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 it’s 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

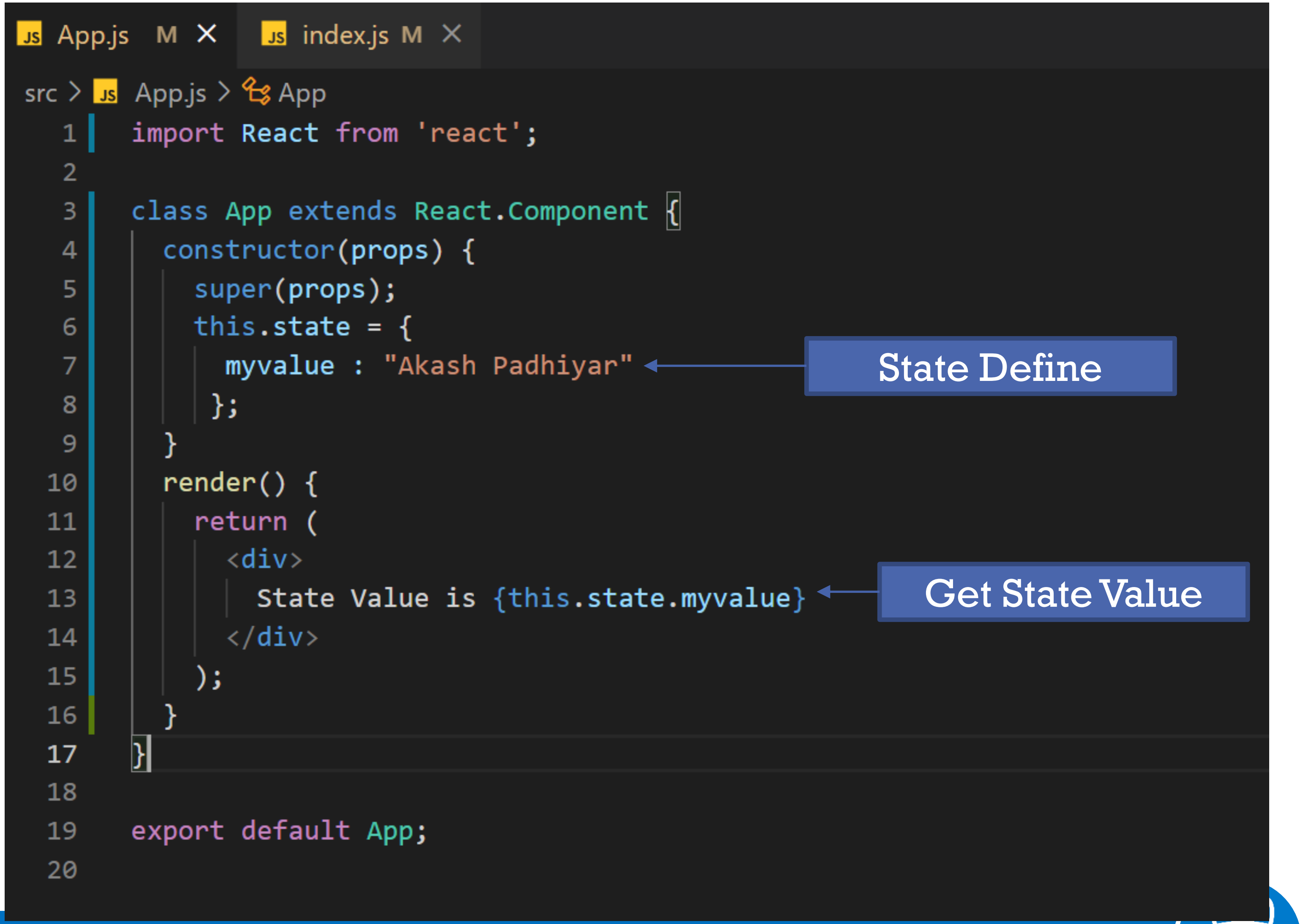
- Short Cut Key :
  - Cccs

- **State Define**

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

- **Print State Value**

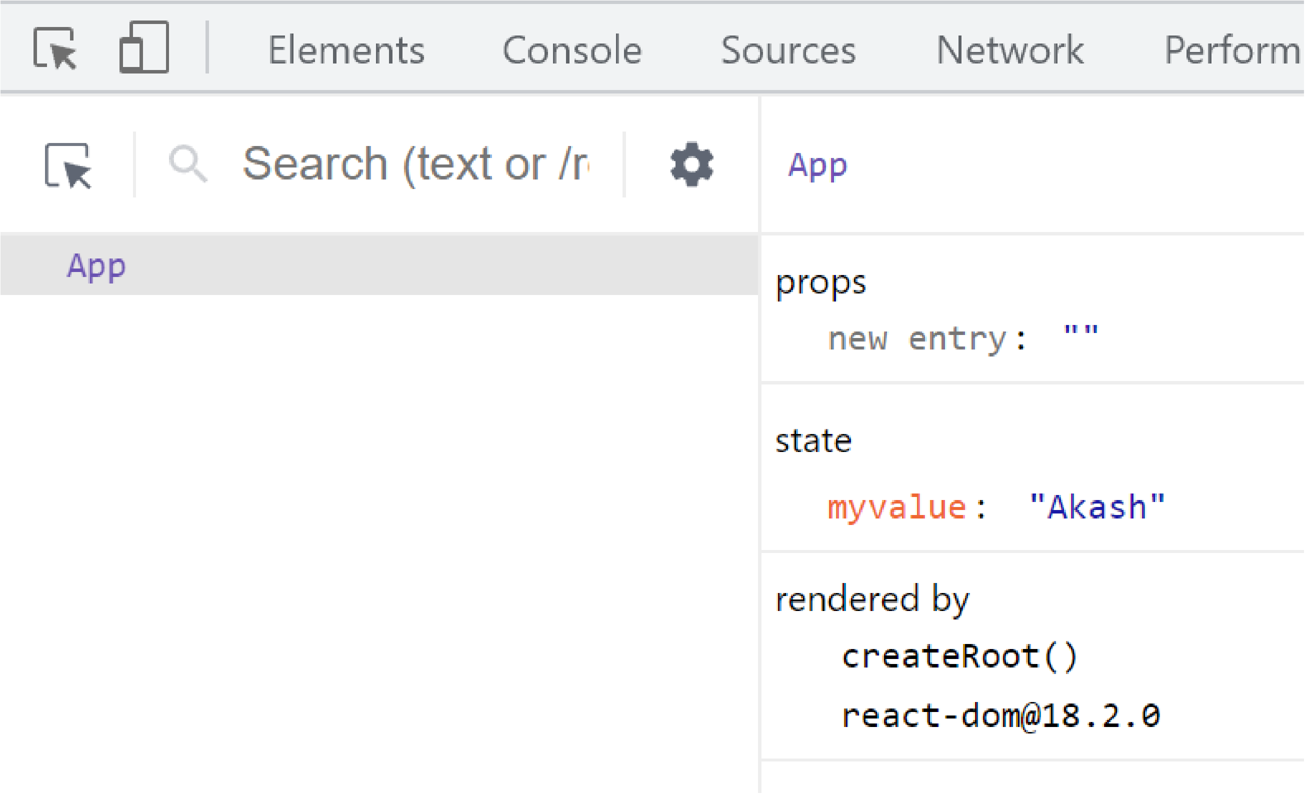
```
{this.state.myvalue}
```



```
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;

```

Troubleshoot / Debug

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



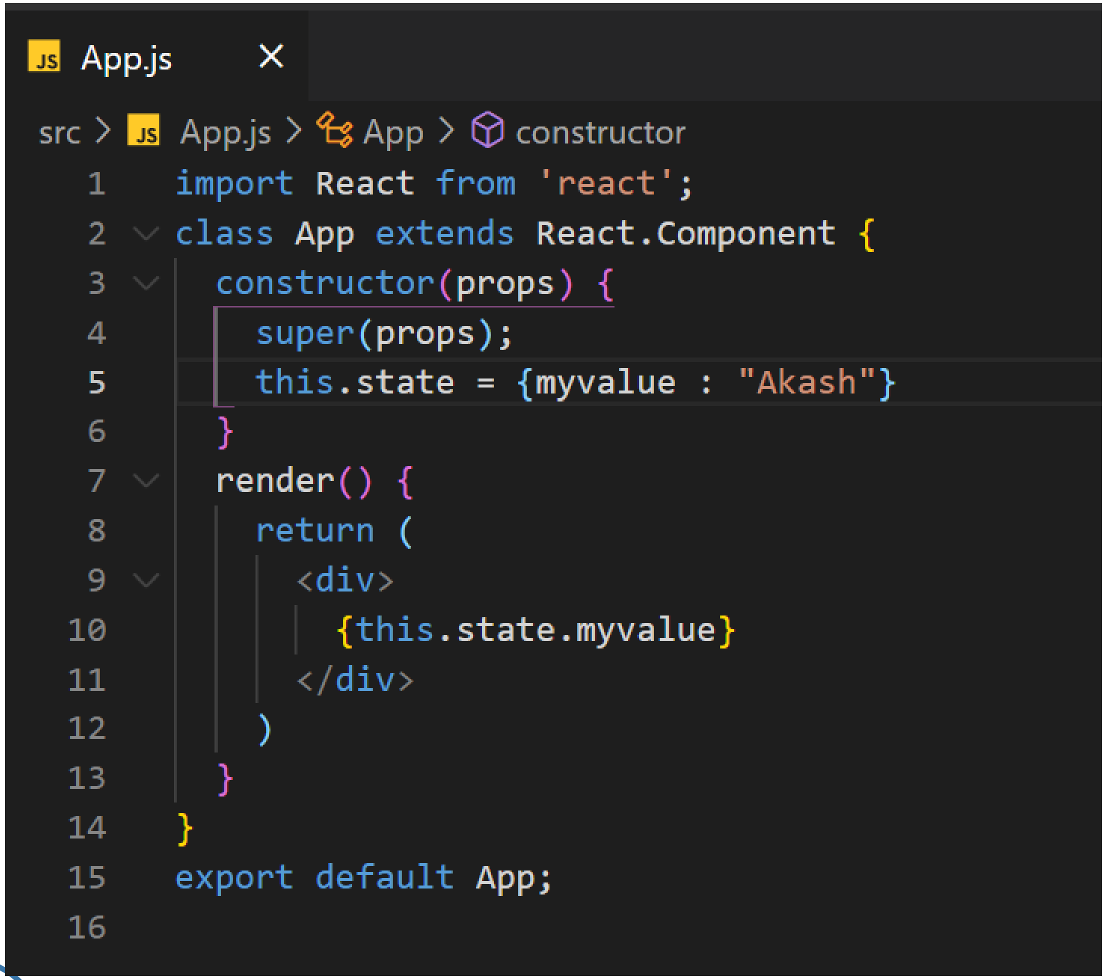
props
new entry: ""

state
myvalue: "Akash"

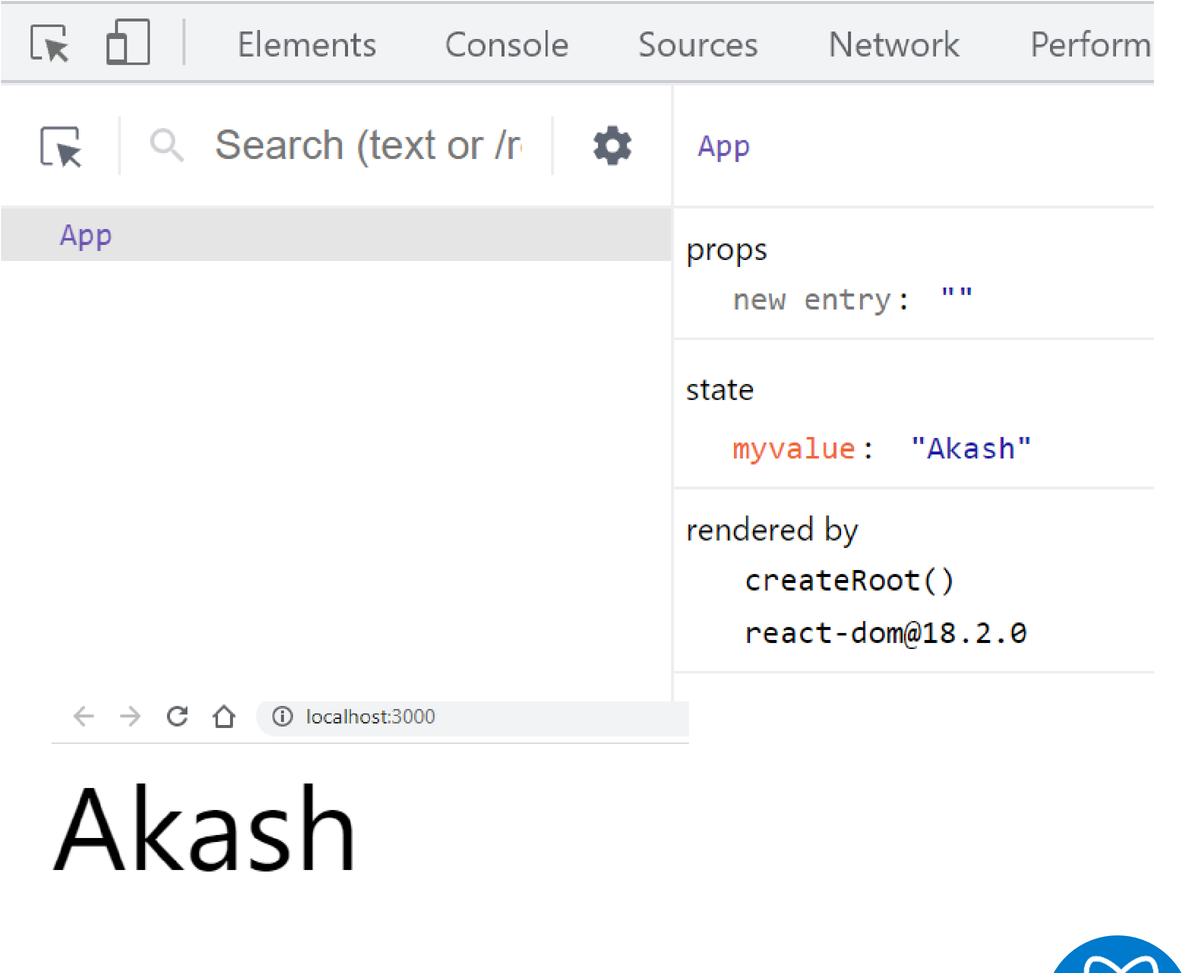
rendered by
createRoot()
react-dom@18.2.0

Set and Get State Value

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



```
import React from 'react';
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {myvalue : "Akash"}
  }
  render() {
    return (
      <div>
        {this.state.myvalue}
      </div>
    )
  }
}
export default App;

```



props
new entry: ""

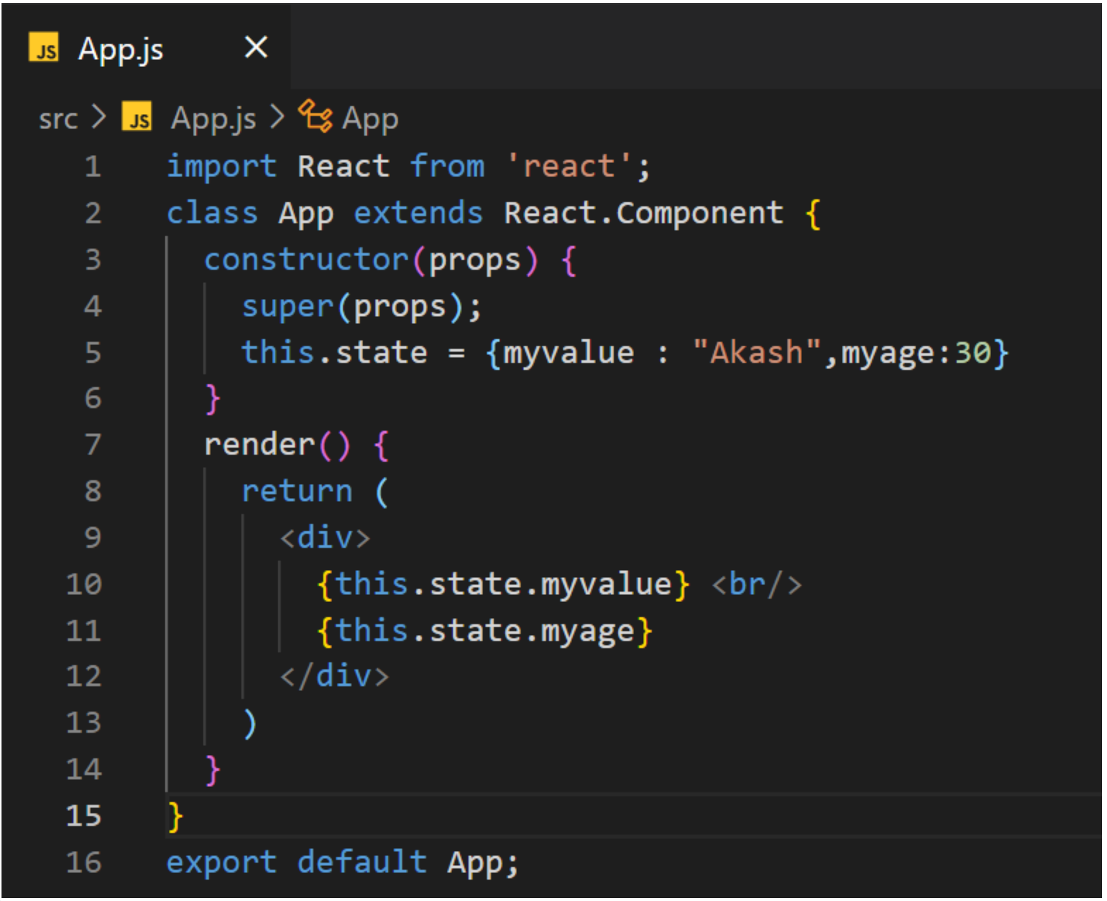
state
myvalue: "Akash"

rendered by
createRoot()
react-dom@18.2.0

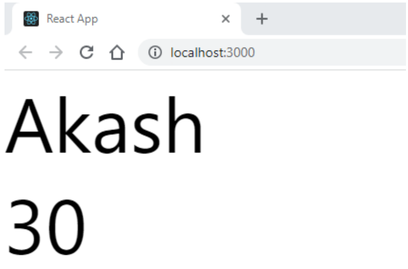
Akash

Multiple Value Set and Get

- We can assign multiple values In State.

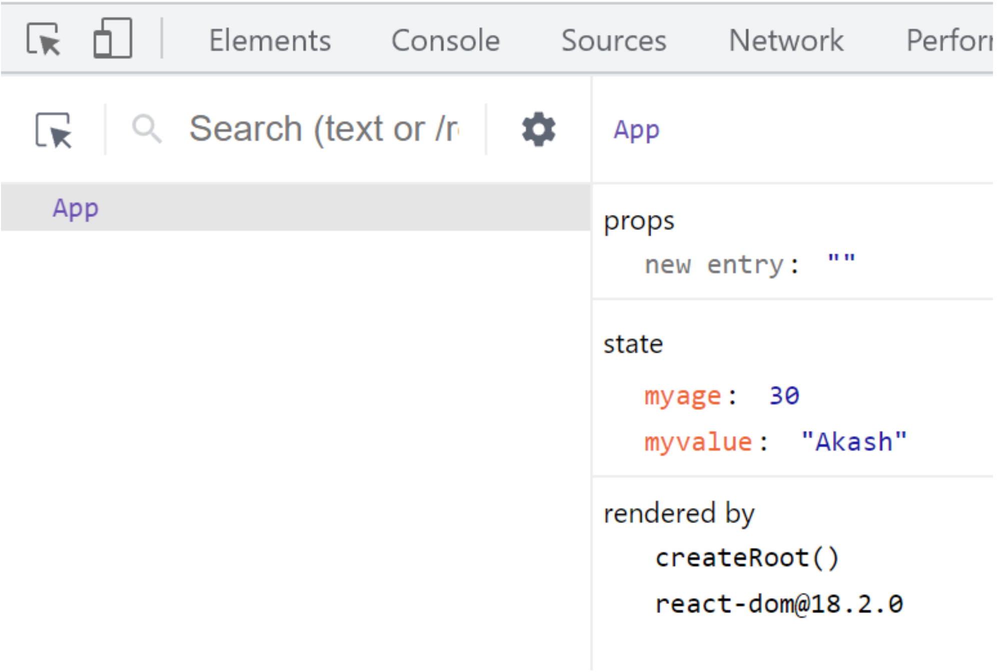


```
import React from 'react';
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {myvalue : "Akash",myage:30}
  }
  render() {
    return (
      <div>
        {this.state.myvalue} <br/>
        {this.state.myage}
      </div>
    )
  }
}
export default App;
```



Akash

30



props
new entry: ""

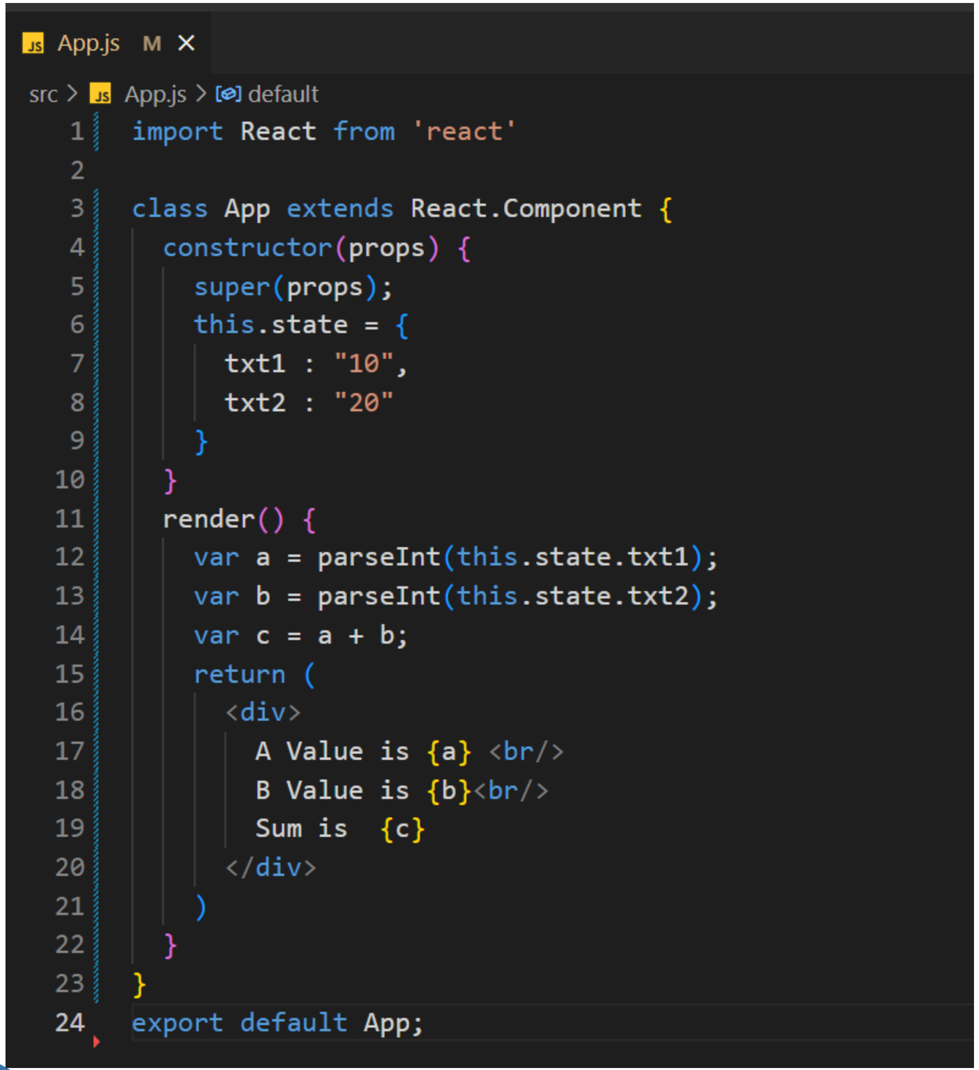
state
myage: 30
myvalue: "Akash"

rendered by
createRoot()
react-dom@18.2.0

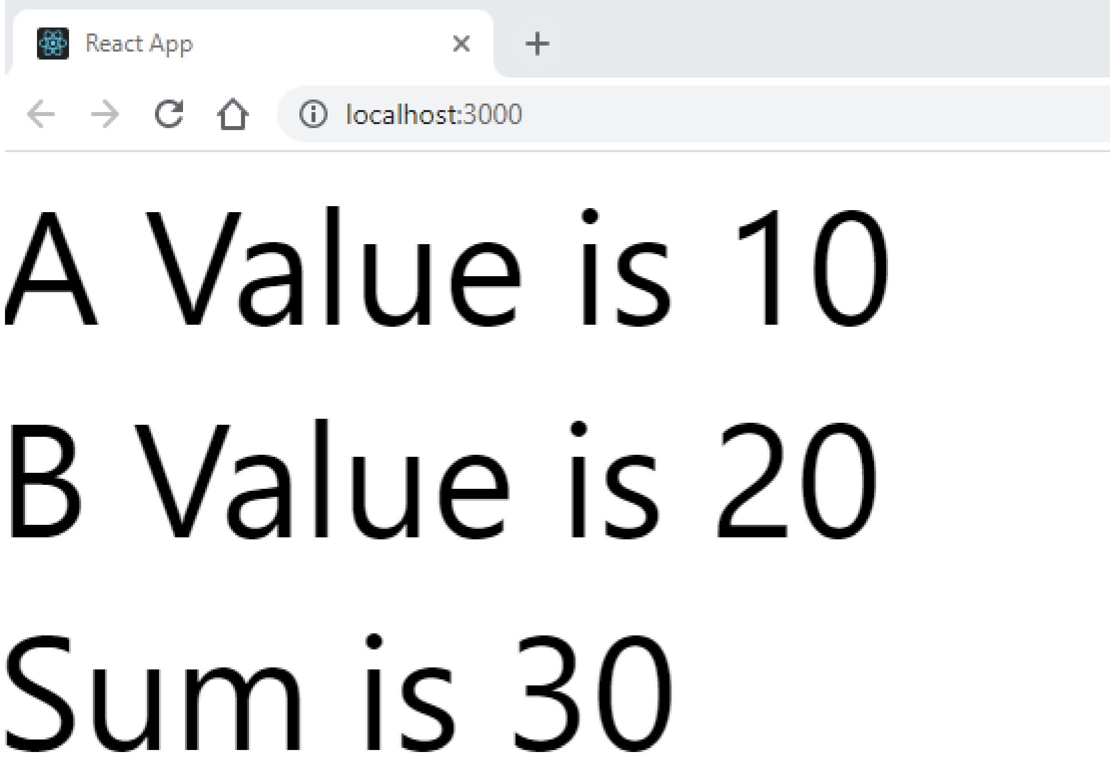
Example

```
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



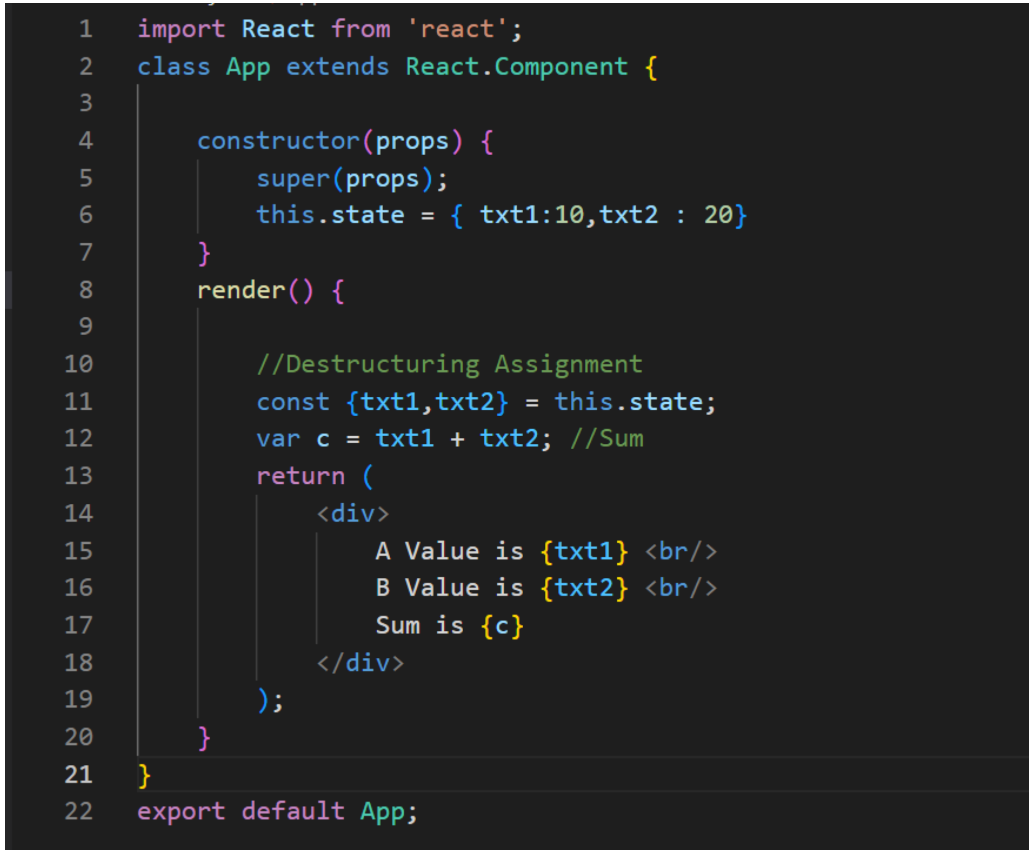
```
import React from 'react'

class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      txt1 : "10",
      txt2 : "20"
    }
  }
  render() {
    var a = parseInt(this.state.txt1);
    var b = parseInt(this.state.txt2);
    var c = a + b;
    return (
      <div>
        A Value is {a} <br/>
        B Value is {b}<br/>
        Sum is  {c}
      </div>
    )
  }
}
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



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

    constructor(props) {
        super(props);
        this.state = { txt1:10,txt2 : 20}
    }
    render() {

        //Destructuring Assignment
        const {txt1,txt2} = this.state;
        var c = txt1 + txt2; //Sum
        return (
            <div>
                A Value is {txt1} <br/>
                B Value is {txt2} <br/>
                Sum is {c}
            </div>
        );
    }
}
export default App;
```

```
A Value is 10
B Value is 20
Sum is 30
```

setState
(Update State)

setState

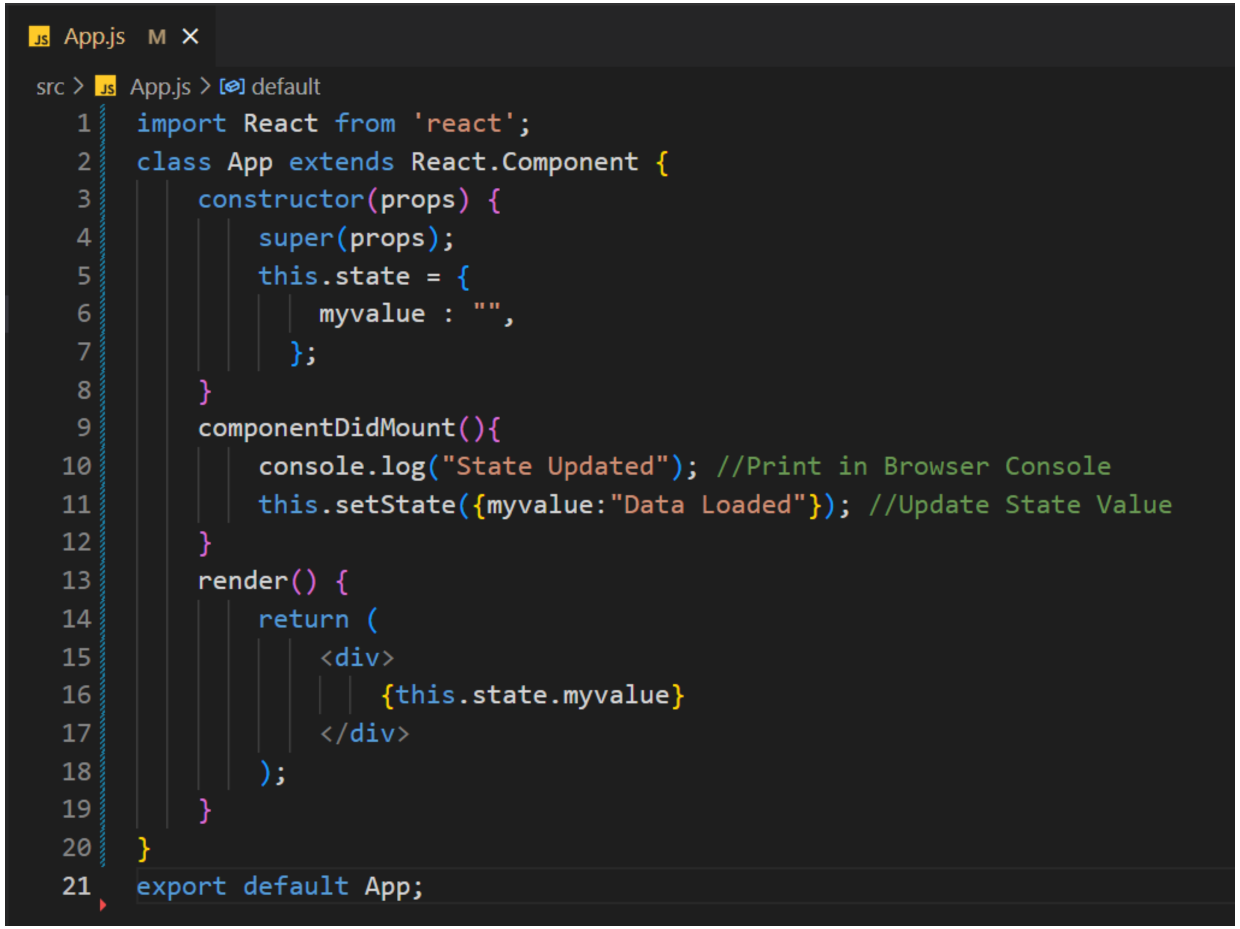
- In many cases we need to update the state.
- To update the State we have to use setState method with new Value
- Example :

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

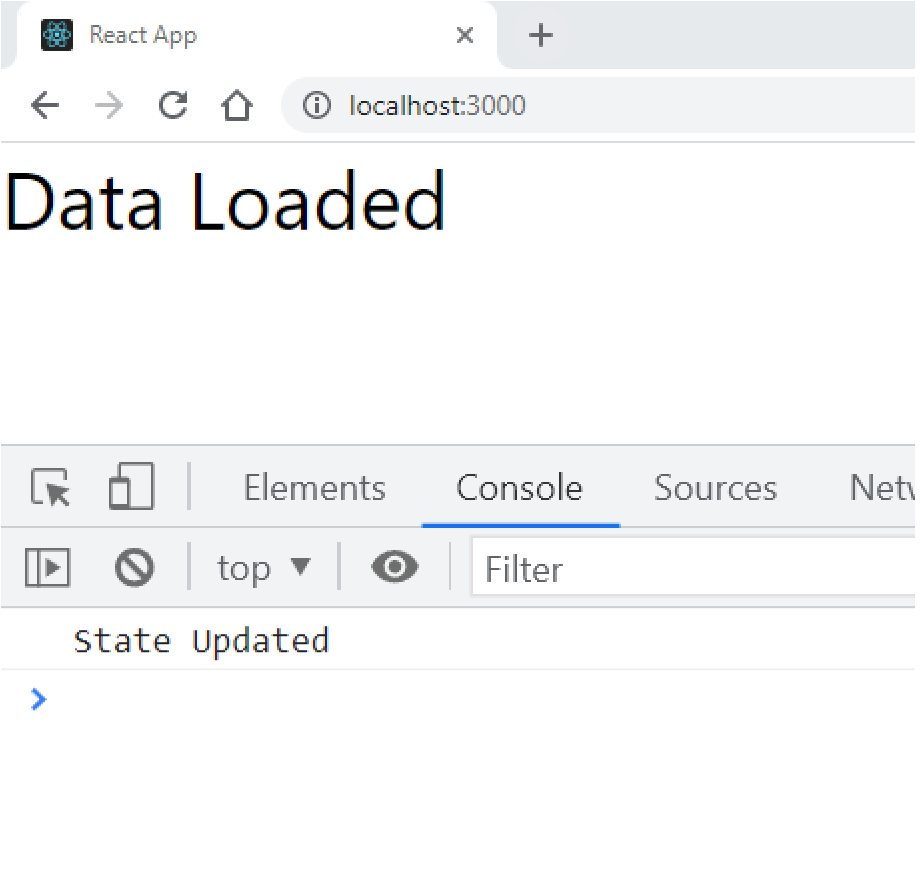
- Do not use this
  - this.state = {'key': 'value'} is strictly prohibited.

On Component Load Event Update State Value

- OnLoading of Component Value will be Assigned to State



```
import React from 'react';
class App extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            myvalue : "",
        };
    }
    componentDidMount(){
        console.log("State Updated"); //Print in Browser Console
        this.setState({myvalue:"Data Loaded"}); //Update State Value
    }
    render() {
        return (
            <div>
                {this.state.myvalue}
            </div>
        );
    }
}
export default App;
```



Data Loaded

```
State Updated
```

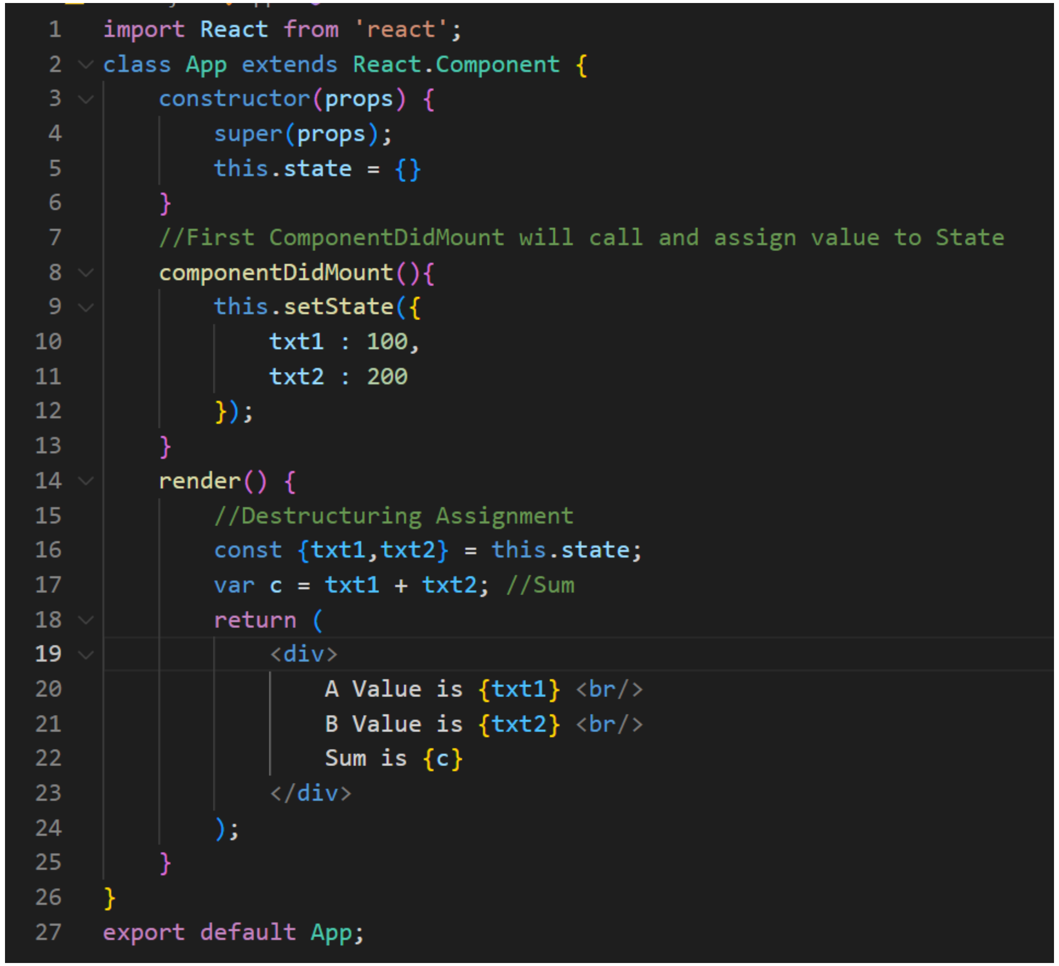
Update State Value

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

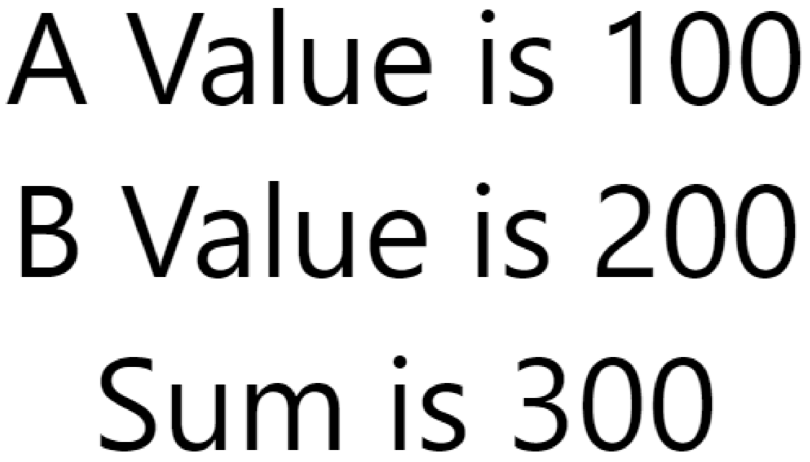
```
handleClick(){

   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

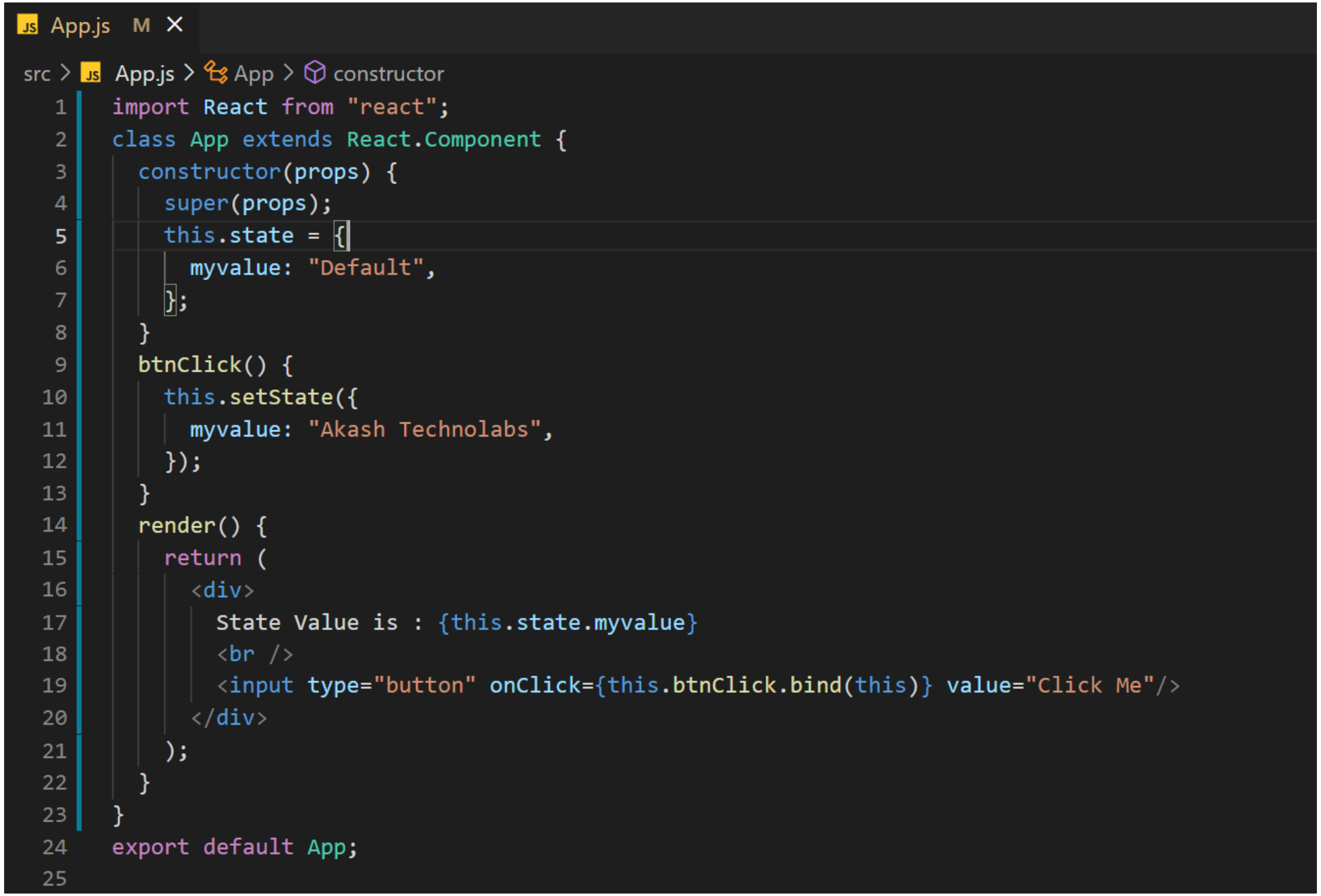


```
import React from 'react';
class App extends React.Component {
    constructor(props) {
        super(props);
        this.state = {}
    }
    //First ComponentDidMount will call and assign value to State
    componentDidMount(){
        this.setState({
            txt1 : 100,
            txt2 : 200
        });
    }
    render() {
        //Destructuring Assignment
        const {txt1,txt2} = this.state;
        var c = txt1 + txt2; //Sum
        return (
            <div>
                A Value is {txt1} <br/>
                B Value is {txt2} <br/>
                Sum is {c}
            </div>
        );
    }
}
export default App;
```



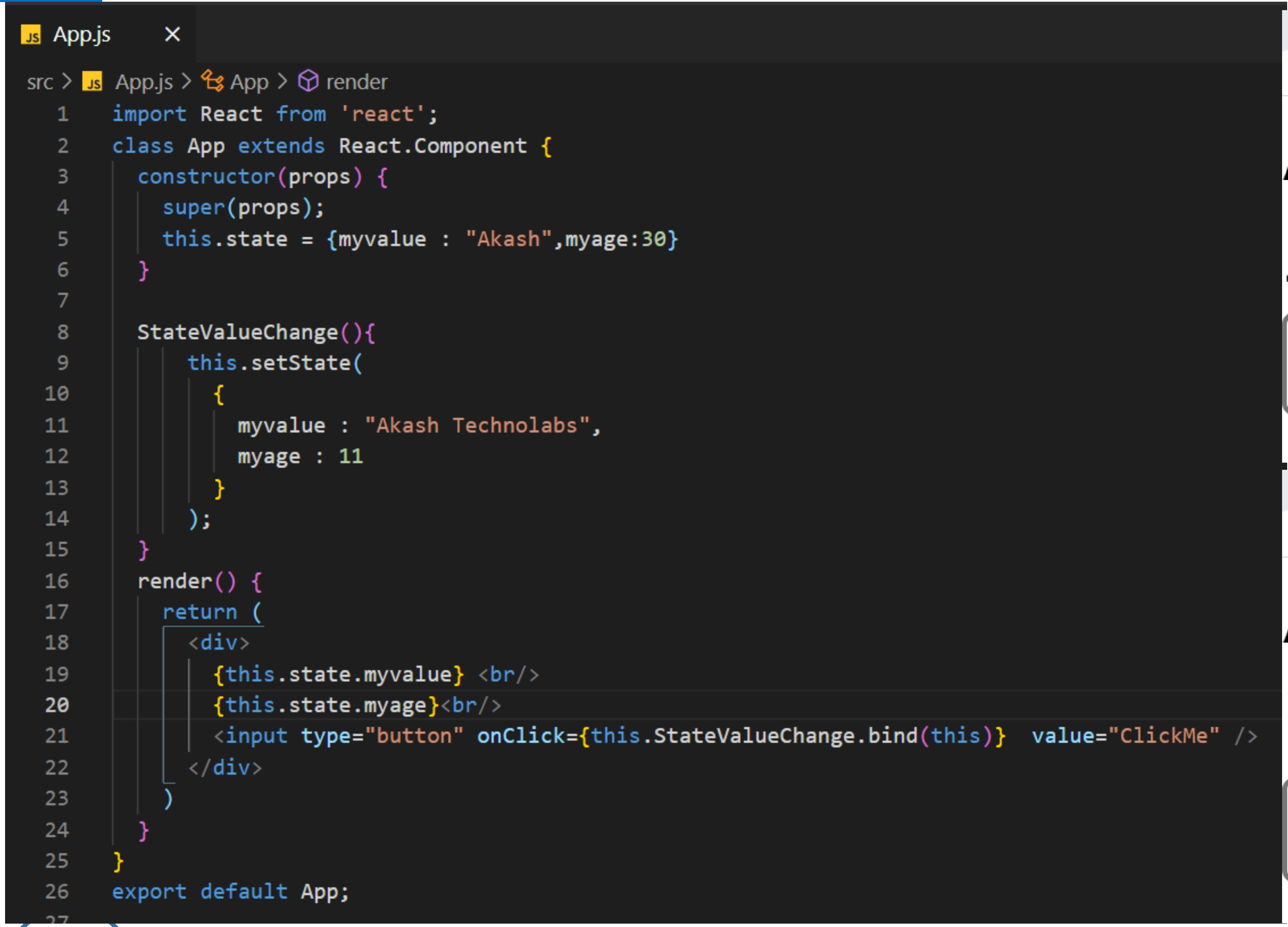
```
A Value is 100
B Value is 200
Sum is 300
```

Update State Value using “setState”

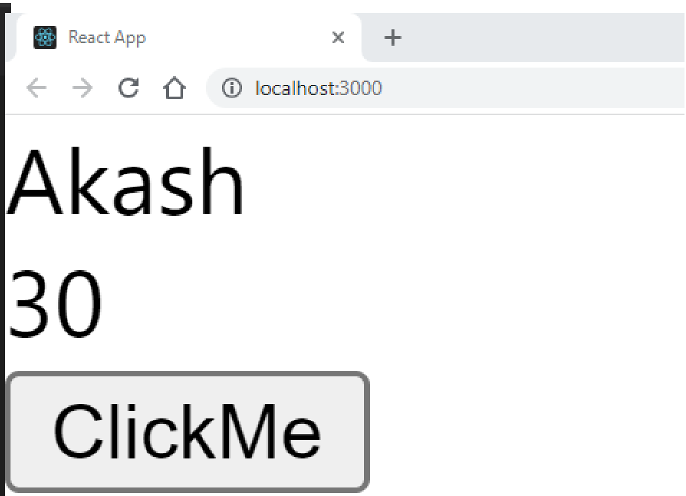


```
import React from "react";
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      myvalue: "Default",
    };
  }
  btnClick() {
    this.setState({
      myvalue: "Akash Technolabs",
    });
  }
  render() {
    return (
      <div>
        State Value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

```

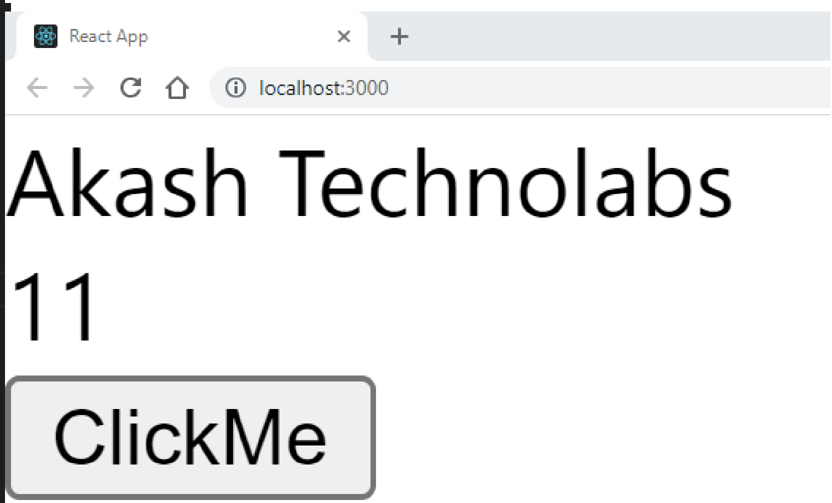
setState Example



```
import React from 'react';
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {myvalue : "Akash",myage:30}
  }

  StateValueChange(){
      this.setState(
        {
          myvalue : "Akash Technolabs",
          myage : 11
        }
      );
  }
  render() {
    return (
      <div>
        {this.state.myvalue} <br/>
        {this.state.myage}<br/>
        <input type="button" onClick={this.StateValueChange.bind(this)}  value="ClickMe" />
      </div>
    )
  }
}
export default App;
```

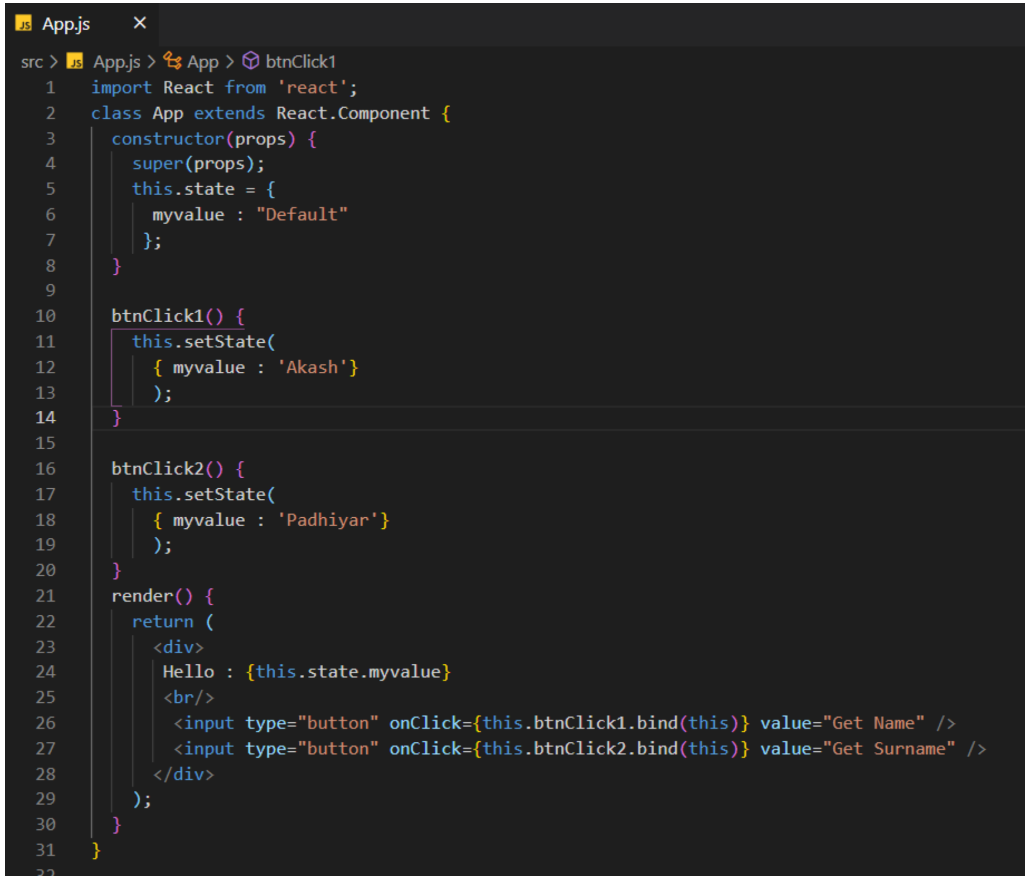


```
Akash
30
ClickMe
```

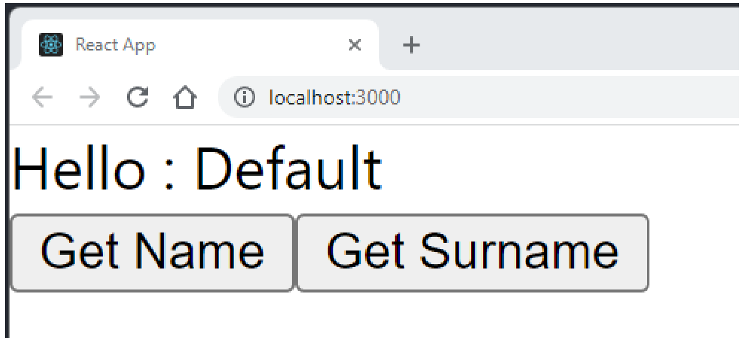


```
Akash Technolabs
11
ClickMe
```

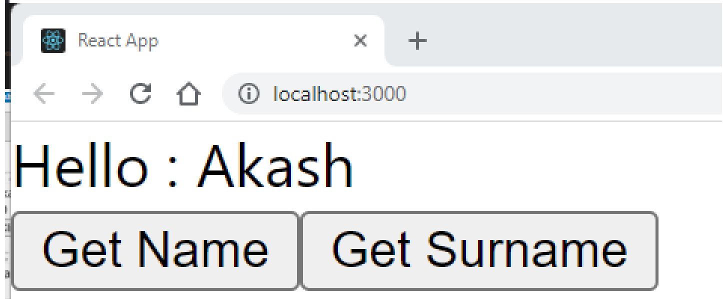
State Update with 2 Buttons



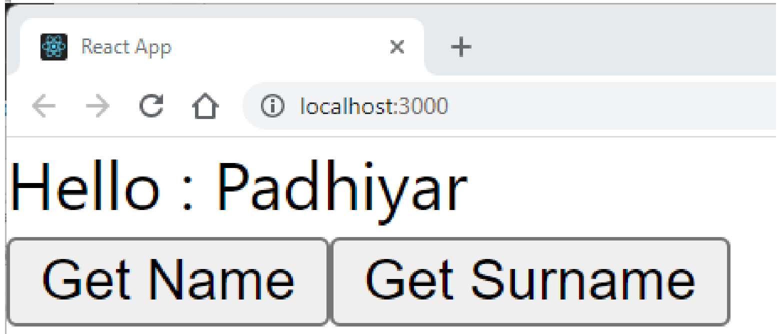
```jsx
import React from 'react';
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      myvalue : "Default"
     };
  }

  btnClick1() {
    this.setState(
      { myvalue : 'Akash'}
      );
  }

  btnClick2() {
    this.setState(
      { myvalue : 'Padhiyar'}
      );
  }
  render() {
    return (
      <div>
       Hello : {this.state.myvalue}
       <br/>
        <input type="button" onClick={this.btnClick1.bind(this)} value="Get Name" />
        <input type="button" onClick={this.btnClick2.bind(this)} value="Get Surname" />
      </div>
    );
  }
}
```



Hello : Default



Hello : Akash



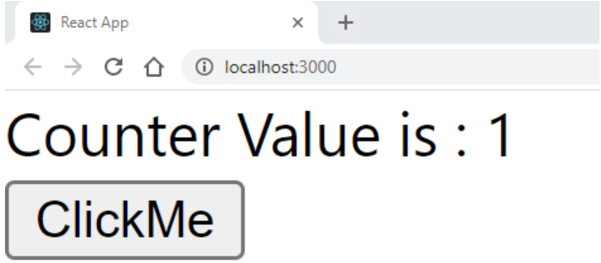
Hello : Padhiyar

Example Code

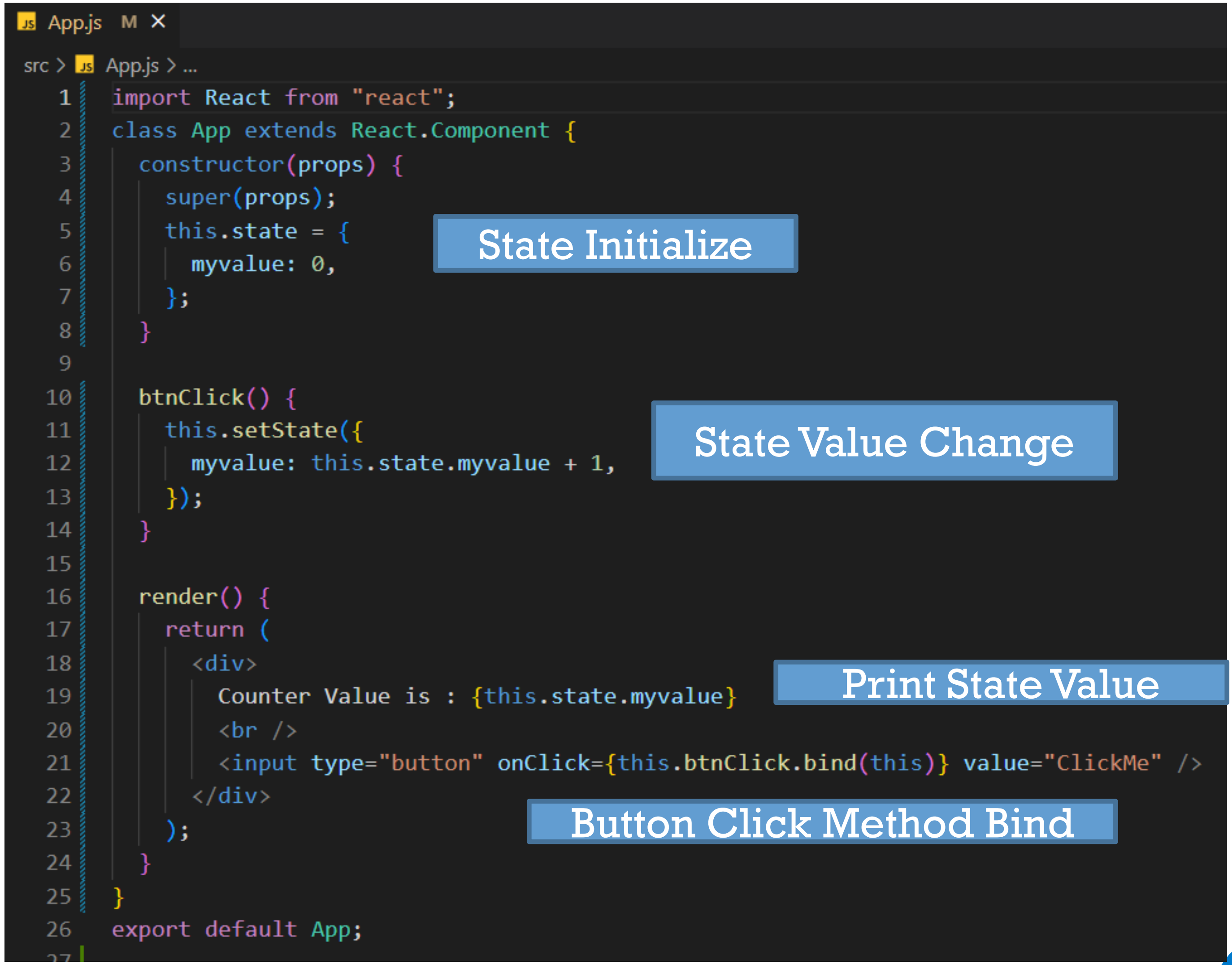
```
import React from "react";
class App extends React.Component {
 constructor(props) {
  super(props);
  this.state = {
   myvalue: "Default",
  };
 }
 btnClick() {
  this.setState({
   myvalue: "Akash Technolabs",
  });
 }
 render() {
  return (
   <div>
    State Value is : {this.state.myvalue}
    <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;
```

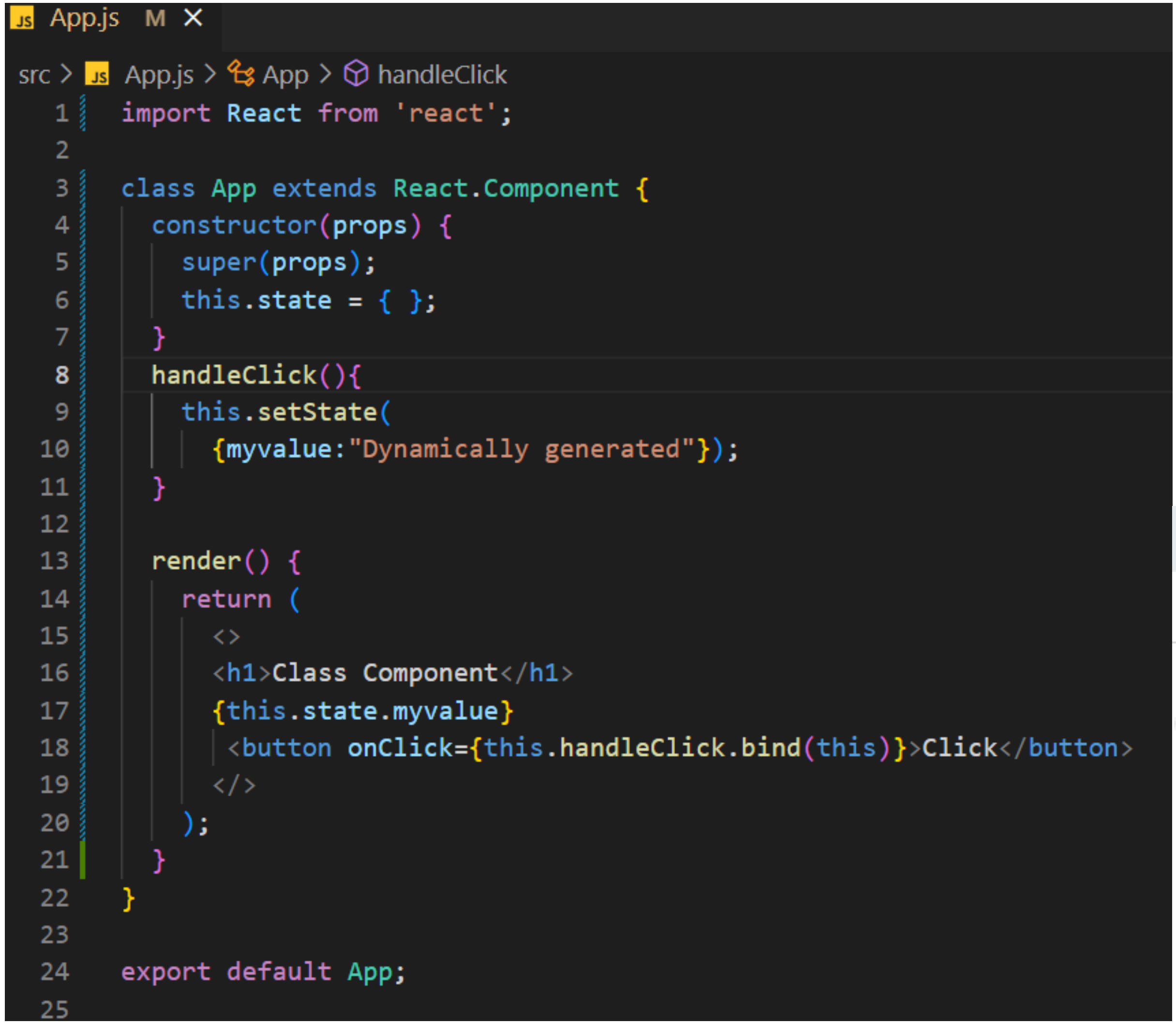


Counter Value is : 1

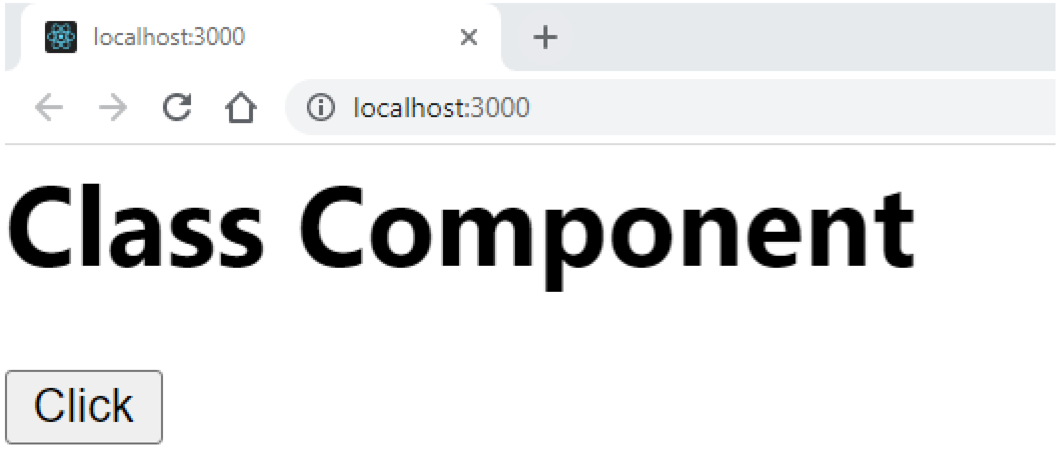


```
import React from "react";
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      myvalue: 0,
    };
  }

  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;
```

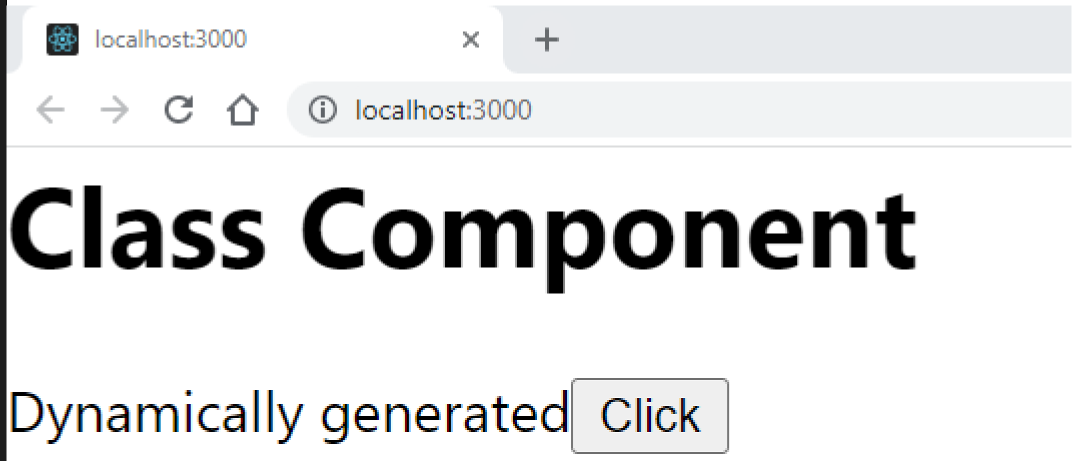
Dynamic Add State



```
import React from 'react';

class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = { };
  }
  handleClick(){
    this.setState(
      {myvalue:"Dynamically generated"});
  }

  render() {
    return (
      <>
      <h1>Class Component</h1>
      {this.state.myvalue}
       <button onClick={this.handleClick.bind(this)}>Click</button>
      </>
    );
  }
}

export default App;

```



```
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;
```

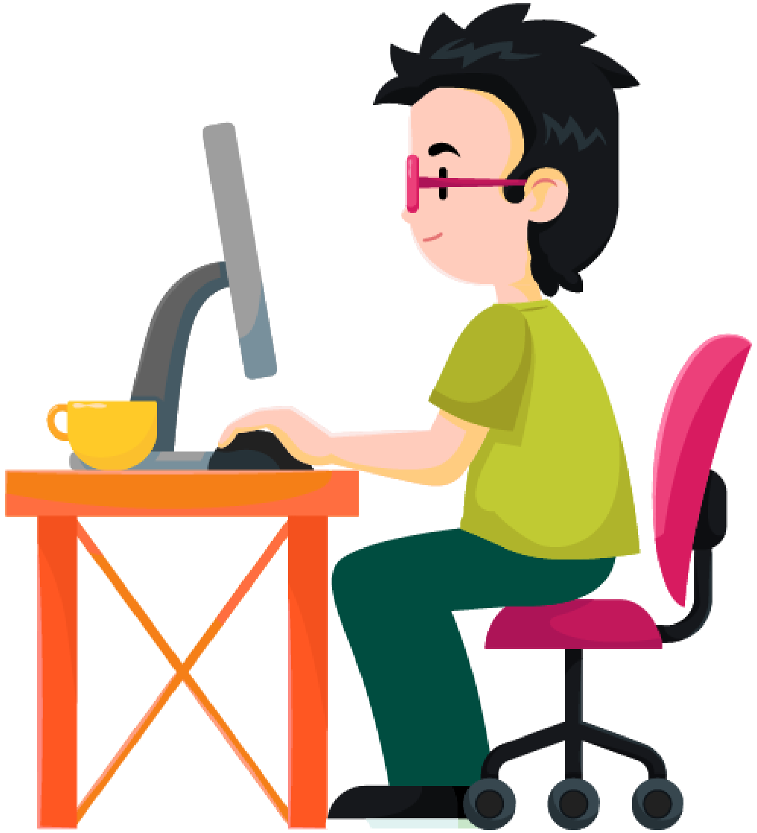
Poprs Vs State

- 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.aptutorials.com

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

Get More Details

www.akashsir.com

If You Liked It !

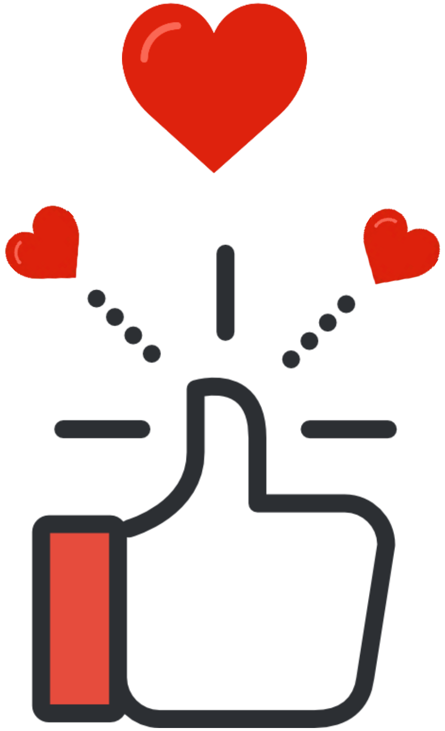
Rating Us Now

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.akashtechnolabs.com
www.akashpadhiyar.com
www.aptutorials.com

# Social Info

Akash.padhiyar

Akashpadhiyar

Akash_padhiyar

+91 99786-21654

#Akashpadhiyar
#aptutorials