ReactJS
State and Props

Agenda

- State
- Props
- Props vs State
- Default Props
- Validating Props
- Exercise

State

- The state is a data structure that starts with a default value when a Component mounts.
- It may be mutated across time, mostly as a result of user events.
- A Component manages its own state internally. Besides setting an initial state, it has no business fiddling with the state of its children. You might conceptualize state as private to that component.
- Example : https://github.com/priya1091992/React--State-and-Props

State

Default State -> ES5

```
var MyComponent = React.createClass({
getInitialState() {
return { /* initial state */ };
 },
});
```

State

Default State -> ES6

```
class MyComponent extends React.Component {
   constructor(props) {
      super(props);
      this.state = { /* initial state */ };
   }
}
```

State

State modification and component updates

```
class example extends React.Component{
  constructor(props){
    super(props);
      this.state={
        inputValue: " "
      }
  }
  onChange(e) {
     this.setState({ inputValue: e.target.value });
  }
```

State

State modification and component updates

```
render() {
    return (
      <input
        type='text'
        value={this.state.inputValue}
        onChange={this.onChange} />
    )}
}
```

Whenever this.state is updated the component will be re-rendered, causing the input value to reflect what the user typed.

Props

- Props (short for properties) are a Component's configuration.
- Received from parent and immutable.
- A Component cannot change its props, but it is responsible for putting together the props of its child Components.
- Props do not have to just be data -- callback functions may be passed in as props.

Default Props

- To set the default props use getDefaultProps in the component. (ES5)

```
var Greeting = React.createClass({
  propTypes: {
    name: React.PropTypes.string
  },
  getDefaultProps: function() {
    return {
      name: 'Mary'
    }; },
});
```

Default Props

- To set the default props set the class.defaultProps in the component. (ES6)

```
class Greeting extends React.Component {
  // ...
}
Greeting.defaultProps = {
  name: 'Mary'
};
```

Validating Props

- We can also specify what type of props a component is expecting by specifying the **proptypes**.
- Styntax :
- <ClassName>.proptypes = {
- <propName> : <propType>
- }

Validating Props

If types does not match, react warns us about react validation failing.

```
App.propTypes = {
    propArray: React.PropTypes.array.isRequired,
    propBool: React.PropTypes.bool.isRequired,
    propFunc: React.PropTypes.func,
    propNumber: React.PropTypes.number,
    propString: React.PropTypes.string,
    propObject: React.PropTypes.object
}
```

```
Warning: Failed propType: Invalid prop `propArray` of type `number` supplied to
`App`, expected `array`.
```

Changing props and state

	props	state
Can get initial value from parent Component?	Yes	Yes
Can set default values inside Component?*	Yes	Yes
Can change inside Component?	No	Yes
Can set initial value for child Components?	Yes	Yes
Can change in child Components?	Yes	No

- Note that both props and state initial values received from parents override default values defined inside a Component.

Your Queries Please!!!