React Component Interaction

SENG 4640
Software Engineering for Web Apps
Winter 2023

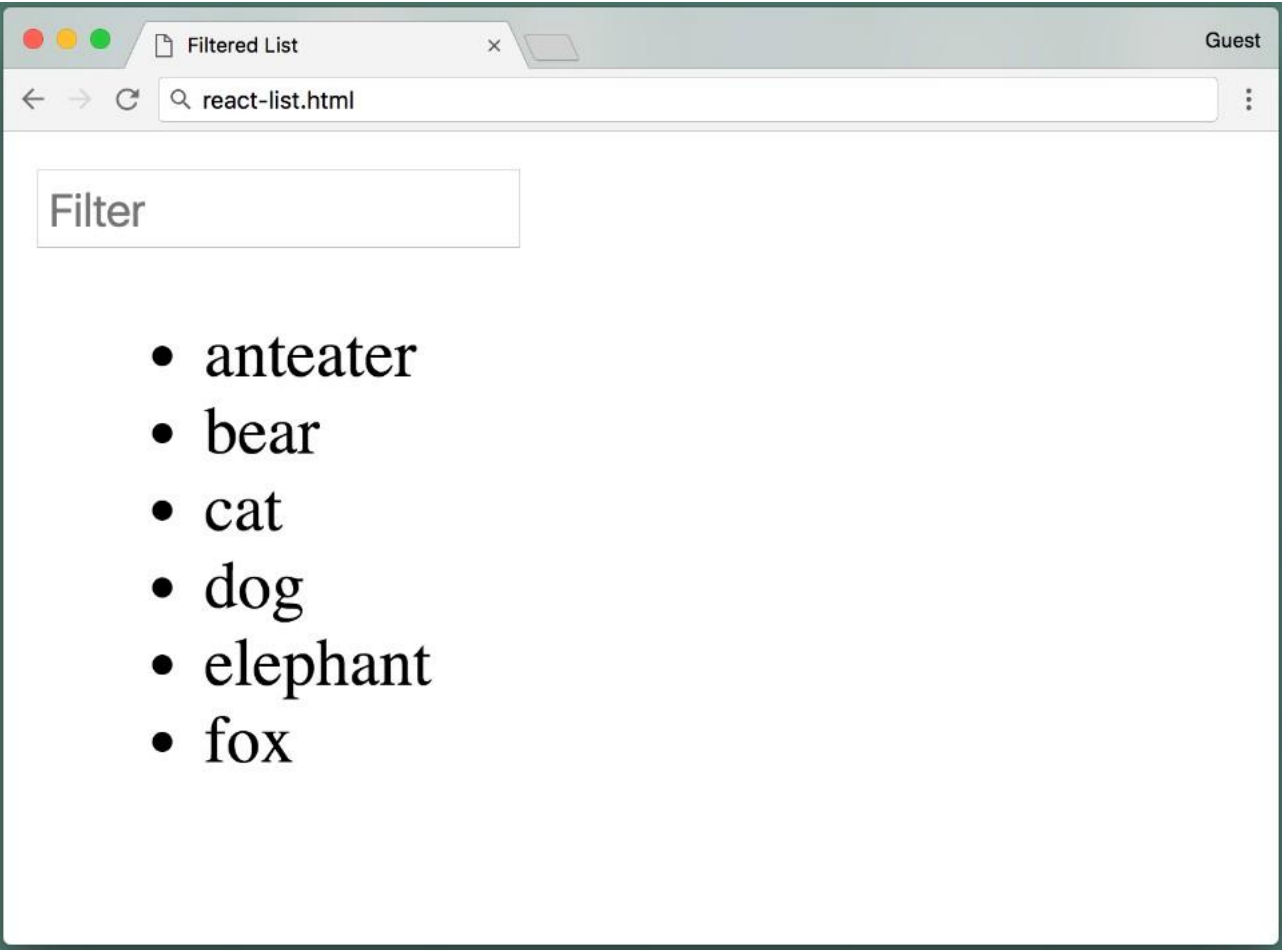
Sina Keshvadi
Thompson Rivers University

Review

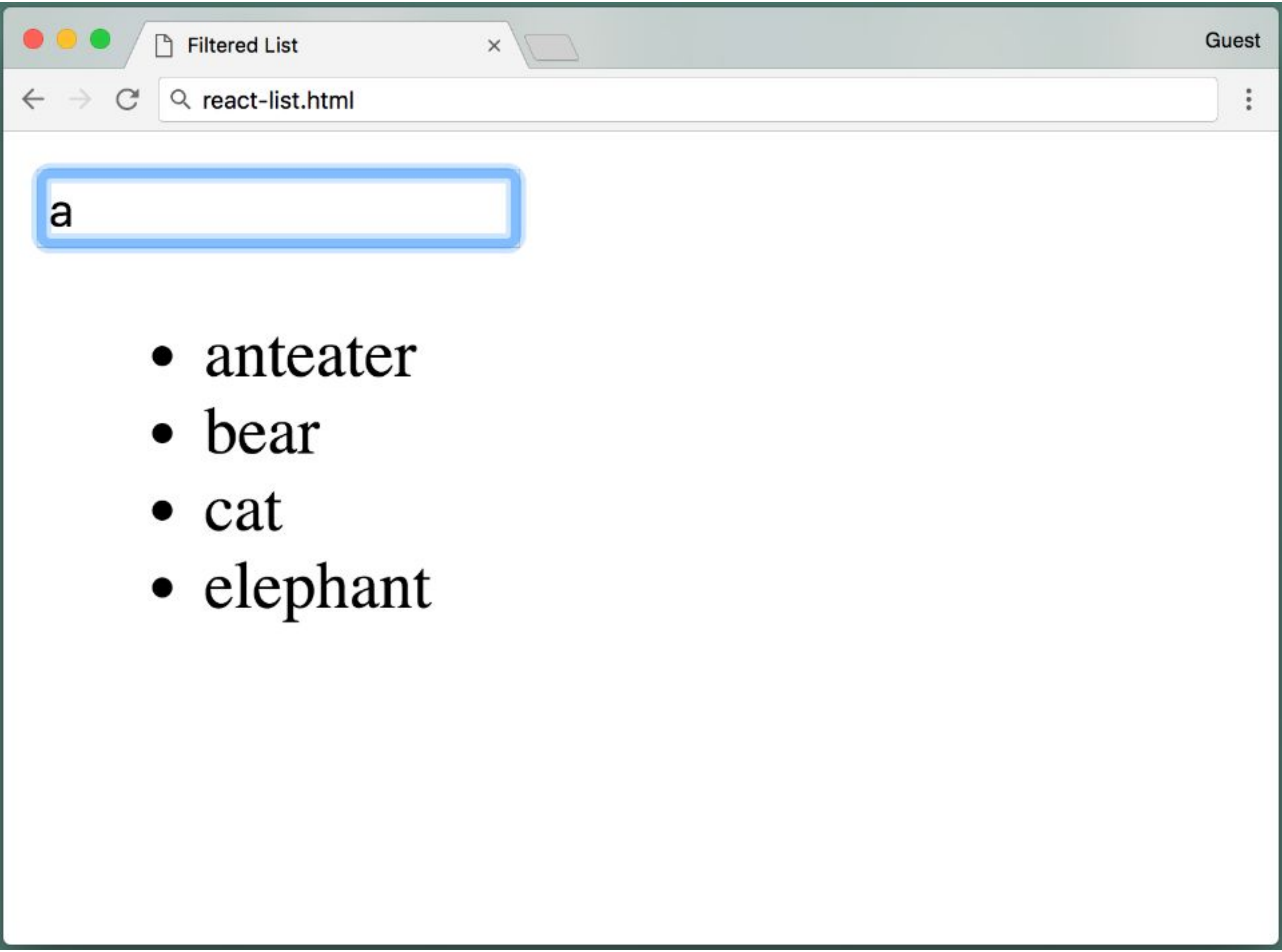
- We can bind user events in HTML elements to callback functions in React components

- 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

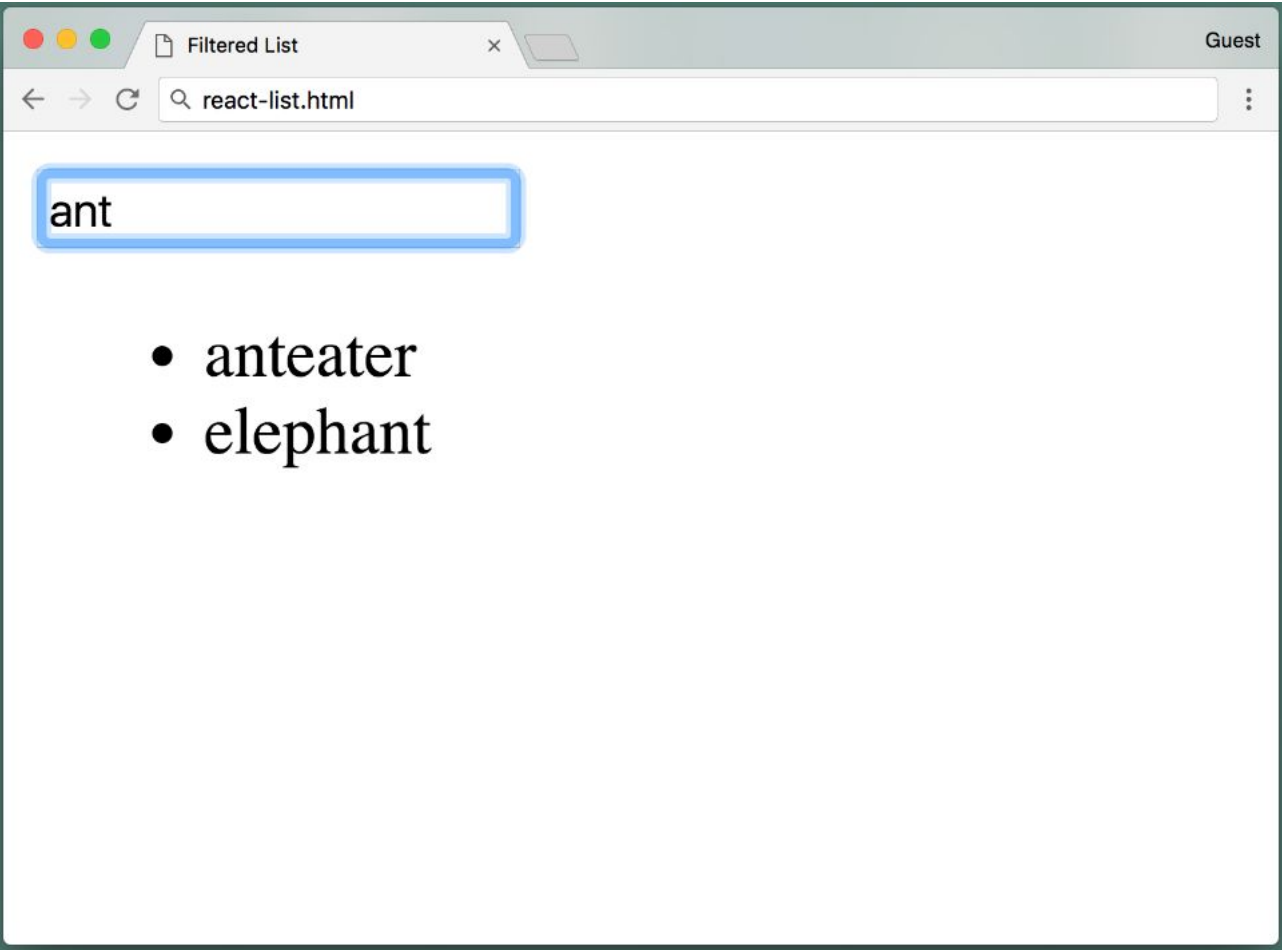


- anteater
- bear
- cat
- dog
- elephant
- fox



a

- anteater
- bear
- cat
- elephant



ant

- anteater
- elephant

```
<body>
    <div id="container"></div>
    <script type="text/babel">
         // React here
    </script>
</body>
```

ex08.html

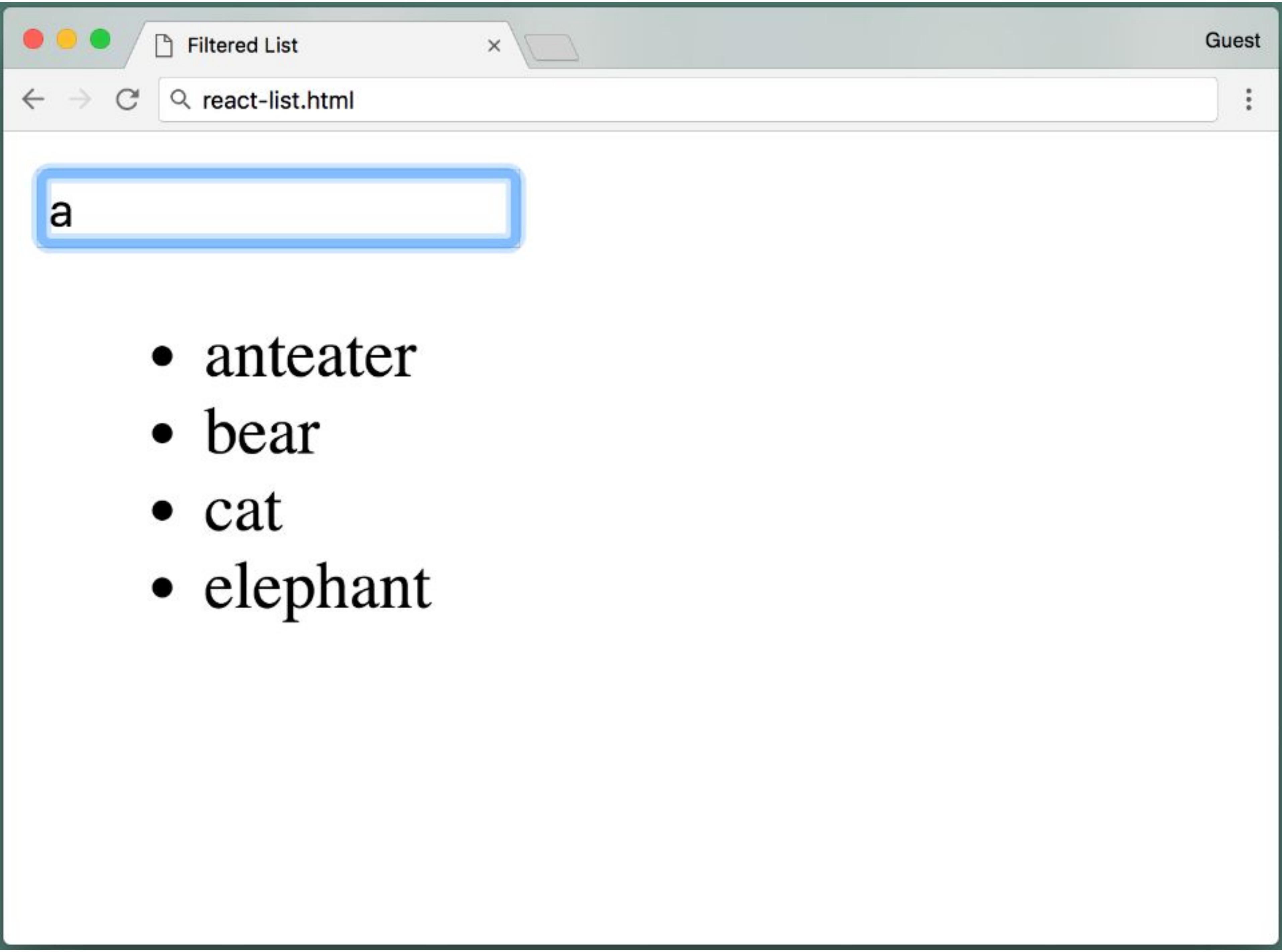
```
<body>
    <div id="container"></div>
    <script type="text/babel">
        class FilteredList extends React.Component{
            // not implemented yet
        };

        class ListItems extends React.Component{
            // not implemented yet
        };


        ReactDOM.createRoot(document.getElementById("container")).render(
            <FilteredList />
        );

    </script>
</body>
```

ex08.html

```
<body>
    <div id="container"></div>
    <script type="text/babel">
        class FilteredList extends React.Component{
            // not implemented yet
        };

        class ListItems extends React.Component{
            // not implemented yet
        };

        ReactDOM.createRoot(document.getElementById("container")).render(
            <FilteredList />
        );

    </script>
</body>
```

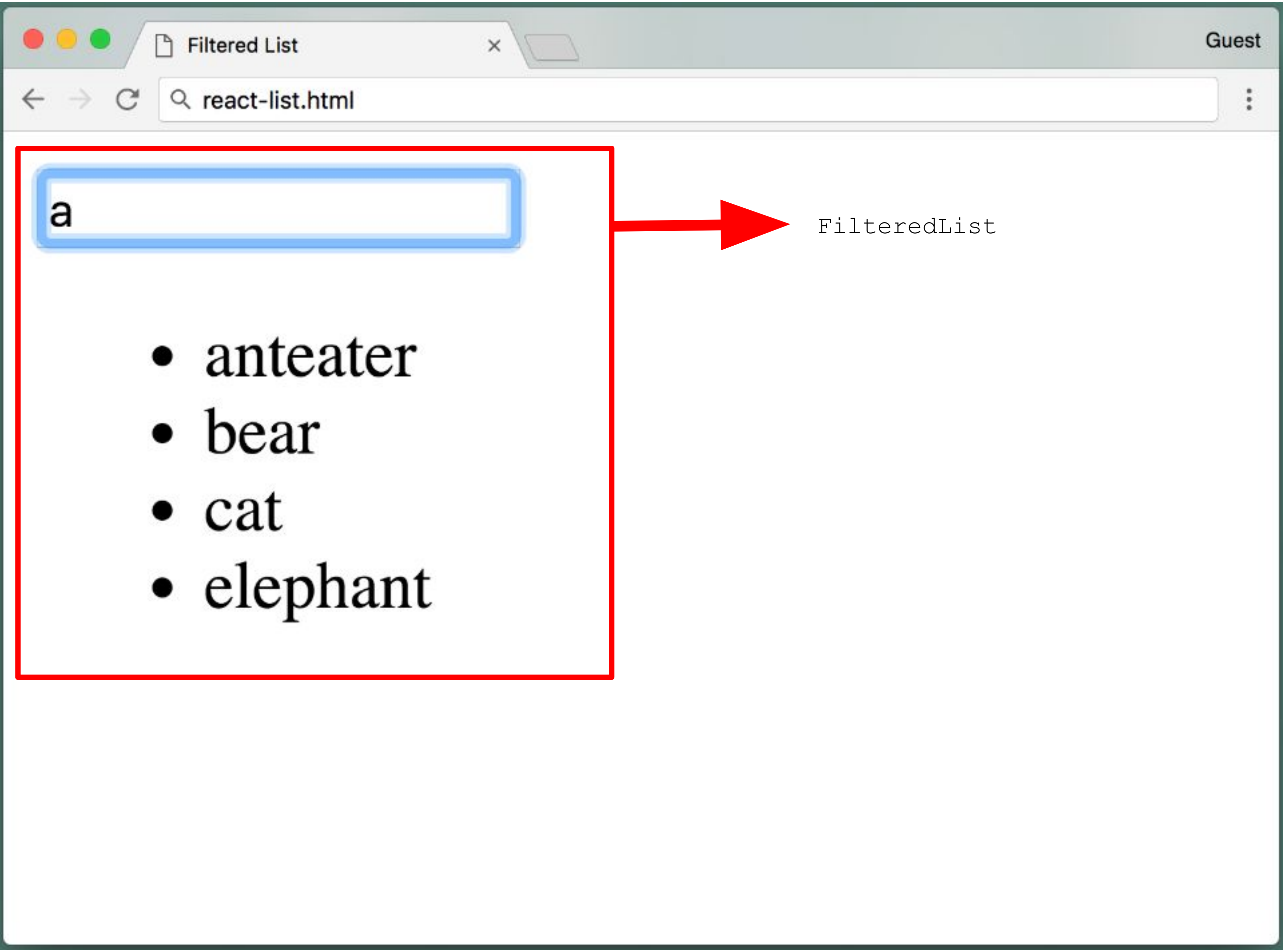
ex08.html

```
<body>
    <div id="container"></div>
    <script type="text/babel">
        class FilteredList extends React.Component{
            // not implemented yet
        };

        class ListItems extends React.Component{
            // not implemented yet
        };

        ReactDOM.createRoot(document.getElementById("container")).render(
            <FilteredList />
        );

    </script>
</body>
```



a

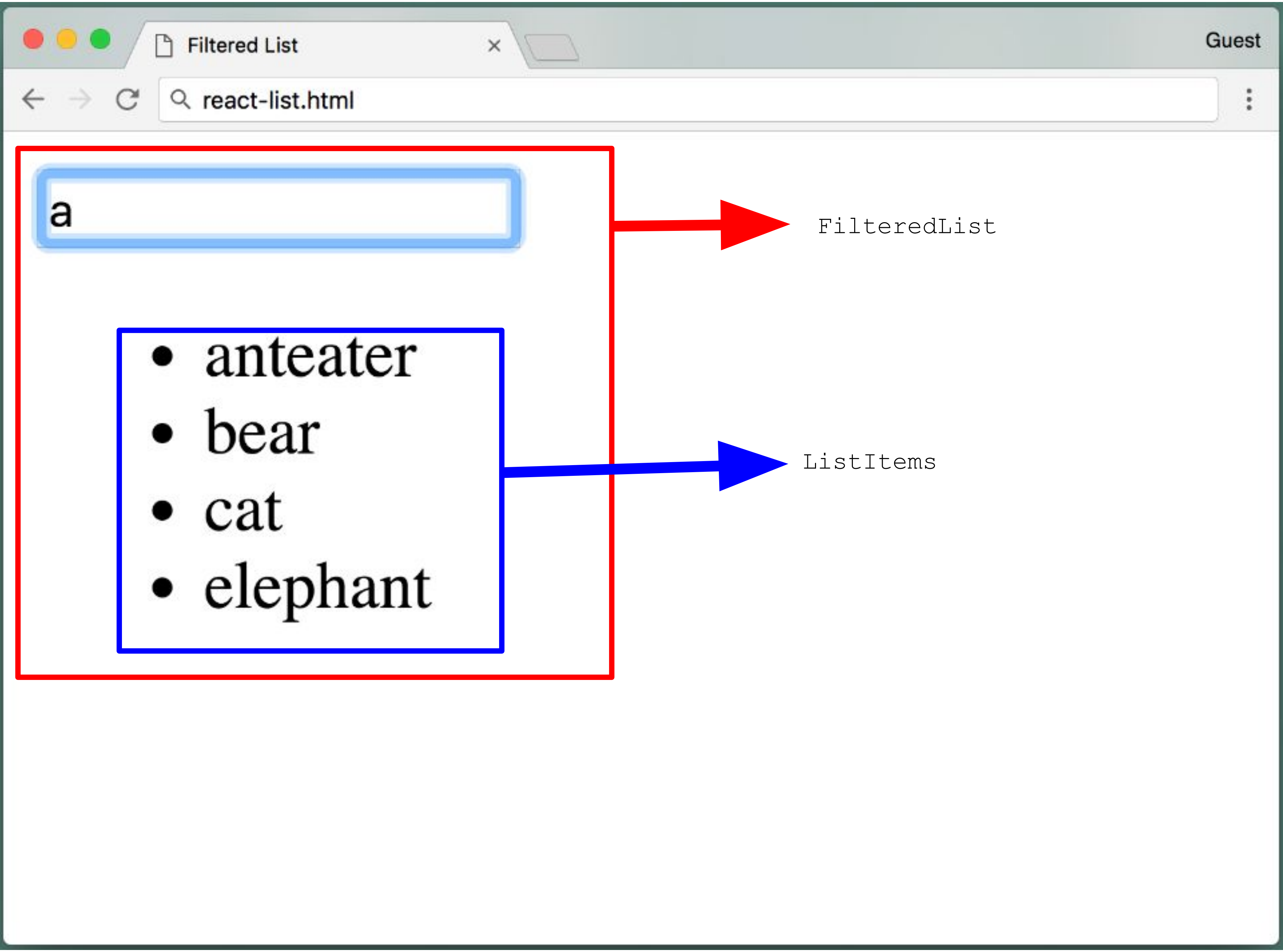
- anteater
- bear
- cat
- elephant



a

- anteater
- bear
- cat
- elephant

FilteredList



a

- anteater
- bear
- cat
- elephant

```
<body>
    <div id="container"></div>
    <script type="text/babel">
        class FilteredList extends React.Component{
            // not implemented yet
        };

        class ListItems extends React.Component{
            // not implemented yet
        };

        ReactDOM.createRoot(document.getElementById("container")).render(
            <FilteredList />
        );

    </script>
</body>
```

```
class FilteredList extends React.Component {
    constructor(props) {
        super(props);
        let allItems = ["Anteater", "Bear", "Cat", "Elephant", "Dog", "Fox"];
        this.state = { initialItems: allItems, currentItems: allItems };
    }

    filterList(inp) {
        let updatedList = this.state.initialItems;
        updatedList = updatedList.filter(function (item) {
            return item.search(inp.target.value) !== -1;
            });
        this.setState({ currentItems: updatedList });
    }

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

```
class FilteredList extends React.Component {
    constructor(props) {
        super(props);
        let allItems = ["Anteater", "Bear", "Cat", "Elephant", "Dog", "Fox"];
        this.state = { initialItems: allItems, currentItems: allItems };
    }

    filterList(inp) {
        let updatedList = this.state.initialItems;
        updatedList = updatedList.filter(function (item) {
            return item.search(inp.target.value) !== -1;
            });
        this.setState({ currentItems: updatedList });
    }

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

```
class FilteredList extends React.Component {
    constructor(props) {
        super(props);
        let allItems = ["Anteater", "Bear", "Cat", "Elephant", "Dog", "Fox"];
        this.state = { initialItems: allItems, currentItems: allItems };
    }

    filterList(inp) {
        let updatedList = this.state.initialItems;
        updatedList = updatedList.filter(function (item) {
            return item.search(inp.target.value) !== -1;
            });
        this.setState({ currentItems: updatedList });
    }

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

```
class FilteredList extends React.Component {
    constructor(props) {
        super(props);
        let allItems = ["Anteater", "Bear", "Cat", "Elephant", "Dog", "Fox"];
        this.state = { initialItems: allItems, currentItems: allItems };
    }

    filterList(inp) {
        let updatedList = this.state.initialItems;
        updatedList = updatedList.filter(function (item) {
            return item.search(inp.target.value) !== -1;
            });
        this.setState({ currentItems: updatedList });
    }

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

```
class FilteredList extends React.Component {
    constructor(props) {
        super(props);
        let allItems = ["Anteater", "Bear", "Cat", "Elephant", "Dog", "Fox"];
        this.state = { initialItems: allItems, currentItems: allItems };
    }

    filterList(inp) {
        let updatedList = this.state.initialItems;
        updatedList = updatedList.filter(function (item) {
            return item.search(inp.target.value) !== -1;
            });
        this.setState({ currentItems: updatedList });
    }

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

```
class FilteredList extends React.Component {
    constructor(props) {
        super(props);
        let allItems = ["Anteater", "Bear", "Cat", "Elephant", "Dog", "Fox"];
        this.state = { initialItems: allItems, currentItems: allItems };
    }

    filterList(inp) {
        let updatedList = this.state.initialItems;
        updatedList = updatedList.filter(function (item) {
            return item.search(inp.target.value) !== -1;
            });
        this.setState({ currentItems: updatedList });
    }

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

```
class FilteredList extends React.Component {
    constructor(props) {
        super(props);
        let allItems = ["Anteater", "Bear", "Cat", "Elephant", "Dog", "Fox"];
        this.state = { initialItems: allItems, currentItems: allItems };
    }

    filterList(inp) {
        let updatedList = this.state.initialItems;
        updatedList = updatedList.filter(function (item) {
            return item.search(inp.target.value) !== -1;
            });
        this.setState({ currentItems: updatedList });
    }

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

```
class FilteredList extends React.Component {
    constructor(props) {
        super(props);
        let allItems = ["Anteater", "Bear", "Cat", "Elephant", "Dog", "Fox"];
        this.state = { initialItems: allItems, currentItems: allItems };
    }

    filterList(inp) {
        let updatedList = this.state.initialItems;
        updatedList = updatedList.filter(function (item) {
            return item.search(inp.target.value) !== -1;
            });
        this.setState({ currentItems: updatedList });
    }

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

```
class FilteredList extends React.Component {
    constructor(props) {
        super(props);
        let allItems = ["Anteater", "Bear", "Cat", "Elephant", "Dog", "Fox"];
        this.state = { initialItems: allItems, currentItems: allItems };
    }

    filterList(inp) {
        let updatedList = this.state.initialItems;
        updatedList = updatedList.filter(function (item) {
            return item.search(inp.target.value) !== -1;
            });
        this.setState({ currentItems: updatedList });
    }

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

```
class FilteredList extends React.Component {
    constructor(props) {
        super(props);
        let allItems = ["Anteater", "Bear", "Cat", "Elephant", "Dog", "Fox"];
        this.state = { initialItems: allItems, currentItems: allItems };
    }

    filterList(inp) {
        let updatedList = this.state.initialItems;
        updatedList = updatedList.filter(function (item) {
            return item.search(inp.target.value) !== -1;
            });
        this.setState({ currentItems: updatedList });
    }

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

```
class FilteredList extends React.Component {
    constructor(props) {
        super(props);
        let allItems = ["Anteater", "Bear", "Cat", "Elephant", "Dog", "Fox"];
        this.state = { initialItems: allItems, currentItems: allItems };
    }

    filterList(inp) {
        let updatedList = this.state.initialItems;
        updatedList = updatedList.filter(function (item) {
            return item.search(inp.target.value) !== -1;
            });
        this.setState({ currentItems: updatedList });
    }

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

```
class FilteredList extends React.Component {
    constructor(props) {
        super(props);
        let allItems = ["Anteater", "Bear", "Cat", "Elephant", "Dog", "Fox"];
        this.state = { initialItems: allItems, currentItems: allItems };
    }

    filterList(inp) {
        let updatedList = this.state.initialItems;
        updatedList = updatedList.filter(function (item) {
            return item.search(inp.target.value) !== -1;
            });
        this.setState({ currentItems: updatedList });
    }

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

```
class FilteredList extends React.Component {
    constructor(props) {
        super(props);
        let allItems = ["Anteater", "Bear", "Cat", "Elephant", "Dog", "Fox"];
        this.state = { initialItems: allItems, currentItems: allItems };
    }

    filterList(inp) {
        let updatedList = this.state.initialItems;
        updatedList = updatedList.filter(function (item) {
            return item.search(inp.target.value) !== -1;
            });
        this.setState({ currentItems: updatedList });
    }

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

```
class FilteredList extends React.Component {
    constructor(props) {
        super(props);
        let allItems = ["Anteater", "Bear", "Cat", "Elephant", "Dog", "Fox"];
        this.state = { initialItems: allItems, currentItems: allItems };
    }

    filterList(inp) {
        let updatedList = this.state.initialItems;
        updatedList = updatedList.filter(function (item) {
            return item.search(inp.target.value) !== -1;
            });
        this.setState({ currentItems: updatedList });
    }

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

```
class FilteredList extends React.Component {
    constructor(props) {
        super(props);
        let allItems = ["Anteater", "Bear", "Cat", "Elephant", "Dog", "Fox"];
        this.state = { initialItems: allItems, currentItems: allItems };
    }

    filterList(inp) {
        let updatedList = this.state.initialItems;
        updatedList = updatedList.filter(function (item) {
            return item.search(inp.target.value) !== -1;
            });
        this.setState({ currentItems: updatedList });
    }

    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>
            );
        }
    };
```

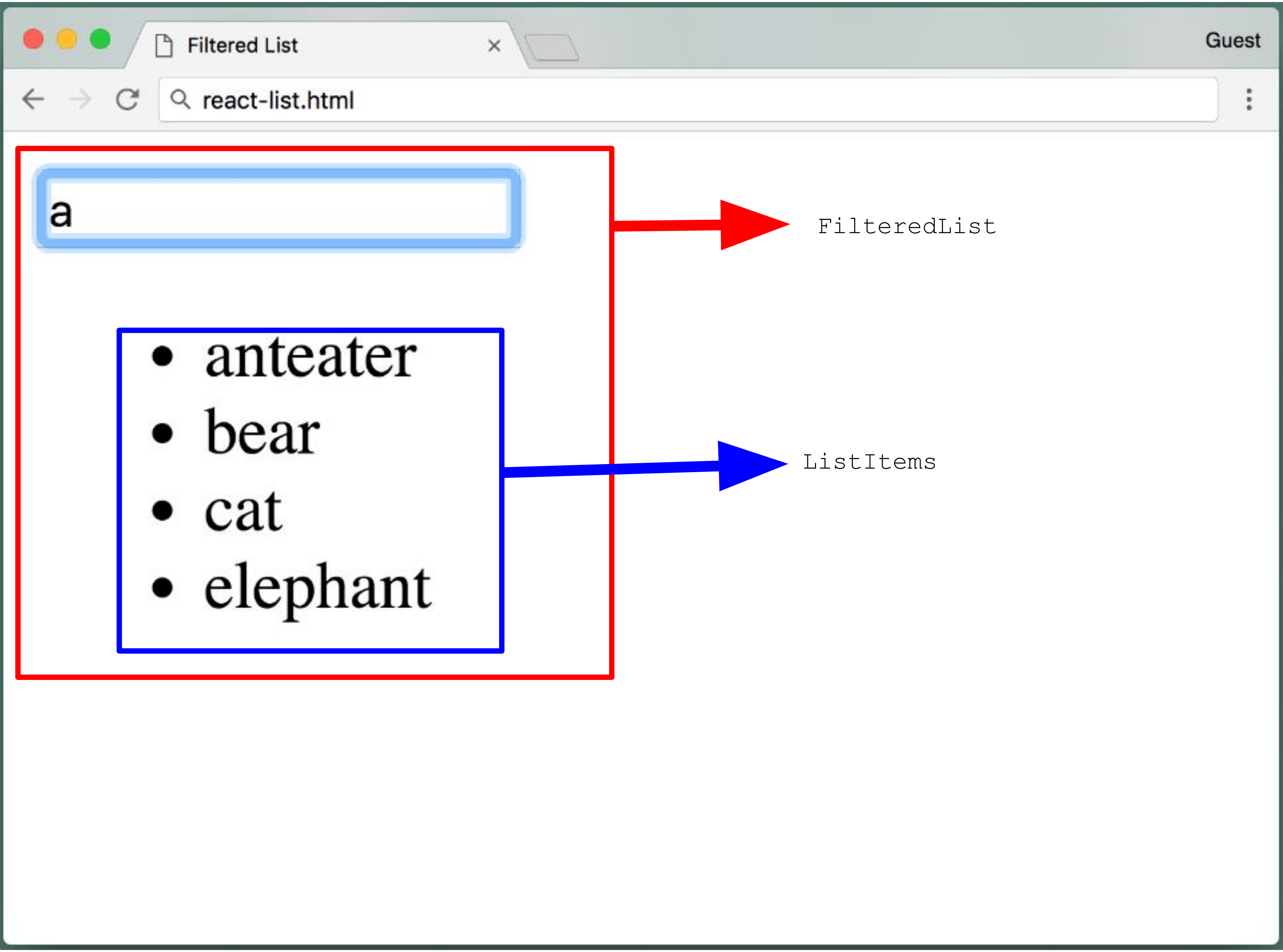
```
class ListItems extends React.Component {
    render() {
        return (
            <ul> {
                this.props.items.map(function (item) {
                    return <li key={item}>{item}</li>
                })
            } </ul>
        );
    }
};
```

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

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

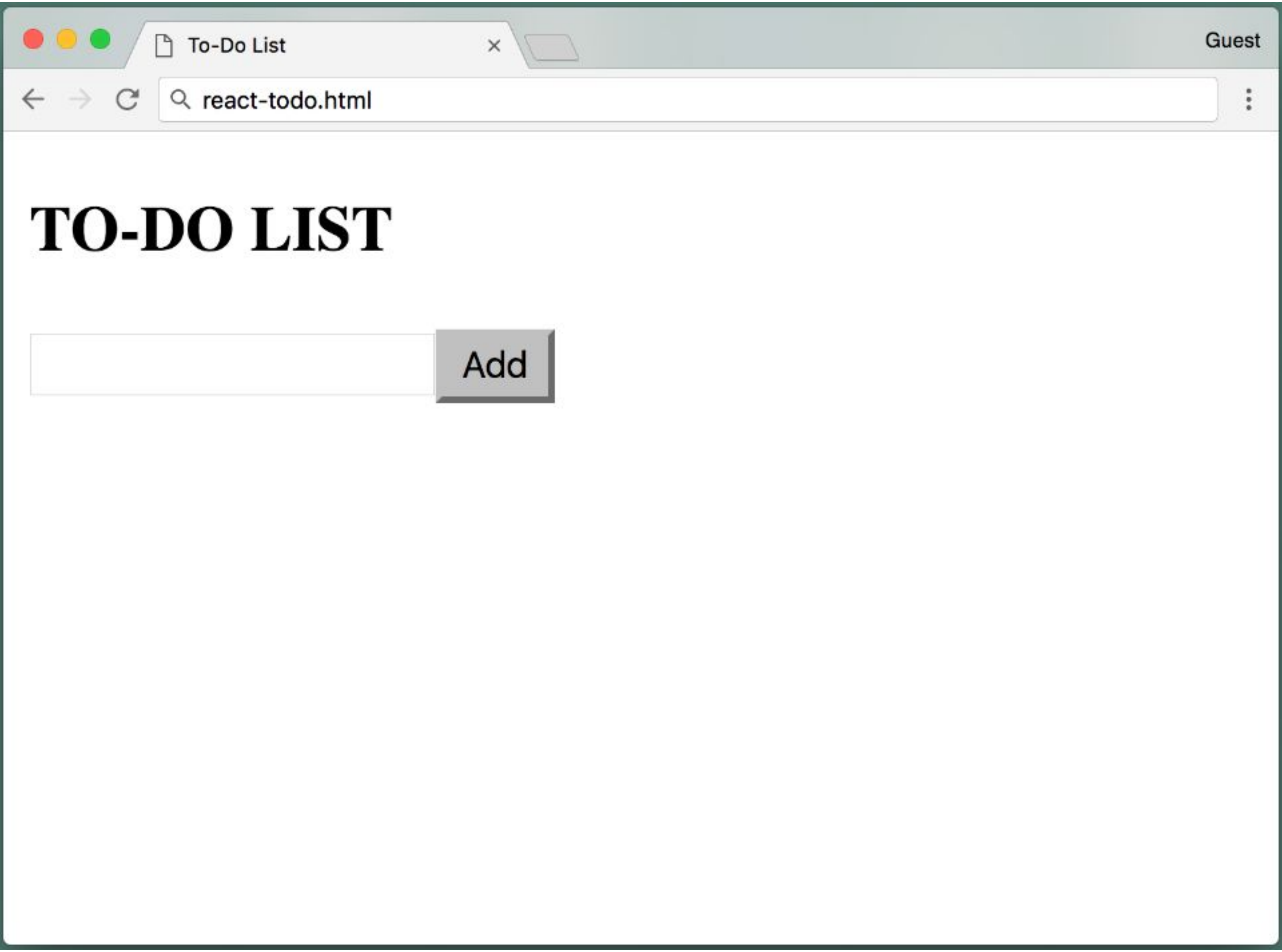
```
class ListItems extends React.Component {
    render() {
        return (
            <ul> {
                this.props.items.map(function (item) {
                    return <li key={item}>{item}</li>
                })
            } </ul>
        );
    }
};
```

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

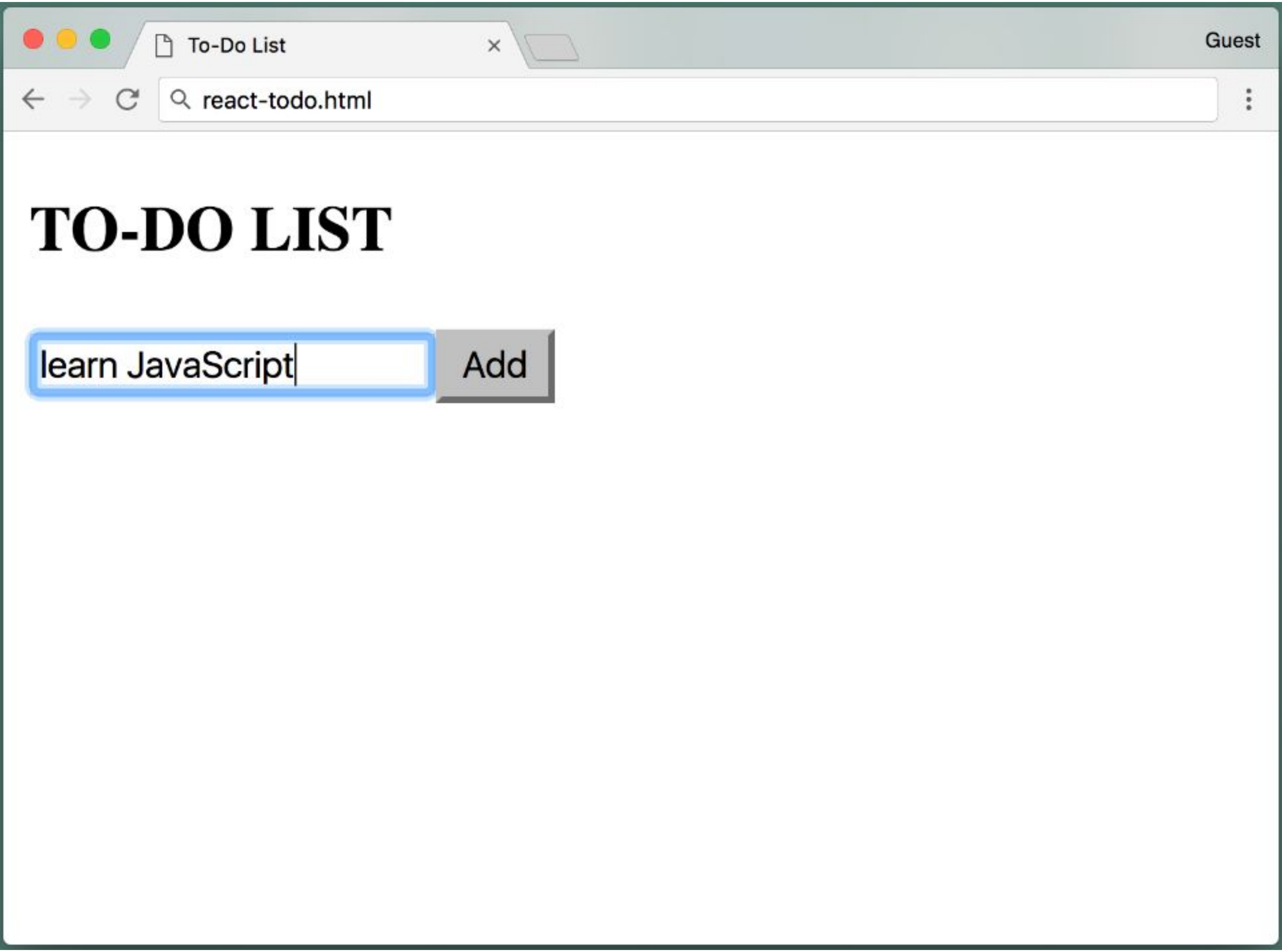


- anteater
- bear
- cat
- elephant

```
<body>
    <div id="container"></div>
    <script type="text/babel">
        class FilteredList extends React.Component {
           constructor(props) {
                super(props);
                …
                this.state = { initialItems: allItems, currentItems: allItems };}

           filterList(inp) {
                      …
                this.setState({ currentItems: updatedList });}

            render() {
                return (
                    <div><input type="text" 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.createRoot(document.getElementById("container")).render(
            <FilteredList />
        );

    </script>
</body>
```

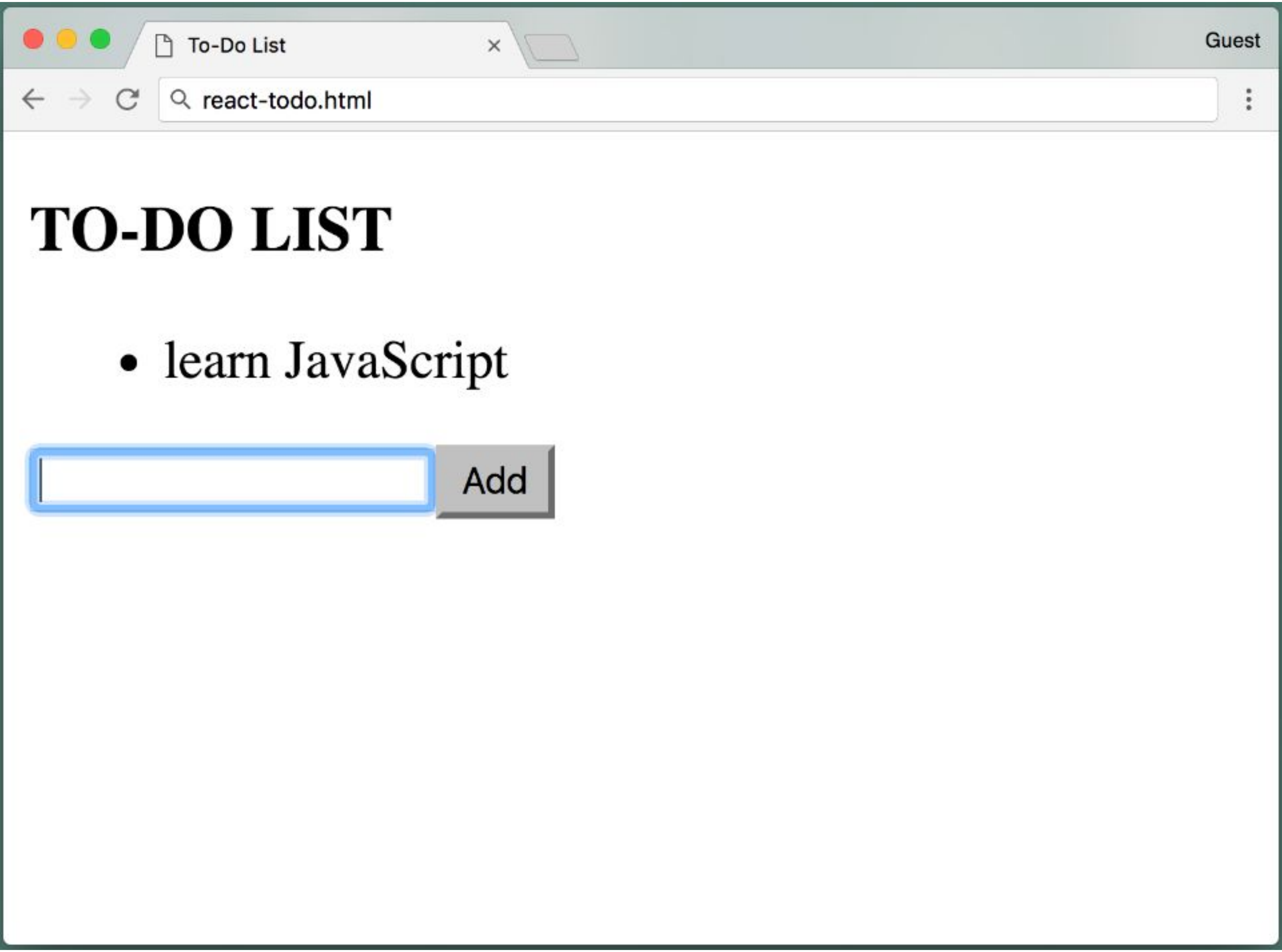


TO-DO LIST



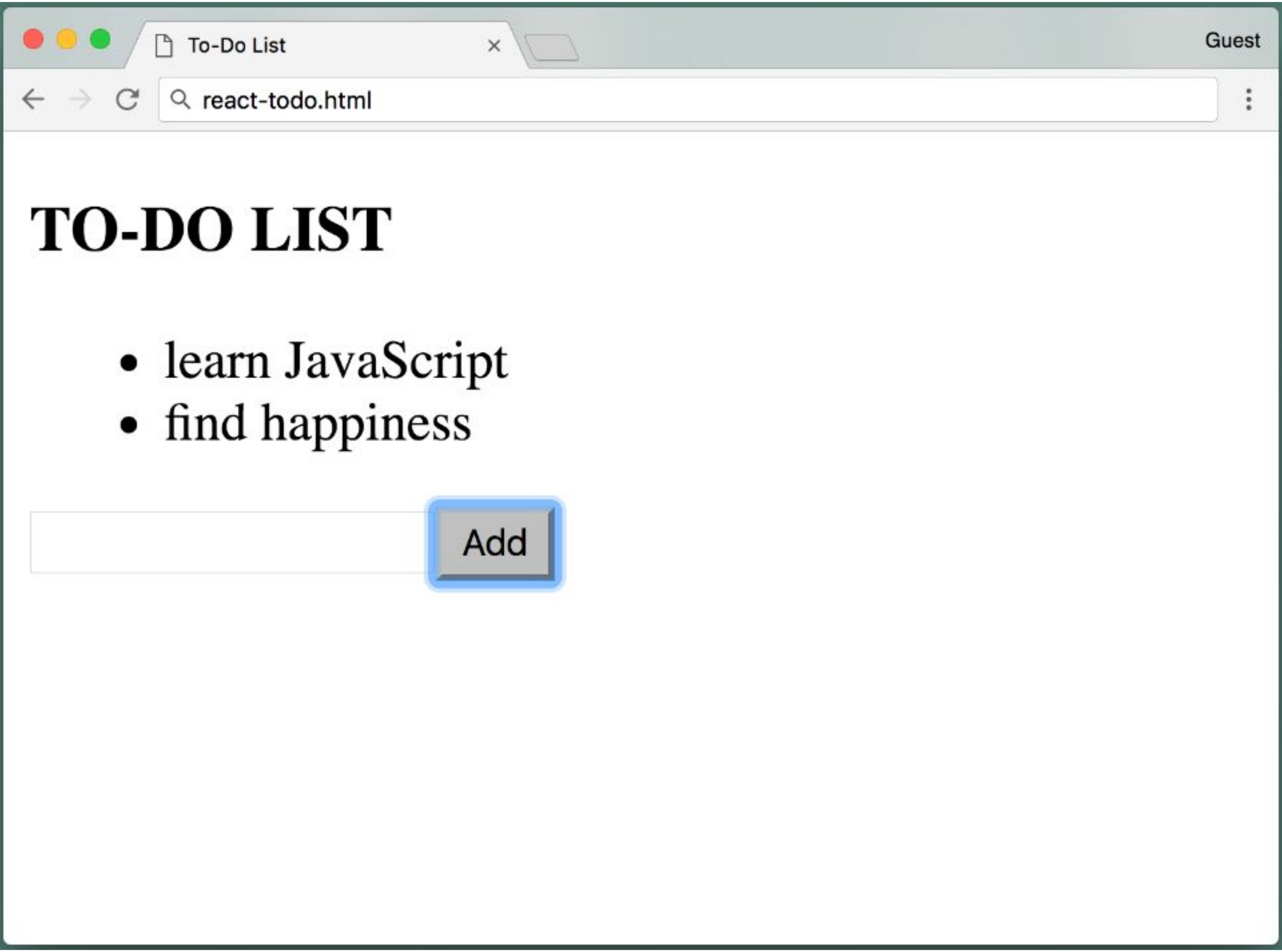
TO-DO LIST

learn JavaScript



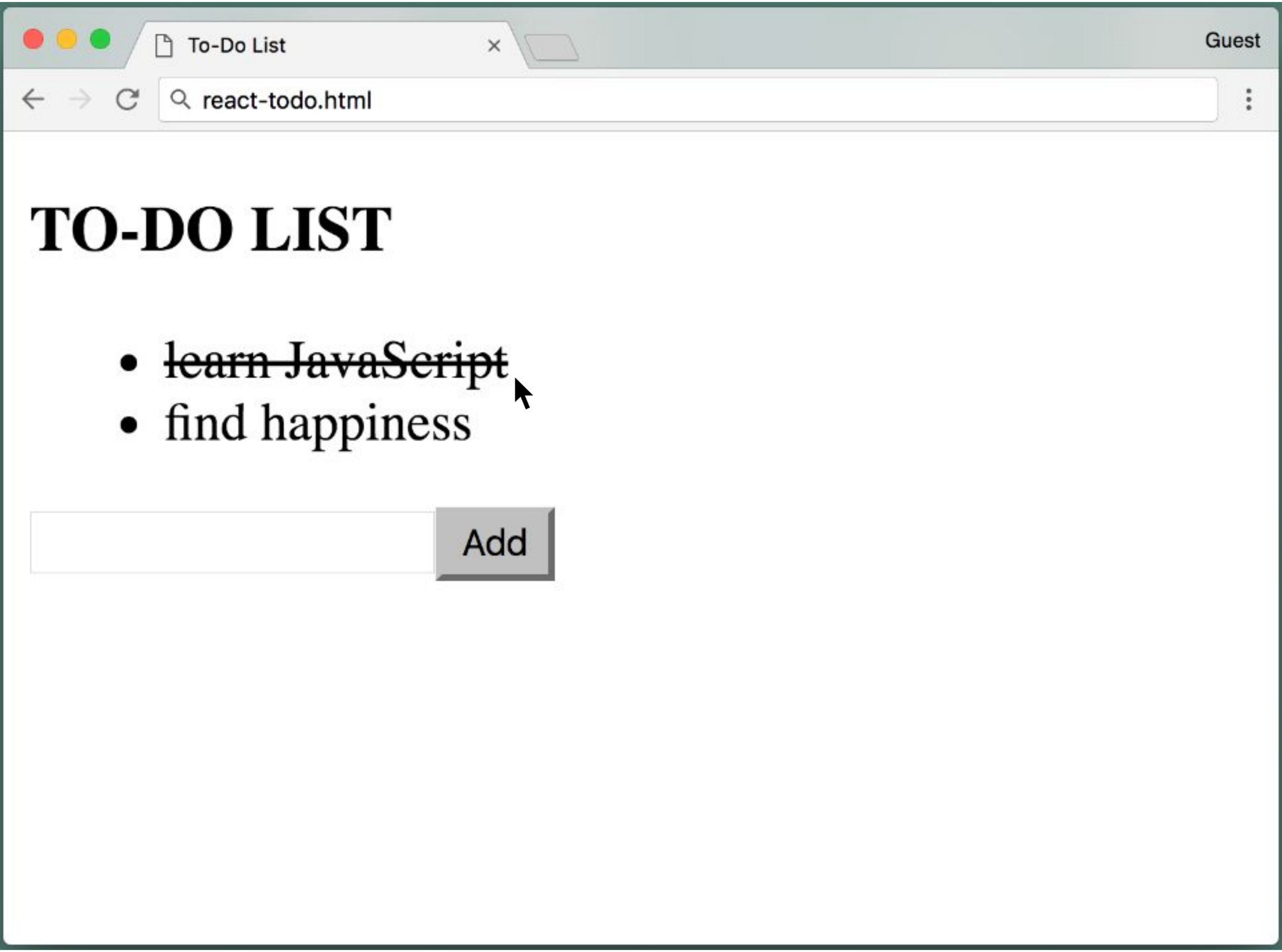
TO-DO LIST

- learn JavaScript



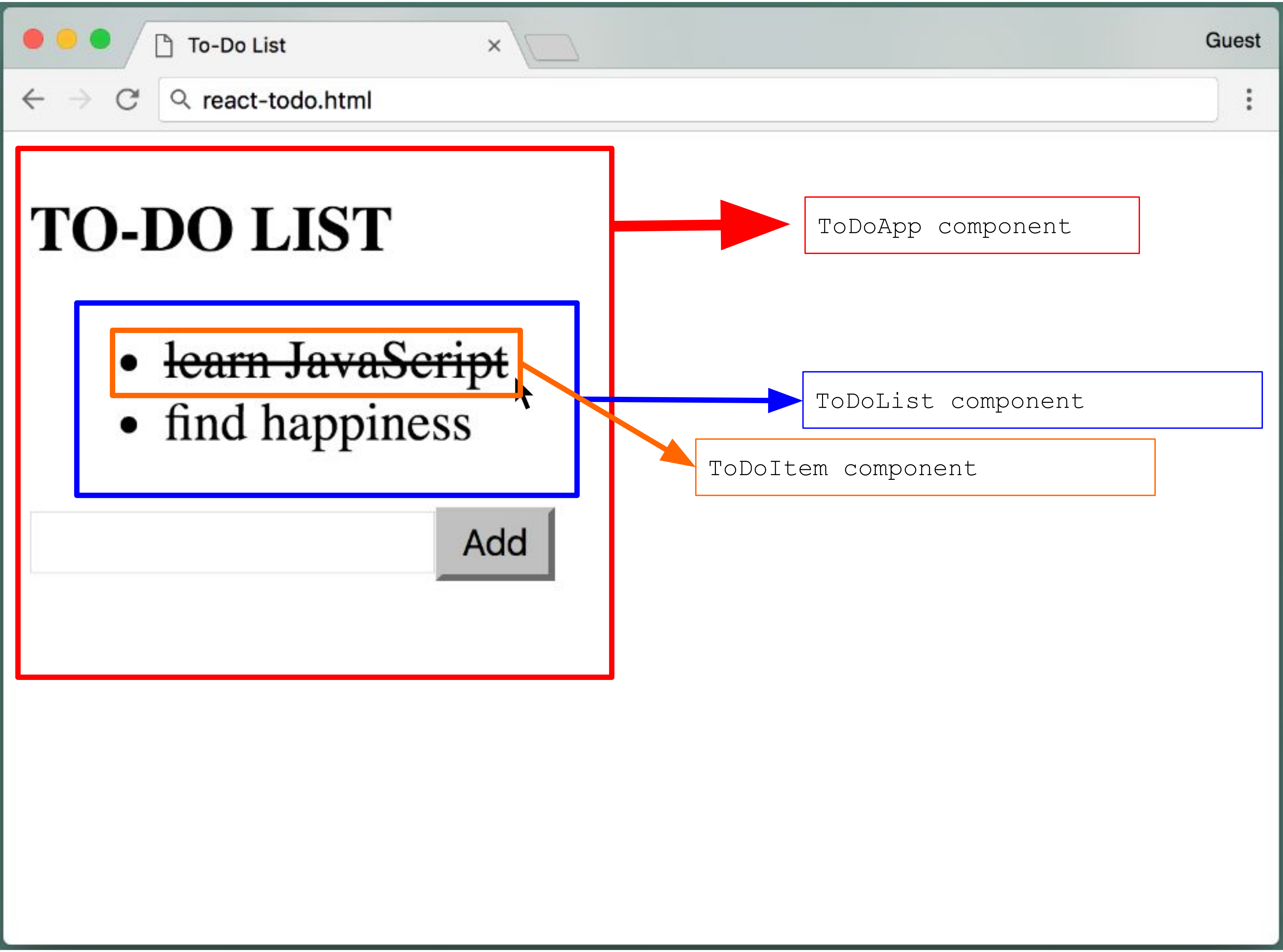
TO-DO LIST

- learn JavaScript
- find happiness



TO-DO LIST

- ~~learn JavaScript~~
- find happiness


To-Do List
Guest
react-todo.html
TO-DO LIST
~~learn JavaScript~~
find happiness
Add
ToDoApp component
ToDoList component
ToDoItem component

```
<body>
   <div id="container"></div>
   <script type="text/babel">
       class ToDoApp extends React.Component {
           // to be implemented
       };

       class ToDoList extends React.Component {
           // to be implemented
       };

       class ToDoItem extends React.Component {
           // to be implemented
       };

       ReactDOM.createRoot(document.getElementById('container')).render(<ToDoApp />);
   </script>
</body>
```

ex09.html

```
<body>
   <div id="container"></div>
   <script type="text/babel">
       class ToDoApp extends React.Component {
           // to be implemented
       };

       class ToDoList extends React.Component {
           // to be implemented
       };

       class ToDoItem extends React.Component {
           // to be implemented
       };

       ReactDOM.createRoot(document.getElementById('container'))
         .render(<ToDoApp />);
   </script>
</body>
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                     <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                     <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                     <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

```
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();
        let newItem = { id: this.state.id, text: this.state.text };
        this.setState({ items: this.state.items.concat(newItem),
                        text: " ", id: this.state.id + 1 });
    }

    render() {
        return (
            <div>
                <h3>To-Do List</h3>
                <form onSubmit={this.handleSubmit.bind(this)}>
                    <input onChange={this.handleChange.bind(this)}
                           value={this.state.text} />
                    <button>Add</button>
                </form>
                <ToDoList items={this.state.items} />
            </div>
        );
    }
};
```

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

```
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 (
             ...
             <ToDoItem id={item.id} text={item.text} />
             ...
```

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

```
class ToDoList extends React.Component {
    render() {
        return (
             ...
             <ToDoItem id={item.id} text={item.text} />
             ...
```

```
class ToDoItem extends React.Component {
    constructor(props) {
        super(props);
        this.state = { done: false };
    }
    handleClick() {
        this.setState({ done: !this.state.done });
    }
    render() {
        let line = this.state.done ? "line-through" : "none";
        return (
            <li key={this.props.id} onClick={this.handleClick.bind(this)}
                style={{ textDecoration: line }}>
                {this.props.text}
            </li>
        );
    }
};
```

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

```
class ToDoList extends React.Component {
    render() {
        return (
             ...
             <ToDoItem id={item.id} text={item.text} />
             ...
```

```
class ToDoItem extends React.Component {
    constructor(props) {
        super(props);
        this.state = { done: false };
    }
    handleClick() {
        this.setState({ done: !this.state.done });
    }
    render() {
        let line = this.state.done ? "line-through" : "none";
        return (
            <li key={this.props.id} onClick={this.handleClick.bind(this)}
                style={{ textDecoration: line }}>
                {this.props.text}
            </li>
        );
    }
};
```

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

```
class ToDoList extends React.Component {
    render() {
        return (
             ...
             <ToDoItem id={item.id} text={item.text} />
             ...
```

```
class ToDoItem extends React.Component {
    constructor(props) {
        super(props);
        this.state = { done: false };
    }
    handleClick() {
        this.setState({ done: !this.state.done });
    }
    render() {
        let line = this.state.done ? "line-through" : "none";
        return (
            <li key={this.props.id} onClick={this.handleClick.bind(this)}
                style={{ textDecoration: line }}>
                {this.props.text}
            </li>
        );
    }
};
```

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

```
class ToDoList extends React.Component {
    render() {
        return (
             ...
             <ToDoItem id={item.id} text={item.text} />
             ...
```

```
class ToDoItem extends React.Component {
    constructor(props) {
        super(props);
        this.state = { done: false };
    }
    handleClick() {
        this.setState({ done: !this.state.done });
    }
    render() {
        let line = this.state.done ? "line-through" : "none";
        return (
            <li key={this.props.id} onClick={this.handleClick.bind(this)}
                style={{ textDecoration: line }}>
                {this.props.text}
            </li>
        );
    }
};
```

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

```
class ToDoList extends React.Component {
    render() {
        return (
             ...
             <ToDoItem id={item.id} text={item.text} />
             ...
```

```
class ToDoItem extends React.Component {
    constructor(props) {
        super(props);
        this.state = { done: false };
    }
    handleClick() {
        this.setState({ done: !this.state.done });
    }
    render() {
        let line = this.state.done ? "line-through" : "none";
        return (
            <li key={this.props.id} onClick={this.handleClick.bind(this)}
                style={{ textDecoration: line }}>
                {this.props.text}
            </li>
        );
    }
};
```

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

```
class ToDoList extends React.Component {
    render() {
        return (
             ...
             <ToDoItem id={item.id} text={item.text} />
             ...
```

```
class ToDoItem extends React.Component {
    constructor(props) {
        super(props);
        this.state = { done: false };
    }
    handleClick() {
        this.setState({ done: !this.state.done });
    }
    render() {
        let line = this.state.done ? "line-through" : "none";
        return (
            <li key={this.props.id} onClick={this.handleClick.bind(this)}
                style={{ textDecoration: line }}>
                {this.props.text}
            </li>
        );
    }
};
```

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

```
class ToDoList extends React.Component {
    render() {
        return (
             ...
             <ToDoItem id={item.id} text={item.text} />
             ...
```

```
class ToDoItem extends React.Component {
    constructor(props) {
        super(props);
        this.state = { done: false };
    }
    handleClick() {
        this.setState({ done: !this.state.done });
    }
    render() {
        let line = this.state.done ? "line-through" : "none";
        return (
            <li key={this.props.id} onClick={this.handleClick.bind(this)}
                style={{ textDecoration: line }}>
                {this.props.text}
            </li>
        );
    }
};
```

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

```
class ToDoList extends React.Component {
    render() {
        return (
             ...
             <ToDoItem id={item.id} text={item.text} />
             ...
```

```
class ToDoItem extends React.Component {
    constructor(props) {
        super(props);
        this.state = { done: false };
    }
    handleClick() {
        this.setState({ done: !this.state.done });
    }
    render() {
        let line = this.state.done ? "line-through" : "none";
        return (
            <li key={this.props.id} onClick={this.handleClick.bind(this)}
                style={{ textDecoration: line }}>
                {this.props.text}
            </li>
        );
    }
};
```

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

```
class ToDoList extends React.Component {
    render() {
        return (
             ...
             <ToDoItem id={item.id} text={item.text} />
             ...
```

```
class ToDoItem extends React.Component {
    constructor(props) {
        super(props);
        this.state = { done: false };
    }
    handleClick() {
        this.setState({ done: !this.state.done });
    }
    render() {
        let line = this.state.done ? "line-through" : "none";
        return (
            <li key={this.props.id} onClick={this.handleClick.bind(this)}
                style={{ textDecoration: line }}>
                {this.props.text}
            </li>
        );
    }
};
```

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

```
class ToDoList extends React.Component {
    render() {
        return (
             ...
             <ToDoItem id={item.id} text={item.text} />
             ...
```

```
class ToDoItem extends React.Component {
    constructor(props) {
        super(props);
        this.state = { done: false };
    }
    handleClick() {
        this.setState({ done: !this.state.done });
    }
    render() {
        let line = this.state.done ? "line-through" : "none";
        return (
            <li key={this.props.id} onClick={this.handleClick.bind(this)}
                style={{ textDecoration: line }}>
                {this.props.text}
            </li>
        );
    }
};
```

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

```
class ToDoList extends React.Component {
    render() {
        return (
             ...
             <ToDoItem id={item.id} text={item.text} />
             ...
```

```
class ToDoItem extends React.Component {
    constructor(props) {
        super(props);
        this.state = { done: false };
    }
    handleClick() {
        this.setState({ done: !this.state.done });
    }
    render() {
        let line = this.state.done ? "line-through" : "none";
        return (
            <li key={this.props.id} onClick={this.handleClick.bind(this)}
                style={{ textDecoration: line }}>
                {this.props.text}
            </li>
        );
    }
};
```

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

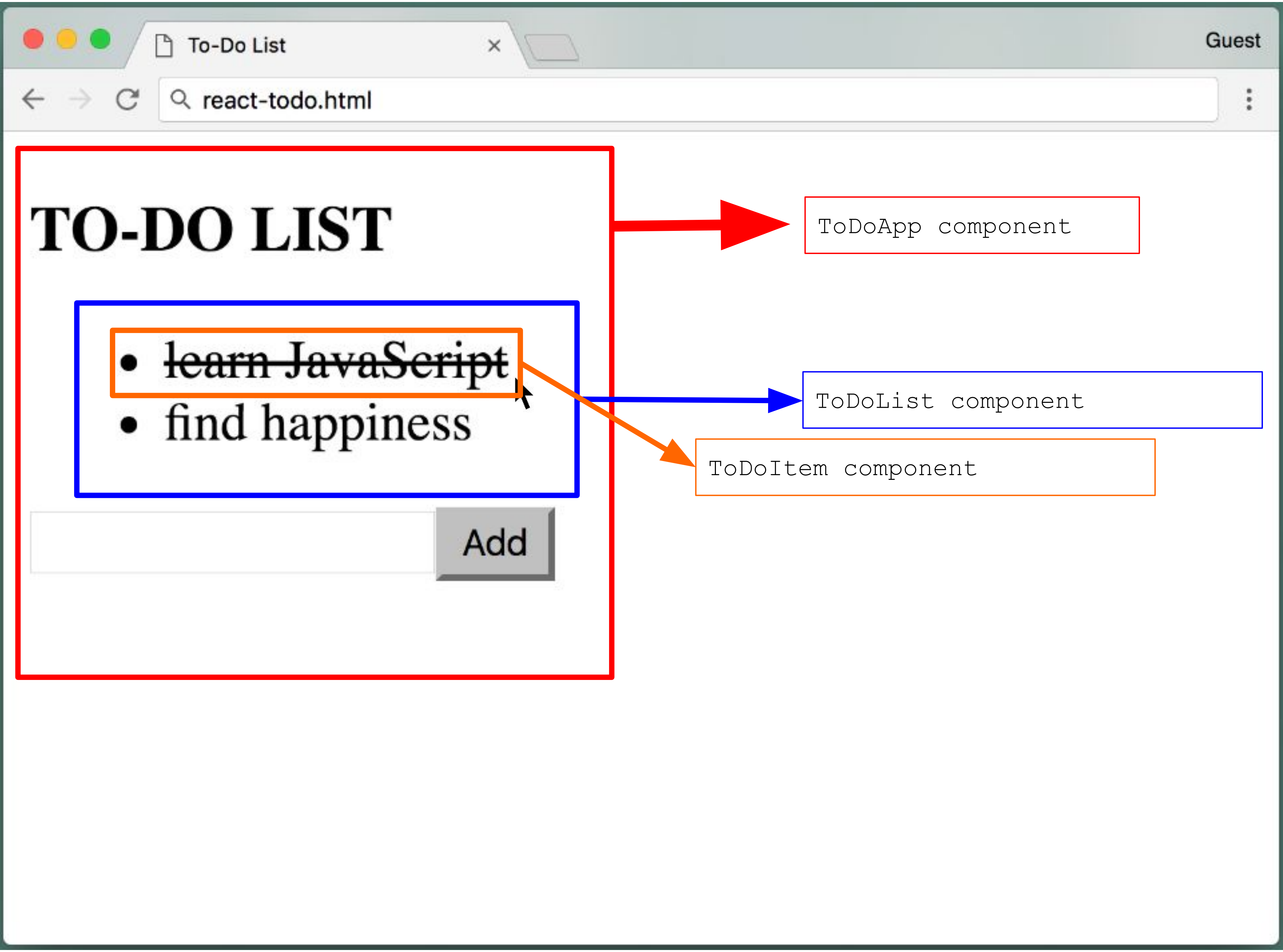
```
class ToDoList extends React.Component {
    render() {
        return (
             ...
             <ToDoItem id={item.id} text={item.text} />
             ...
```

```
class ToDoItem extends React.Component {
    constructor(props) {
        super(props);
        this.state = { done: false };
    }
    handleClick() {
        this.setState({ done: !this.state.done });
    }
    render() {
        let line = this.state.done ? "line-through" : "none";
        return (
            <li key={this.props.id} onClick={this.handleClick.bind(this)}
                style={{ textDecoration: line }}>
                {this.props.text}
            </li>
        );
    }
};
```

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

```
class ToDoList extends React.Component {
    render() {
        return (
             ...
             <ToDoItem id={item.id} text={item.text} />
             ...
```

```
class ToDoItem extends React.Component {
    constructor(props) {
        super(props);
        this.state = { done: false };
    }
    handleClick() {
        this.setState({ done: !this.state.done });
    }
    render() {
        let line = this.state.done ? "line-through" : "none";
        return (
            <li key={this.props.id} onClick={this.handleClick.bind(this)}
                style={{ textDecoration: line }}>
                {this.props.text}
            </li>
        );
    }
};
```



TO-DO LIST

- ~~learn JavaScript~~
- find happiness

ToDoApp component

ToDoList component

ToDoItem component

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