <React.StrictMode>

    <CartItem title={order.title}

      image={order.image}

      initialQty={order.initialQty}

      price={order.price}/>,

  </React.StrictMode>,

  document.getElementById('root')

);
```

App.js

```
import React from 'react';
```

```
import {render} from 'react-dom';

//import PropTypes from 'prop-types';

export default class CartItem extends React.Component {
  constructor(props) {
    super(props);

    this.state = {
      qty: props.initialQty,
      total: 0
    };
  }

  componentWillMount() {
    this.recalculateTotal();
  }

  increaseQty() {
    this.setState({qty: this.state.qty + 1}, this.recalculateTotal);
  }

  decreaseQty() {
    let newQty = this.state.qty > 0 ? this.state.qty - 1 : 0;
    this.setState({qty: newQty}, this.recalculateTotal);
  }

  recalculateTotal() {
    this.setState({total: this.state.qty * this.props.price});
  }
}
```

```
}
```

```
// previous code we wrote here
```

```
render() {
```

```
  return <article className="row large-4">
```

```
    <figure className="text-center">
```

```
      <p>
```

```
        <img src={this.props.image}/>
```

```
      </p>
```

```
      <figcaption>
```

```
        <h2>{this.props.title}</h2>
```

```
      </figcaption>
```

```
    </figure>
```

```
    <p className="large-4 column"><strong>Quantity:  
{this.state.qty}</strong></p>
```

```
    <p className="large-4 column">
```

```
      <button onClick={this.increaseQty.bind(this)} className="button  
success">+</button>
```

```
      <button onClick={this.decreaseQty.bind(this)} className="button  
alert">-</button>
```

```
    </p>
```

```
<p className="large-4 column"><strong>Price per item:</strong>
${this.props.price}</p>
```

```
<h3 className="large-12 column text-center">
```

```
  Total: ${this.state.total}
```

```
</h3>
```

```
</article>;
```

```
}
```

```
}
```