

# Higher Order Component example

## HOC.js

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

export default function Hoc(HocComponent){
  return class extends Component{
    render(){
      return (
        <div>
          <HocComponent> </HocComponent>
        </div>
      );
    }
  }
}
```

## App.js

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

class App extends Component {
  render() {
    return (
      <div>
        <h2>HOC Example</h2>
        JavaTpoint provides best CS tutorials.
      </div>
    )
  }
}

App = Hoc(App);
export default App;
```

# Hooks

Hooks are essential when working with Functional Components  
Most frequently used hooks: **useState**, **useEffect**.

## **useState:**

```
const [name, setName] =  
useState('Tom')
```

Is used to as a hook to state,  
this state will be tracked and  
your page re-render whenever  
the state is changed.

```
useEffect(() => {  
  //do stuff  
}, [dependencyArray])
```

- Most commonly used with a dependencyArray
- Replaces the need to use LifeCycle hooks
- It is designed to fire effects, what are effects? Some kind of event that will change your state, API calls, click events