React js

What is React.js

- React is a Javascript Library (Not a framework)
- Created by Facebook
- Is used to create complex user interfaces by composing components

Why React

- Virtual DOM
- One-Way data binding
- Component Based Architecture

Prior Knowledge

- HTML
- CSS/Bootstrap
- Javascript
- ES-6

Topics we will cover

- Components
- State & Props
- JSX
- Lifecycle Methods
- Styled Components
- Handling Events
- Controlled Components
- Type Checking
- React Hooks
- Context
- Refs
- React Memo
- Profiler
- Higher Order Components
- Render Props
- Code Splitting
- Error Boundaries
- ES-6
- React Router
- Axios
- Redux
- Thunk
- Jest & Enzyme

Common Libraries & Tools

- Axios for Ajax request
- React-router for Routing
- Redux for State Management
- Babel for ES6/JSX compilation to ES5
- Webpack for build process

Components

- Components are reusable and isolated piece of code
- In React we have two kind of components– Class and Functional Compoents
- All Component Name should start with capital letter

Component structure

Import React from ‘react’;

Components…….

Export default component;

Class Components

```
Class HeaderComponent extends React.Component {
    Constructor(props){
      super(props)
      …….
    }
    render() {
       return (……..……)
    }
}
```

Functional Component

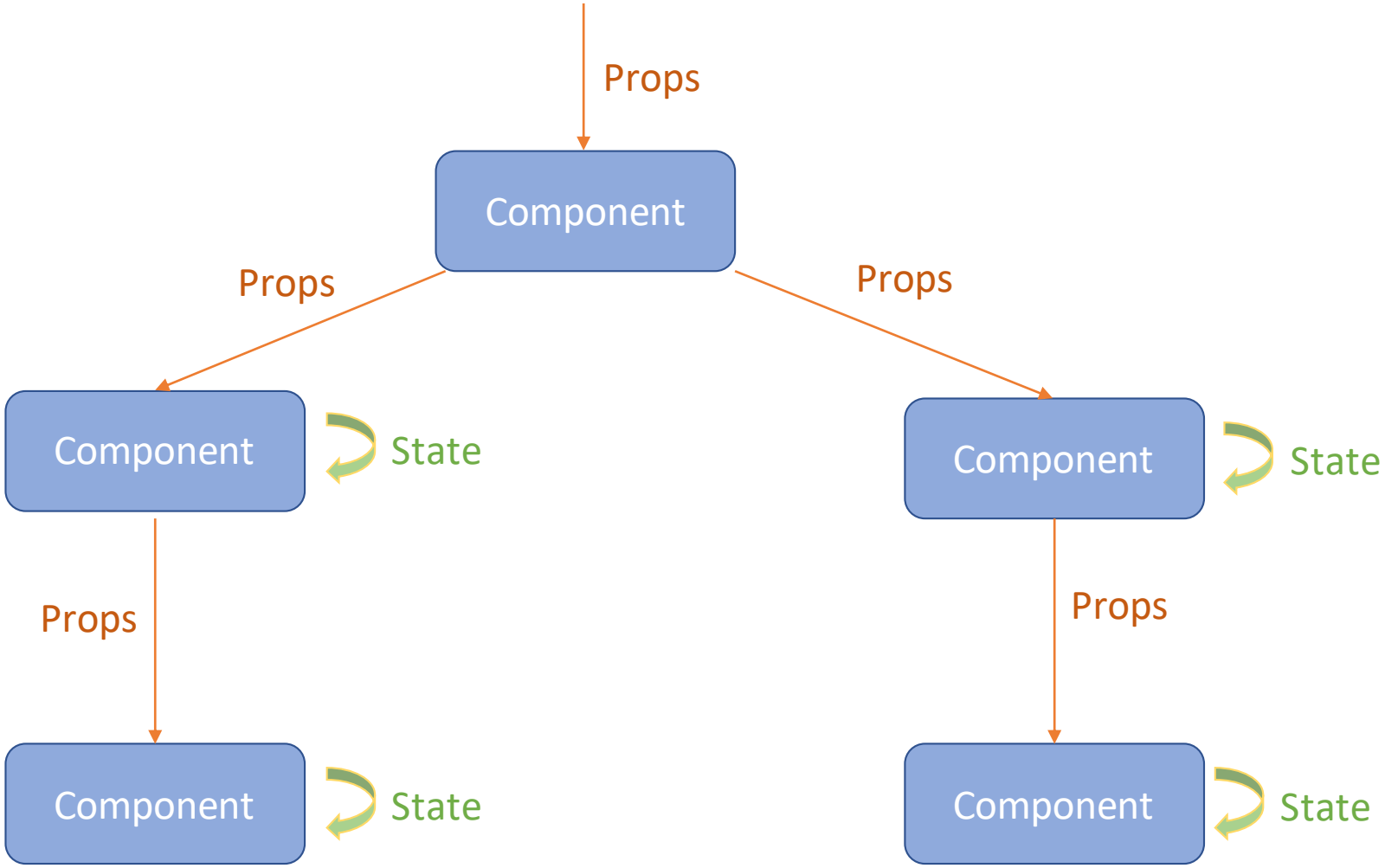
```
const HeaderComponent = ( ) => {
    return (
        ..................
     )
}
```

State & Props

State and props are data that we use or pass in components

State	**Props**
Mutable	Immutable
Can be owned by component only	Can be passed to child component
Can be used in class component only (can be used with functional component using hooks)	Can be used in both class and functional component
this.state.name	this.props.name

Data flow in React



JSX

JSX is Syntax extension to Javascript. JXS has HTML like syntax

```
const HeaderComponent = ( ) => {
    return (
        <div className="header" >Header</div>
    )
}
```

JSX rules

- Use <lowercase /> tags for DOM elements and <Capitalized /> Tags for components
- Attribute names should be camelCase like className=“ “
- Use quotes “ “ for string and braces { } for variables in attribute values
- Multiple JSX code should be wrapped with parantheses
- JSX should have single element at parent level. In case of multiple elements, wrap them with div or react fragment < >……. </ >
- All JSX tags should be properly closed <tag /> or <tag>…….</tag>

```
Import  React from ‘react’;

Class Header extends React.Component {
   Constructor(props){
     super(props)
     this.state = { name : “xyz” }
   }
   render() {
      return (
           <div>{this.state.name} </div>
     )
   }
}

Export default HeaderComponent;
```

In React class component constructor method is optional and is used only to bind event handlers and/or to initialize local state of component. Constructor method is fired automatically before the component is mounted.

We use super( ) method in constructor to call constructor of parent class which is React.Component

Initialize State

```
Constructor(props){
    super(props)
    this.state = {
        firstName : “Vikrant”,
        lastName  :  “Rana”
     }
}

Render ( ) {
  return <div>Hello {this.state.firstName}</div>
```

Update State

```
Const handleClick = ( ) => {
      this.setState  = ( {
          firstName : “Sahil”
      } );
}
```

Event Handler

```
Constructor(props){
      super(props)
      this.handleClick = this.bind.handleClick(this);
}

function handleClick( ) {  ……..……. }

Render( ) {
   return <button onClick={this.handleClick} > Submit  </button>
}
```

Event Handler using arrow functions

If we use arrow functions for event handlers, we would not need to bind this in constructor.

```
Constructor(props){
      super(props)
}

const handleClick = ( ) => {  ..... }

Render( ) {
   return <button onClick={this.handleClick} > Submit  </button>
}
```