Penn Engineering

ONLINE LEARNING

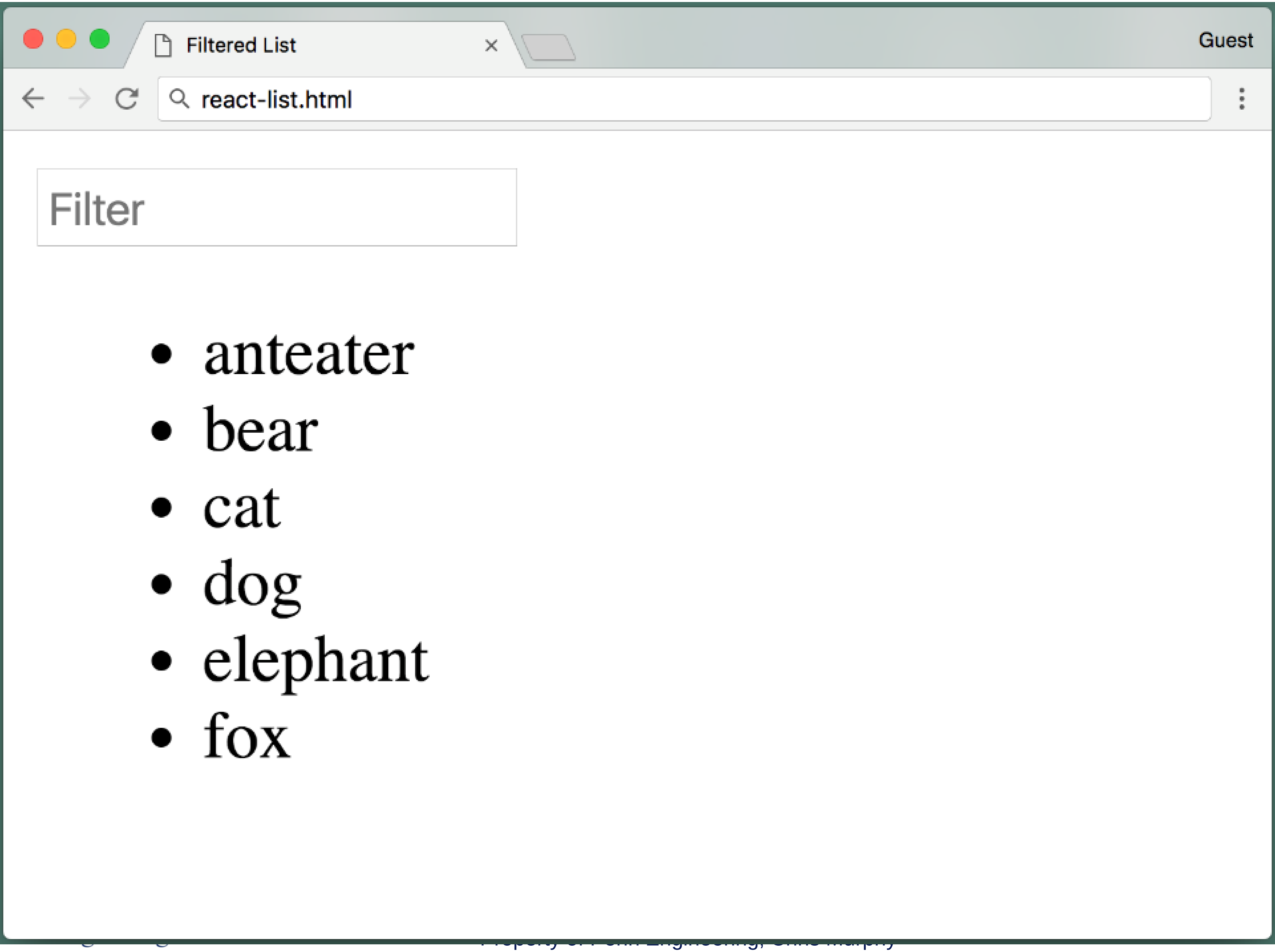
Video 3.4

React Component Interaction

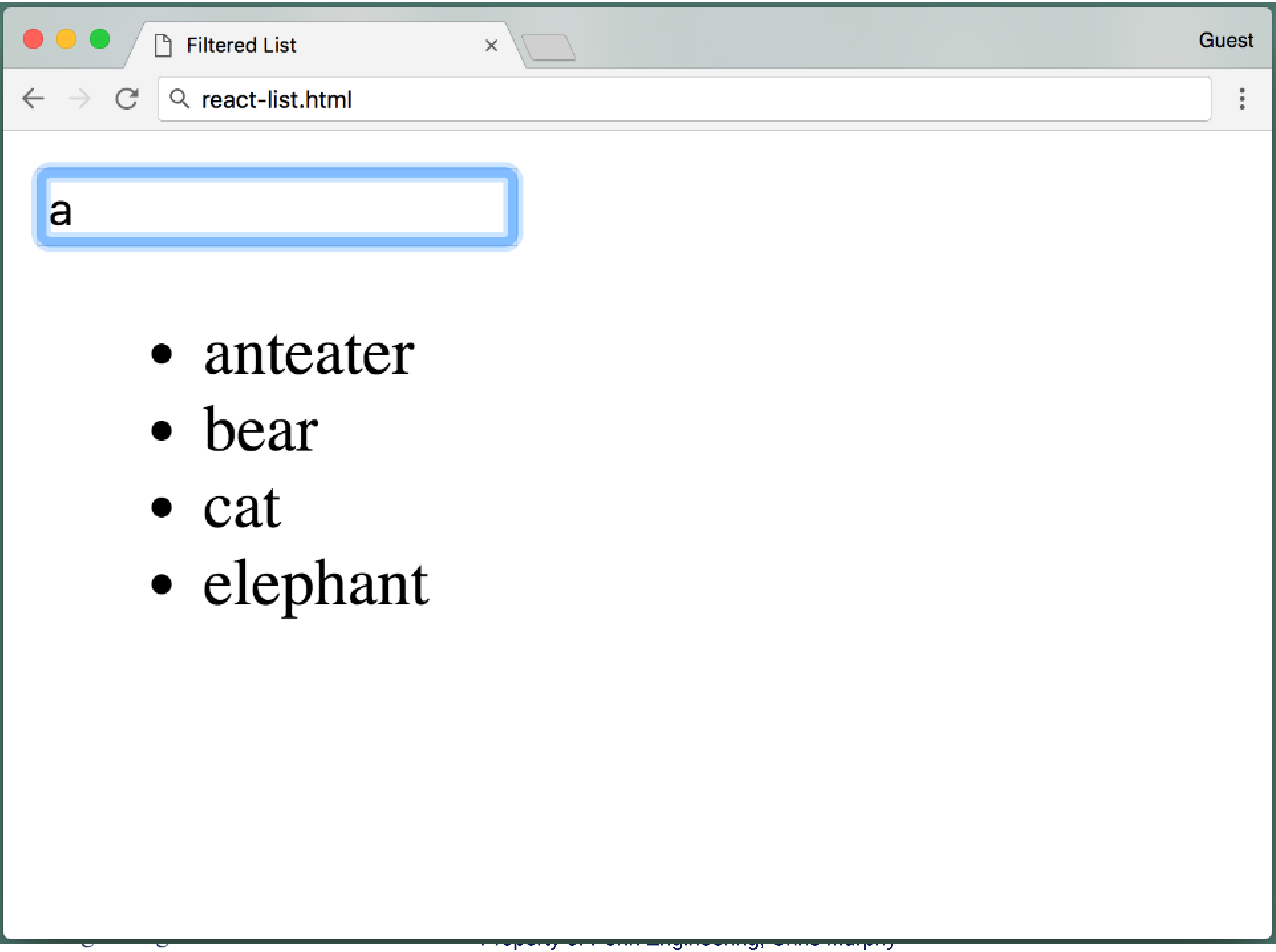
Chris Murphy

Review

- We can bind user events in HTML elements to callback functions in React components
- When we invoke a component's **`setState`** function, the **`render`** function will automatically be called and the component's appearance can change accordingly
- Re-rendering one component may necessitate re-rendering others as well

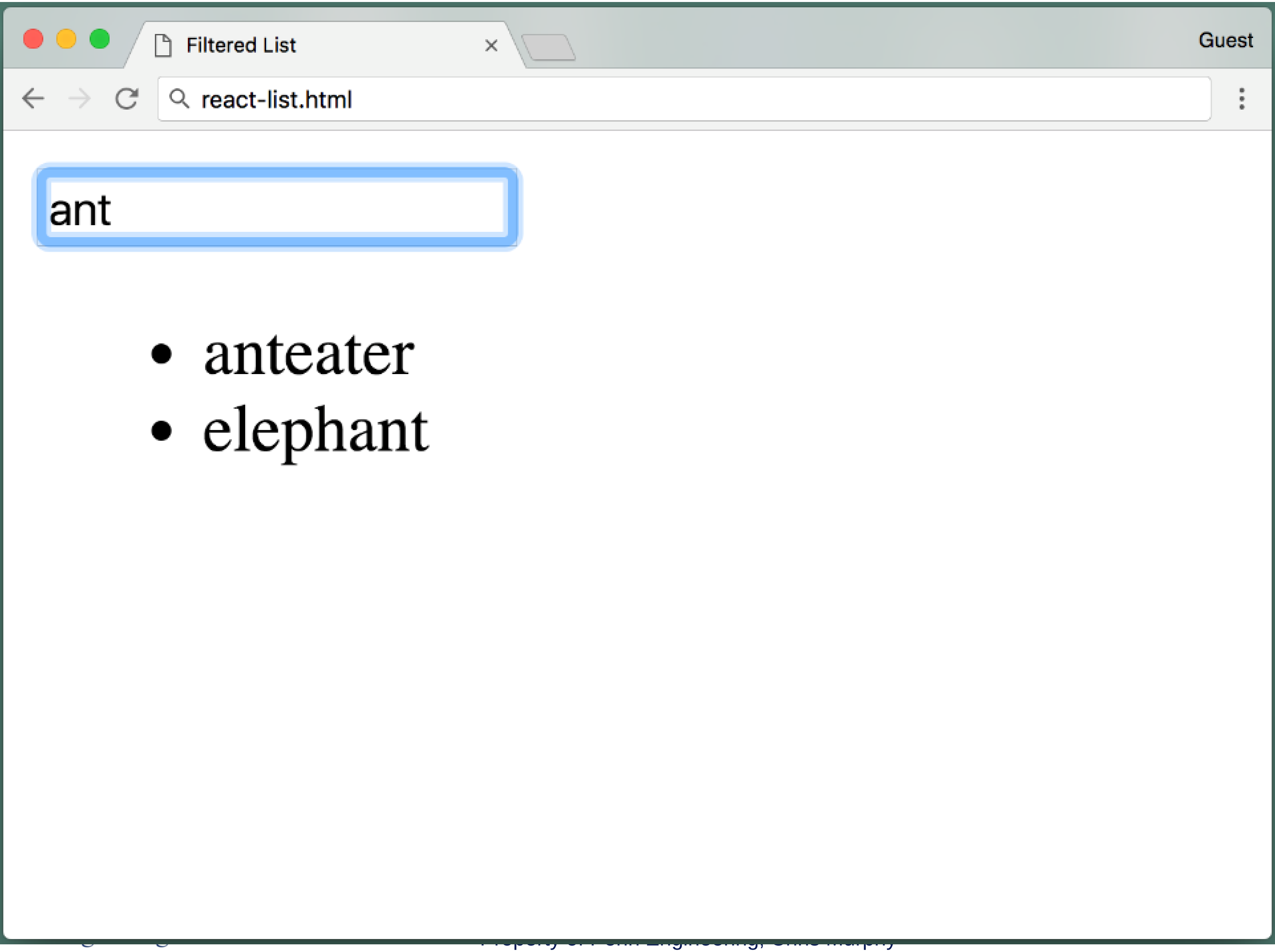


- anteater
- bear
- cat
- dog
- elephant
- fox



a

- anteater
- bear
- cat
- elephant



ant

- anteater
- elephant

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                       "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                    currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
            return item.search(input.target.value) !== -1;
        });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                       "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                    currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
            return item.search(input.target.value) !== -1;
        });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                       "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                    currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
            return item.search(input.target.value) !== -1;
        });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                        "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                     currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
             return item.search(input.target.value) !== -1;
         });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                        "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                     currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
            return item.search(input.target.value) !== -1;
        });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                       "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                    currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
            return item.search(input.target.value) !== -1;
        });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                        "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                    currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
            return item.search(input.target.value) !== -1;
         });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                        "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                     currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
             return item.search(input.target.value) !== -1;
         });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                        "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                     currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
             return item.search(input.target.value) !== -1;
         });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                        "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                     currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
             return item.search(input.target.value) !== -1;
         });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                        "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                     currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
             return item.search(input.target.value) !== -1;
         });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                        "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                     currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
             return item.search(input.target.value) !== -1;
         });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                        "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                     currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
             return item.search(input.target.value) !== -1;
         });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                       "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                    currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
            return item.search(input.target.value) !== -1;
         });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                        "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                     currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
             return item.search(input.target.value) !== -1;
         });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                        "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                     currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
             return item.search(input.target.value) !== -1;
         });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                        "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                     currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
            return item.search(input.target.value) !== -1;
        });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                       "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                     currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
             return item.search(input.target.value) !== -1;
         });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {

  constructor(props) {
     super(props);
     var allItems = [ "Anteater", "Bear", "Cat",
                       "Dog", "Elephant" ];
     this.state = { initialItems: allItems,
                     currentItems: allItems };
  }

  filterList(input) { // callback function
    var updatedList = this.state.initialItems;

    updatedList = updatedList.filter(function(item){
            return item.search(input.target.value) !== -1;
         });

    this.setState( { currentItems: updatedList } );
  }


  . . .
```

```
class FilteredList extends React.Component {
  . . .
  render(){
   return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
    );
  }
};

class ListItems extends React.Component {
   render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
   }
};

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

```
class FilteredList extends React.Component {
  . . .
 render(){
   return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
    );
  }
};

class ListItems extends React.Component {
  render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
  }
};

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

```
class FilteredList extends React.Component {
   . . .
  render(){
    return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
     );
  }
};

class ListItems extends React.Component {
  render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
   }
};

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

```
class FilteredList extends React.Component {
   . . .
  render(){
    return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
     );
  }
};

class ListItems extends React.Component {
   render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
   }
};

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

```
class FilteredList extends React.Component {
   . . .
  render(){
    return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
     );
  }
};

class ListItems extends React.Component {
   render(){
      return (
         <ul> { this.props.items.map(function(item) {
               return <li key={item}> {item} </li> } ) } </ul>
      );
   }
};

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

```
class FilteredList extends React.Component {
  . . .
  render(){
    return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
          <ListItems items={this.state.currentItems}/>
      </div>
     );
  }
};

class ListItems extends React.Component {
   render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
   }
};

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

```
class FilteredList extends React.Component {
  . . .
  render(){
   return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
    );
  }
};

class ListItems extends React.Component {
  render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
  }
};

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

```
class FilteredList extends React.Component {
  . . .
 render(){
   return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
    );
 }
};

class ListItems extends React.Component {
  render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
  }
};

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

```
class FilteredList extends React.Component {
   . . .
  render(){
   return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
    );
  }
};

class ListItems extends React.Component {
  render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
  }
};

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

```
class FilteredList extends React.Component {
  . . .
  render(){
   return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
    );
  }
};

class ListItems extends React.Component {
   render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
   }
};

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

```
class FilteredList extends React.Component {
   . . .
  render(){
    return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
    );
  }
};

class ListItems extends React.Component {
  render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
  }
};

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

```
class FilteredList extends React.Component {
  . . .
 render(){
   return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
    );
  }
};

class ListItems extends React.Component {
  render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
  }
};

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

```
class FilteredList extends React.Component {
  . . .
  render(){
    return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
     );
  }
};

class ListItems extends React.Component {
   render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
   }
};

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

```
class FilteredList extends React.Component {
   . . .
  render(){
    return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
    );
  }
};

class ListItems extends React.Component {
   render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
   }
};

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

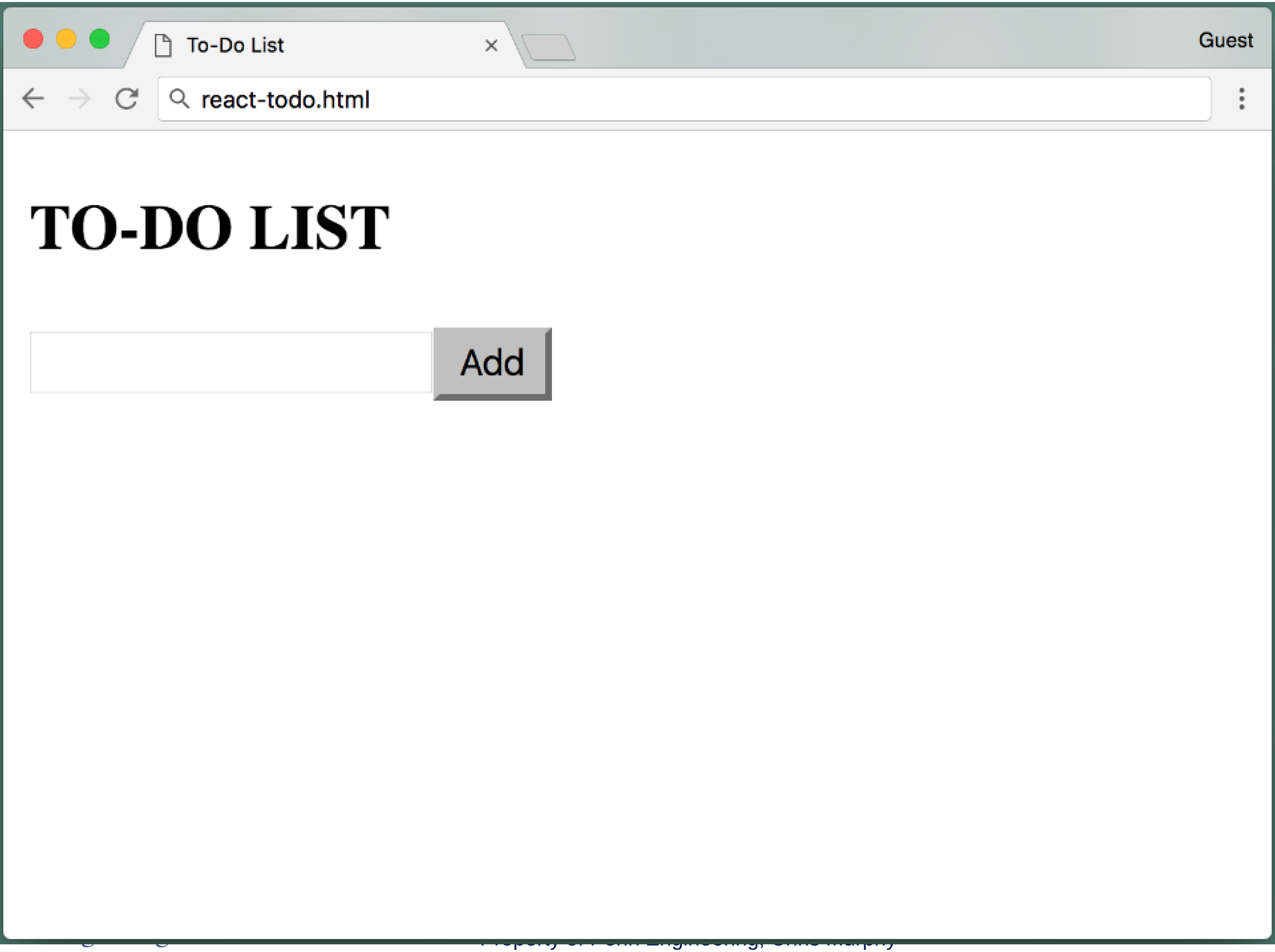
```
class FilteredList extends React.Component {
   . . .
  render(){
    return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
    );
  }
};

class ListItems extends React.Component {
   render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
   }
};

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

```
class FilteredList extends React.Component {
   . . .
  render(){
    return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
    );
  }
};

class ListItems extends React.Component {
  render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
  }
};

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

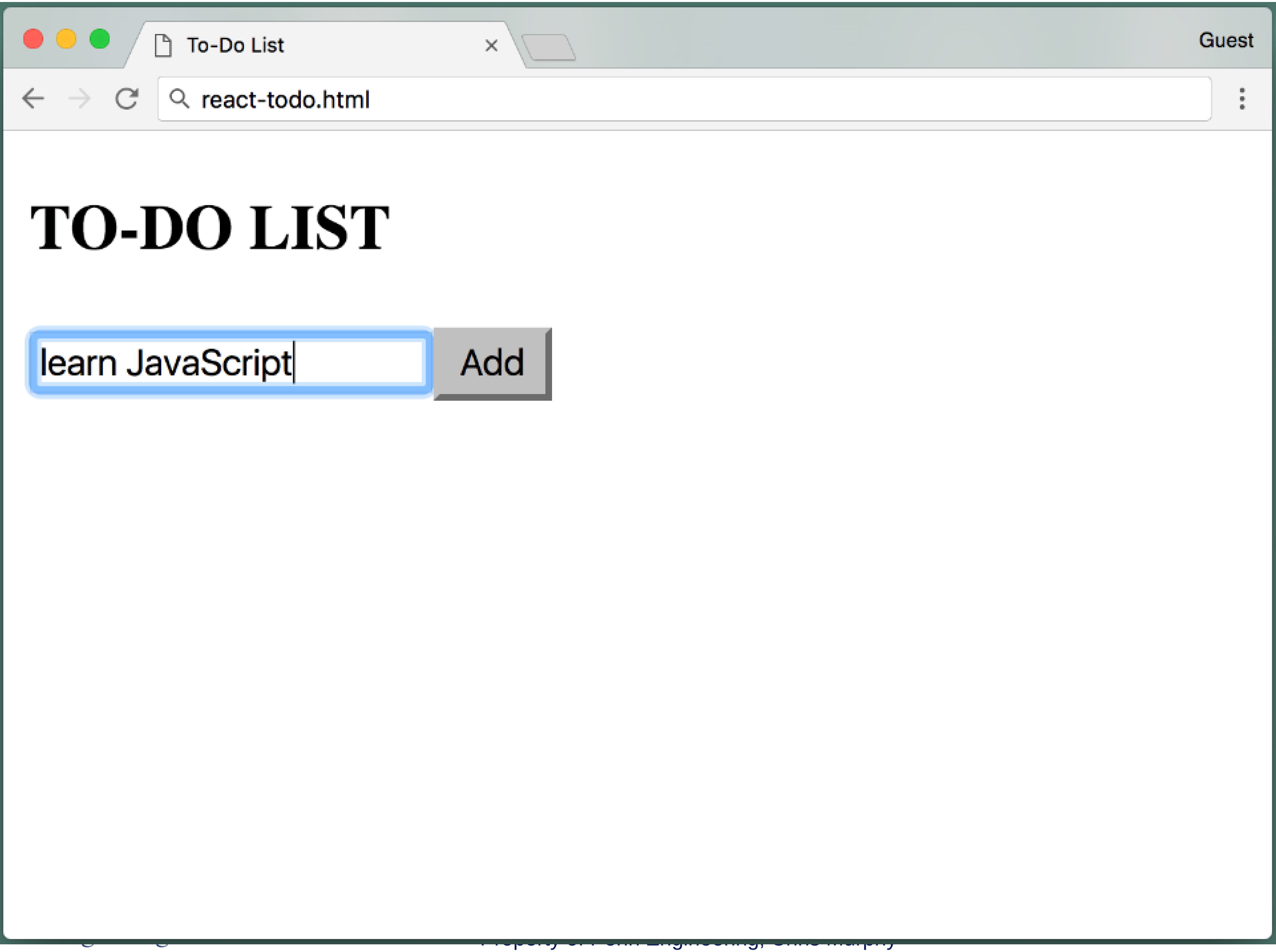
```
class FilteredList extends React.Component {
  . . .
  render(){
    return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
    );
  }
};

class ListItems extends React.Component {
  render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
  }
};

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

```
class FilteredList extends React.Component {
   . . .
  render(){
   return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
    );
  }
};

class ListItems extends React.Component {
  render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
  }
};

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

```
class FilteredList extends React.Component {
   . . .
  render(){
    return (
      <div><input type="text" placeholder="Search"
                  onChange={this.filterList.bind(this)}/>
           <ListItems items={this.state.currentItems}/>
      </div>
    );
  }
};

class ListItems extends React.Component {
  render(){
      return (
         <ul> { this.props.items.map(function(item) {
                return <li key={item}> {item} </li> } ) } </ul>
      );
   }
};

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

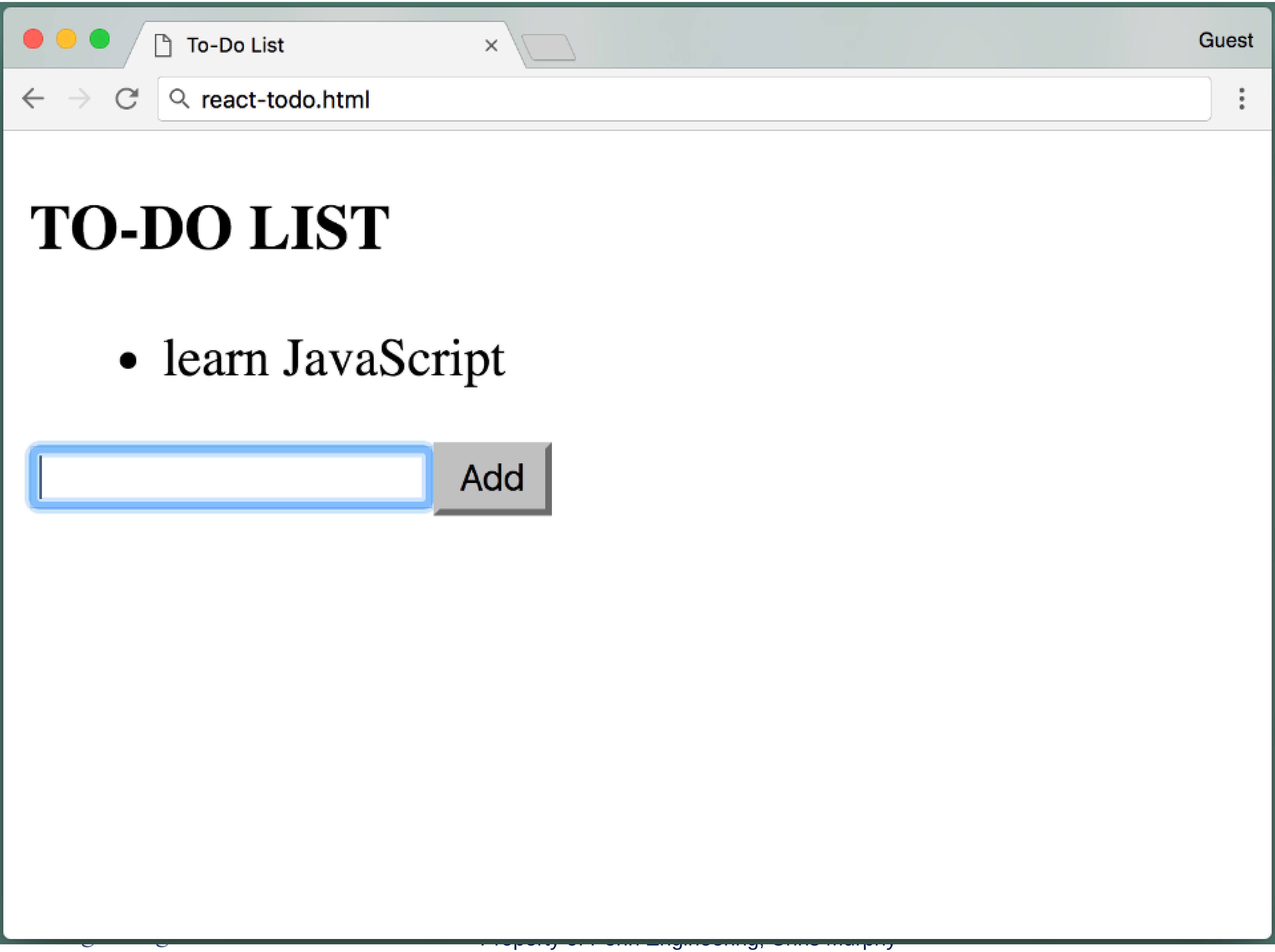


TO-DO LIST



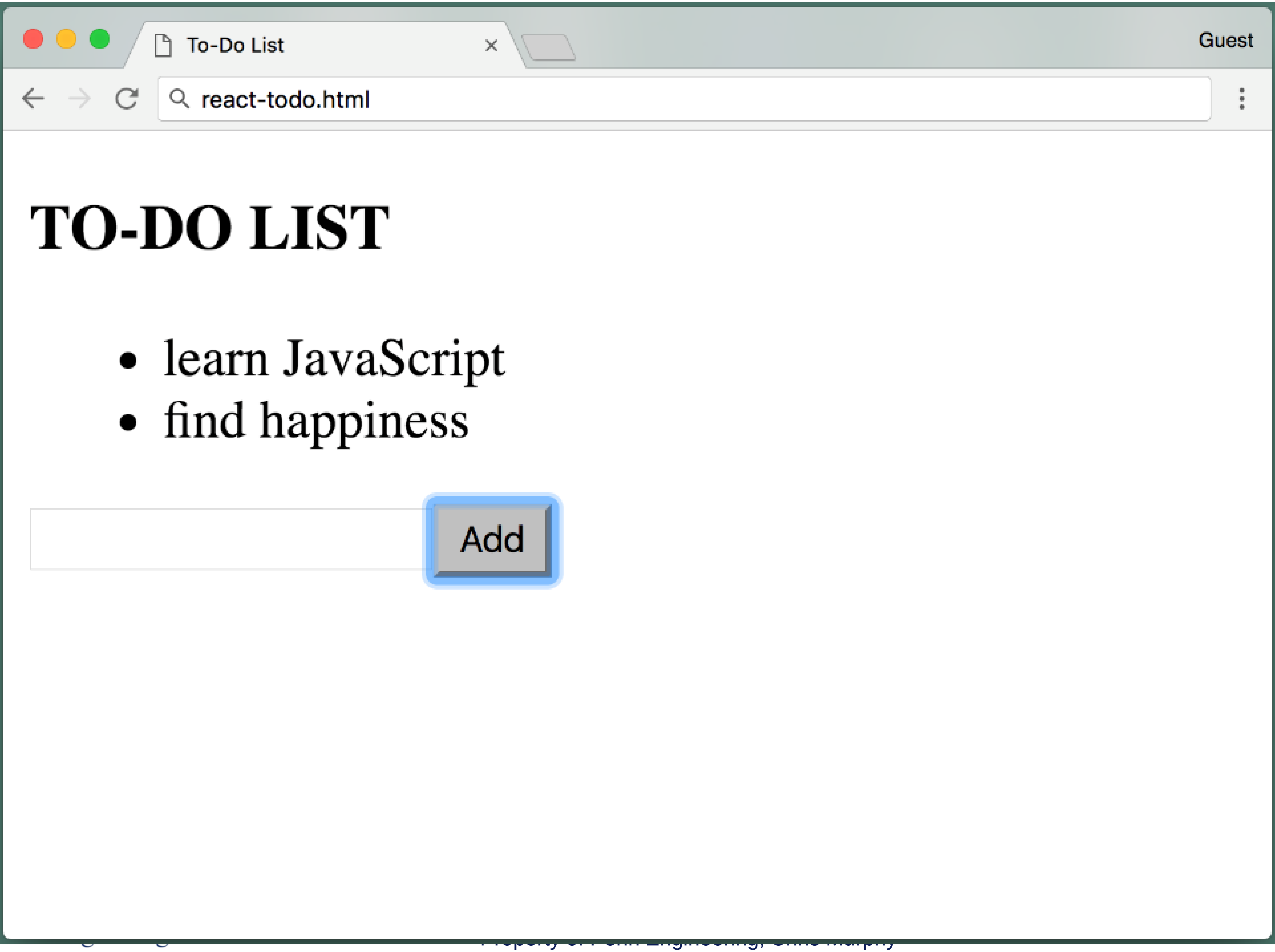
TO-DO LIST

learn JavaScript



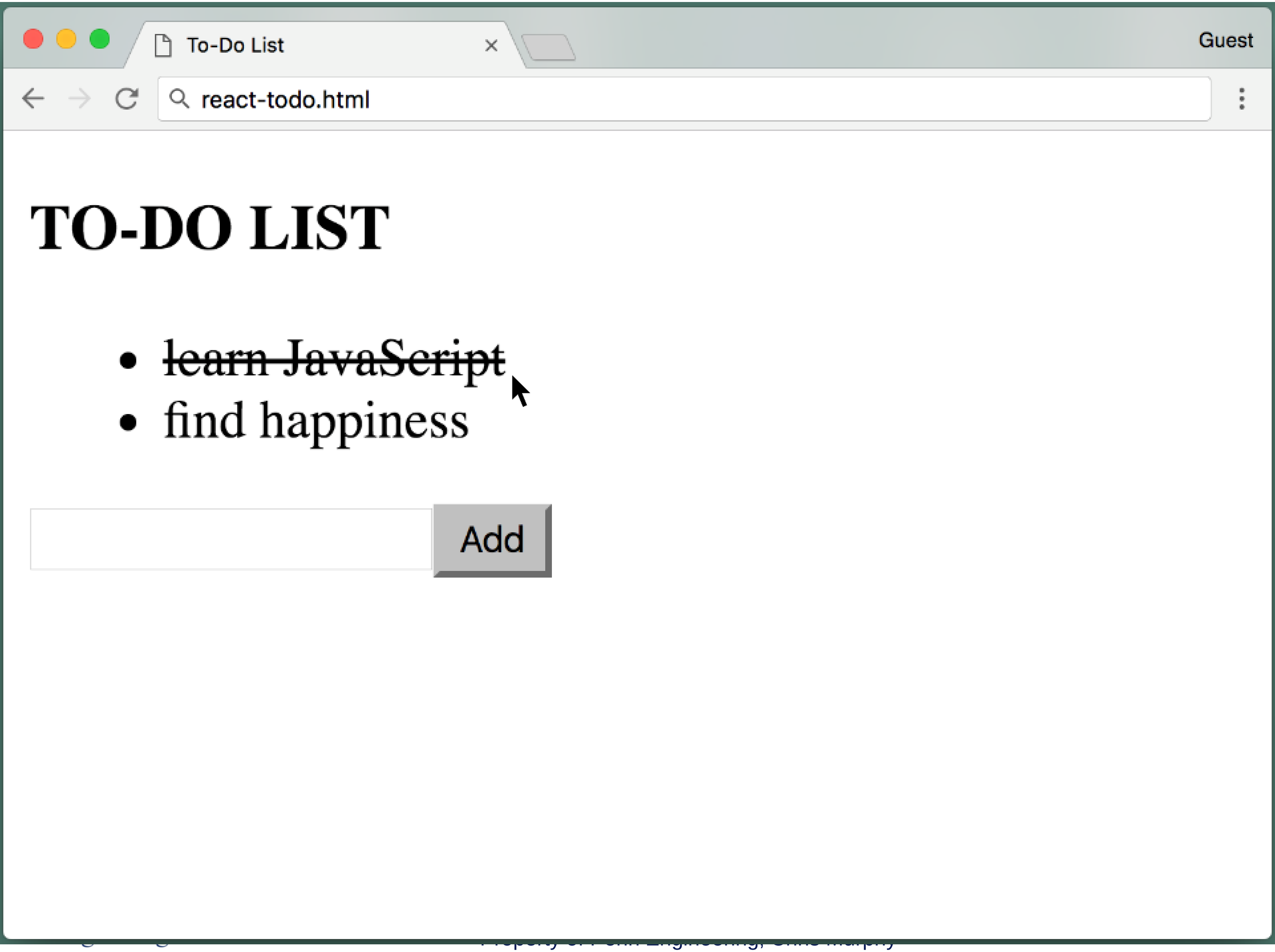
TO-DO LIST

- learn JavaScript



TO-DO LIST

- learn JavaScript
- find happiness



TO-DO LIST

- ~~learn JavaScript~~
- find happiness

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
```

```
class TodoApp extends React.Component {
 constructor(props) {
   super(props);
   this.state = {items: [], text: ' ', id: 0};
 }

 handleChange(e) {
   this.setState( { text: e.target.value } );
 }

 handleSubmit(e) {
   e.preventDefault(); // so as not to reload the page
   var newItem = { text: this.state.text,
                     id: this.state.id };
   this.setState( {
     items: this.state.items.concat(newItem),
     text: ' ',
     id: this.state.id + 1
   });
 }
 class TodoApp extends React.Component {
   . . .

  render() {
    return (
      <div>
        <h3>TO-DO LIST</h3>
        <TodoList items={this.state.items} />
        <form onSubmit={this.handleSubmit.bind(this)}>
        <input onChange={this.handleChange.bind(this)}
               value={this.state.text} />
        <button>Add</button>
        </form>
      </div>
    );
  }
};
```

```
 class TodoApp extends React.Component {
   . . .

  render() {
    return (
      <div>
        <h3>TO-DO LIST</h3>
        <TodoList items={this.state.items} />
        <form onSubmit={this.handleSubmit.bind(this)}>
        <input onChange={this.handleChange.bind(this)}
               value={this.state.text} />
        <button>Add</button>
        </form>
      </div>
    );
  }
};
```

```
 class TodoApp extends React.Component {
   . . .

  render() {
    return (
      <div>
        <h3>TO-DO LIST</h3>
        <TodoList items={this.state.items} />
        <form onSubmit={this.handleSubmit.bind(this)}>
        <input onChange={this.handleChange.bind(this)}
               value={this.state.text} />
        <button>Add</button>
        </form>
      </div>
    );
  }
};
```

```
 class TodoApp extends React.Component {
   . . .

  render() {
    return (
      <div>
        <h3>TO-DO LIST</h3>
        <TodoList items={this.state.items} />
        <form onSubmit={this.handleSubmit.bind(this)}>
        <input onChange={this.handleChange.bind(this)}
               value={this.state.text} />
        <button>Add</button>
        </form>
      </div>
    );
  }
};
```

```
 class TodoApp extends React.Component {
   . . .

  render() {
    return (
      <div>
        <h3>TO-DO LIST</h3>
        <TodoList items={this.state.items} />
        <form onSubmit={this.handleSubmit.bind(this)}>
        <input onChange={this.handleChange.bind(this)}
               value={this.state.text} />
        <button>Add</button>
        </form>
      </div>
    );
  }
};
```

```
 class TodoApp extends React.Component {
   . . .

  render() {
    return (
      <div>
        <h3>TO-DO LIST</h3>
        <TodoList items={this.state.items} />
        <form onSubmit={this.handleSubmit.bind(this)}>
        <input onChange={this.handleChange.bind(this)}
               value={this.state.text} />
        <button>Add</button>
        </form>
      </div>
    );
  }
};
```

```
 class TodoApp extends React.Component {
   . . .

  render() {
    return (
      <div>
        <h3>TO-DO LIST</h3>
        <TodoList items={this.state.items} />
        <form onSubmit={this.handleSubmit.bind(this)}>
        <input onChange={this.handleChange.bind(this)}
               value={this.state.text} />
        <button>Add</button>
        </form>
      </div>
    );
  }
};
```

```
 class TodoApp extends React.Component {
   . . .

  render() {
    return (
      <div>
        <h3>TO-DO LIST</h3>
        <TodoList items={this.state.items} />
        <form onSubmit={this.handleSubmit.bind(this)}>
        <input onChange={this.handleChange.bind(this)}
               value={this.state.text} />
        <button>Add</button>
        </form>
      </div>
    );
  }
};
```

```
 class TodoApp extends React.Component {
   . . .

  render() {
    return (
      <div>
        <h3>TO-DO LIST</h3>
        <TodoList items={this.state.items} />
        <form onSubmit={this.handleSubmit.bind(this)}>
        <input onChange={this.handleChange.bind(this)}
               value={this.state.text} />
        <button>Add</button>
        </form>
      </div>
    );
  }
};
```

```
 class TodoApp extends React.Component {
   . . .

  render() {
    return (
      <div>
        <h3>TO-DO LIST</h3>
        <TodoList items={this.state.items} />
        <form onSubmit={this.handleSubmit.bind(this)}>
        <input onChange={this.handleChange.bind(this)}
               value={this.state.text} />
        <button>Add</button>
        </form>
      </div>
    );
  }
};
```

```
class TodoApp extends React.Component {
  . . .
 render() {
   return (
        . . .
       <TodoList items={this.state.items} />
        . . .
```

```
class TodoList extends React.Component {
 render() {
   return (
     <ul>
       {this.props.items.map(function (item) {
         return
           <TodoItem id={item.id} text={item.text}/>
       })}
     </ul>
   );
 }
}
```

```
class TodoApp extends React.Component {
  . . .
 render() {
   return (
        . . .
       <TodoList items={this.state.items} />
        . . .
```

```
class TodoList extends React.Component {
 render() {
   return (
     <ul>
       {this.props.items.map(function (item) {
         return
           <TodoItem id={item.id} text={item.text}/>
       })}
     </ul>
   );
 }
}
```

```
class TodoApp extends React.Component {
  . . .
 render() {
   return (
        . . .
       <TodoList items={this.state.items} />
        . . .
```

```
class TodoList extends React.Component {
 render() {
   return (
     <ul>
       {this.props.items.map(function (item) {
         return
           <TodoItem id={item.id} text={item.text}/>
       })}
     </ul>
   );
 }
}
```

```
class TodoApp extends React.Component {
  . . .
 render() {
   return (
        . . .
       <TodoList items={this.state.items} />
        . . .
```

```
class TodoList extends React.Component {
 render() {
   return (
     <ul>
       {this.props.items.map(function (item) {
         return
           <TodoItem id={item.id} text={item.text}/>
       })}
     </ul>
   );
 }
}
```

```
class TodoApp extends React.Component {
  . . .
 render() {
   return (
        . . .
       <TodoList items={this.state.items} />
        . . .
```

```
class TodoList extends React.Component {
 render() {
   return (
     <ul>
       {this.props.items.map(function (item) {
         return
           <TodoItem id={item.id} text={item.text}/>
       })}
     </ul>
   );
 }
}
```

```
class TodoApp extends React.Component {
  . . .
 render() {
   return (
        . . .
       <TodoList items={this.state.items} />
        . . .
```

```
class TodoList extends React.Component {
 render() {
   return (
     <ul>
       {this.props.items.map(function (item) {
         return
           <TodoItem id={item.id} text={item.text}/>
       })}
     </ul>
   );
 }
}
```

```
class TodoApp extends React.Component {
  . . .
 render() {
   return (
        . . .
       <TodoList items={this.state.items} />
        . . .
```

```
class TodoList extends React.Component {
 render() {
   return (
     <ul>
       {this.props.items.map(function (item) {
         return
           <TodoItem id={item.id} text={item.text}/>
       })}
     </ul>
   );
 }
}
```

```
class TodoApp extends React.Component {
  . . .
 render() {
   return (
        . . .
       <TodoList items={this.state.items} />
        . . .
```

```
class TodoList extends React.Component {
 render() {
   return (
     <ul>
       {this.props.items.map(function (item) {
         return
           <TodoItem id={item.id} text={item.text}/>
       })}
     </ul>
   );
 }
}
```

```
class TodoApp extends React.Component {
  . . .
 render() {
   return (
        . . .
       <TodoList items={this.state.items} />
        . . .
```

```
class TodoList extends React.Component {
 render() {
   return (
     <ul>
       {this.props.items.map(function (item) {
         return
           <TodoItem id={item.id} text={item.text}/>
       })}
     </ul>
   );
 }
}
```

```
class TodoApp extends React.Component {
  . . .
 render() {
   return (
        . . .
       <TodoList items={this.state.items} />
        . . .
```

```
class TodoList extends React.Component {
 render() {
   return (
     <ul>
       {this.props.items.map(function (item) {
         return
           <TodoItem id={item.id} text={item.text}/>
       })}
     </ul>
   );
 }
}
```

```
class TodoApp extends React.Component {
  . . .
 render() {
   return (
        . . .
       <TodoList items={this.state.items} />
        . . .
```

```
class TodoList extends React.Component {
 render() {
   return (
     <ul>
       {this.props.items.map(function (item) {
         return
           <TodoItem id={item.id} text={item.text}/>
       })}
     </ul>
   );
 }
}
```

```
class TodoApp extends React.Component {
  . . .
 render() {
   return (
        . . .
       <TodoList items={this.state.items} />
        . . .
```

```
class TodoList extends React.Component {
 render() {
   return (
     <ul>
       {this.props.items.map(function (item) {
         return
           <TodoItem id={item.id} text={item.text}/>
       })}
     </ul>
   );
 }
}
```

```
class TodoApp extends React.Component {
  . . .
 render() {
   return (
        . . .
       <TodoList items={this.state.items} />
        . . .
```

```
class TodoList extends React.Component {
 render() {
   return (
     <ul>
       {this.props.items.map(function (item) {
         return
           <TodoItem id={item.id} text={item.text}/>
       })}
     </ul>
   );
  }
}
```

```
class TodoItem extends React.Component {
 constructor(props) {
   super(props);
   this.state = { amDone: false };
 }

 handleClick() {
  this.setState( { amDone: !this.state.amDone} );
 }

 render() {
  var line = this.state.amDone ? 'line-through' : 'none';
  return (
   <li key={this.props.id}
        onClick={this.handleClick.bind(this)}
        style={{textDecoration:line}}>
     {this.props.text} </li>
  );
 }
}
```

```
class TodoItem extends React.Component {
 constructor(props) {
   super(props);
   this.state = { amDone: false };
 }

 handleClick() {
  this.setState( { amDone: !this.state.amDone} );
 }

 render() {
  var line = this.state.amDone ? 'line-through' : 'none';
  return (
   <li key={this.props.id}
        onClick={this.handleClick.bind(this)}
        style={{textDecoration:line}}>
     {this.props.text} </li>
  );
 }
}
```

```
class TodoItem extends React.Component {
 constructor(props) {
   super(props);
   this.state = { amDone: false };
 }

 handleClick() {
  this.setState( { amDone: !this.state.amDone} );
 }

 render() {
  var line = this.state.amDone ? 'line-through' : 'none';
  return (
   <li key={this.props.id}
        onClick={this.handleClick.bind(this)}
        style={{textDecoration:line}}>
     {this.props.text} </li>
  );
 }
}
```

```
class TodoItem extends React.Component {
 constructor(props) {
   super(props);
   this.state = { amDone: false };
 }

 handleClick() {
  this.setState( { amDone: !this.state.amDone} );
 }

 render() {
  var line = this.state.amDone ? 'line-through' : 'none';
  return (
   <li key={this.props.id}
        onClick={this.handleClick.bind(this)}
        style={{textDecoration:line}}>
     {this.props.text} </li>
  );
 }
}
```

```
class TodoItem extends React.Component {
 constructor(props) {
   super(props);
   this.state = { amDone: false };
 }

 handleClick() {
  this.setState( { amDone: !this.state.amDone} );
 }

 render() {
  var line = this.state.amDone ? 'line-through' : 'none';
  return (
   <li key={this.props.id}
        onClick={this.handleClick.bind(this)}
        style={{textDecoration:line}}>
     {this.props.text} </li>
  );
 }
}
```

```
class TodoItem extends React.Component {
 constructor(props) {
   super(props);
   this.state = { amDone: false };
 }

 handleClick() {
  this.setState( { amDone: !this.state.amDone} );
 }

 render() {
  var line = this.state.amDone ? 'line-through' : 'none';
  return (
   <li key={this.props.id}
        onClick={this.handleClick.bind(this)}
        style={{textDecoration:line}}>
     {this.props.text} </li>
  );
 }
}
```

```
class TodoItem extends React.Component {
 constructor(props) {
   super(props);
   this.state = { amDone: false };
 }

 handleClick() {
  this.setState( { amDone: !this.state.amDone} );
 }

 render() {
  var line = this.state.amDone ? 'line-through' : 'none';
  return (
   <li key={this.props.id}
        onClick={this.handleClick.bind(this)}
        style={{textDecoration:line}}>
     {this.props.text} </li>
  );
 }
}
```

```
class TodoItem extends React.Component {
 constructor(props) {
   super(props);
   this.state = { amDone: false };
 }

 handleClick() {
  this.setState( { amDone: !this.state.amDone} );
 }

 render() {
  var line = this.state.amDone ? 'line-through' : 'none';
  return (
   <li key={this.props.id}
        onClick={this.handleClick.bind(this)}
        style={{textDecoration:line}}>
     {this.props.text} </li>
  );
 }
}
```

```
class TodoItem extends React.Component {
 constructor(props) {
   super(props);
   this.state = { amDone: false };
 }

 handleClick() {
  this.setState( { amDone: !this.state.amDone} );
 }

 render() {
  var line = this.state.amDone ? 'line-through' : 'none';
  return (
   <li key={this.props.id}
        onClick={this.handleClick.bind(this)}
        style={{textDecoration:line}}>
     {this.props.text} </li>
  );
 }
}
```

```
class TodoItem extends React.Component {
 constructor(props) {
   super(props);
   this.state = { amDone: false };
 }

 handleClick() {
  this.setState( { amDone: !this.state.amDone} );
 }

 render() {
  var line = this.state.amDone ? 'line-through' : 'none';
  return (
   <li key={this.props.id}
        onClick={this.handleClick.bind(this)}
        style={{textDecoration:line}}>
     {this.props.text} </li>
  );
 }
}
```

```
class TodoItem extends React.Component {
 constructor(props) {
   super(props);
   this.state = { amDone: false };
 }

 handleClick() {
  this.setState( { amDone: !this.state.amDone} );
 }

 render() {
  var line = this.state.amDone ? 'line-through' : 'none';
  return (
   <li key={this.props.id}
        onClick={this.handleClick.bind(this)}
        style={{textDecoration:line}}>
     {this.props.text} </li>
  );
 }
}
```

```
class TodoItem extends React.Component {
 constructor(props) {
   super(props);
   this.state = { amDone: false };
 }

 handleClick() {
  this.setState( { amDone: !this.state.amDone} );
 }

 render() {
  var line = this.state.amDone ? 'line-through' : 'none';
  return (
   <li key={this.props.id}
        onClick={this.handleClick.bind(this)}
        style={{textDecoration:line}}>
     {this.props.text} </li>
  );
 }
}
```

```
class TodoItem extends React.Component {
 constructor(props) {
   super(props);
   this.state = { amDone: false };
 }

 handleClick() {
  this.setState( { amDone: !this.state.amDone} );
 }

 render() {
  var line = this.state.amDone ? 'line-through' : 'none';
  return (
   <li key={this.props.id}
        onClick={this.handleClick.bind(this)}
        style={{textDecoration:line}}>
     {this.props.text} </li>
  );
 }
}
```

```
class TodoItem extends React.Component {
 constructor(props) {
   super(props);
   this.state = { amDone: false };
 }

 handleClick() {
  this.setState( { amDone: !this.state.amDone} );
 }

 render() {
  var line = this.state.amDone ? 'line-through' : 'none';
  return (
   <li key={this.props.id}
        onClick={this.handleClick.bind(this)}
        style={{textDecoration:line}}>
     {this.props.text} </li>
  );
 }
}
```

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