Video 3.5

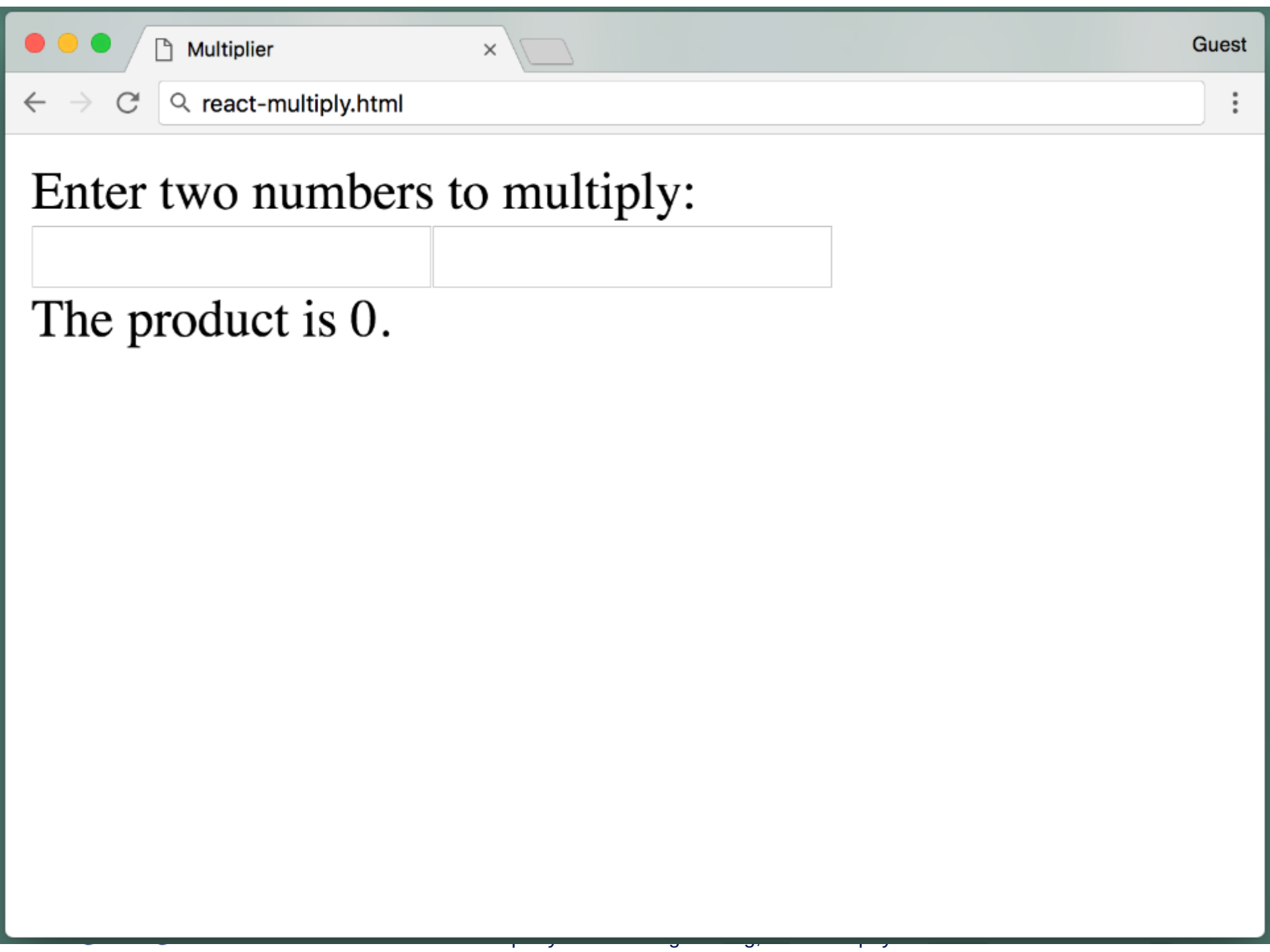
React Component Interaction – part 2

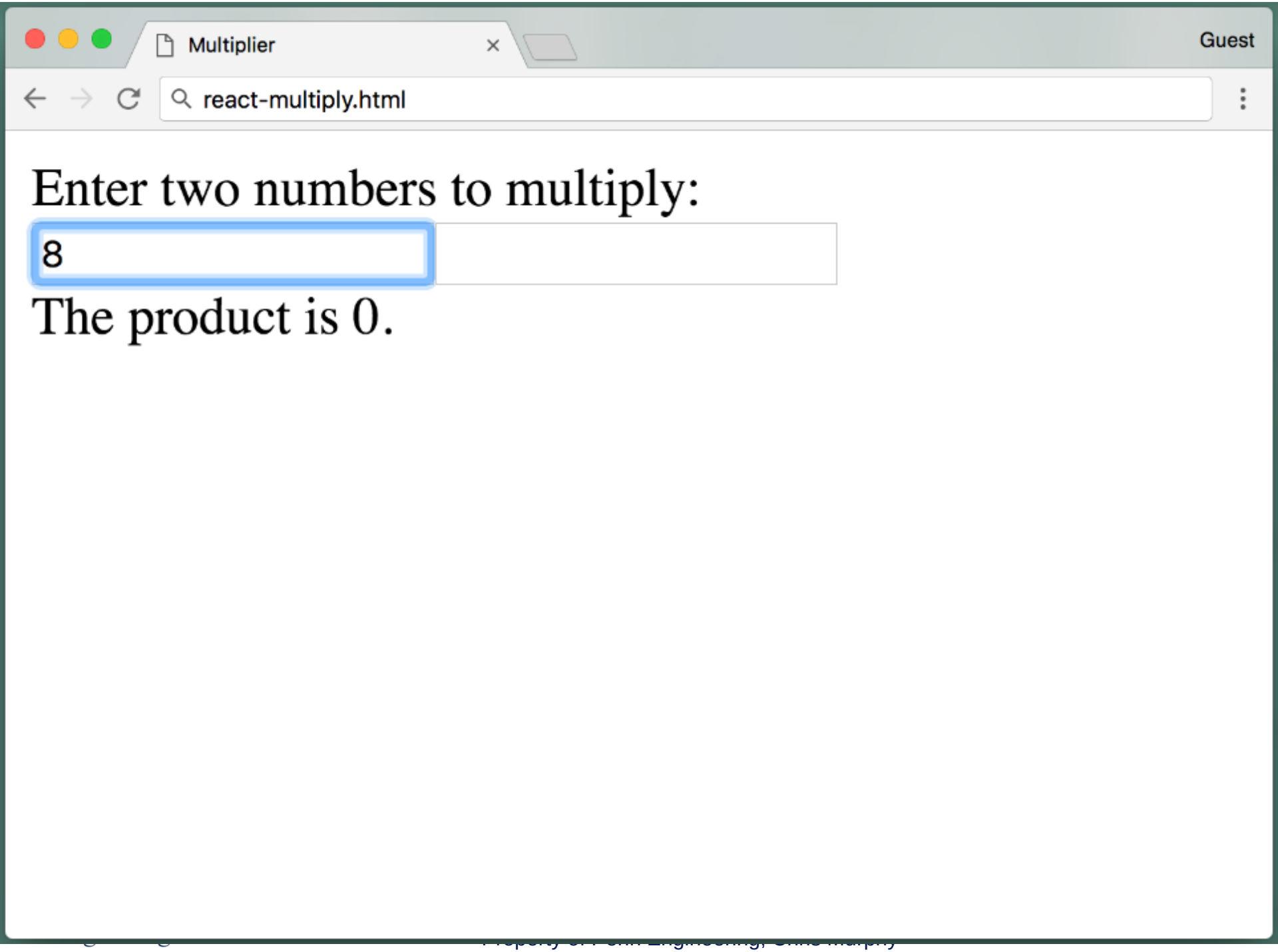
Chris Murphy

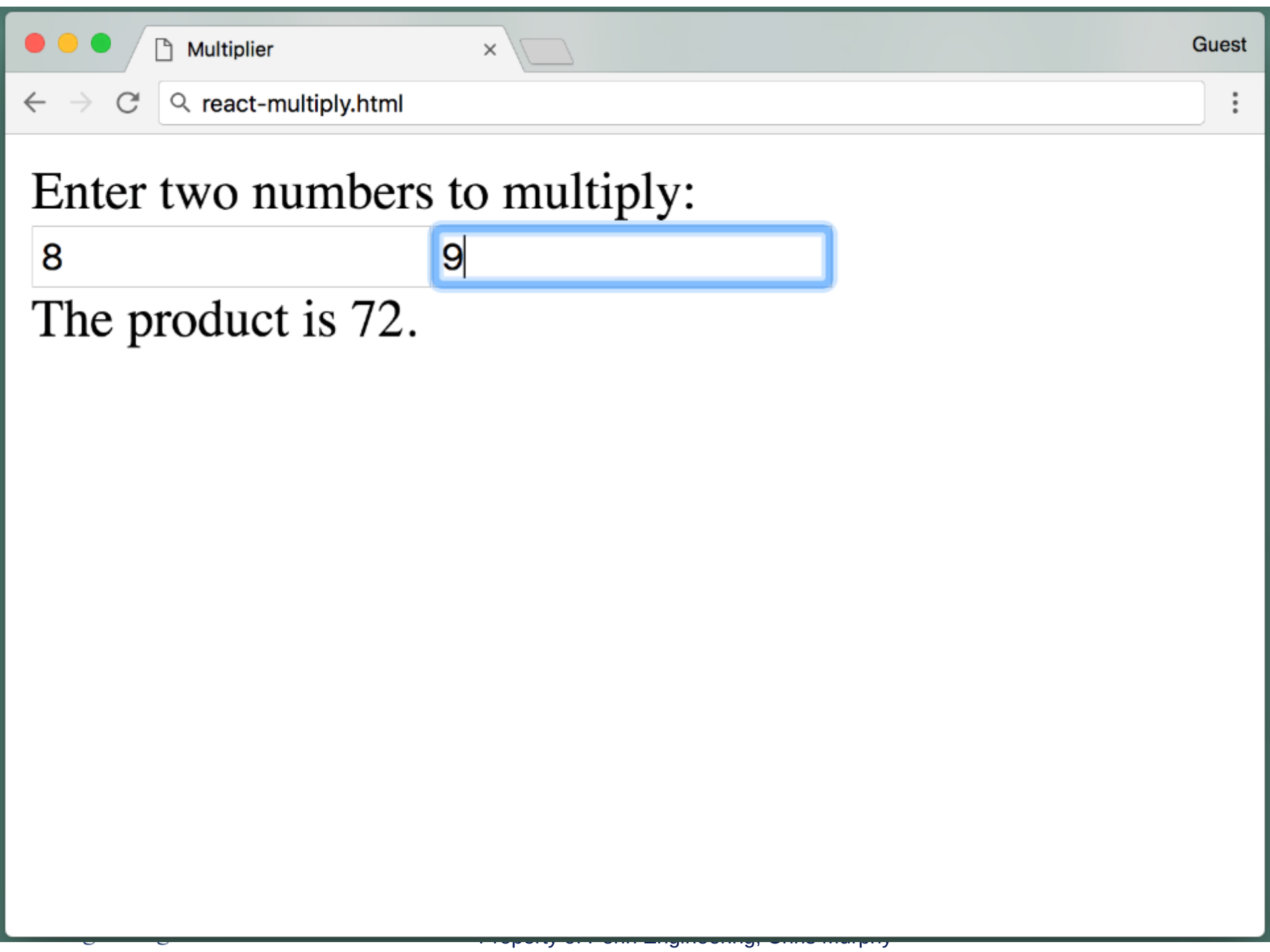
Review

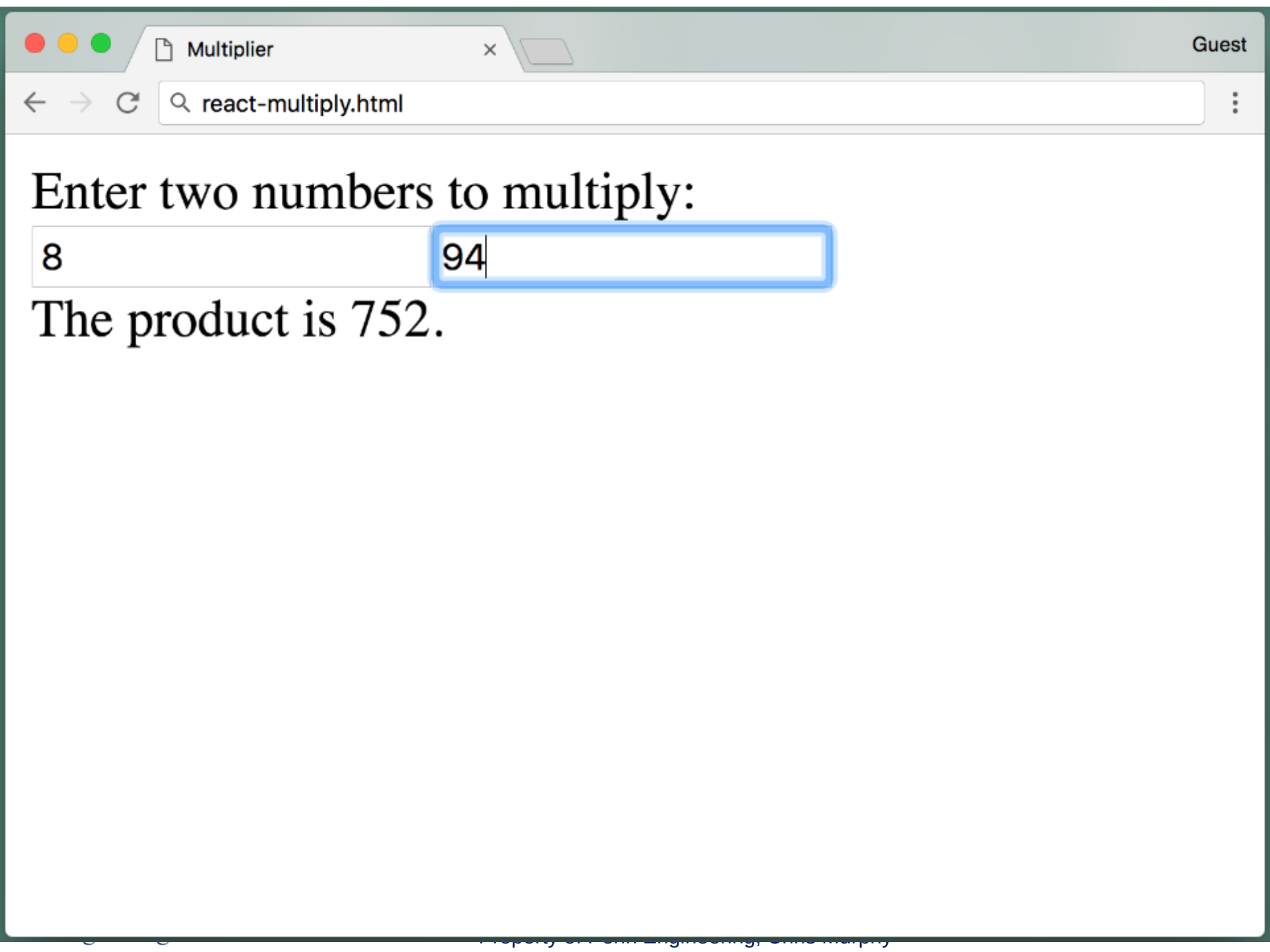
- React allows us to create reusable, modularized components that can be combined to form web applications

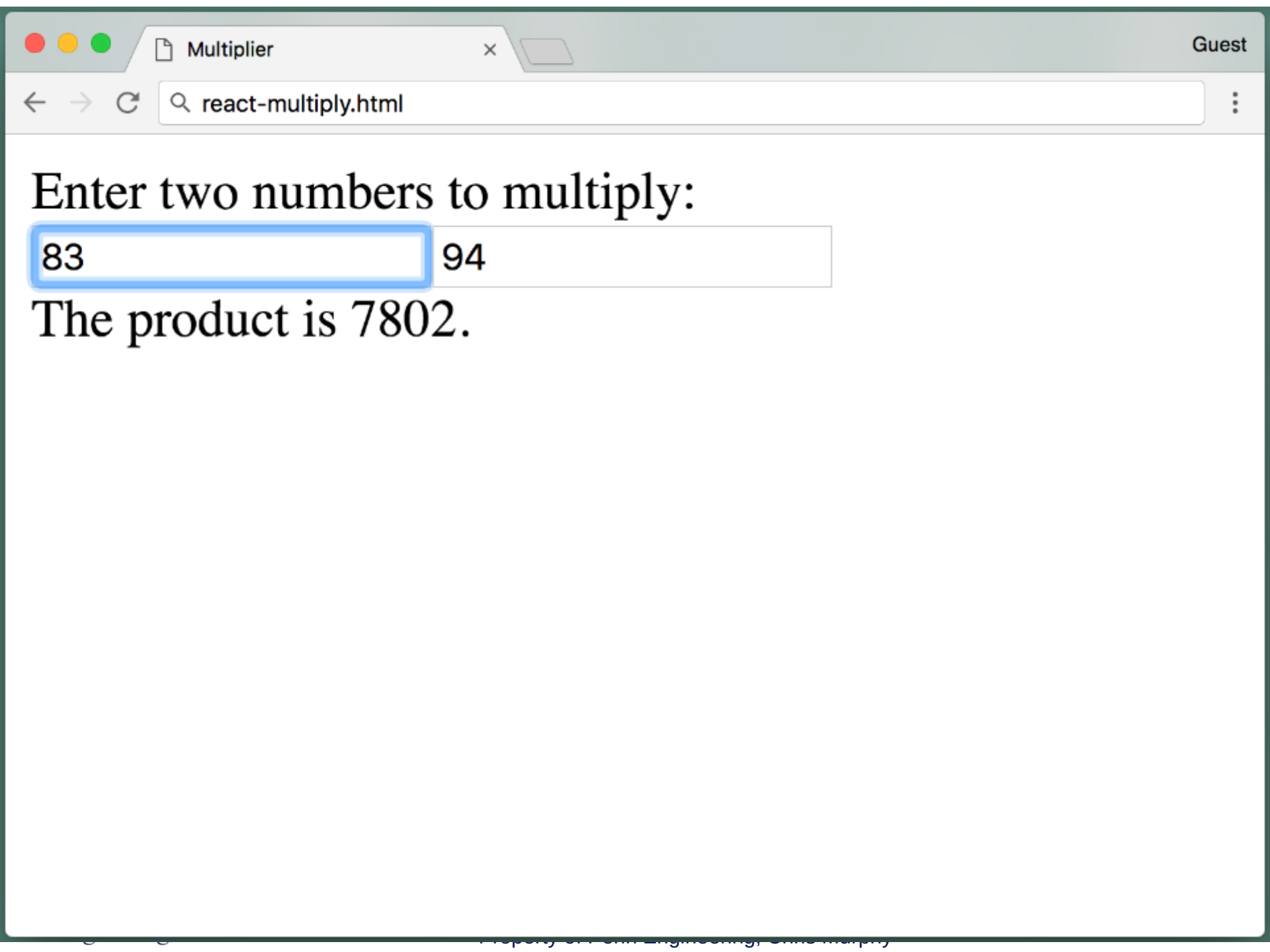
- React handles re-rendering of components based on the structure of VirtualDOM

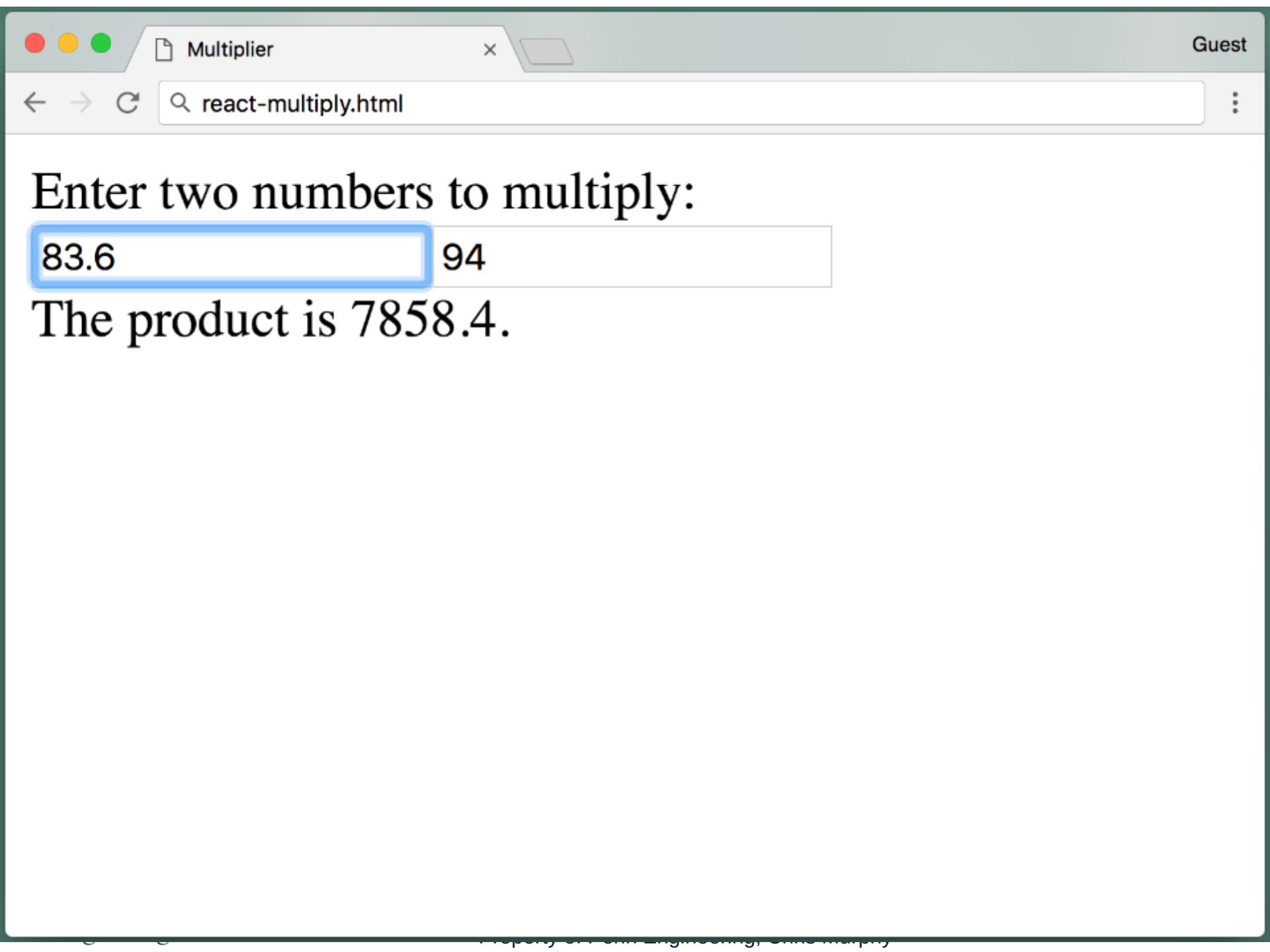

Multiplier
Guest
react-multiply.html
Enter two numbers to multiply:
The product is 0.

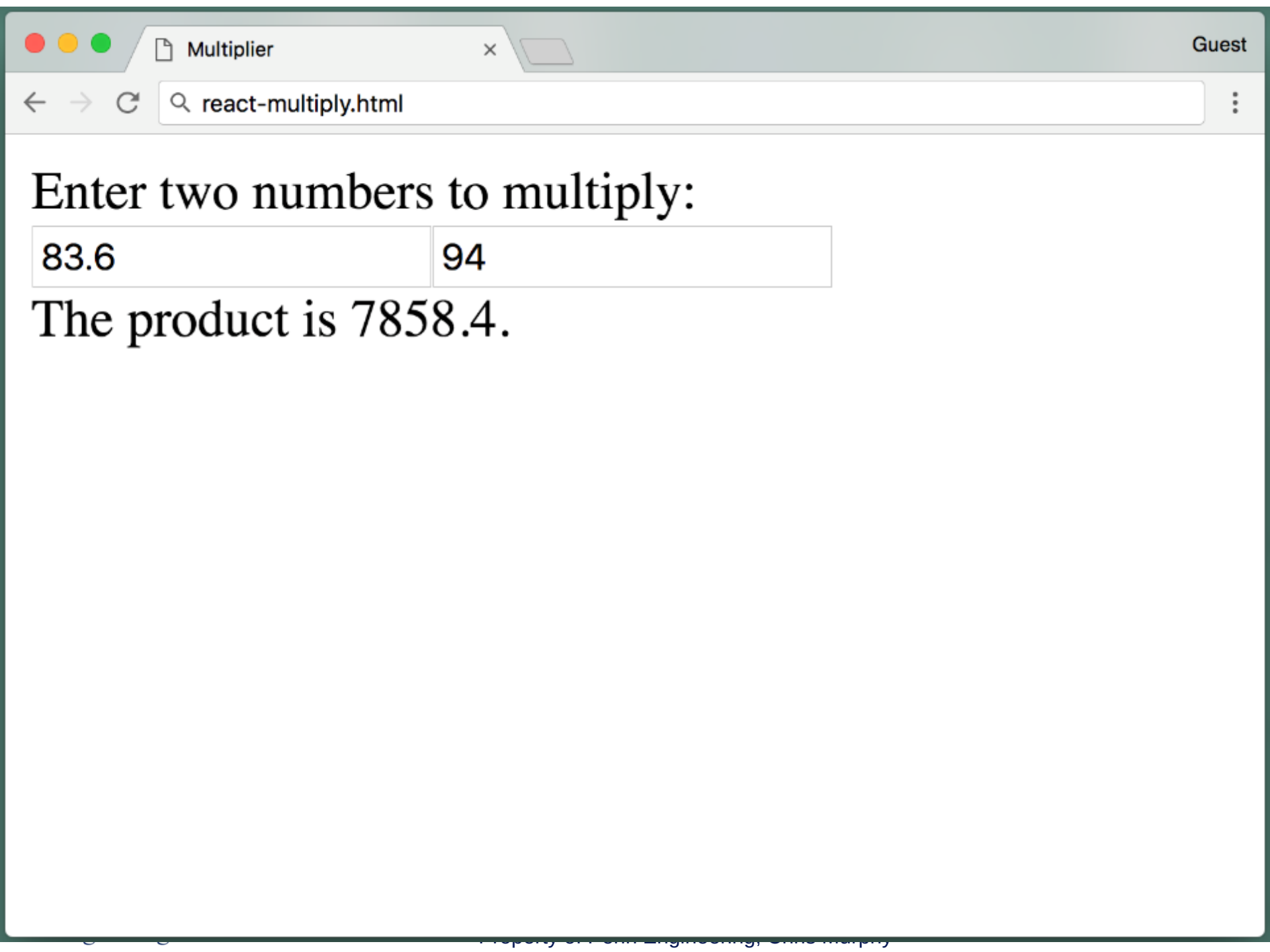

Multiplier
Guest
react-multiply.html
Enter two numbers to multiply:
8
The product is 0.

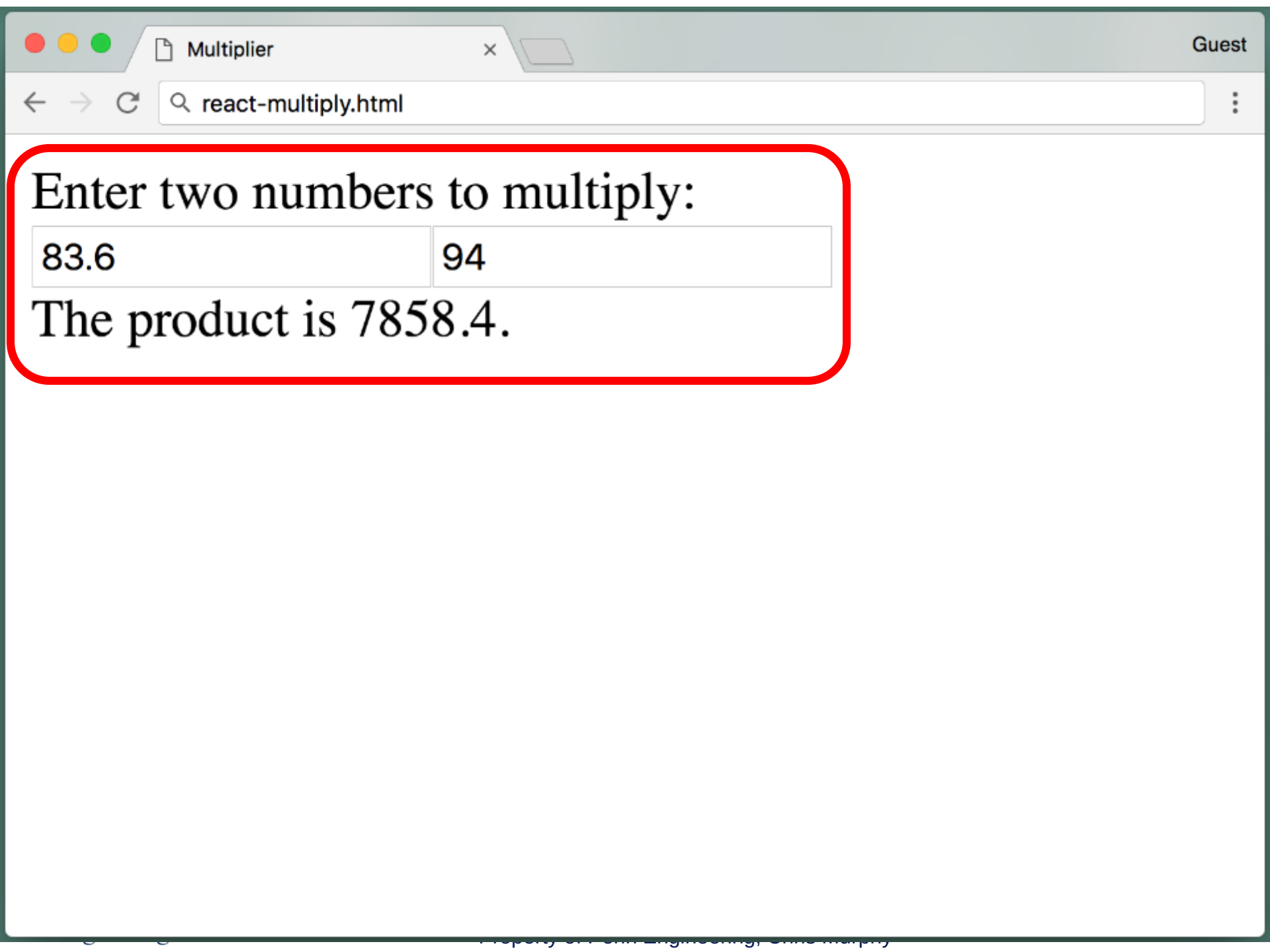

Multiplier
Guest
react-multiply.html
Enter two numbers to multiply:
8
9
The product is 72.

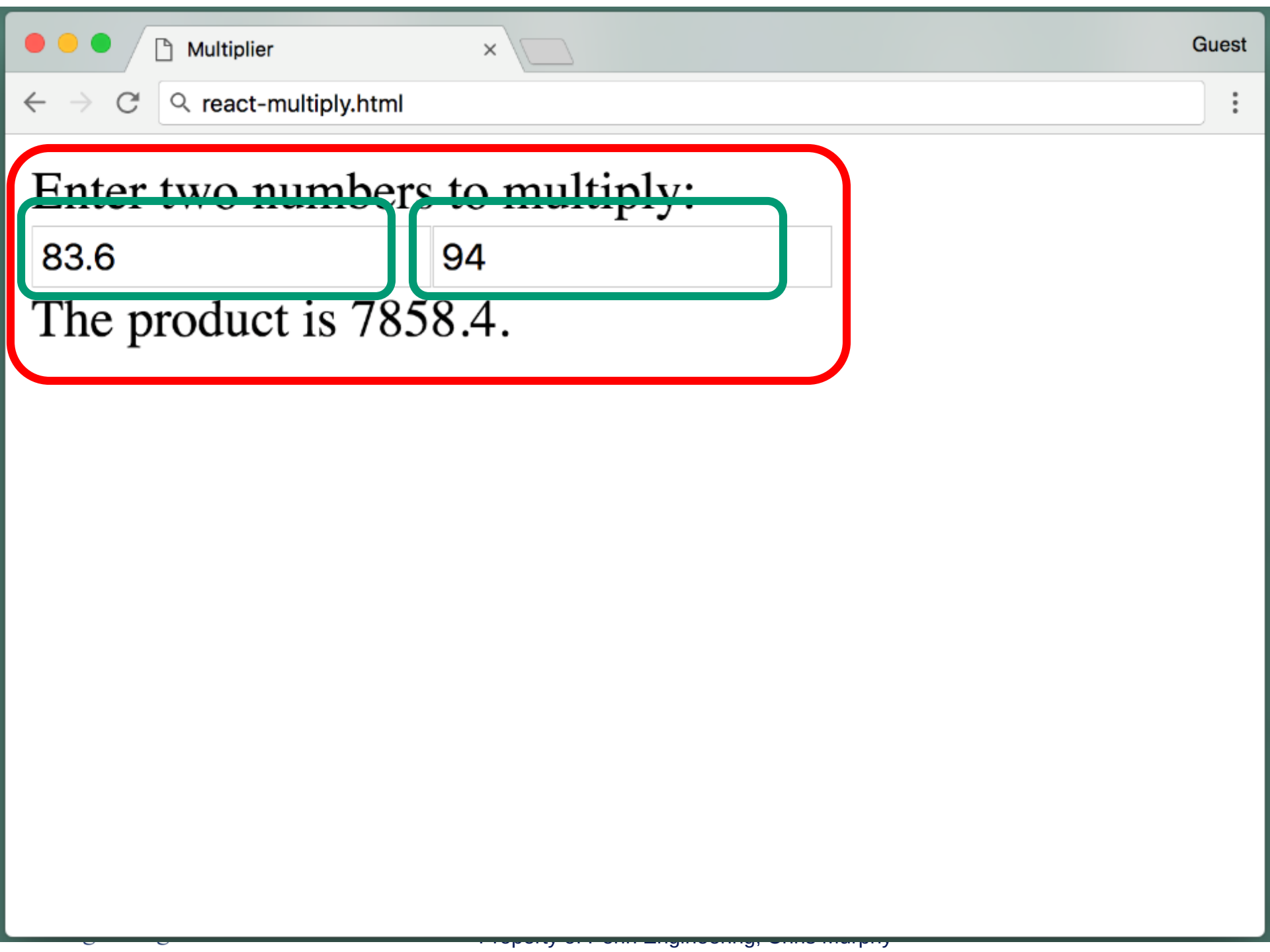

Multiplier
Guest
react-multiply.html
Enter two numbers to multiply:
8
94
The product is 752.

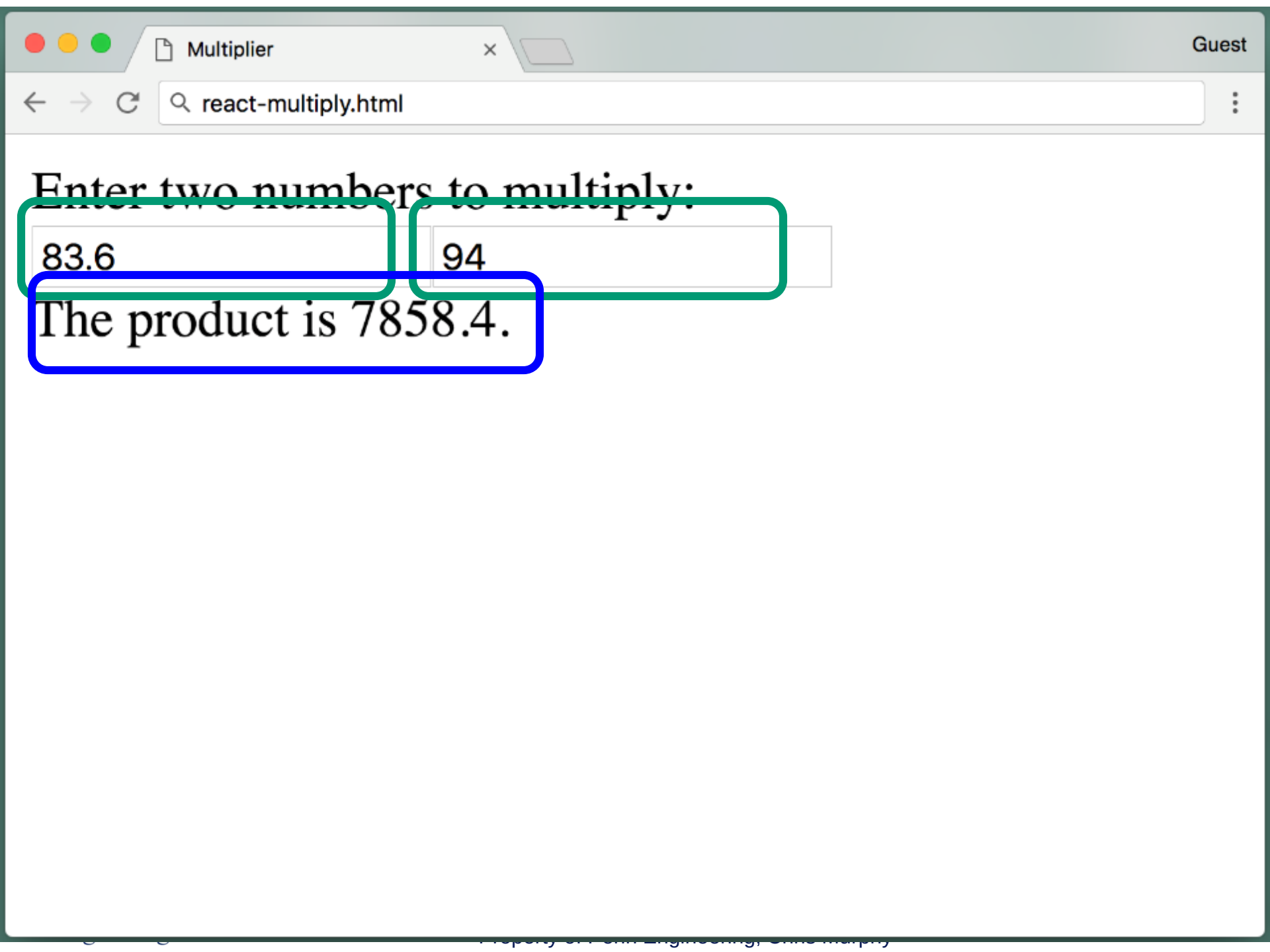

Multiplier
Guest
react-multiply.html
Enter two numbers to multiply:
83
94
The product is 7802.

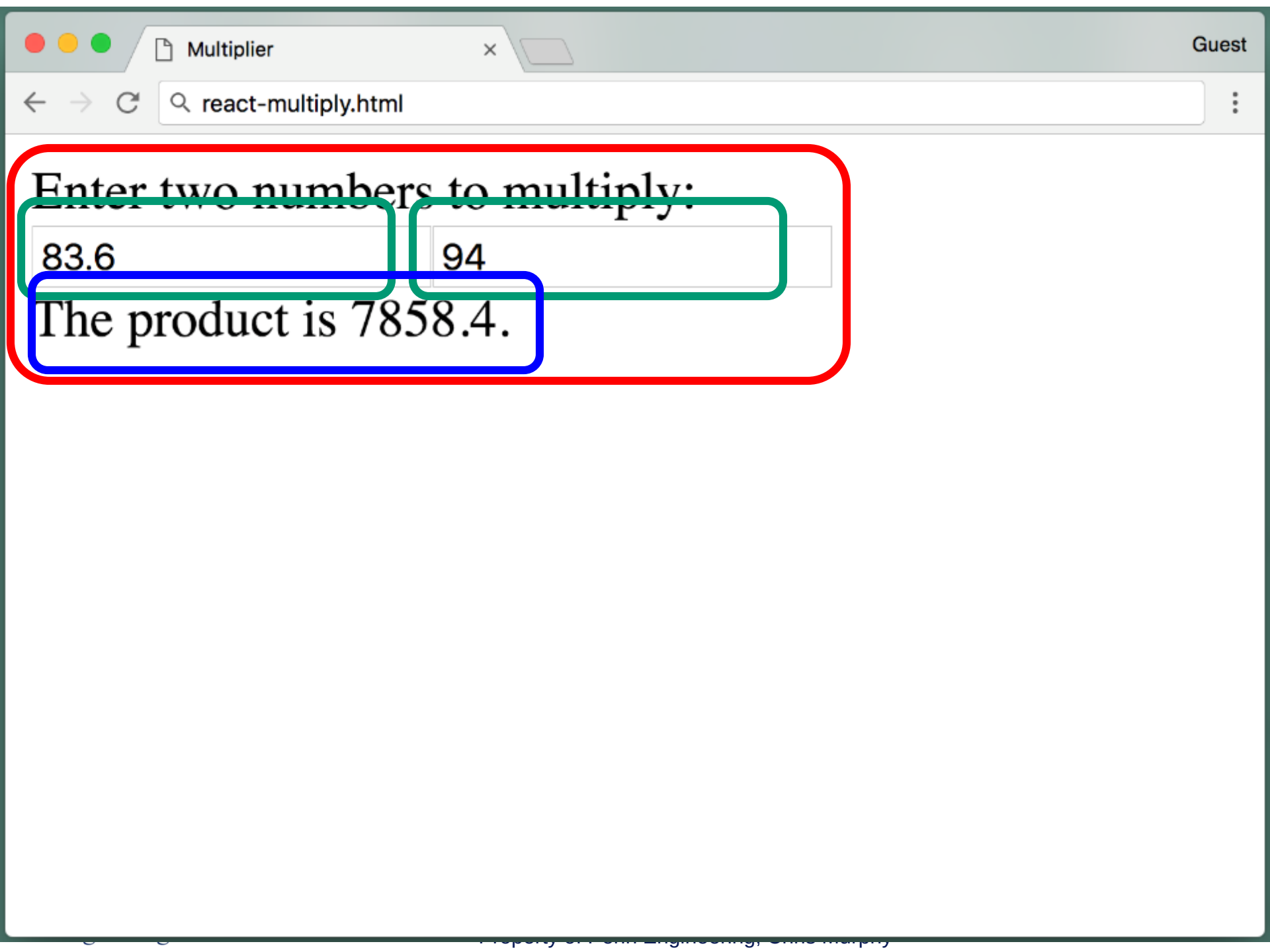

Multiplier
Guest
react-multiply.html
Enter two numbers to multiply:
83.6
94
The product is 7858.4.


Multiplier
Guest
react-multiply.html
Enter two numbers to multiply:
83.6
94
The product is 7858.4.


Multiplier
Guest
react-multiply.html
Enter two numbers to multiply:
83.6
94
The product is 7858.4.


Multiplier
Guest
react-multiply.html
Enter two numbers to multiply:
83.6
94
The product is 7858.4.


Multiplier
Guest
react-multiply.html
Enter two numbers to multiply:
83.6
94
The product is 7858.4.


Multiplier
Guest
react-multiply.html
Enter two numbers to multiply:
83.6
94
The product is 7858.4.

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                 product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                 product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                 product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                 product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                 product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                 product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                 product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                 product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                 product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                 product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                 product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                 product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                 product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                 product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                 product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                 product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                 product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                 product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                      product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                      product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                  product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                  product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                 product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                 product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                 product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                 product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                 product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                 product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                 product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                 product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 constructor(props) {
   super(props);
   this.state = { input1: 0, input2: 0, product: 0 };
   this.multiply = this.multiply.bind(this);
 }


 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                      product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                      product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

  . . .

render() {
    return (
      <div>
      <NumberInputField id="1" action={this.multiply}/>
      <NumberInputField id="2" action={this.multiply}/>
      <OutputField product={this.state.product}/>
      </div>
    );
  }
}
```

```
class Multiplier extends React.Component {

  . . .

render() {
    return (
      <div>
      <NumberInputField id="1" action={this.multiply}/>
      <NumberInputField id="2" action={this.multiply}/>
      <OutputField product={this.state.product}/>
      </div>
    );
  }
}
```

```
class Multiplier extends React.Component {

  . . .

render() {
    return (
      <div>
      <NumberInputField id="1" action={this.multiply}/>
      <NumberInputField id="2" action={this.multiply}/>
      <OutputField product={this.state.product}/>
      </div>
    );
  }
}
```

```
class Multiplier extends React.Component {

  . . .

render() {
    return (
      <div>
      <NumberInputField id="1" action={this.multiply}/>
      <NumberInputField id="2" action={this.multiply}/>
      <OutputField product={this.state.product}/>
      </div>
    );
  }
}
```

```
class Multiplier extends React.Component {

  . . .

render() {
    return (
      <div>
      <NumberInputField id="1" action={this.multiply}/>
      <NumberInputField id="2" action={this.multiply}/>
      <OutputField product={this.state.product}/>
      </div>
    );
  }
}
```

```
class Multiplier extends React.Component {

  . . .

render() {
    return (
      <div>
      <NumberInputField id="1" action={this.multiply}/>
      <NumberInputField id="2" action={this.multiply}/>
      <OutputField product={this.state.product}/>
      </div>
    );
  }
}
```

```
class Multiplier extends React.Component {

  . . .

render() {
    return (
      <div>
      <NumberInputField id="1" action={this.multiply}/>
      <NumberInputField id="2" action={this.multiply}/>
      <OutputField product={this.state.product}/>
      </div>
    );
  }
}
```

```
class Multiplier extends React.Component {
 . . .
   <NumberInputField id="1" action={this.multiply}/>
     . . .
```

```
class NumberInputField extends React.Component {
 constructor(props) {
   super(props);
   this.handleChange = this.handleChange.bind(this);
 }

 handleChange(e) {
   this.props.action(this.props.id, e.target.value);
 }

 render() {
   return(
     <input onChange={this.handleChange}></input>
   );
 }
}
```

```
class Multiplier extends React.Component {
 . . .
   <NumberInputField id="1" action={this.multiply}/>
     . . .
```

```
class NumberInputField extends React.Component {
 constructor(props) {
   super(props);
   this.handleChange = this.handleChange.bind(this);
 }

 handleChange(e) {
   this.props.action(this.props.id, e.target.value);
 }

 render() {
   return(
     <input onChange={this.handleChange}></input>
   );
 }
}
```

```
class Multiplier extends React.Component {
 . . .
   <NumberInputField id="1" action={this.multiply}/>
     . . .
```

```
class NumberInputField extends React.Component {
 constructor(props) {
   super(props);
   this.handleChange = this.handleChange.bind(this);
 }

 handleChange(e) {
   this.props.action(this.props.id, e.target.value);
 }

 render() {
   return(
     <input onChange={this.handleChange}></input>
   );
 }
}
```

```
class Multiplier extends React.Component {
 . . .
   <NumberInputField id="1" action={this.multiply}/>
     . . .
```

```
class NumberInputField extends React.Component {
 constructor(props) {
   super(props);
   this.handleChange = this.handleChange.bind(this);
 }

 handleChange(e) {
   this.props.action(this.props.id, e.target.value);
 }

 render() {
   return(
     <input onChange={this.handleChange}></input>
   );
 }
}
```

```
class Multiplier extends React.Component {
 . . .
   <NumberInputField id="1" action={this.multiply}/>
     . . .
```

```
class NumberInputField extends React.Component {
 constructor(props) {
   super(props);
   this.handleChange = this.handleChange.bind(this);
 }

 handleChange(e) {
   this.props.action(this.props.id, e.target.value);
 }

 render() {
   return(
     <input onChange={this.handleChange}></input>
   );
 }
}
```

```
class Multiplier extends React.Component {
 . . .
   <NumberInputField id="1" action={this.multiply}/>
     . . .
```

```
class NumberInputField extends React.Component {
 constructor(props) {
   super(props);
   this.handleChange = this.handleChange.bind(this);
 }

 handleChange(e) {
   this.props.action(this.props.id, e.target.value);
 }

 render() {
   return(
     <input onChange={this.handleChange}></input>
   );
 }
}
```

```
class Multiplier extends React.Component {
 . . .
   <NumberInputField id="1" action={this.multiply}/>
     . . .
```

```
class NumberInputField extends React.Component {
 constructor(props) {
   super(props);
   this.handleChange = this.handleChange.bind(this);
 }

 handleChange(e) {
   this.props.action(this.props.id, e.target.value);
 }

 render() {
   return(
     <input onChange={this.handleChange}></input>
   );
 }
}
```

```
class Multiplier extends React.Component {
 . . .
   <NumberInputField id="1" action={this.multiply}/>
     . . .
```

```
class NumberInputField extends React.Component {
 constructor(props) {
   super(props);
   this.handleChange = this.handleChange.bind(this);
 }

 handleChange(e) {
   this.props.action(this.props.id, e.target.value);
 }

 render() {
   return(
     <input onChange={this.handleChange}></input>
   );
 }
}
```

```
class Multiplier extends React.Component {
 . . .
   <NumberInputField id="1" action={this.multiply}/>
     . . .
```

```
class NumberInputField extends React.Component {
 constructor(props) {
   super(props);
   this.handleChange = this.handleChange.bind(this);
 }

 handleChange(e) {
   this.props.action(this.props.id, e.target.value);
 }

 render() {
   return(
     <input onChange={this.handleChange}></input>
   );
 }
}
```

```
class Multiplier extends React.Component {
 . . .
   <NumberInputField id="1" action={this.multiply}/>
     . . .
```

```
class NumberInputField extends React.Component {
 constructor(props) {
   super(props);
   this.handleChange = this.handleChange.bind(this);
 }

 handleChange(e) {
   this.props.action(this.props.id, e.target.value);
 }

 render() {
   return(
     <input onChange={this.handleChange}></input>
   );
 }
}
```

```
class Multiplier extends React.Component {
 . . .
   <NumberInputField id="1" action={this.multiply}/>
     . . .
```

```
class NumberInputField extends React.Component {
 constructor(props) {
   super(props);
   this.handleChange = this.handleChange.bind(this);
 }

 handleChange(e) {
   this.props.action(this.props.id, e.target.value);
 }

 render() {
   return(
     <input onChange={this.handleChange}></input>
   );
 }
}
```

```
class Multiplier extends React.Component {
 . . .
   <NumberInputField id="1" action={this.multiply}/>
     . . .
```

```
class NumberInputField extends React.Component {
 constructor(props) {
   super(props);
   this.handleChange = this.handleChange.bind(this);
 }

 handleChange(e) {
   this.props.action(this.props.id, e.target.value);
 }

 render() {
   return(
     <input onChange={this.handleChange}></input>
   );
 }
}
```

```
class Multiplier extends React.Component {
 . . .
   <NumberInputField id="1" action={this.multiply}/>
     . . .
```

```
class NumberInputField extends React.Component {
 constructor(props) {
   super(props);
   this.handleChange = this.handleChange.bind(this);
 }

 handleChange(e) {
   this.props.action(this.props.id, e.target.value);
 }

 render() {
   return(
     <input onChange={this.handleChange}></input>
   );
 }
}
```

```
class Multiplier extends React.Component {

 . . .

 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                      product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                      product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 . . .

 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                      product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                      product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {

 . . .

 multiply(id, val) {
   if (id == 1) {
     this.setState( { input1: val,
                      product: val * this.state.input2 } );
   }
   else if (id == 2) {
     this.setState( { input2: val,
                      product: this.state.input1 * val } );
   }
 }

. . .
```

```
class Multiplier extends React.Component {
 . . .
    <OutputField product={this.state.product}/>
     . . .
```

```
class OutputField extends React.Component {
  render() {
    return(
       <div>The product is {this.props.product}.
       </div>
    );
  }
}
```

```
ReactDOM.render(<Multiplier/>,
    document.getElementById('container'));
```

```
class Multiplier extends React.Component {
 . . .
    <OutputField product={this.state.product}/>
      . . .
```

```
class OutputField extends React.Component {
  render() {
    return(
       <div>The product is {this.props.product}.
       </div>
    );
  }
}
```

```
ReactDOM.render(<Multiplier/>,
    document.getElementById('container'));
```

```
class Multiplier extends React.Component {
 . . .
    <OutputField product={this.state.product}/>
      . . .
```

```
class OutputField extends React.Component {
  render() {
    return(
       <div>The product is {this.props.product}.
       </div>
    );
  }
}
```

```
ReactDOM.render(<Multiplier/>,
    document.getElementById('container'));
```

```
class Multiplier extends React.Component {
 . . .
    <OutputField product={this.state.product}/>
     . . .
```

```
class OutputField extends React.Component {
  render() {
    return(
       <div>The product is {this.props.product}.
       </div>
    );
  }
}
```

```
ReactDOM.render(<Multiplier/>,
    document.getElementById('container'));
```

```
class Multiplier extends React.Component {
 . . .
    <OutputField product={this.state.product}/>
     . . .
```

```
class OutputField extends React.Component {
  render() {
    return(
       <div>The product is {this.props.product}.
       </div>
    );
  }
}
```

```
ReactDOM.render(<Multiplier/>,
    document.getElementById('container'));
```

```
class Multiplier extends React.Component {
 . . .
    <OutputField product={this.state.product}/>
     . . .
```

```
class OutputField extends React.Component {
  render() {
    return(
       <div>The product is {this.props.product}.
       </div>
    );
  }
}
```

```
ReactDOM.render(<Multiplier/>,
    document.getElementById('container'));
```

```
class Multiplier extends React.Component {
 . . .
    <OutputField product={this.state.product}/>
     . . .
```

```
class OutputField extends React.Component {
  render() {
    return(
       <div>The product is {this.props.product}.
       </div>
    );
  }
}
```

```
ReactDOM.render(<Multiplier/>,
    document.getElementById('container'));
```

```
class Multiplier extends React.Component {
 . . .
    <OutputField product={this.state.product}/>
     . . .
```

```
class OutputField extends React.Component {
  render() {
    return(
       <div>The product is {this.props.product}.
       </div>
    );
  }
}
```

```
ReactDOM.render(<Multiplier/>,
    document.getElementById('container'));
```

Review

- React allows us to create reusable, modularized components that can be combined to form web applications

- Components can communicate with each other via callback methods that are set as props