```
<ActionButton text="Book flight" onAction={someFunc} />
```

```
var ActionButton = React.createClass({
  render: function() {




  }
});
```

```
<ActionButton text="Book flight" onAction={someFunc} />
```

```
var ActionButton = React.createClass({
  render: function() {




  }
});
```

```
<ActionButton text="Book flight" onAction={someFunc} />
```

```
var ActionButton = React.createClass({
  render: function() {
    return (
      <button class="ActionButton">
        <span>button text</span>
      </button>
    );
  }
});
```

```
<ActionButton text="Book flight" onAction={someFunc} />
```

```
var ActionButton = React.createClass({
  render: function() {
    return (
      <button class="ActionButton" onClick={this.props.onAction}>
        <span>{this.props.text}</span>
      </button>
    );
  }
});
```

```
<ActionButton text="Book flight" onAction={someFunc} />
```

```
var ActionButton = React.createClass({
  render: function() {
    return (
      <button class="ActionButton" onClick={this.props.onAction}>
        <span>{this.props.text.toUpperCase()}</span>
      </button>
    );
  }
});
```

```
<ActionButton text="Book flight" onAction={someFunc} />
```

```
var ActionButton = React.createClass({
  render: function() {
    return (
      <button class="ActionButton" onClick={this.props.onAction}>
        <span>{this.props.text}</span>
      </button>
    );
  }
});
```

```
<ActionButton text="Book flight" onAction={someFunc} />
```

```
<Counter initialCount={4} />
```

```
var Counter = React.createClass({
  getInitialState: function() {
    return {count: this.props.initialCount};
  },
  addToCount: function(delta) {
    this.setState({count: this.state.count + delta})
  },
  render: function() {
    return (
      <div>
        <h3>Count: {this.state.count}</h3>
        <ActionButton text="+1" onAction={this.addToCount.bind(this, 1)} />
        <ActionButton text="-1" onAction={this.addToCount.bind(this, -1)} />
      </div>
    );
  }
});
```

```
<Counter initialCount={4} />
```

```
var Counter = React.createClass({
  getInitialState: function() {
    return {count: this.props.initialCount};
  },
  addToCount: function(delta) {
    this.setState({count: this.state.count + delta})
  },
  render: function() {
    return (
      <div>
        <h3>Count: {this.state.count}</h3>
        <ActionButton text="+1" onAction={this.addToCount.bind(this, 1)} />
        <ActionButton text="-1" onAction={this.addToCount.bind(this, -1)} />
      </div>
    );
  }
});
```

```
<Counter initialCount={4} />
```

```
var Counter = React.createClass({
  getInitialState: function() {
    return {count: this.props.initialCount};
  },
  addToCount: function(delta) {
    this.setState({count: this.state.count + delta})
  },
  render: function() {
    return (
      <div>
        <h3>Count: {this.state.count}</h3>
        <ActionButton text="+1" onAction={this.addToCount.bind(this, 1)} />
        <ActionButton text="-1" onAction={this.addToCount.bind(this, -1)} />
      </div>
    );
  }
});
```

```
<Counter initialCount={4} />
```

1. Components, not templates
2. Re-render on update
3. Virtual DOM (and events)

1. Components, not templates

```
var Counter = React.createClass({
  getInitialState: function() {
    return {count: this.props.initialCount};
  },
  addToCount: function(delta) {
    this.setState({count: this.state.count + delta})
  },
  render: function() {
    return (
      <div>
        <h3>Count: {this.state.count}</h3>
        <ActionButton text="+1" onAction={this.addToCount.bind(this, 1)} />
        <ActionButton text="-1" onAction={this.addToCount.bind(this, -1)} />
      </div>
    );
  }
});
```

```
<Counter initialCount={4} />
```

On every update…

- React builds a new virtual DOM subtree
- …diffs it with the old one
- …computes the minimal set of DOM mutations and puts them in a queue
- …and batch executes all updates