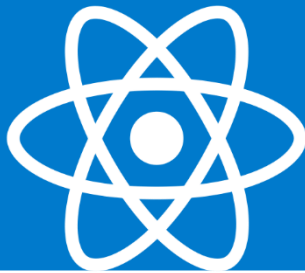




State in React JS

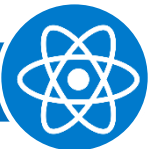
#React Notes

| State



What is State?

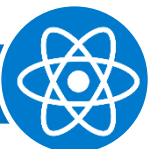
- State is like a **data store** to the ReactJS component.
- It is mostly used to update the component when user performed some action like clicking button, typing some text, pressing some key, etc.



How to Define State?

- React.Component is the base class for all **class based ReactJS components**.
- Whenever a class inherits the class React.Component its constructor will automatically assigns attribute state to the class with initial value is set to null.
- we can change it by overriding the method constructor.

```
Class MyClass extends React.Component
{
  constructor(props)
  {
    super(props);
    this.state = { attribute : "value" };
  }
}
```



Set and Get State Value

- Short Cut Key :

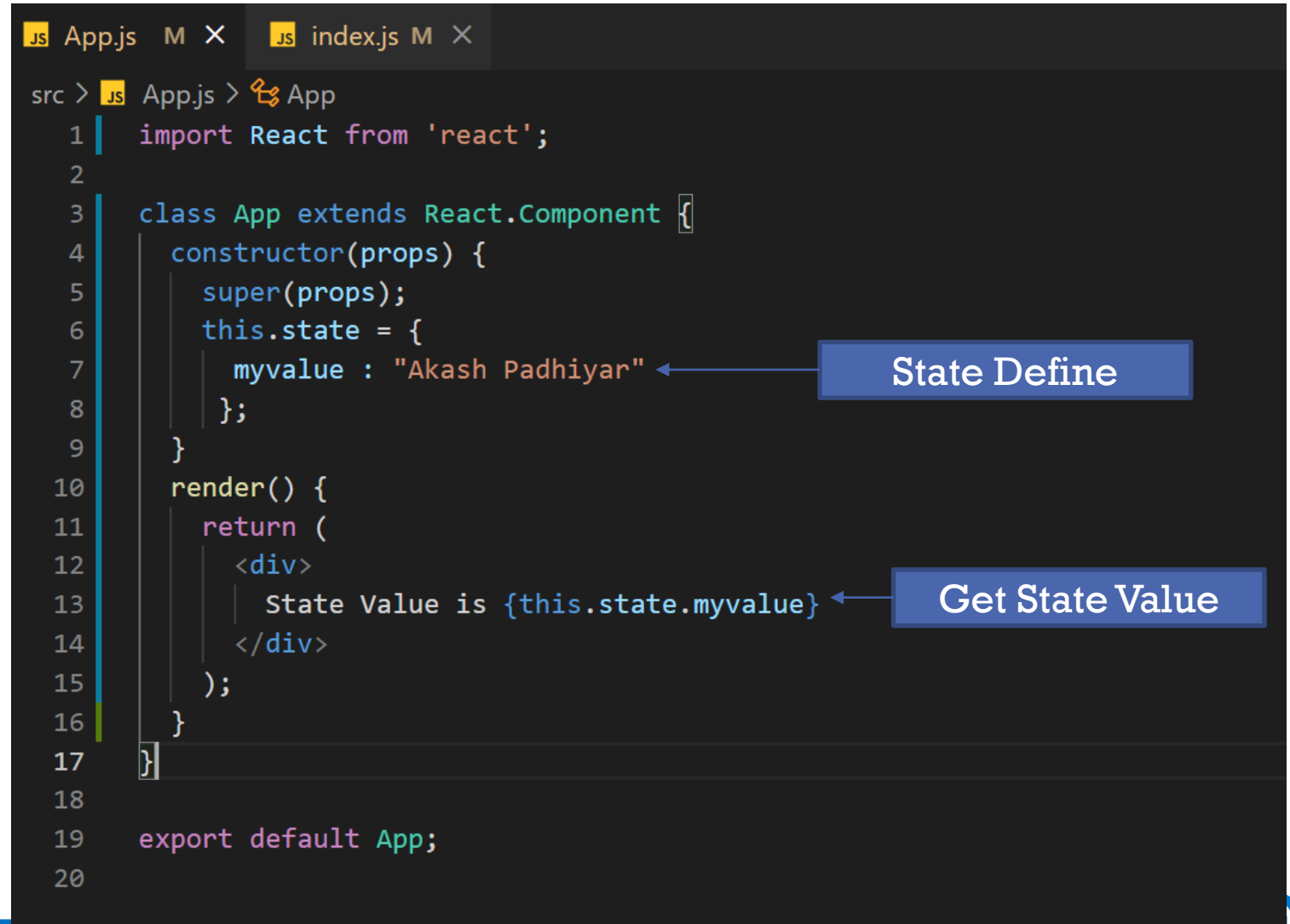
- Cccs

- State Define

```
constructor(props) {  
  super(props);  
  this.state = {  
    myvalue : "Akash"  
  };  
}
```

- Print State Value

```
{this.state.myvalue}
```

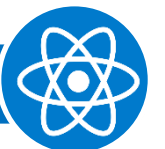
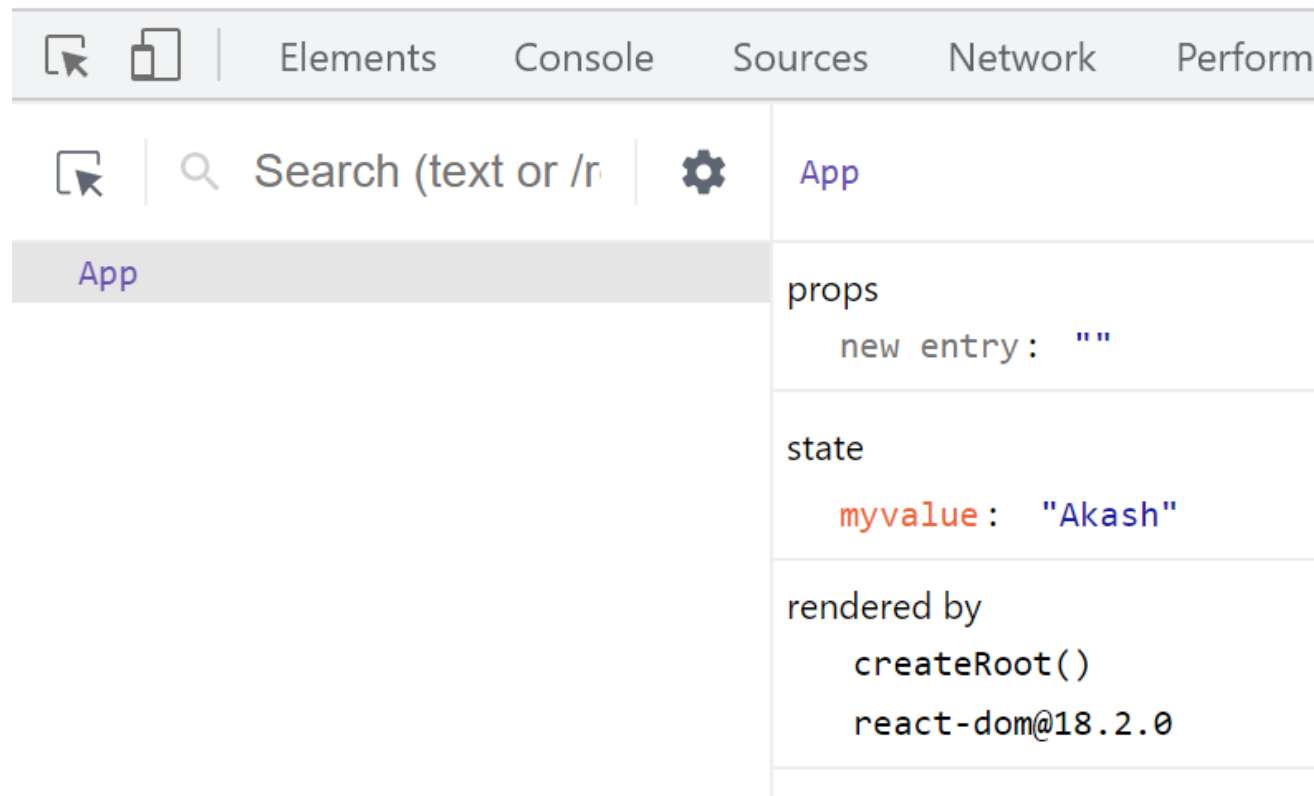


```
JS App.js M X JS index.js M X  
src > JS App.js > App  
1 | import React from 'react';  
2 |  
3 | class App extends React.Component {  
4 |   constructor(props) {  
5 |     super(props);  
6 |     this.state = {  
7 |       myvalue : "Akash Padhiyar" ← State Define  
8 |     };  
9 |   }  
10 |   render() {  
11 |     return (  
12 |       <div>  
13 |         State Value is {this.state.myvalue} ← Get State Value  
14 |       </div>  
15 |     );  
16 |   }  
17 | }  
18 |  
19 | export default App;  
20 |
```



Troubleshoot / Debug

- For Debugging purpose we can use Chrome Extension .
 - React Developer Chrome Extension : [React Developer Tools](#)



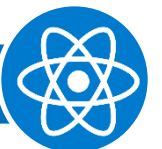
Set and Get State Value

- For Debugging purpose we can view values using React Developer Chrome Extension .

```
Js App.js ×
src > Js App.js > App > constructor
1  import React from 'react';
2  class App extends React.Component {
3    constructor(props) {
4      super(props);
5      this.state = {myvalue : "Akash"}
6    }
7    render() {
8      return (
9        <div>
10         {this.state.myvalue}
11       </div>
12     )
13   }
14 }
15 export default App;
16
```

The screenshot shows the React Developer Chrome Extension interface. The top bar includes tabs for Elements, Console, Sources, Network, and Perform. Below the tabs is a search bar and a settings gear icon. The main panel displays the component 'App' with its props and state. The props section shows 'new entry: ""'. The state section shows 'myvalue: "Akash"'. The rendered by section shows 'createRoot()' and 'react-dom@18.2.0'. The browser address bar at the bottom shows 'localhost:3000'.

Akash



Multiple Value Set and Get

- We can assign multiple values In State.

```
JS App.js ×
src > JS App.js > App
1  import React from 'react';
2  class App extends React.Component {
3    constructor(props) {
4      super(props);
5      this.state = {myvalue : "Akash",myage:30}
6    }
7    render() {
8      return (
9        <div>
10         {this.state.myvalue} <br/>
11         {this.state.myage}
12       </div>
13     )
14   }
15 }
16 export default App;
```

Akash
30

React App
localhost:3000

Akash
30

Elements Console Sources Network Perform

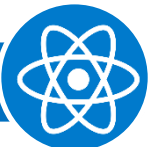
App

App

props
new entry: ""

state
myage: 30
myvalue: "Akash"

rendered by
createRoot()
react-dom@18.2.0



Example

```
import React from 'react';
class App extends React.Component {

  constructor(props) {

    super(props);

    this.state = {

      myvalue : "Akash Padhiyar"

    };
  }

  render() {

    return (

      <div>

        State Value is {this.state.myvalue}

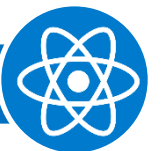
      </div>

    );

  }

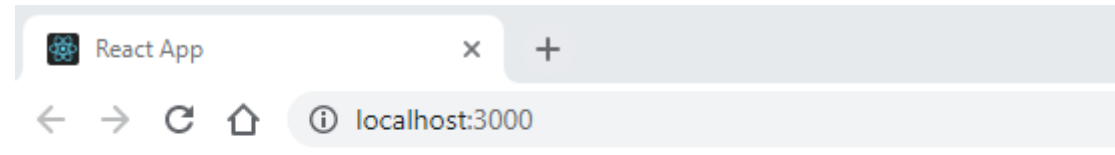
}

export default App;
```



Sum Using State Value

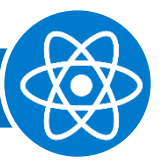
```
JS App.js M X
src > JS App.js > default
1 import React from 'react'
2
3 class App extends React.Component {
4   constructor(props) {
5     super(props);
6     this.state = {
7       txt1 : "10",
8       txt2 : "20"
9     }
10  }
11  render() {
12    var a = parseInt(this.state.txt1);
13    var b = parseInt(this.state.txt2);
14    var c = a + b;
15    return (
16      <div>
17        A Value is {a} <br/>
18        B Value is {b}<br/>
19        Sum is {c}
20      </div>
21    )
22  }
23 }
24 export default App;
```



A Value is 10

B Value is 20

Sum is 30

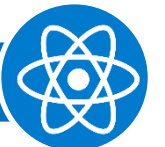


Destructuring Assignment

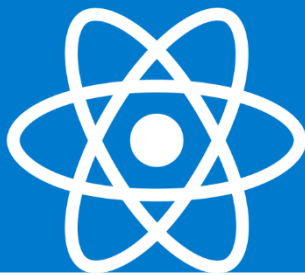
- Access State Value using Destructuring Assignment
- Name must be same

```
1  import React from 'react';
2  class App extends React.Component {
3
4      constructor(props) {
5          super(props);
6          this.state = { txt1:10,txt2 : 20}
7      }
8      render() {
9
10         //Destructuring Assignment
11         const {txt1,txt2} = this.state;
12         var c = txt1 + txt2; //Sum
13         return (
14             <div>
15                 A Value is {txt1} <br/>
16                 B Value is {txt2} <br/>
17                 Sum is {c}
18             </div>
19         );
20     }
21 }
22 export default App;
```

A Value is 10
B Value is 20
Sum is 30



| setState (Update State)

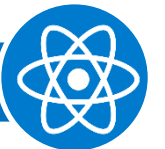


setState

- In many cases we need to update the state.
- To update the State we have to use setState method with new Value
- Example :

```
    this.setState(  
      {myvalue:"Aarav"  
    });
```

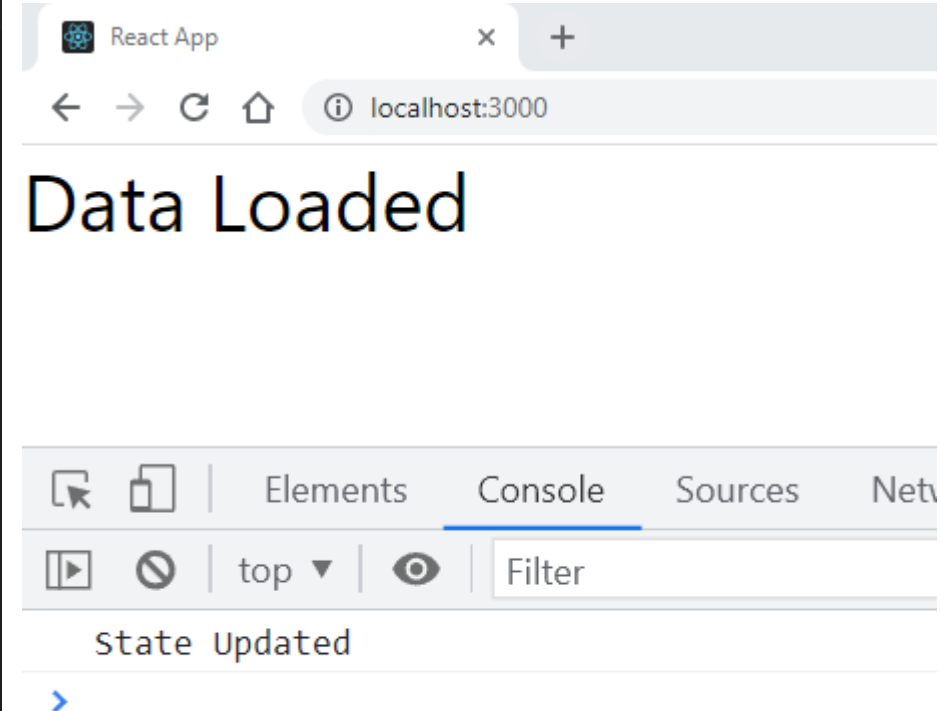
- Do not use this
 - `this.state = {'key': 'value'}` is strictly prohibited.



On Component Load Event Update State Value

- OnLoading of Component Value will be Assigned to State

```
App.js M X
src > JS App.js > [default]
1  import React from 'react';
2  class App extends React.Component {
3      constructor(props) {
4          super(props);
5          this.state = {
6              myvalue : "",
7          };
8      }
9      componentDidMount(){
10         console.log("State Updated"); //Print in Browser Console
11         this.setState({myvalue:"Data Loaded"}); //Update State Value
12     }
13     render() {
14         return (
15             <div>
16                 {this.state.myvalue}
17             </div>
18         );
19     }
20 }
21 export default App;
```

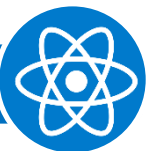


Update State Value

- To Update State Value we need to define button click / Change Event.

```
handleClick(){  
  
  this.setState(  
    {myvalue:"Aarav"  
  });  
  
}
```

```
<button onClick={this.handleClick.bind(this)}>Click</button>
```



Sum using Component Did Mount

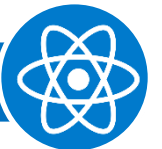
- OnLoading of Component Value will be assign to State and Print Sum

```
1  import React from 'react';
2  class App extends React.Component {
3    constructor(props) {
4      super(props);
5      this.state = {}
6    }
7    //First ComponentDidMount will call and assign value to State
8    componentDidMount(){
9      this.setState({
10        txt1 : 100,
11        txt2 : 200
12      });
13    }
14    render() {
15      //Destructuring Assignment
16      const {txt1,txt2} = this.state;
17      var c = txt1 + txt2; //Sum
18      return (
19        <div>
20          A Value is {txt1} <br/>
21          B Value is {txt2} <br/>
22          Sum is {c}
23        </div>
24      );
25    }
26  }
27  export default App;
```

A Value is 100

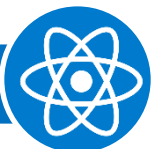
B Value is 200

Sum is 300



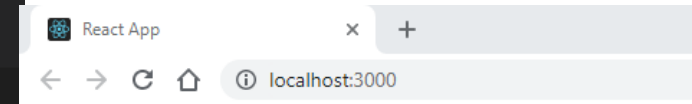
Update State Value using “setState”

```
Js App.js M X
src > Js App.js > App > constructor
1  import React from "react";
2  class App extends React.Component {
3    constructor(props) {
4      super(props);
5      this.state = {
6        myvalue: "Default",
7      };
8    }
9    btnClick() {
10     this.setState({
11       myvalue: "Akash Technolabs",
12     });
13   }
14   render() {
15     return (
16       <div>
17         State Value is : {this.state.myvalue}
18         <br />
19         <input type="button" onClick={this.btnClick.bind(this)} value="Click Me"/>
20       </div>
21     );
22   }
23 }
24 export default App;
25
```



setState Example

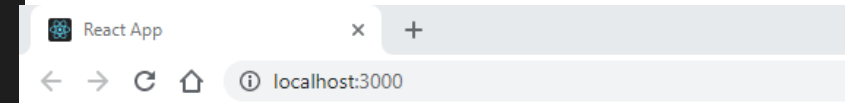
```
App.js
src > App.js > App > render
1 import React from 'react';
2 class App extends React.Component {
3   constructor(props) {
4     super(props);
5     this.state = {myvalue : "Akash",myage:30}
6   }
7
8   StateValueChange(){
9     this.setState(
10      {
11        myvalue : "Akash Technolabs",
12        myage : 11
13      }
14    );
15  }
16  render() {
17    return (
18      <div>
19        {this.state.myvalue} <br/>
20        {this.state.myage}<br/>
21        <input type="button" onClick={this.StateValueChange.bind(this)} value="ClickMe" />
22      </div>
23    )
24  }
25 }
26 export default App;
```



Akash

30

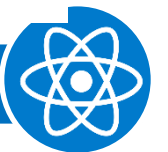
ClickMe



Akash Technolabs

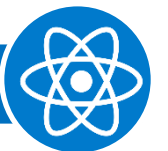
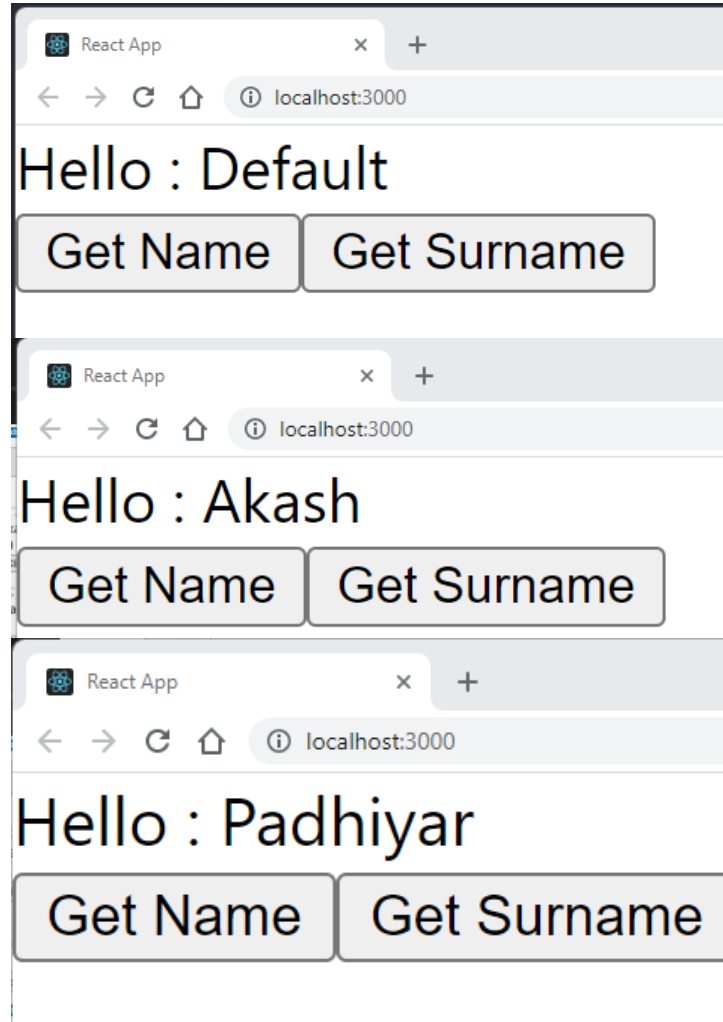
11

ClickMe



State Update with 2 Buttons

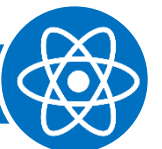
```
App.js
src > App.js > App > btnClick1
1 import React from 'react';
2 class App extends React.Component {
3   constructor(props) {
4     super(props);
5     this.state = {
6       myvalue : "Default"
7     };
8   }
9
10  btnClick1() {
11    this.setState(
12      { myvalue : 'Akash' }
13    );
14  }
15
16  btnClick2() {
17    this.setState(
18      { myvalue : 'Padhiyar' }
19    );
20  }
21  render() {
22    return (
23      <div>
24        Hello : {this.state.myvalue}
25      <br/>
26      <input type="button" onClick={this.btnClick1.bind(this)} value="Get Name" />
27      <input type="button" onClick={this.btnClick2.bind(this)} value="Get Surname" />
28    </div>
29  );
30 }
31 }
```



Example Code

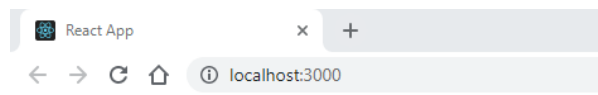
```
import React from "react";
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      myvalue: "Default",
    };
  }
  btnClick() {
    this.setState({
      myvalue: "Akash Technolabs",
    });
  }
}
```

```
render() {
  return (
    <div>
      State Value is : {this.state.myvalue}
      <br />
      <input type="button" onClick={this.btnClick.bind(this)} value="Click Me"/>
    </div>
  );
}
export default App;
```



Counter Demo

```
import React from "react";
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      myvalue: 0,
    };
  }
  btnClick() {
    this.setState({
      myvalue: this.state.myvalue + 1,
    });
  }
  render() {
    return (
      <div>
        Counter Value is : {this.state.myvalue}
        <br />
        <input type="button" onClick={this.btnClick.bind(this)} value="ClickMe" />
      </div>
    );
  }
}
export default App;
```



Counter Value is : 1

ClickMe

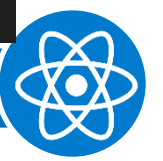
```
App.js M X
src > App.js > ...
1  import React from "react";
2  class App extends React.Component {
3    constructor(props) {
4      super(props);
5      this.state = {
6        myvalue: 0,
7      };
8    }
9
10   btnClick() {
11     this.setState({
12       myvalue: this.state.myvalue + 1,
13     });
14   }
15
16   render() {
17     return (
18       <div>
19         Counter Value is : {this.state.myvalue}
20         <br />
21         <input type="button" onClick={this.btnClick.bind(this)} value="ClickMe" />
22       </div>
23     );
24   }
25 }
26 export default App;
27
```

State Initialize

State Value Change

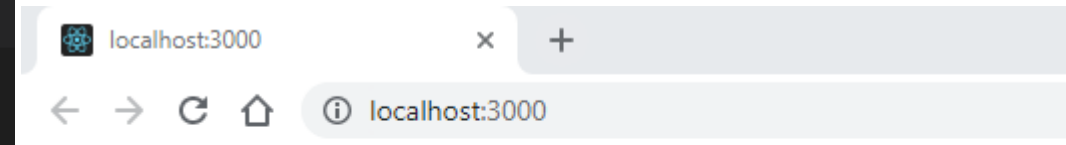
Print State Value

Button Click Method Bind



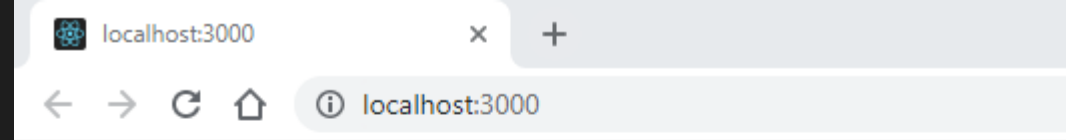
Dynamic Add State

```
App.js M X
src > App.js > App > handleClick
1  import React from 'react';
2
3  class App extends React.Component {
4    constructor(props) {
5      super(props);
6      this.state = { };
7    }
8    handleClick(){
9      this.setState(
10       {myvalue:"Dynamically generated"});
11    }
12
13    render() {
14      return (
15        <>
16        <h1>Class Component</h1>
17        {this.state.myvalue}
18        <button onClick={this.handleClick.bind(this)}>Click</button>
19        </>
20      );
21    }
22  }
23
24  export default App;
25
```



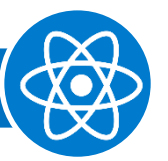
Class Component

Click



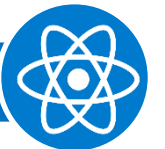
Class Component

Dynamically generated Click



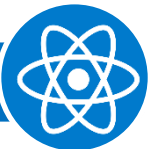
Task

- Make Counter Program with 2 Buttons
 - Increment (On Button Click Increment Value by + 1)
 - Decrement (On Button Click Increment Value by - 1)



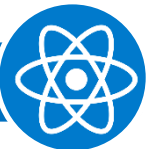
```
import React from "react";
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      myvalue: 0,
    };
  }
  btnClick() {
    this.setState({
      myvalue: this.state.myvalue + 1,
    });
  }
  render() {
    return (
      <div>
        Counter Value is : {this.state.myvalue}
        <br />
        <input type="button" onClick={this.btnClick.bind(this)} value="Click Me"/>
      </div>
    );
  }
}
```

```
export default App;
```



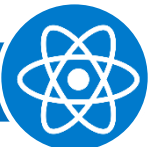
Props Vs State

- Props Means Passing Data From Parent to Child (No Action Only Data pass)
- State used to Change Variable Data (Click Event Action Preform)



What are the differences between props and state?

- Props are used to pass data(Between Component), whereas state is for managing data (Inside Component)
- State data can be modified by its own component, but is private (cannot be accessed from outside)
- Modifying state should happen with the setState () method
- Props can only be passed from parent component to child (unidirectional flow)
- Components receive data from outside with props, whereas they can create and manage their own data with state
- Data from props is read-only, and cannot be modified by a component that is receiving it from outside

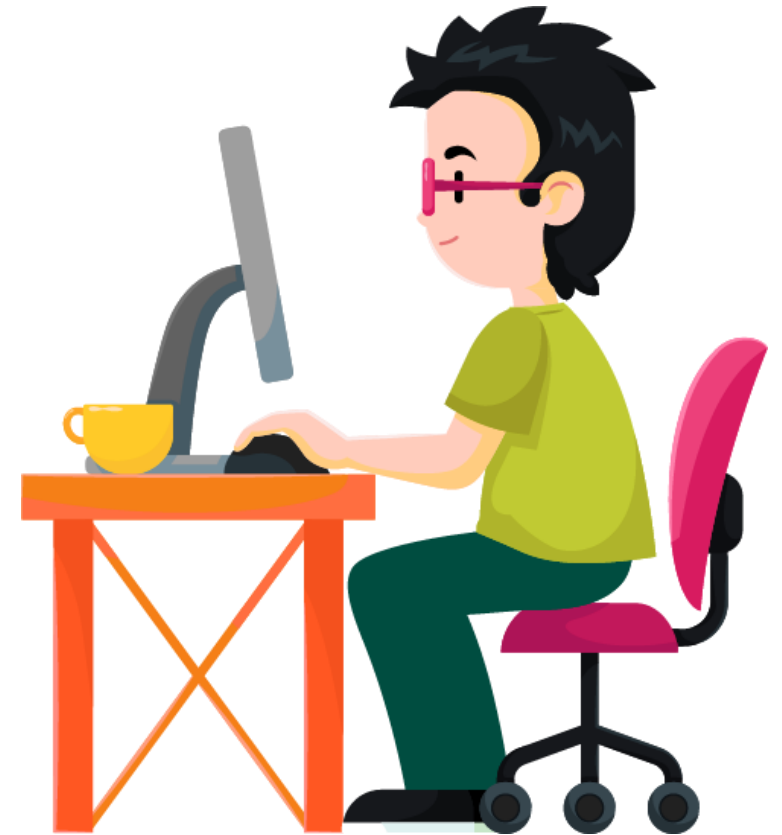


Get Exclusive Video Tutorials



www.apptutorials.com

<https://www.youtube.com/user/Akashtips>





Get More Details

www.akashsir.com



If You Liked It !

Rating Us Now



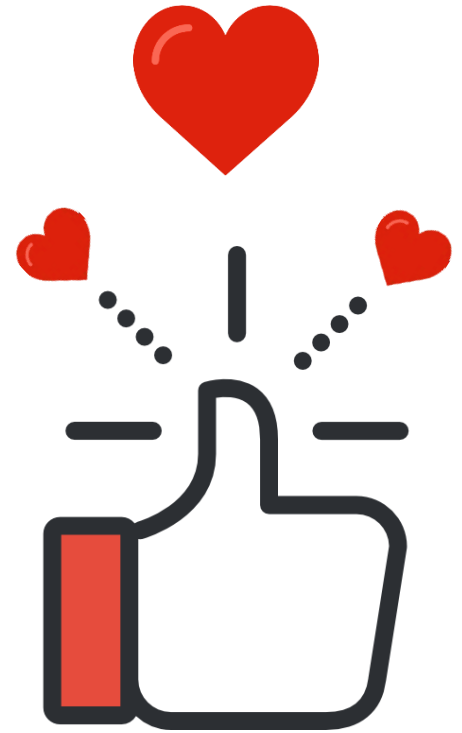
Just Dial

https://www.justdial.com/Ahmedabad/Akash-Technolabs-Navrangpura-Bus-Stop-Navrangpura/079PXX79-XX79-170615221520-S5C4_BZDET



Sulekha

<https://www.sulekha.com/akash-technolabs-navrangpura-ahmedabad-contact-address/ahmedabad>



Connect With Me



Akash Padhiyar
#AkashSir

www.akashsir.com

www.akashtechlabs.com

www.akashpadhiyar.com

www.apptutorials.com

Social Info



Akash.padhiyar



Akashpadhiyar



Akash_padhiyar



+91 99786-21654



#Akashpadhiyar

#apptutorials