STATE MANAGEMENT:

State represents the value of a dynamic properties of a React component at a given instance. React provides a dynamic data store for each component. The internal data represents the state of a React component and can be accessed using this.state member variable of the component. Whenever the state of the component is changed, the component will re-render itself by calling the *render()* method along with the new state.

React components have a built-in state object. The state is encapsulated data where you store assets that are persistent between component renderings.

The state is just a fancy term for a JavaScript data structure. If a user changes state by interacting with your application, the UI may look completely different afterwards, because it's represented by this new state rather than the old state.

App.js

```
import React ,{Components}from 'react';
import './App.css';
import Welcome from './Components/Welcome';
import Greet from './Components/Greet';
import Message from './Components/Message';
function App() {
return (
<div className="App">
<Message/>
</div>
);
}

export default App;
```

Message.js

```
import React ,{Component} from "react";

class Message extends Component{
constructor(){
super()
this.state ={
message:"Welcome to Internship"
}
}
changeMessage(){
this.setState({
message:"Thank you for joining"
})
}
```

```
render(){
return (
<div>
<h1>this.state.message</h1>
<button onClick ={()=> this.changeMessage()}>Welcome</button>
</div>

)
}
}
export default Message
```

Props vs State

props:

1.props get passed to the component

2.function parameters

3.props are immutable

4.props- functional components

this.props-Class components

App.js

```
import React ,{Component}from 'react';
import Display from './Display';

export default class App extends Component{
state ={
name:"Tuplescale"
}
render(){
return (
<div>
<center>
<Display name={this.state.name}/>
</center>
</div>
)
}
}
```

```
import React ,{Component}from 'react';

export default class Display extends Component{

render(){
return (
<div>
<h2>{this.props.name}</h2>
</div>
);
}
}
```

state:

1.State is managed within the component

2.varibles declared in the function body

3.state can be changed

4.useState Hook-Functional components

this.state-Class component

```
import React ,{Components}from 'react';

export default class App extends Components{
state ={
name:"Tuplescale"
}

render(){
return (
<div>
<center>
```

```
<h1> {this.state.name}</h1>
</center>
<Message/>
</div>
);
}
}
```

output:

Tuplescale