Validating Props

In this example we are creating **App** component with all the **props** that we need. **App.propTypes** is used for props validation. If some of the props aren't using correct type that we assigned, we will get console warning. After we specified validation patterns, we are setting **App.defaultProps**.

### App.jsx

import React from 'react';

class App extends React.Component {

render() {

return (

<div>

<h3>Array: {this.props.propArray}</h3>

<h3>Bool: {this.props.propBool ? "True..." : "False..."}</h3>

<h3>Func: {this.props.propFunc(3)}</h3>

<h3>Number: {this.props.propNumber}</h3>

<h3>String: {this.props.propString}</h3>

<h3>Object: {this.props.propObject.objectName1}</h3>

<h3>Object: {this.props.propObject.objectName2}</h3>

<h3>Object: {this.props.propObject.objectName3}</h3>

</div>

);

}

}

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

}

App.defaultProps = {

propArray: [1,2,3,4,5],

propBool: true,

propFunc: function(e){return e},

propNumber: 1,

propString: "String value...",

propObject: {

objectName1:"objectValue1",

objectName2: "objectValue2",

objectName3: "objectValue3"

}

}

export default App;

### main.js

import React from 'react';

import ReactDOM from 'react-dom';

import App from './App.jsx';

ReactDOM.render(<App/>, document.getElementById('app'));

**Set State**

**setState()** method is used for updating the state of the component. This method will not replace the state but only add changes to original state.