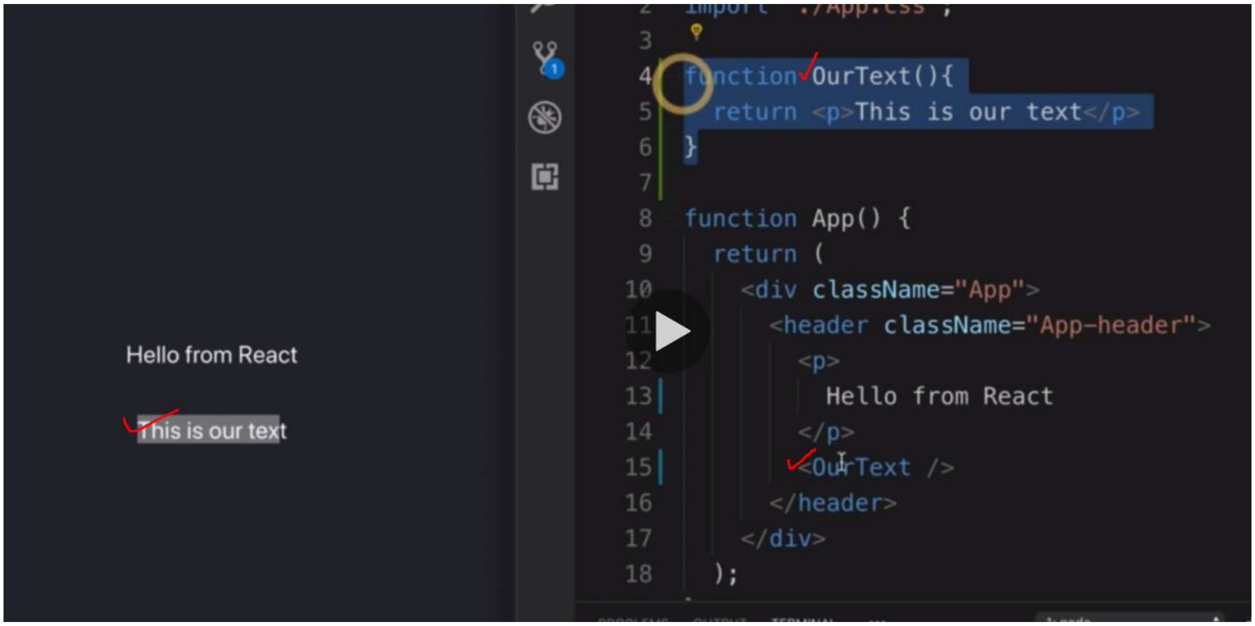
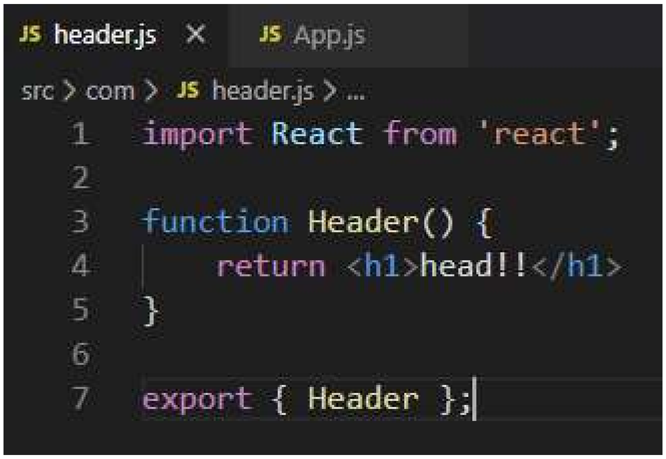
Component



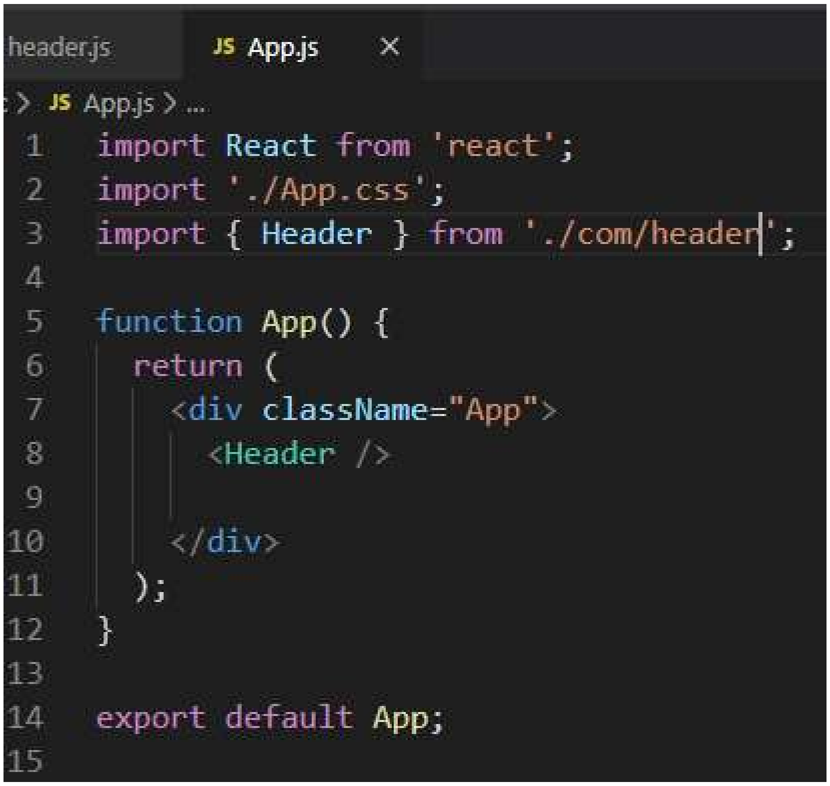
Hello from React

This is our text

```
import './App.css';

function OurText(){
  return <p>This is our text</p>
}

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <p>
          Hello from React
        </p>
        <OurText />
      </header>
    </div>
  );
```

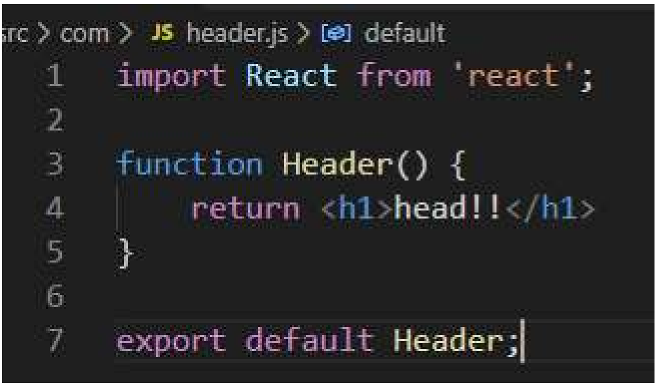


```js
import React from 'react';

function Header() {
    return <h1>head!!</h1>
}

export { Header };
```



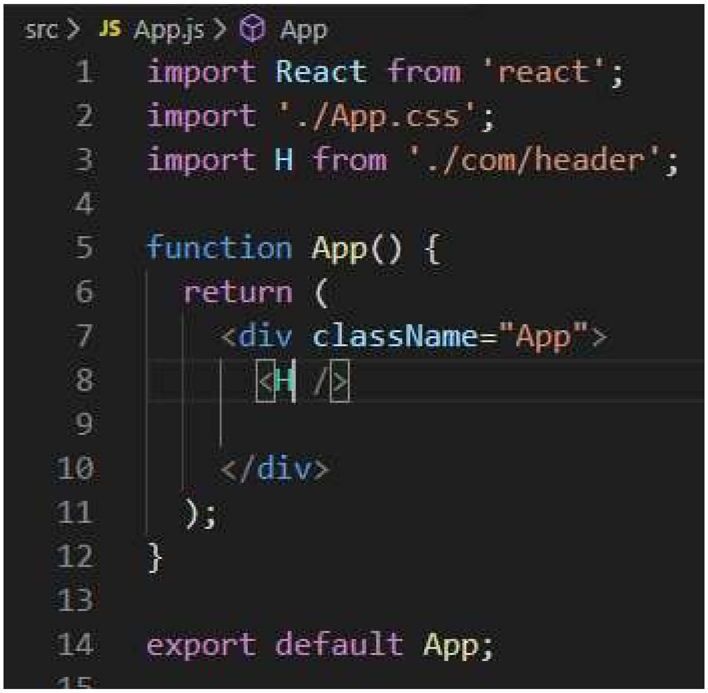
```js
import React from 'react';
import './App.css';
import { Header } from './com/header';

function App() {
  return (
    <div className="App">
      <Header />

    </div>
  );
}

export default App;

```

With export default you can rename

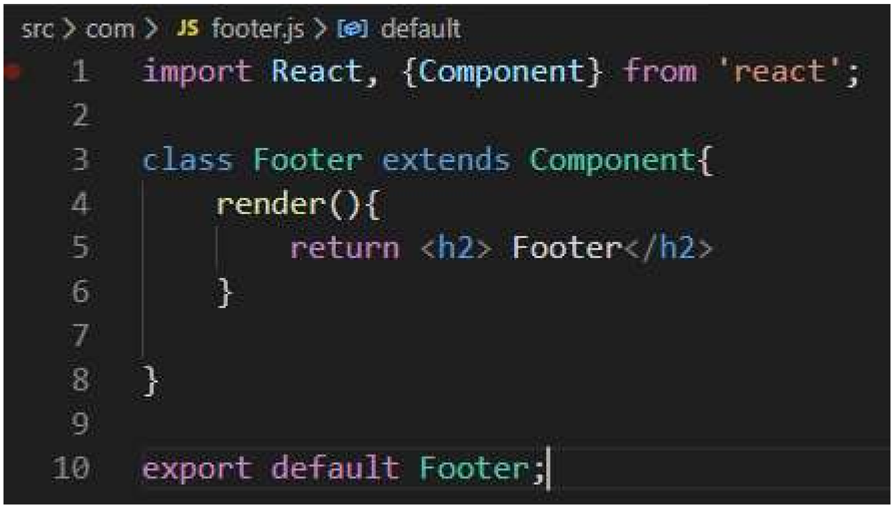


```
import React from 'react';

function Header() {
    return <h1>head!!</h1>
}

export default Header;
```

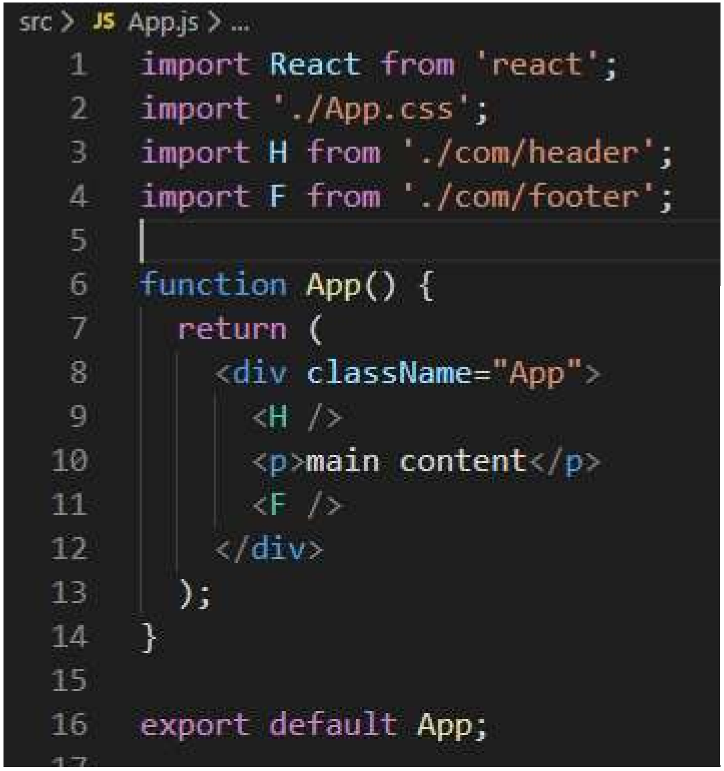
d



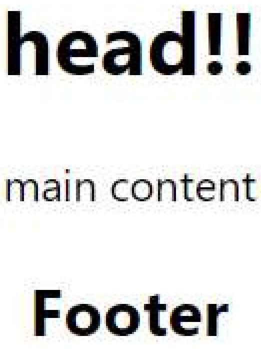
```
import React from 'react';
import './App.css';
import H from './com/header';

function App() {
  return (
    <div className="App">
      <H />

    </div>
  );
}

export default App;
```



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

class Footer extends Component{
    render(){
        return <h2> Footer</h2>
    }

}

export default Footer;
```

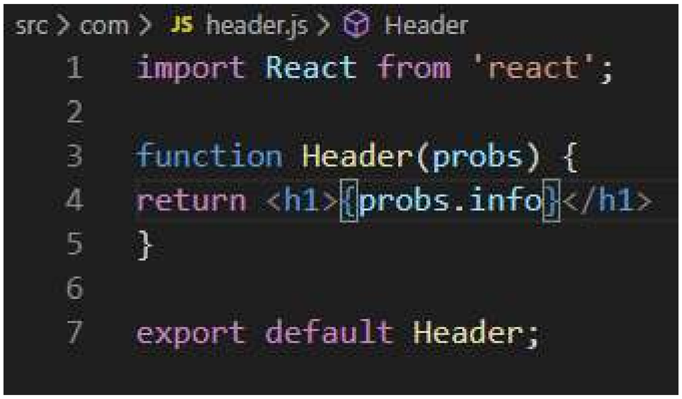


```javascript
import React from 'react';
import './App.css';
import H from './com/header';
import F from './com/footer';

function App() {
  return (
    <div className="App">
      <H />
      <p>main content</p>
      <F />
    </div>
  );
}

export default App;
```

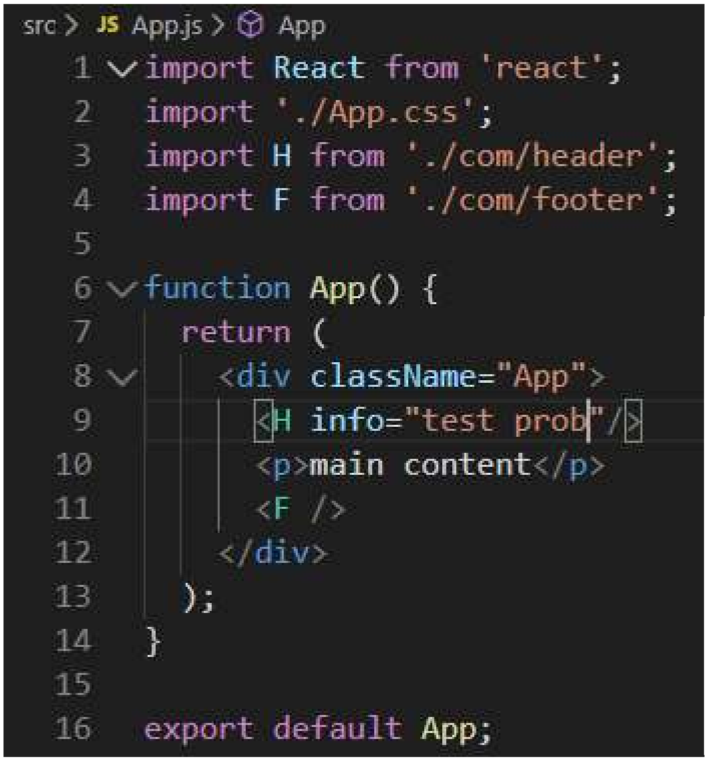


```
head!!

main content

Footer
```

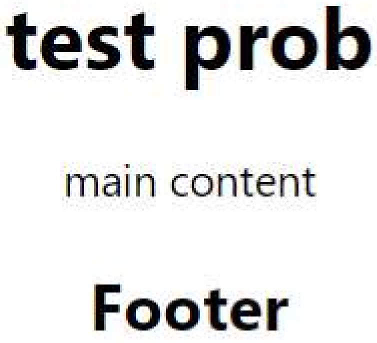
Probs: deliver message



```
import React from 'react';

function Header(probs) {
return <h1>{probs.info}</h1>
}

export default Header;
```

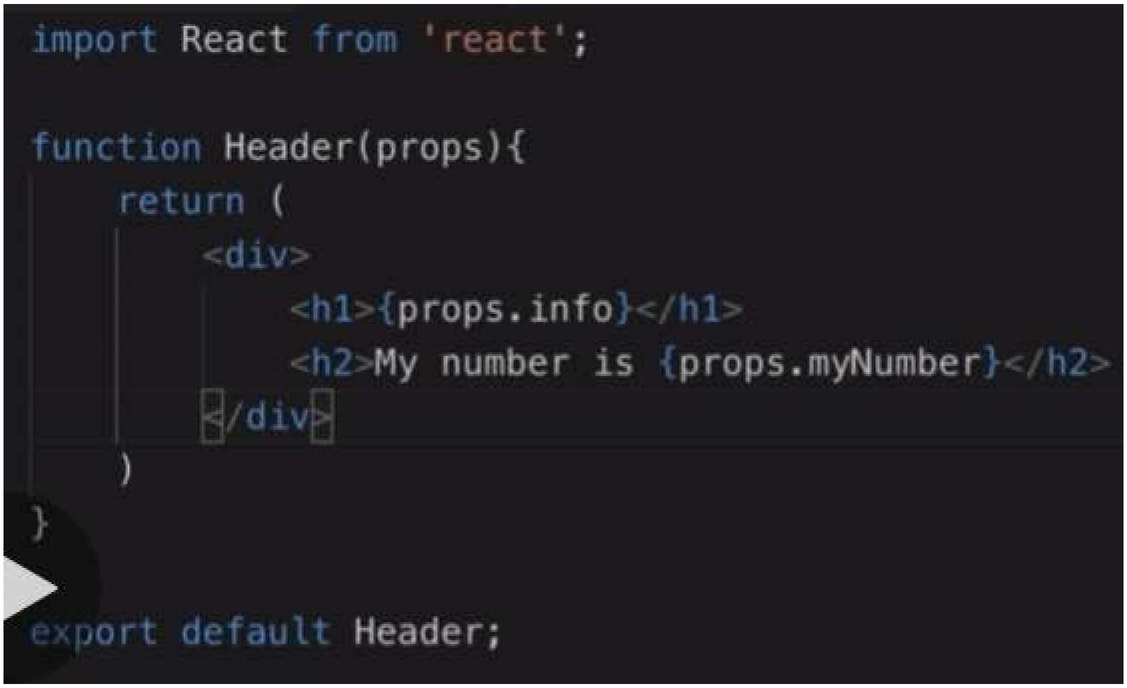


```
import React from 'react';
import './App.css';
import H from './com/header';
import F from './com/footer';

function App() {
  return (
    <div className="App">
      <H info="test prob"/>
      <p>main content</p>
      <F />
    </div>
  );
}

export default App;
```

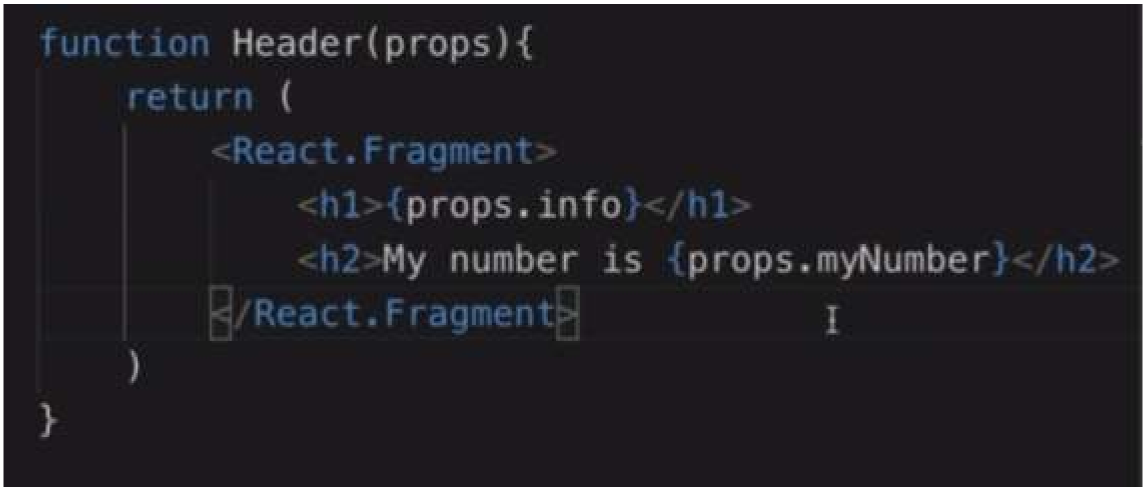


```
test prob

main content

Footer
```

If want to return more than one we should use <div> or <React.Fragment>

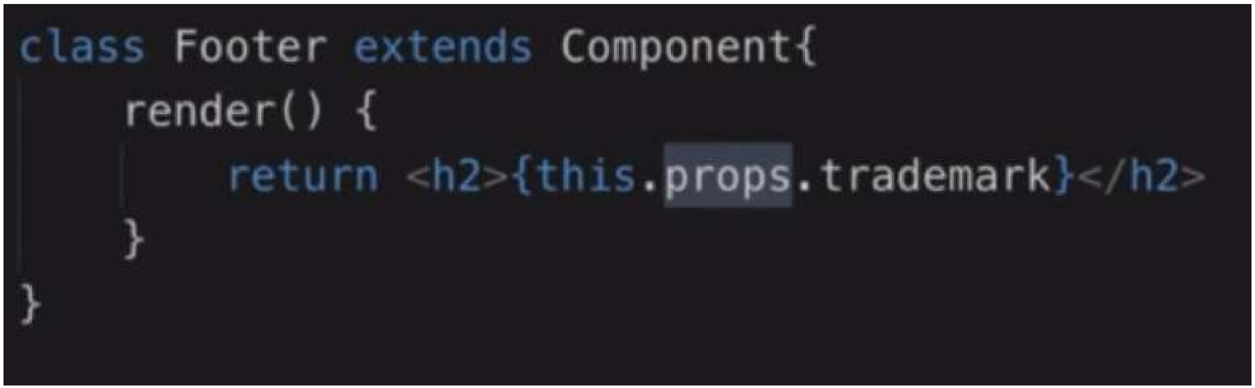


```
import React from 'react';

function Header(props){
    return (
        <div>
            <h1>{props.info}</h1>
            <h2>My number is {props.myNumber}</h2>
        </div>
    )
}

export default Header;
```

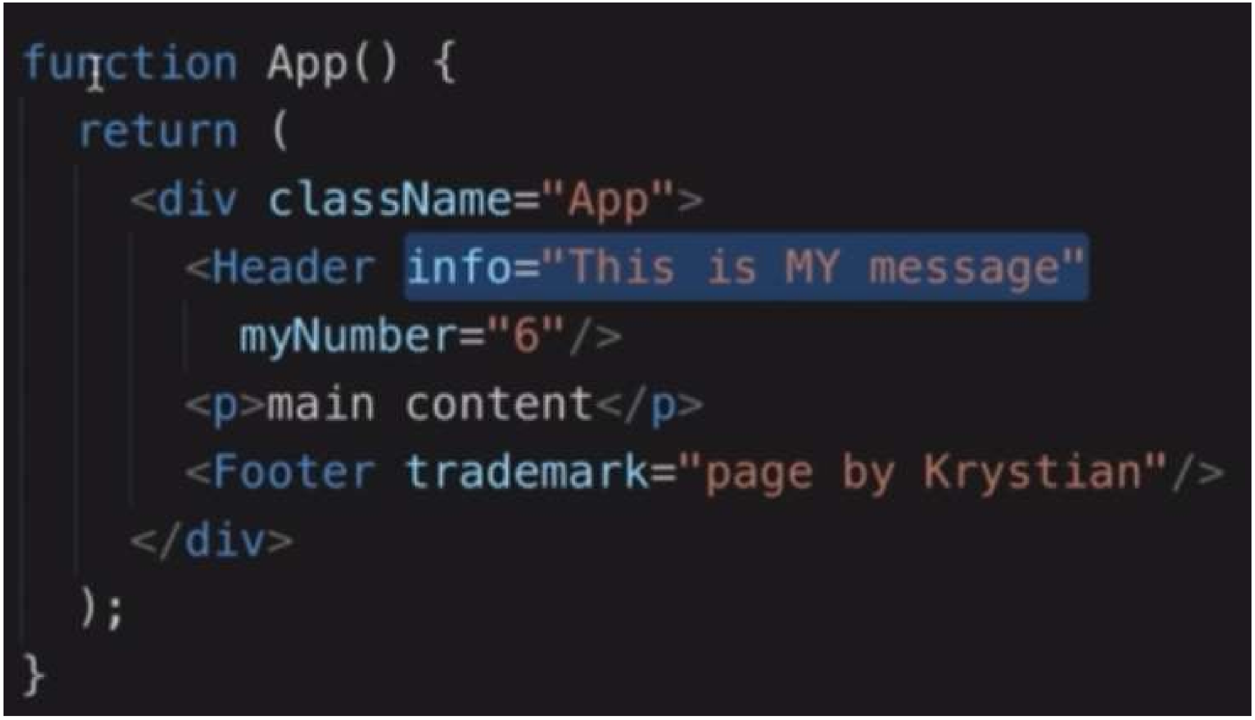


```
function Header(props){
    return (
        <React.Fragment>
            <h1>{props.info}</h1>
            <h2>My number is {props.myNumber}</h2>
        </React.Fragment>
    )
}
```

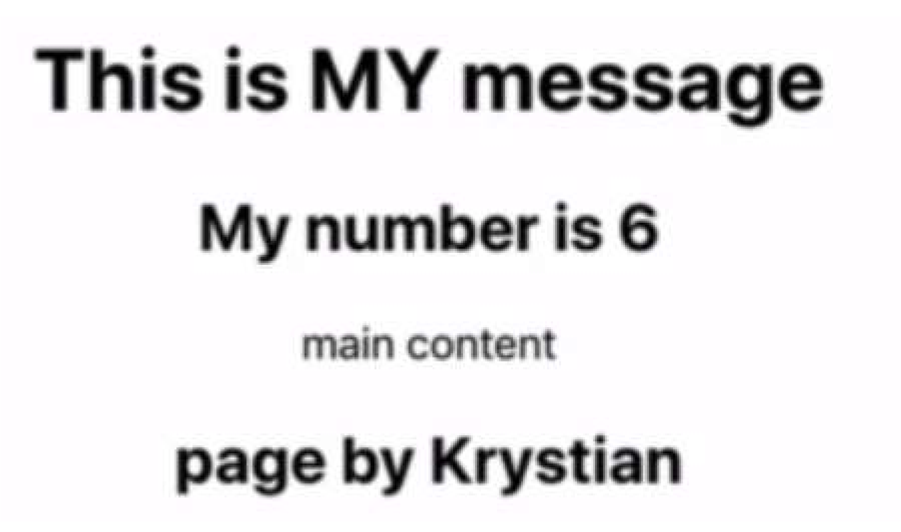
Use in class



```
class Footer extends Component{
    render() {
        return <h2>{this.props.trademark}</h2>
    }
}
```



```
function App() {
  return (
    <div className="App">
      <Header info="This is MY message"
        myNumber="6"/>
      <p>main content</p>
      <Footer trademark="page by Krystian"/>
    </div>
  );
}
```



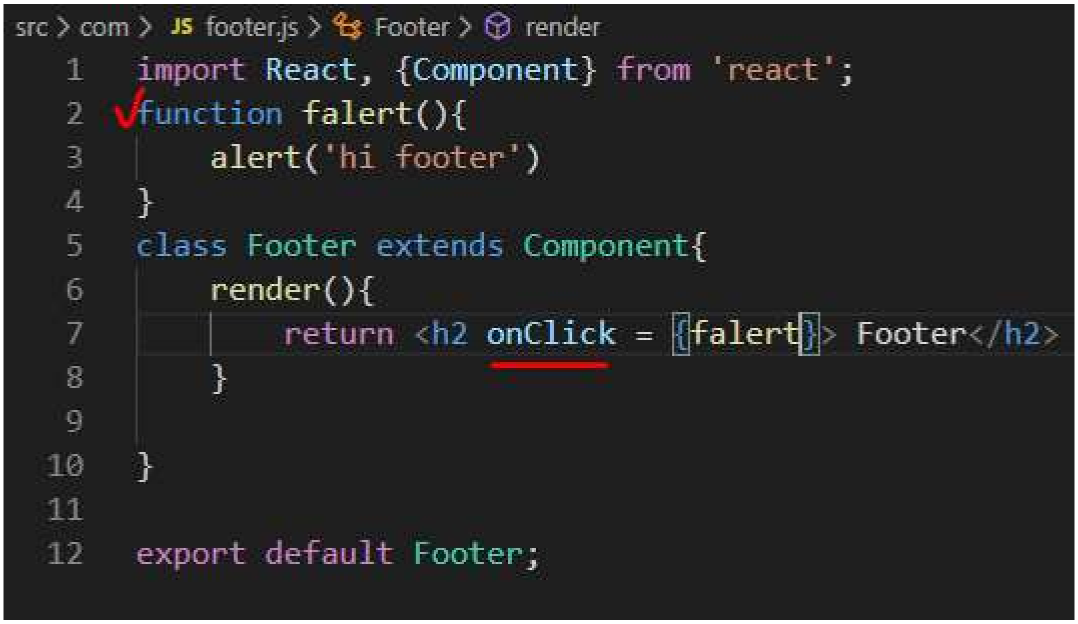
This is MY message

My number is 6

main content

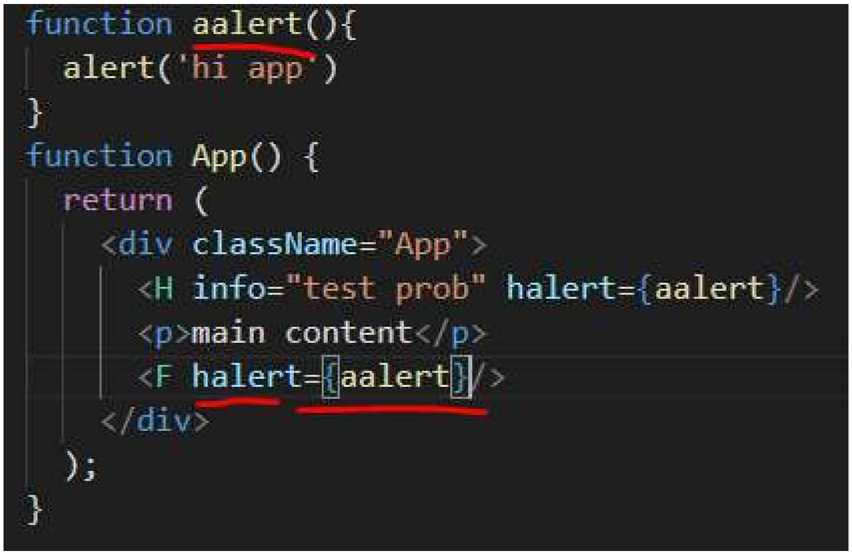
page by Krystian

Event

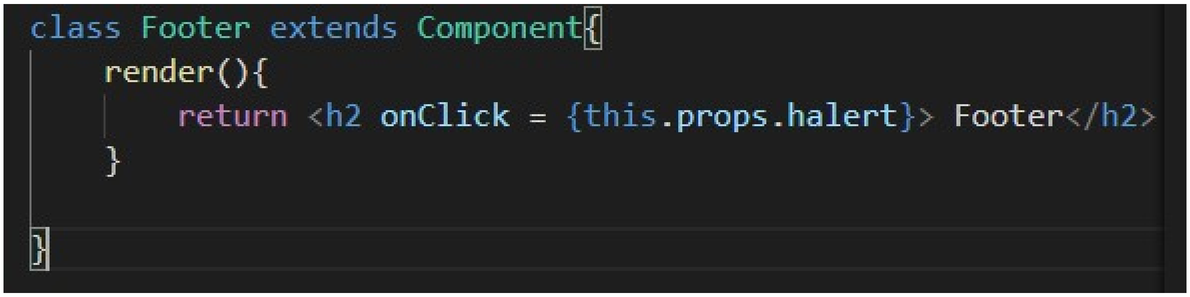


```
import React, {Component} from 'react';
function falert(){
    alert('hi footer')
}
class Footer extends Component{
    render(){
        return <h2 onClick = {falert}> Footer</h2>
    }

}

export default Footer;
```

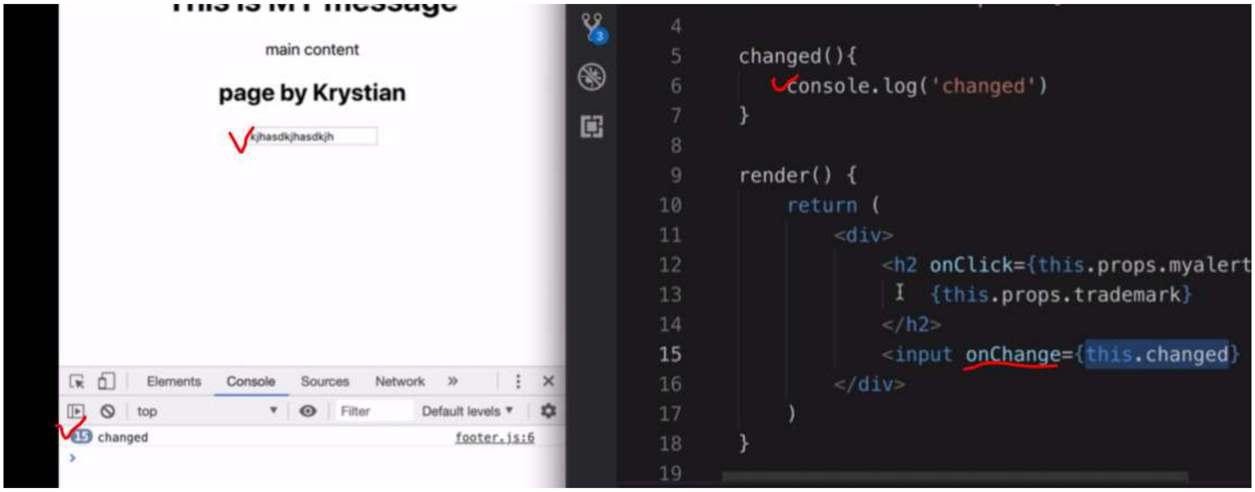
Function from parent s(only class can get the function from parent)



```
function aalert(){
  alert('hi app')
}
function App() {
  return (
    <div className="App">
      <H info="test prob" halert={aalert}/>
      <p>main content</p>
      <F halert={aalert}/>
    </div>
  );
}
```



```
class Footer extends Component{
    render(){
        return <h2 onClick = {this.props.halert}> Footer</h2>
    }

}
```

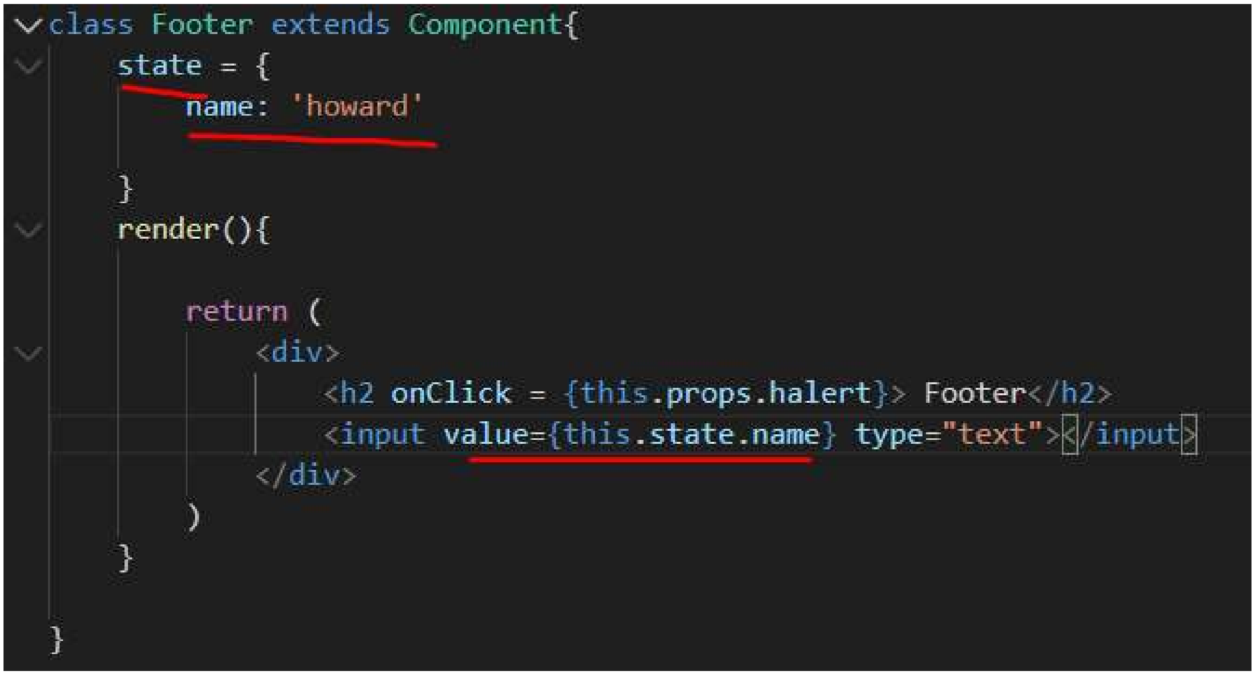


```
15 changed                                              footer.js:6
```

```
    changed(){
        console.log('changed')
    }

    render() {
        return (
            <div>
                <h2 onClick={this.props.myalert
                  {this.props.trademark}
                </h2>
                <input onChange={this.changed}
            </div>
        )
    }
```

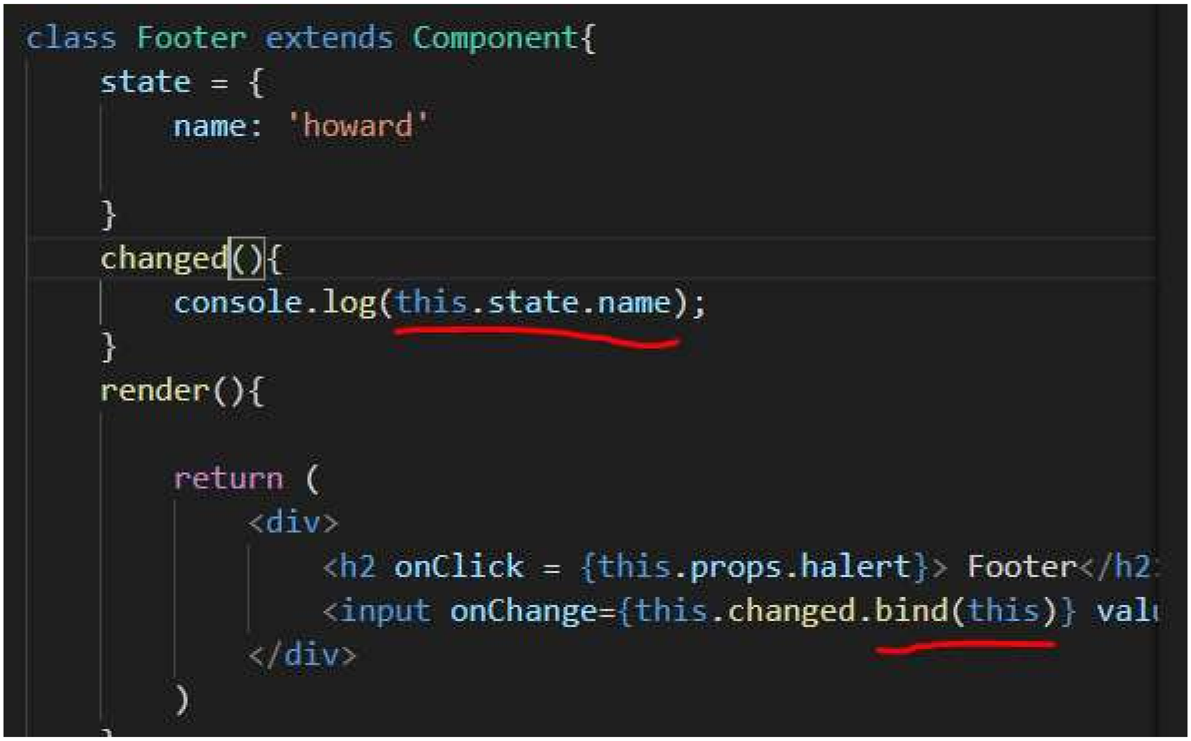
State: can be change

Props is immutable (can't be change)

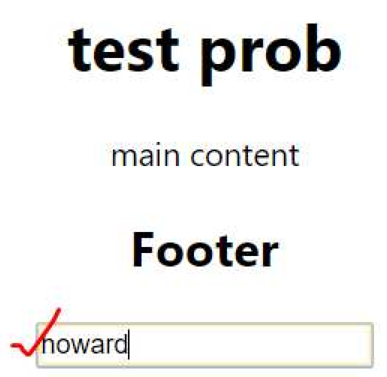


```
class Footer extends Component{
    state = {
        name: 'howard'

    }
    render(){

        return (
            <div>
                <h2 onClick = {this.props.halert}> Footer</h2>
                <input value={this.state.name} type="text"></input>
            </div>
        )
    }

}
```

Bind the class Footer **this** into the function



```
class Footer extends Component{
    state = {
        name: 'howard'

    }
    changed(){
        console.log(this.state.name);
    }
    render(){

        return (
            <div>
                <h2 onClick = {this.props.halert}> Footer</h2
                <input onChange={this.changed.bind(this)} valu
            </div>
        )
    }
```

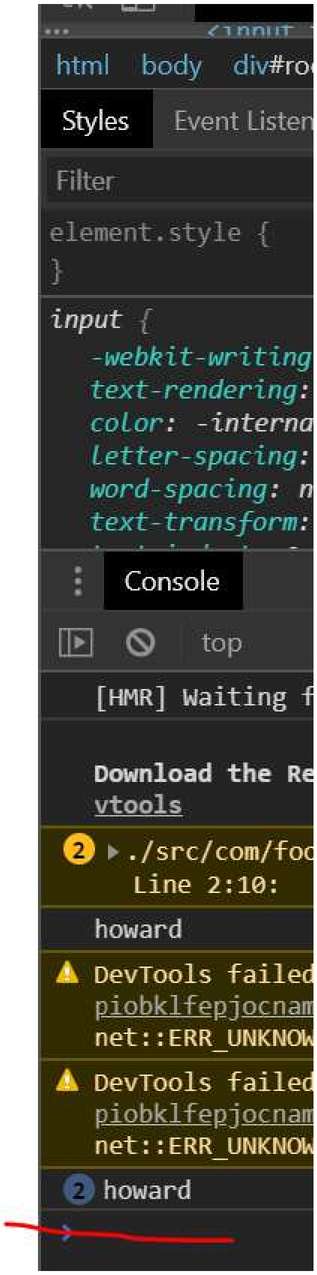


test prob

main content

Footer

howard

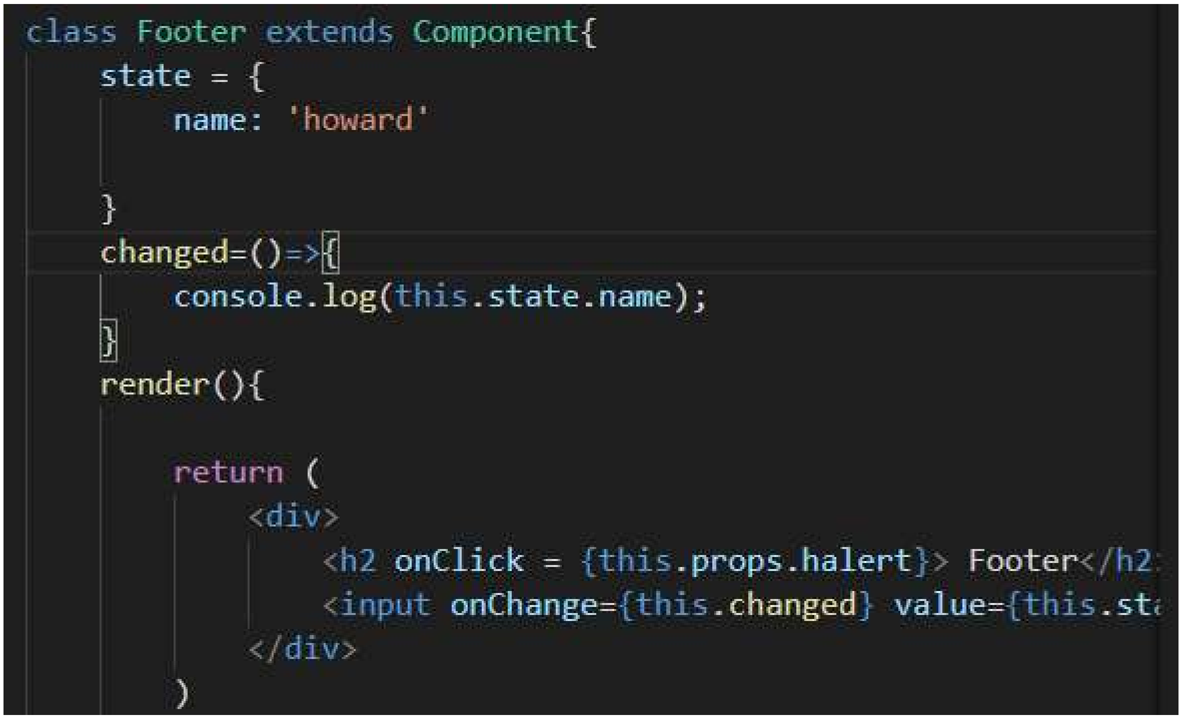


```
element.style {
}
input {
   -webkit-writing
   text-rendering:
   color: -interna
   letter-spacing:
   word-spacing: n
   text-transform:
```

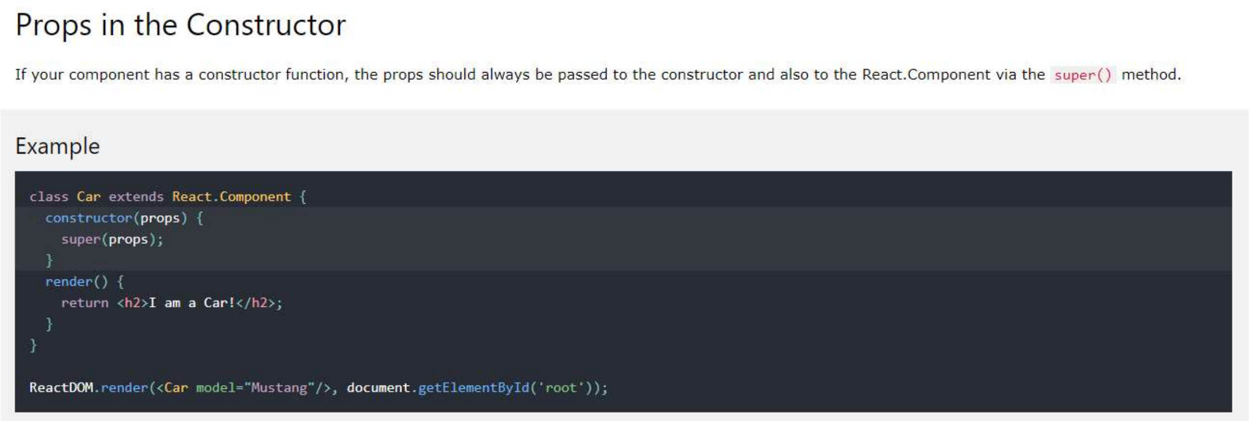
```
[HMR] Waiting f

Download the Re
vtools
2 ▸ ./src/com/foo
   Line 2:10:
howard
DevTools failed
piobklfepjocnam
net::ERR_UNKNOW
DevTools failed
piobklfepjocnam
net::ERR_UNKNOW
2 howard
>
```

Another way to keep **this** (won't change to function **this**

Use arrow function



```
class Footer extends Component{
    state = {
        name: 'howard'

    }
    changed=()=>{
        console.log(this.state.name);
    }
    render(){

        return (
            <div>
                <h2 onClick = {this.props.halert}> Footer</h2
                <input onChange={this.changed} value={this.st
            </div>
        )
```



Props in the Constructor

If your component has a constructor function, the props should always be passed to the constructor and also to the React.Component via the `super()` method.

Example

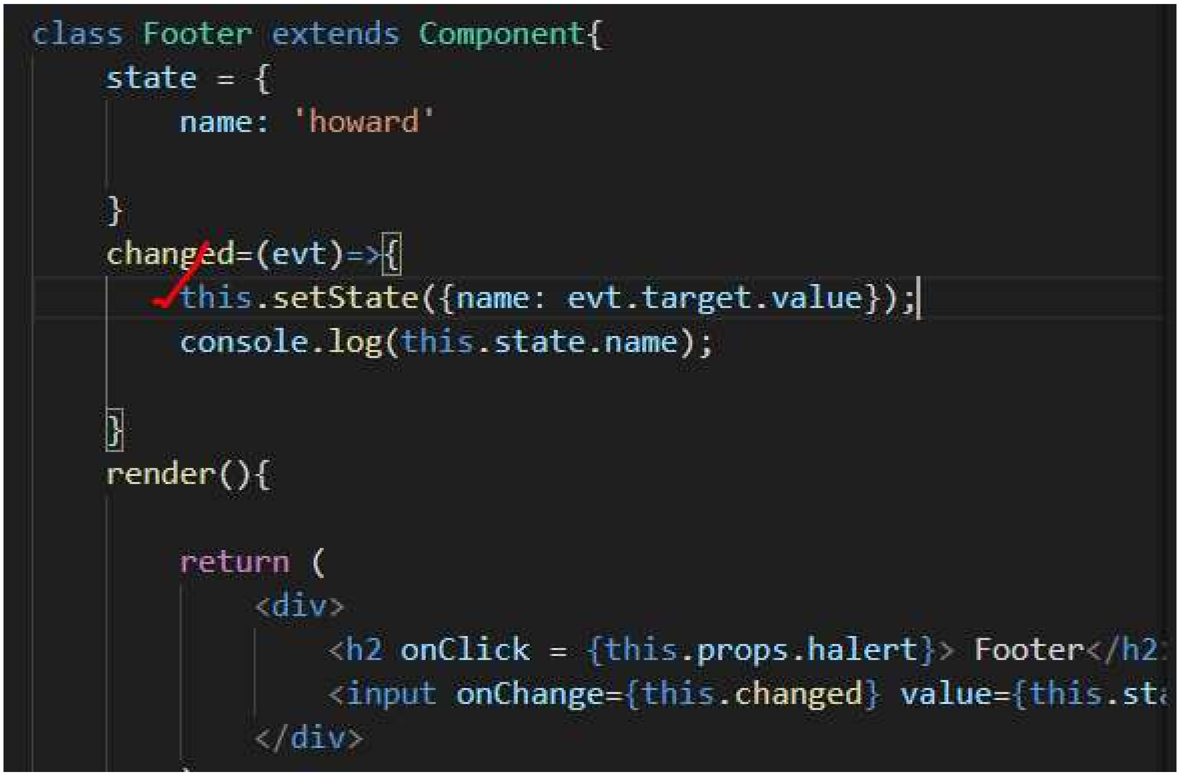
```
class Car extends React.Component {
  constructor(props) {
    super(props);
  }
  render() {
    return <h2>I am a Car!</h2>;
  }
}

ReactDOM.render(<Car model="Mustang"/>, document.getElementById('root'));
```

React components has a built-in `state` object.

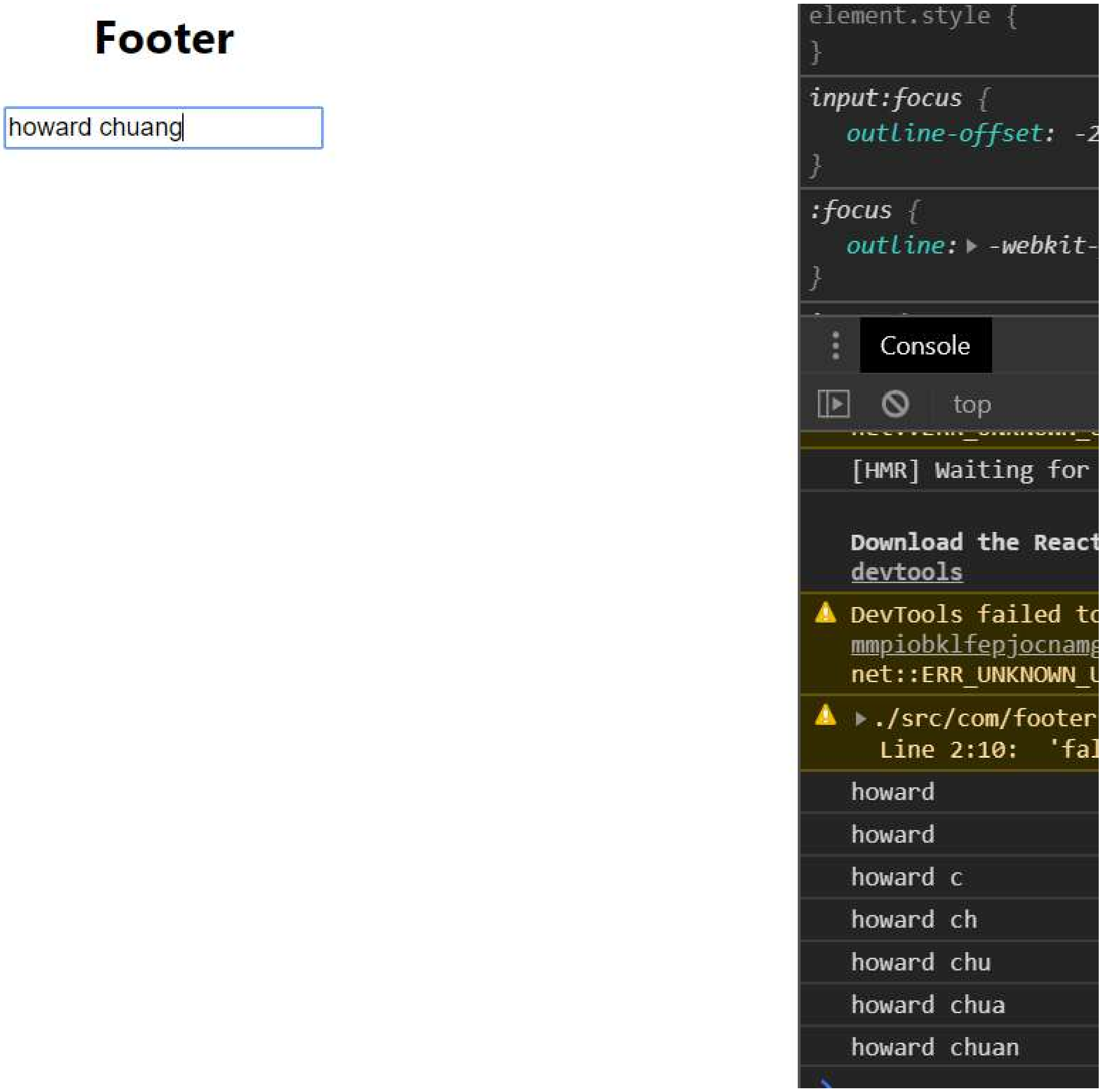
The `state` object is where you store property values that belongs to the component.

When the `state` object changes, the component re-renders.

How to set state



```
class Footer extends Component{
    state = {
        name: 'howard'

    }
    changed=(evt)=>{
        this.setState({name: evt.target.value});
        console.log(this.state.name);

    }
    render(){

        return (
            <div>
                <h2 onClick = {this.props.halert}> Footer</h2
                <input onChange={this.changed} value={this.sta
            </div>
```



Footer

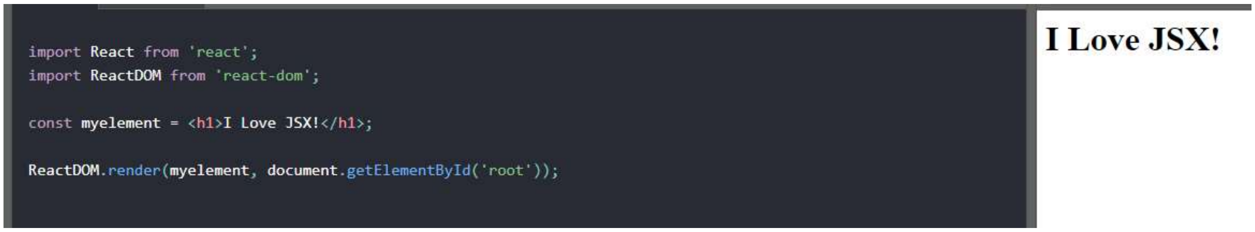
howard chuang

```
element.style {
}
input:focus {
  outline-offset: -2
}
:focus {
  outline: ▸ -webkit-
}
```

```
[HMR] Waiting for

Download the React
devtools
⚠ DevTools failed to
mmpiobklfepjocnamg
net::ERR_UNKNOWN_U
⚠ ▸ ./src/com/footer
    Line 2:10:  'fal
howard
howard
howard c
howard ch
howard chu
howard chua
howard chuan
```

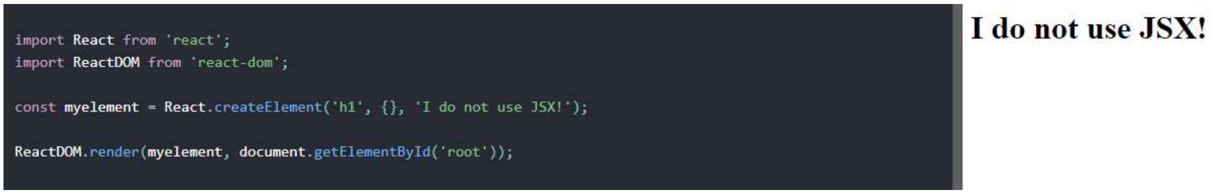
JSX allows us to write HTML elements in JavaScript and place them in the DOM without any `createElement()` and/or `appendChild()` methods.

JSX converts HTML tags into react elements.



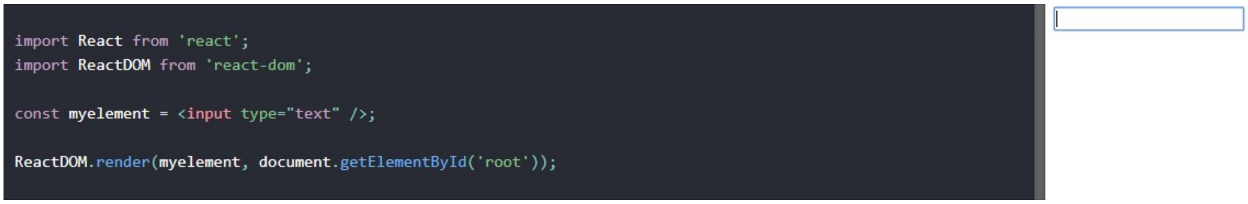
```
import React from 'react';
import ReactDOM from 'react-dom';

const myelement = <h1>I Love JSX!</h1>;

ReactDOM.render(myelement, document.getElementById('root'));
```

I Love JSX!



```
import React from 'react';
import ReactDOM from 'react-dom';

const myelement = React.createElement('h1', {}, 'I do not use JSX!');

ReactDOM.render(myelement, document.getElementById('root'));
```

I do not use JSX!



```
import React from 'react';
import ReactDOM from 'react-dom';

const myelement = <input type="text" />;

ReactDOM.render(myelement, document.getElementById('root'));
```

Lifecycle of Components

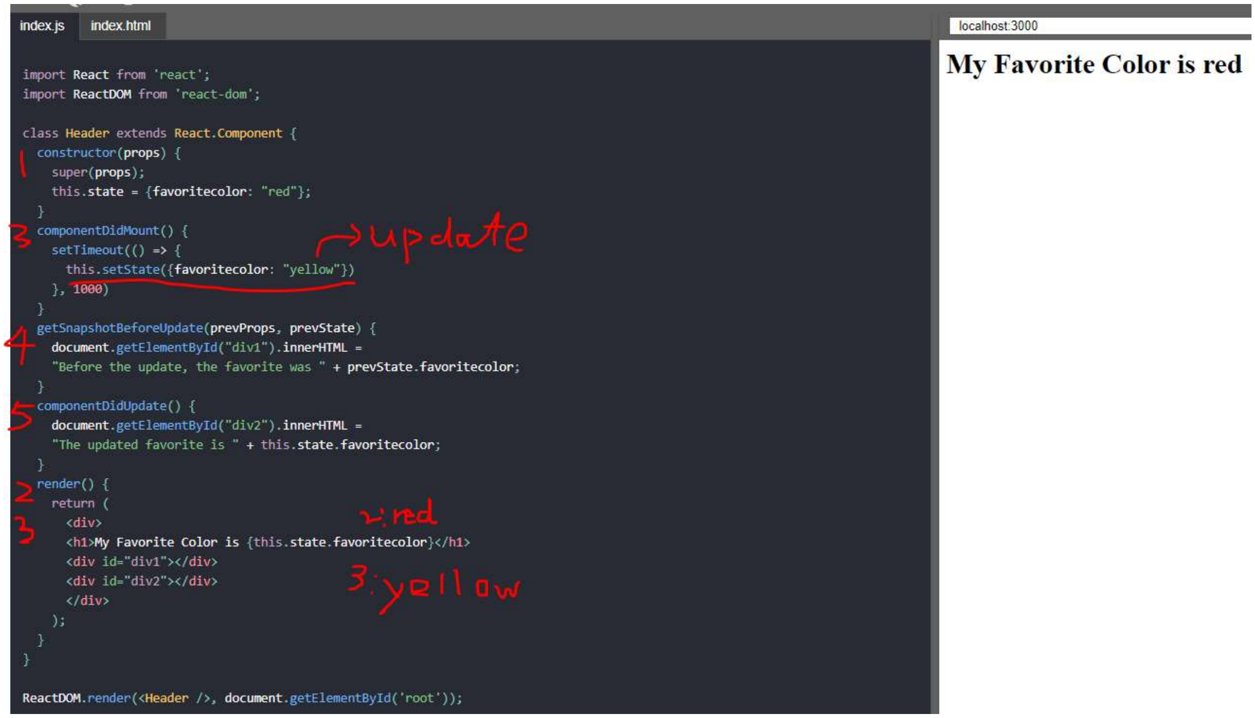
Mounting

1. `constructor()`
2. `getDerivedStateFromProps()`
3. `render()`
4. `componentDidMount()`

Updating

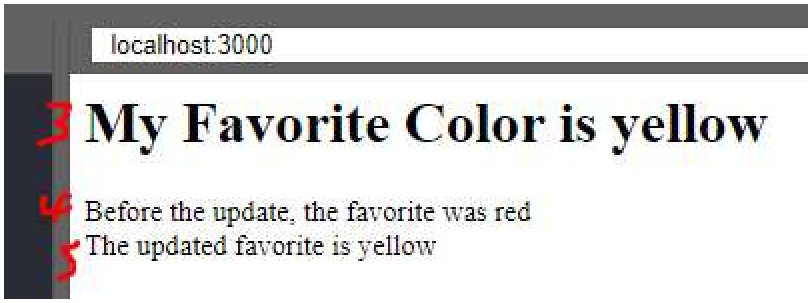
A component is updated whenever there is a change in the component's `state` or `props`.

1. `getDerivedStateFromProps()`
2. `shouldComponentUpdate()`
3. `render()`
4. `getSnapshotBeforeUpdate()`
5. `componentDidUpdate()`



```
import React from 'react';
import ReactDOM from 'react-dom';

class Header extends React.Component {
  constructor(props) {
    super(props);
    this.state = {favoritecolor: "red"};
  }
  componentDidMount() {
    setTimeout(() => {
      this.setState({favoritecolor: "yellow"})
    }, 1000)
  }
  getSnapshotBeforeUpdate(prevProps, prevState) {
    document.getElementById("div1").innerHTML =
    "Before the update, the favorite was " + prevState.favoritecolor;
  }
  componentDidUpdate() {
    document.getElementById("div2").innerHTML =
    "The updated favorite is " + this.state.favoritecolor;
  }
  render() {
    return (
      <div>
      <h1>My Favorite Color is {this.state.favoritecolor}</h1>
      <div id="div1"></div>
      <div id="div2"></div>
      </div>
    );
  }
}

ReactDOM.render(<Header />, document.getElementById('root'));
```

My Favorite Color is red

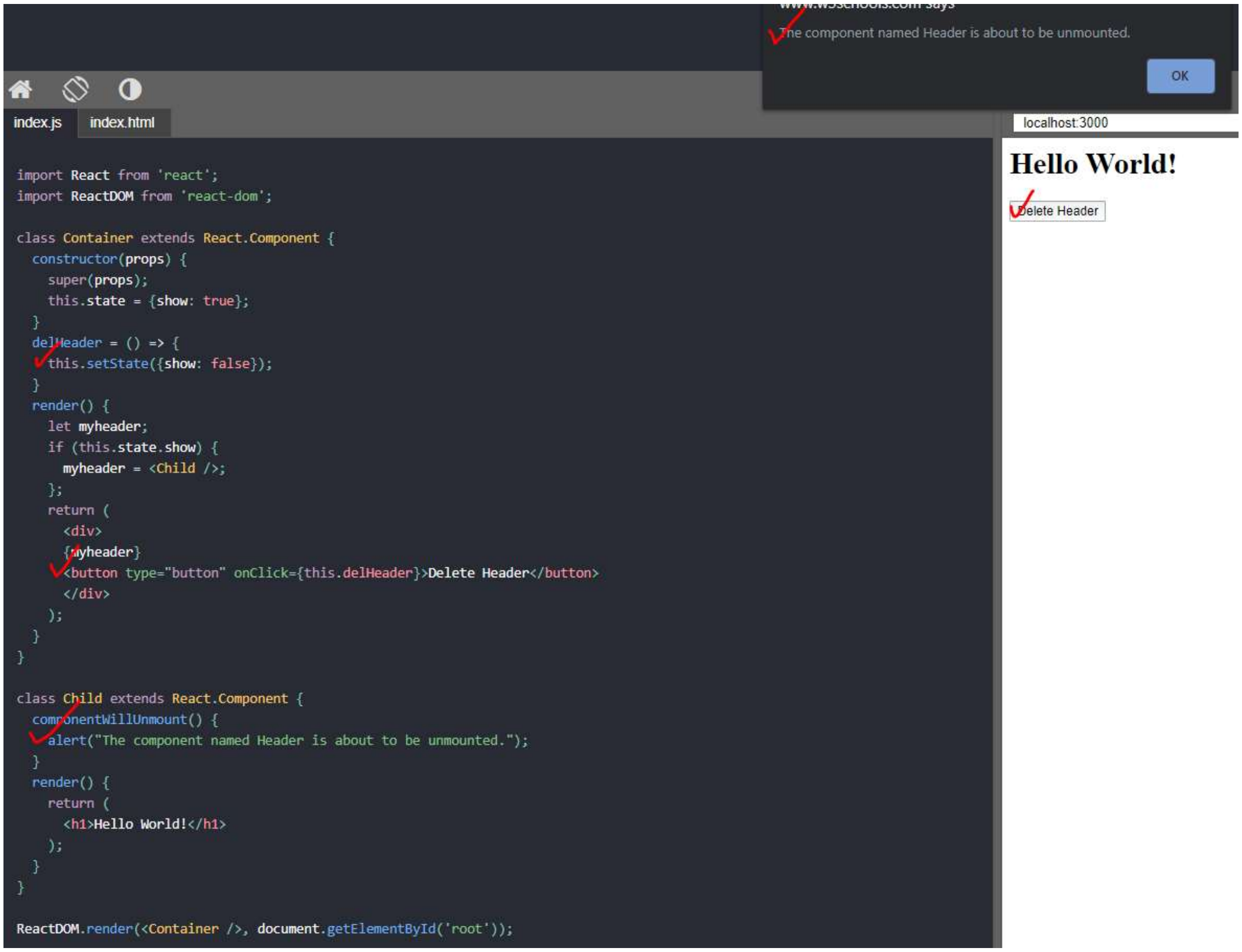


My Favorite Color is yellow

Before the update, the favorite was red
The updated favorite is yellow

Unmounting

`componentWillUnmount()`

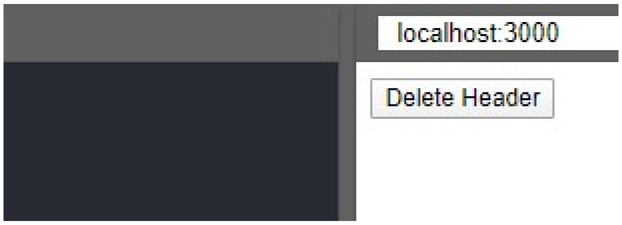


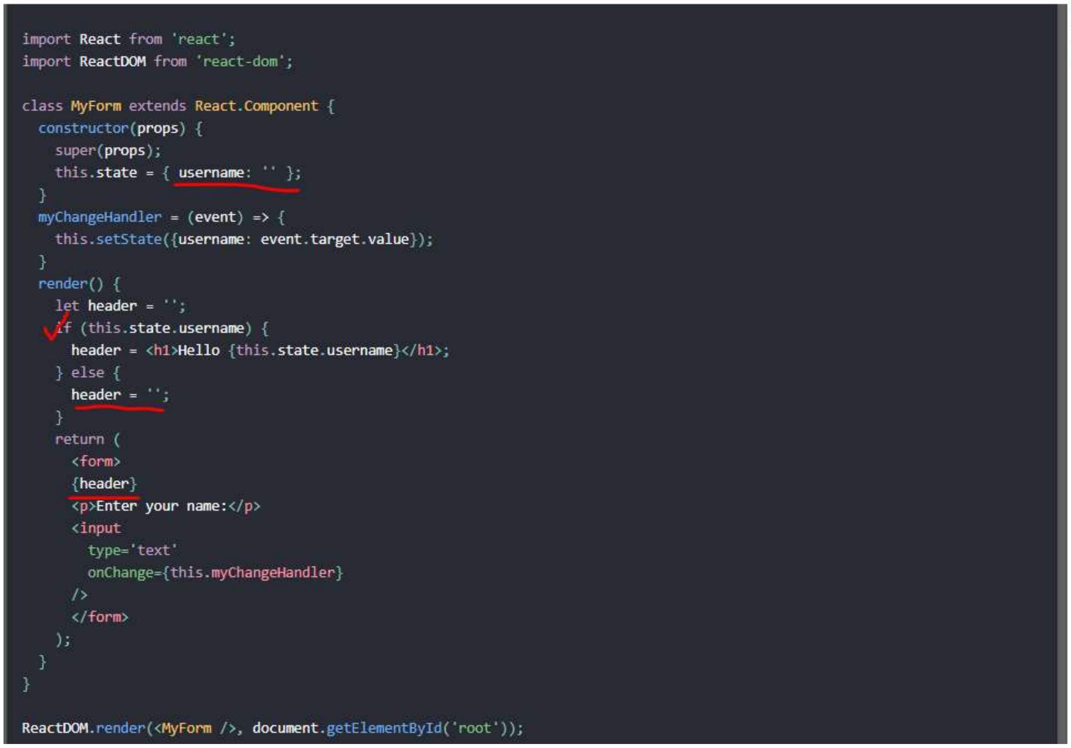
www.w3schools.com says

The component named Header is about to be unmounted.

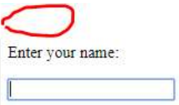
```
import React from 'react';
import ReactDOM from 'react-dom';

class Container extends React.Component {
  constructor(props) {
    super(props);
    this.state = {show: true};
  }
  delHeader = () => {
    this.setState({show: false});
  }
  render() {
    let myheader;
    if (this.state.show) {
      myheader = <Child />;
    };
    return (
      <div>
      {myheader}
      <button type="button" onClick={this.delHeader}>Delete Header</button>
      </div>
    );
  }
}

class Child extends React.Component {
  componentWillUnmount() {
    alert("The component named Header is about to be unmounted.");
  }
  render() {
    return (
      <h1>Hello World!</h1>
    );
  }
}

ReactDOM.render(<Container />, document.getElementById('root'));
```

Hello World!

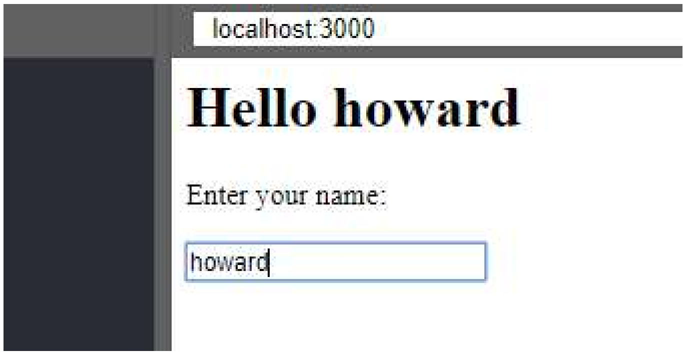




```
import React from 'react';
import ReactDOM from 'react-dom';

class MyForm extends React.Component {
  constructor(props) {
    super(props);
    this.state = { username: '' };
  }
  myChangeHandler = (event) => {
    this.setState({username: event.target.value});
  }
  render() {
    let header = '';
    if (this.state.username) {
      header = <h1>Hello {this.state.username}</h1>;
    } else {
      header = '';
    }
    return (
      <form>
      {header}
      <p>Enter your name:</p>
      <input
        type='text'
        onChange={this.myChangeHandler}
      />
      </form>
    );
  }
}

ReactDOM.render(<MyForm />, document.getElementById('root'));
```



Enter your name:

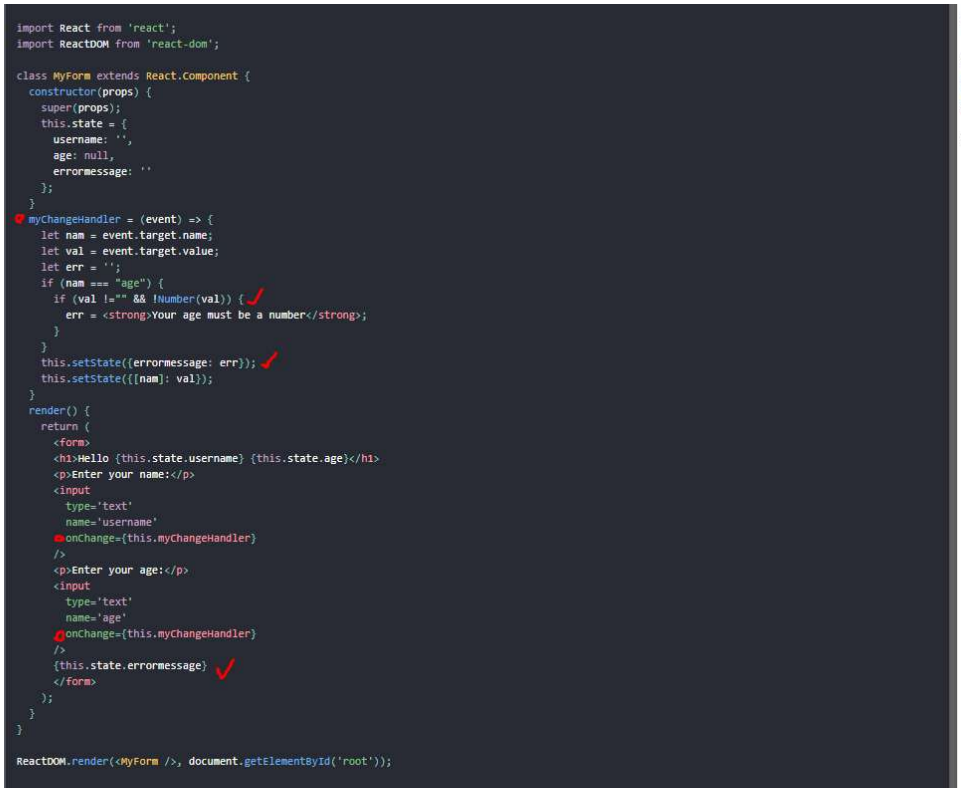


localhost:3000

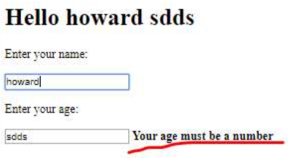
Hello howard

Enter your name:

howard



```
import React from 'react';
import ReactDOM from 'react-dom';

class MyForm extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      username: '',
      age: null,
      errormessage: ''
    };
  }
  myChangeHandler = (event) => {
    let nam = event.target.name;
    let val = event.target.value;
    let err = '';
    if (nam === "age") {
      if (val !="" && !Number(val)) {
        err = <strong>Your age must be a number</strong>;
      }
    }
    this.setState({errormessage: err});
    this.setState({[nam]: val});
  }
  render() {
    return (
      <form>
      <h1>Hello {this.state.username} {this.state.age}</h1>
      <p>Enter your name:</p>
      <input
        type='text'
        name='username'
        onChange={this.myChangeHandler}
      />
      <p>Enter your age:</p>
      <input
        type='text'
        name='age'
        onChange={this.myChangeHandler}
      />
      {this.state.errormessage}
      </form>
    );
  }
}

ReactDOM.render(<MyForm />, document.getElementById('root'));
```



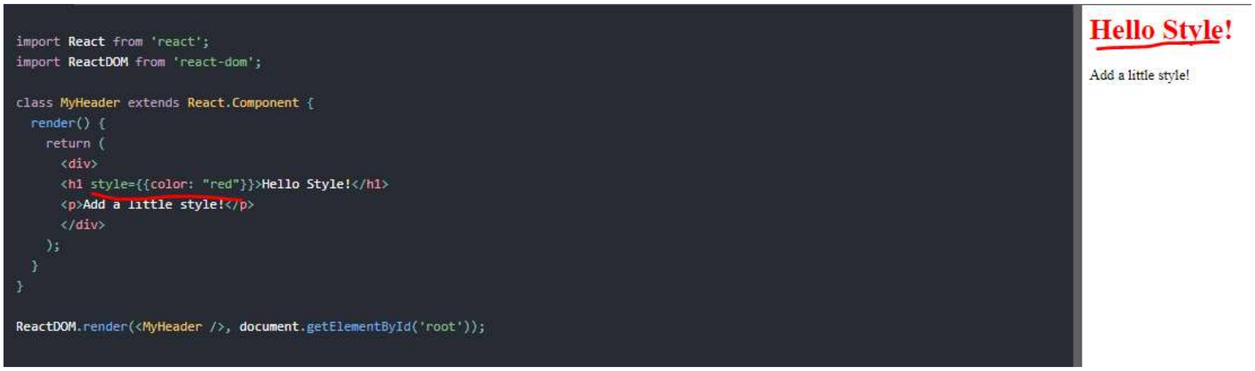
Hello howard sdds

Enter your name:

howard

Enter your age:

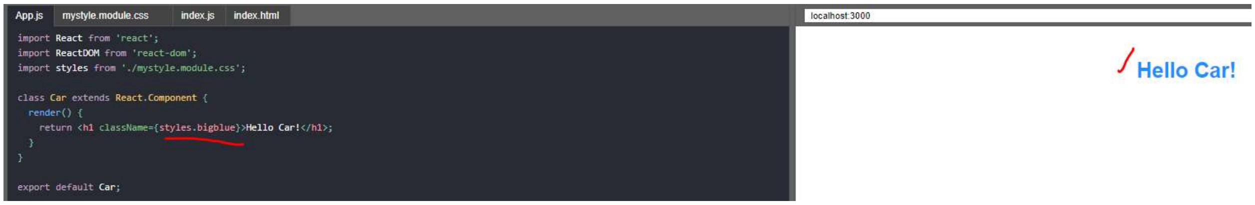
sdds **Your age must be a number**



```
import React from 'react';
import ReactDOM from 'react-dom';

class MyHeader extends React.Component {
  render() {
    return (
      <div>
      <h1 style={{color: "red"}}>Hello Style!</h1>
      <p>Add a little style!</p>
      </div>
    );
  }
}

ReactDOM.render(<MyHeader />, document.getElementById('root'));
```

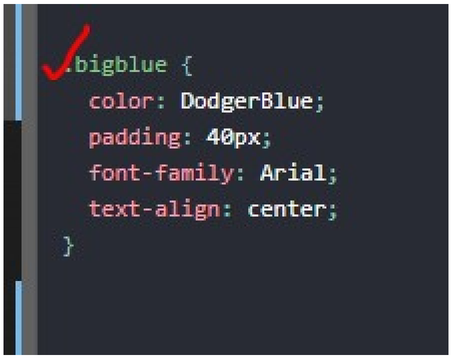
Hello Style!

Add a little style!

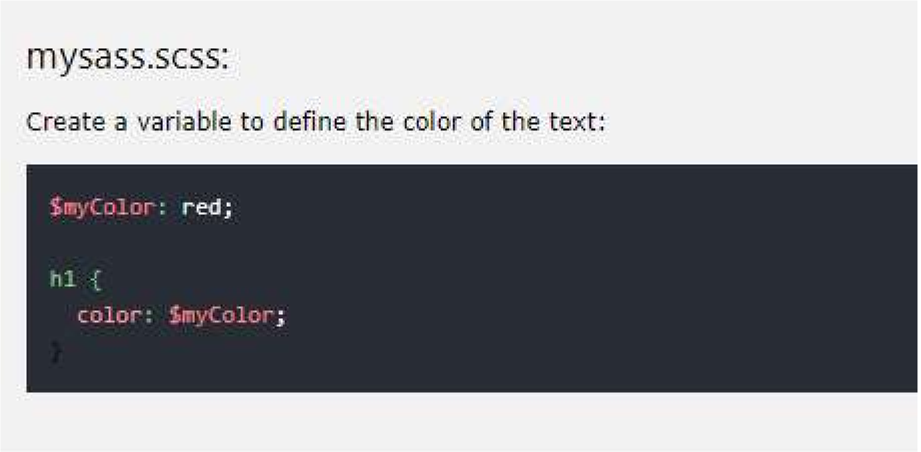


```
import React from 'react';
import ReactDOM from 'react-dom';
import styles from './mystyle.module.css';

class Car extends React.Component {
  render() {
    return <h1 className={styles.bigblue}>Hello Car!</h1>;
  }
}

export default Car;
```

Hello Car!



```
.bigblue {
  color: DodgerBlue;
  padding: 40px;
  font-family: Arial;
  text-align: center;
}
```

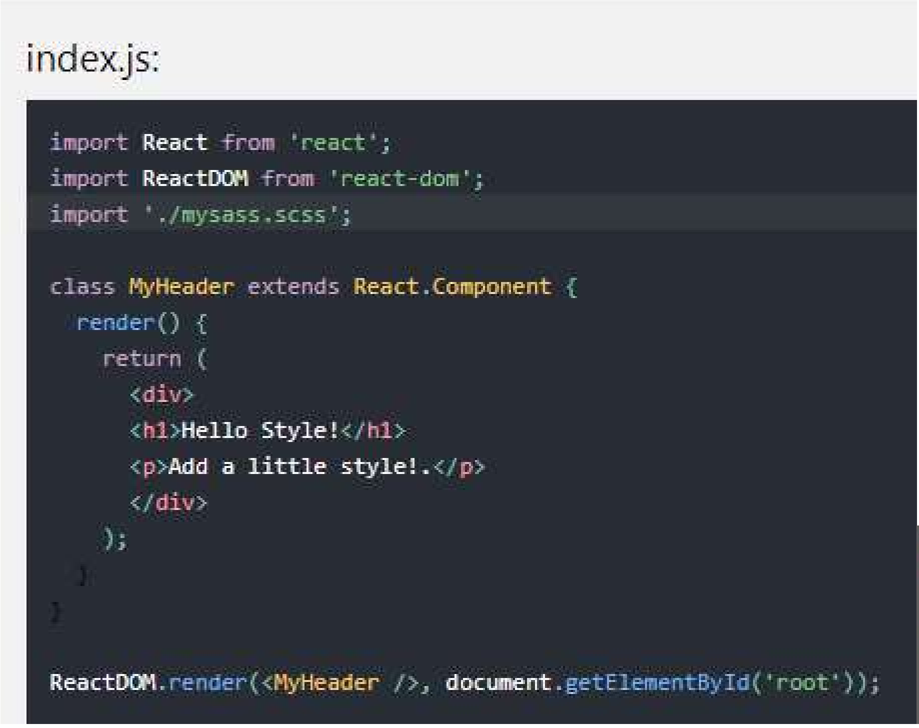


mysass.scss:

Create a variable to define the color of the text:

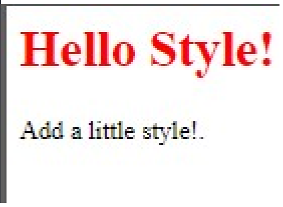
```
$myColor: red;

h1 {
  color: $myColor;
}
```

Import the Sass file the same way as you imported a CSS file:

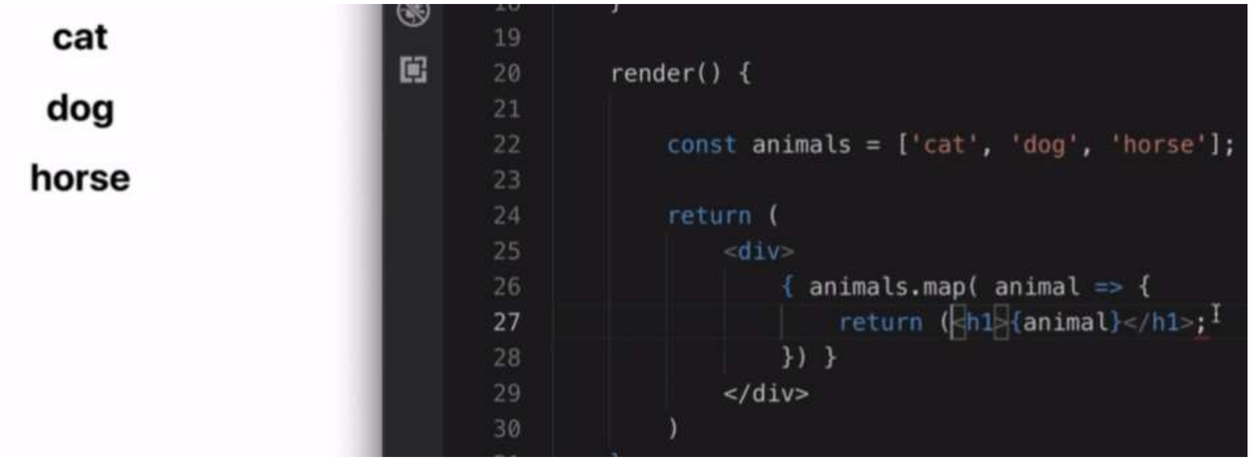


index.js:

```
import React from 'react';
import ReactDOM from 'react-dom';
import './mysass.scss';

class MyHeader extends React.Component {
  render() {
    return (
      <div>
      <h1>Hello Style!</h1>
      <p>Add a little style!.</p>
      </div>
    );
  }
}

ReactDOM.render(<MyHeader />, document.getElementById('root'));
```



```
Hello Style!

Add a little style!.
```



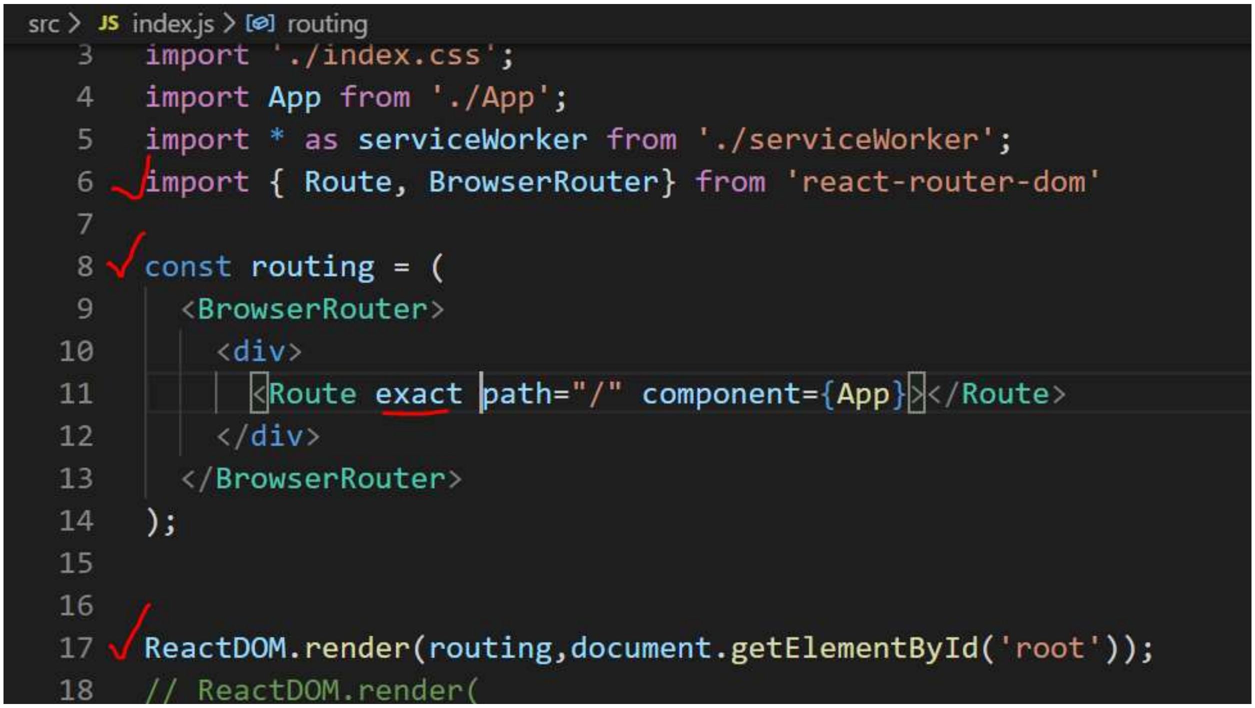
cat

dog

horse

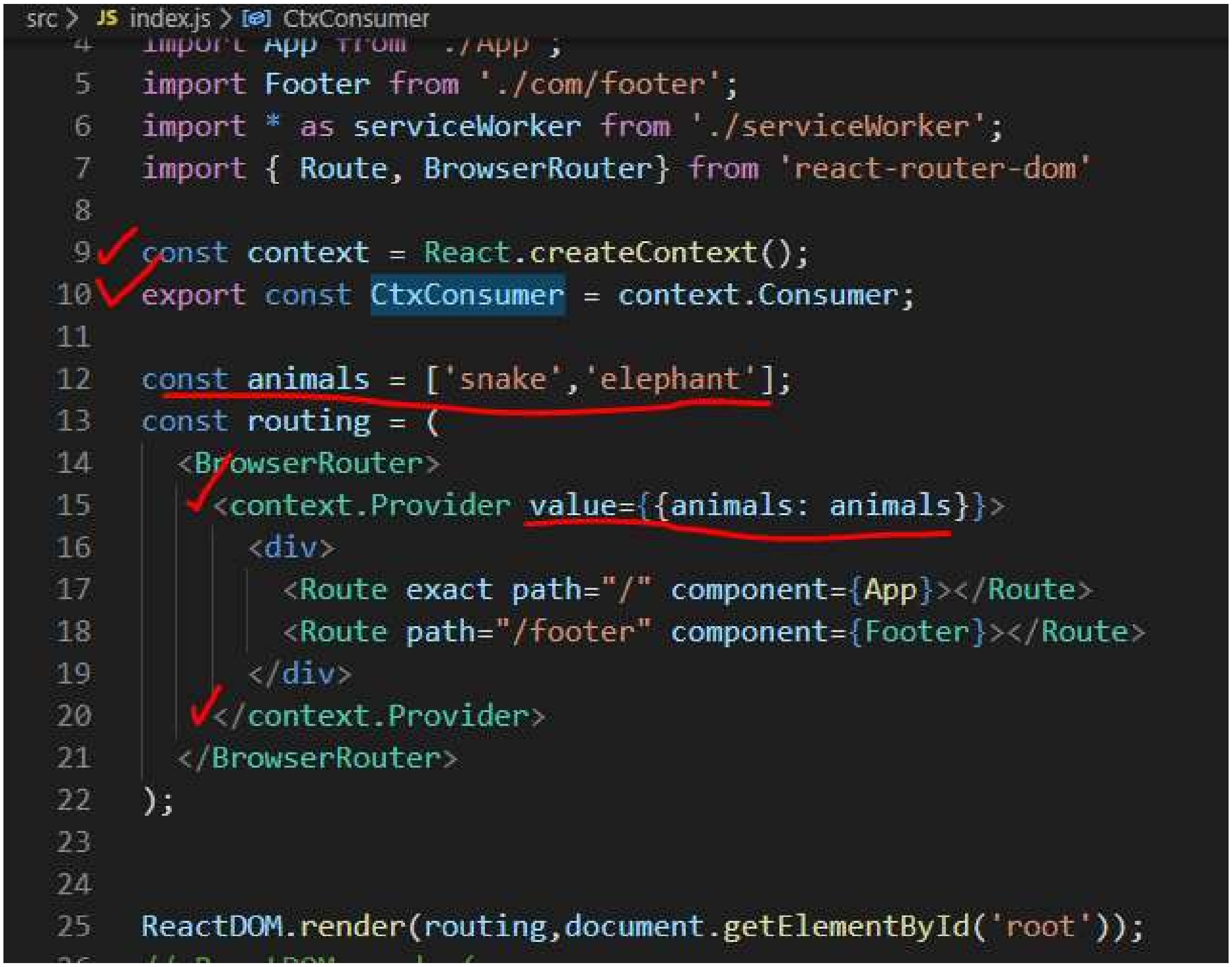
```
    render() {

        const animals = ['cat', 'dog', 'horse'];

        return (
            <div>
                { animals.map( animal => {
                    return (<h1>{animal}</h1>;
                }) }
            </div>
        )
```

Router



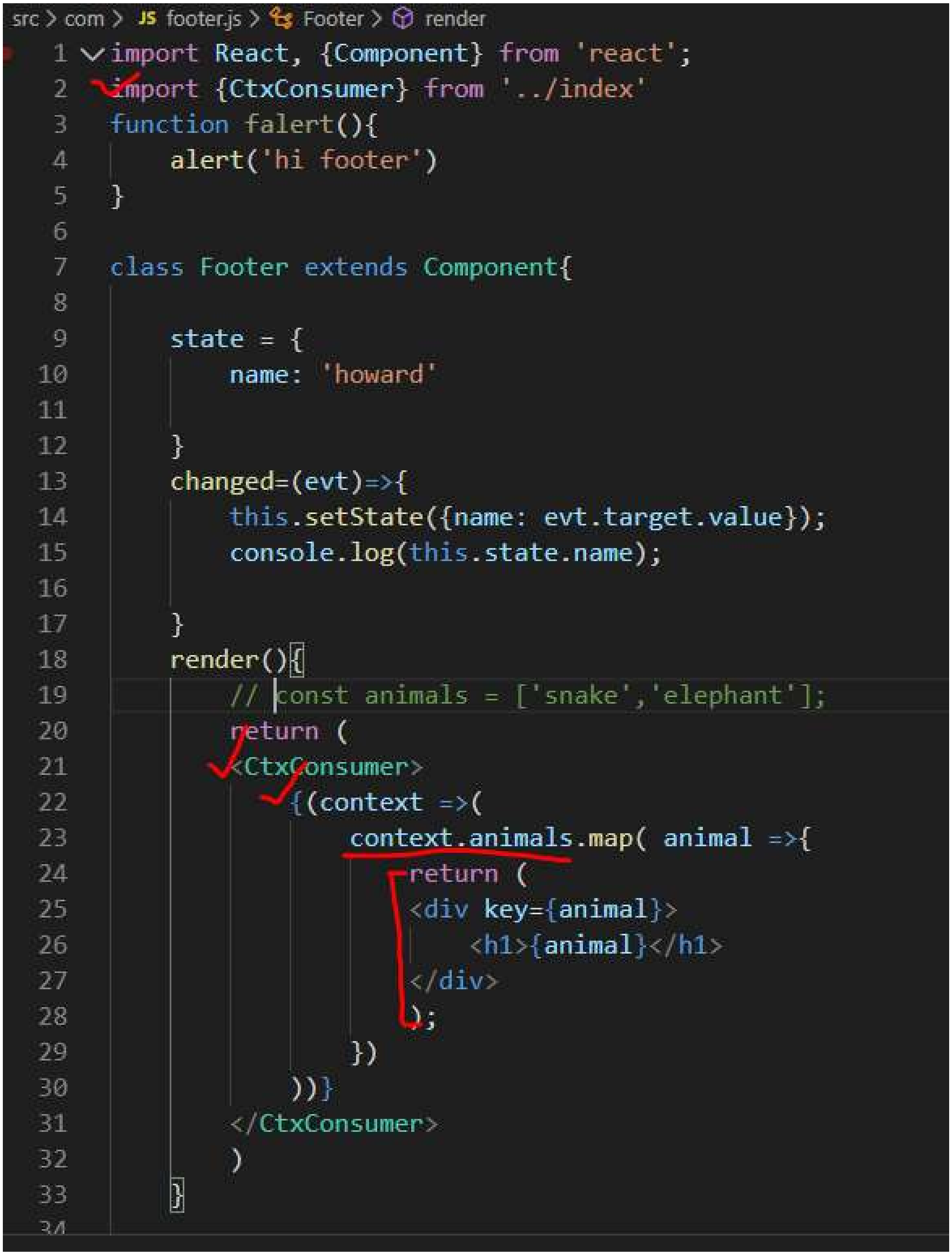
```
import './index.css';
import App from './App';
import * as serviceWorker from './serviceWorker';
import { Route, BrowserRouter} from 'react-router-dom'

const routing = (
  <BrowserRouter>
    <div>
      <Route exact path="/" component={App}></Route>
    </div>
  </BrowserRouter>
);


ReactDOM.render(routing,document.getElementById('root'));
// ReactDOM.render(
```

Context(this is global data=>try not to use)

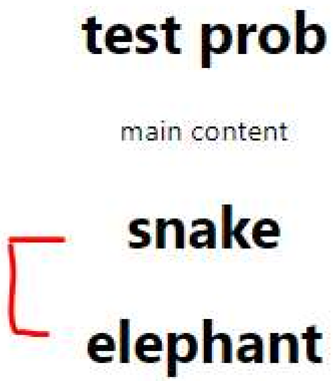


src > JS index.js > CtxConsumer

```
import App from './App';
import Footer from './com/footer';
import * as serviceWorker from './serviceWorker';
import { Route, BrowserRouter} from 'react-router-dom'

const context = React.createContext();
export const CtxConsumer = context.Consumer;

const animals = ['snake','elephant'];
const routing = (
  <BrowserRouter>
    <context.Provider value={{animals: animals}}>
      <div>
        <Route exact path="/" component={App}></Route>
        <Route path="/footer" component={Footer}></Route>
      </div>
    </context.Provider>
  </BrowserRouter>
);


ReactDOM.render(routing,document.getElementById('root'));
// ReactDOM.render(
```



```
import React, {Component} from 'react';
import {CtxConsumer} from '../index'
function falert(){
    alert('hi footer')
}

class Footer extends Component{

    state = {
        name: 'howard'

    }
    changed=(evt)=>{
        this.setState({name: evt.target.value});
        console.log(this.state.name);

    }
    render(){
        // const animals = ['snake','elephant'];
        return (
        <CtxConsumer>
            {(context =>(
                context.animals.map( animal =>{
                    return (
                    <div key={animal}>
                        <h1>{animal}</h1>
                    </div>
                    );
                })
            ))}
        </CtxConsumer>
        )
    }
```



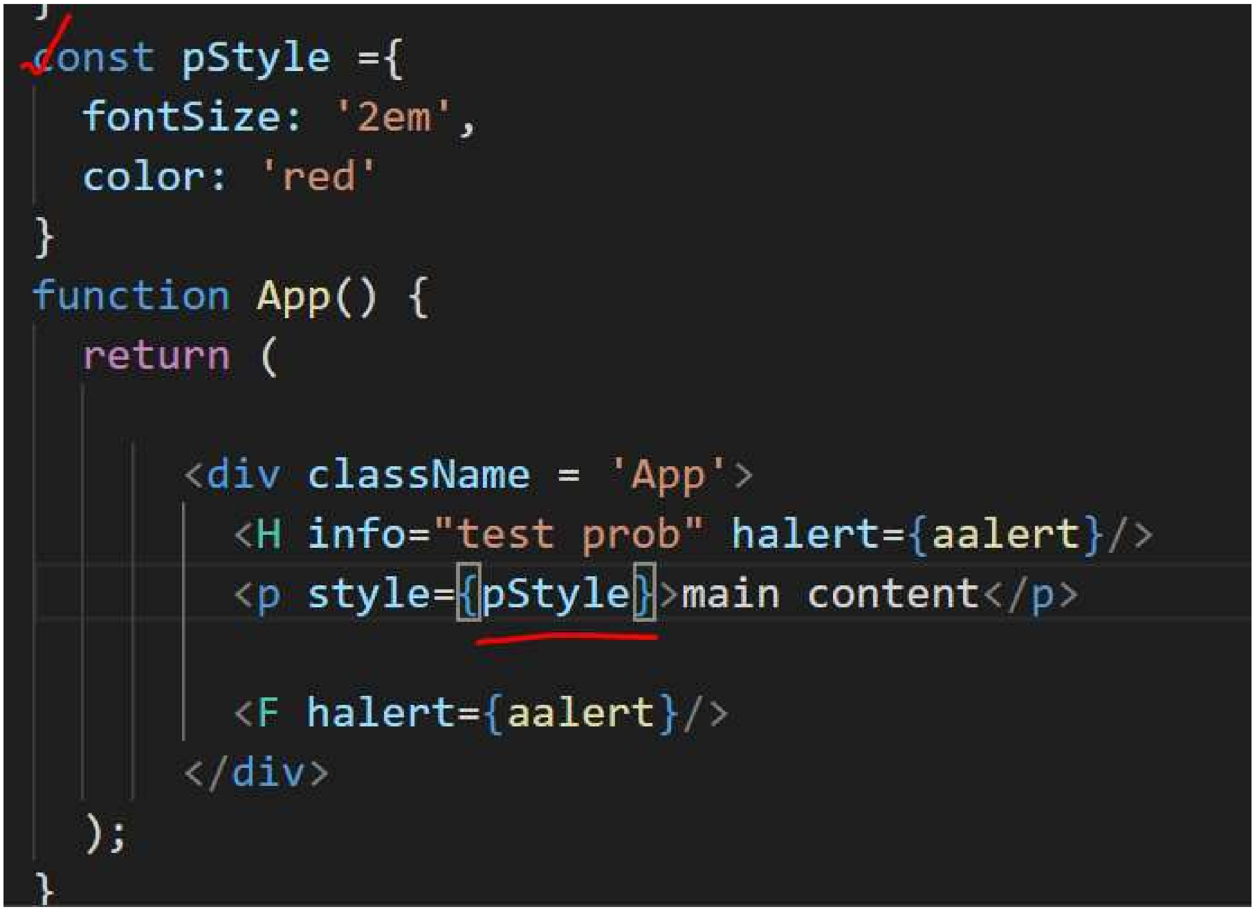
test prob

main content

snake

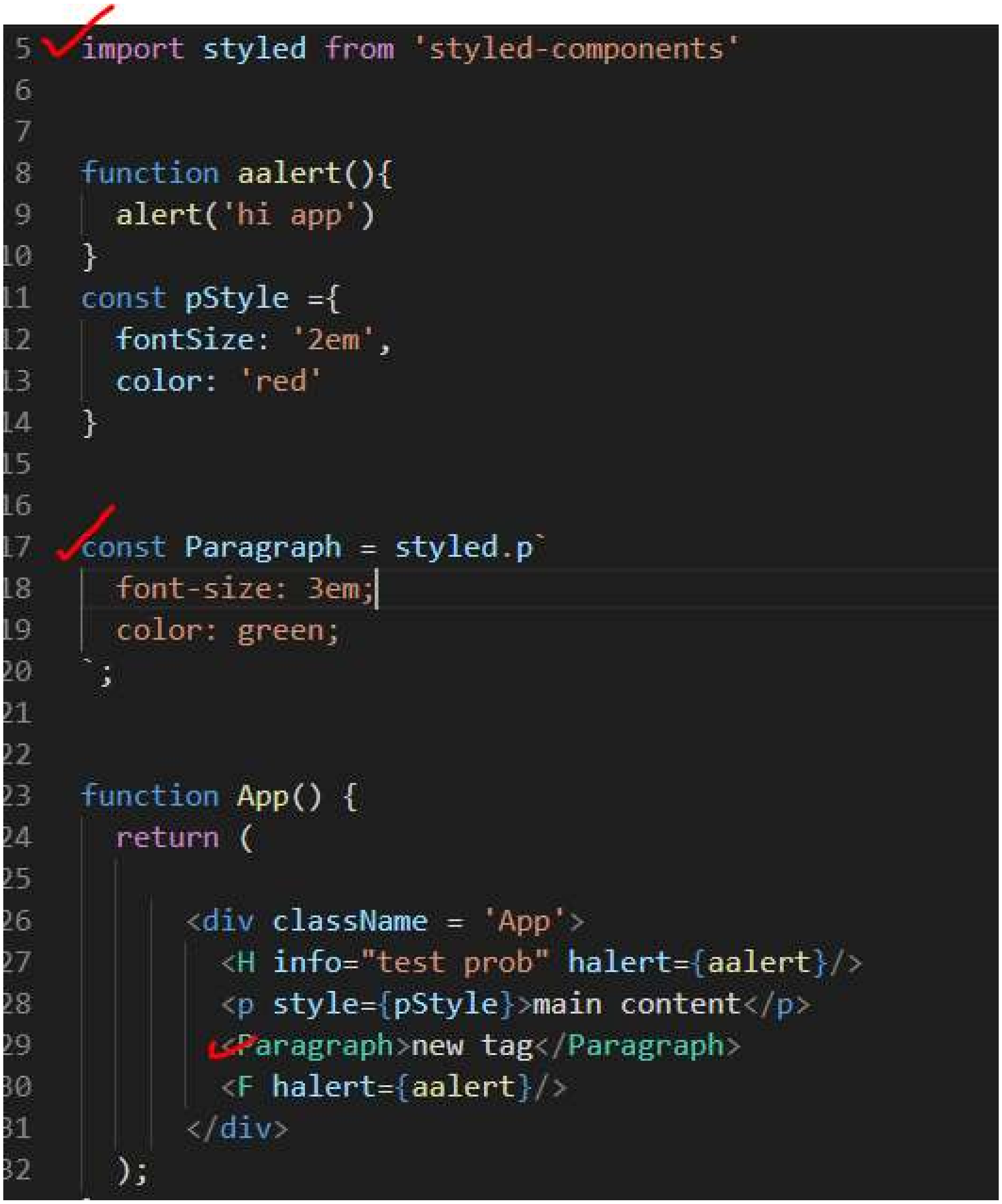
elephant

CSS



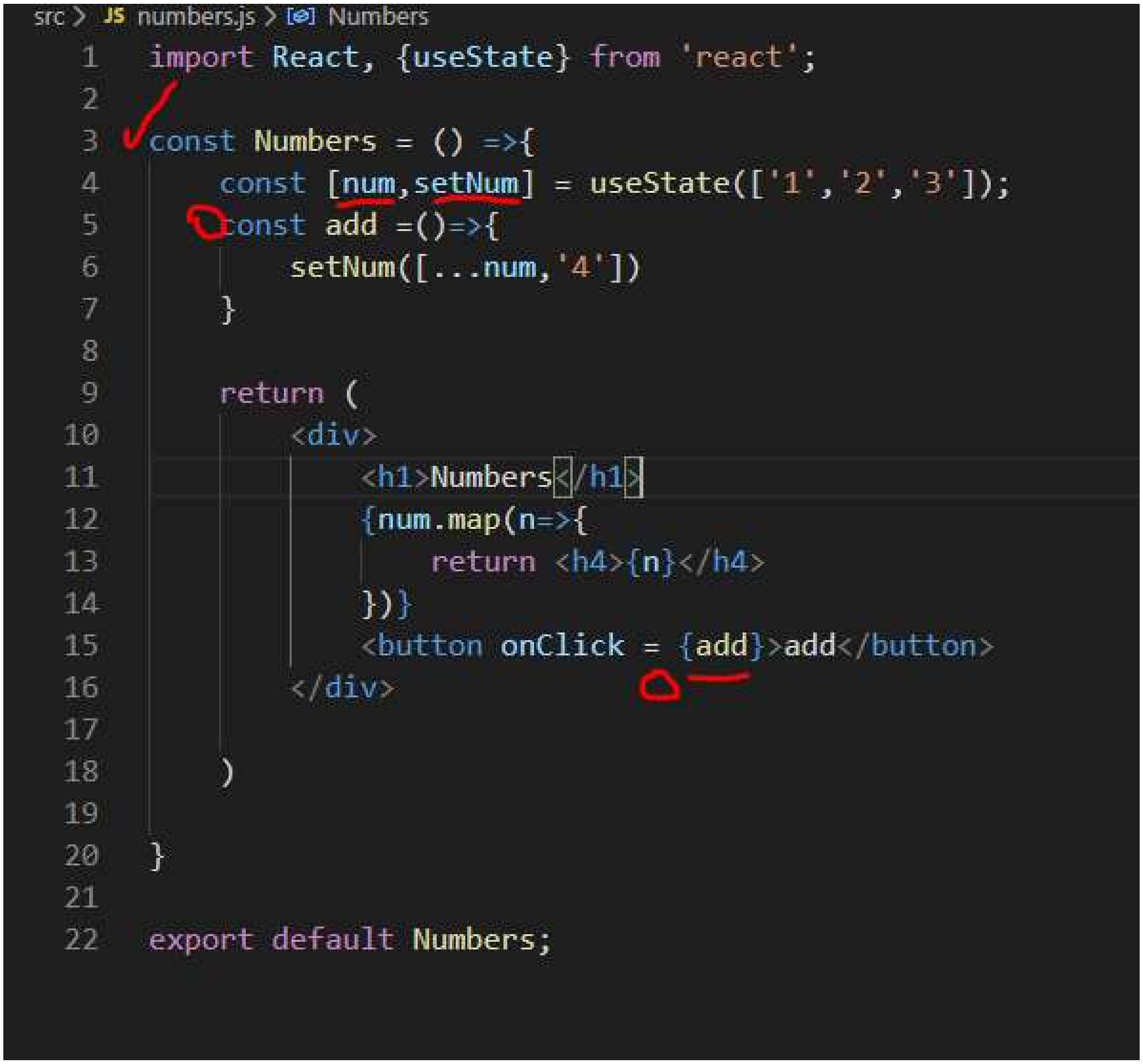
```
}
const pStyle ={
  fontSize: '2em',
  color: 'red'
}
function App() {
  return (

      <div className = 'App'>
        <H info="test prob" halert={aalert}/>
        <p style={pStyle}>main content</p>

        <F halert={aalert}/>
      </div>
  );
}
```

npm install styled-components –save

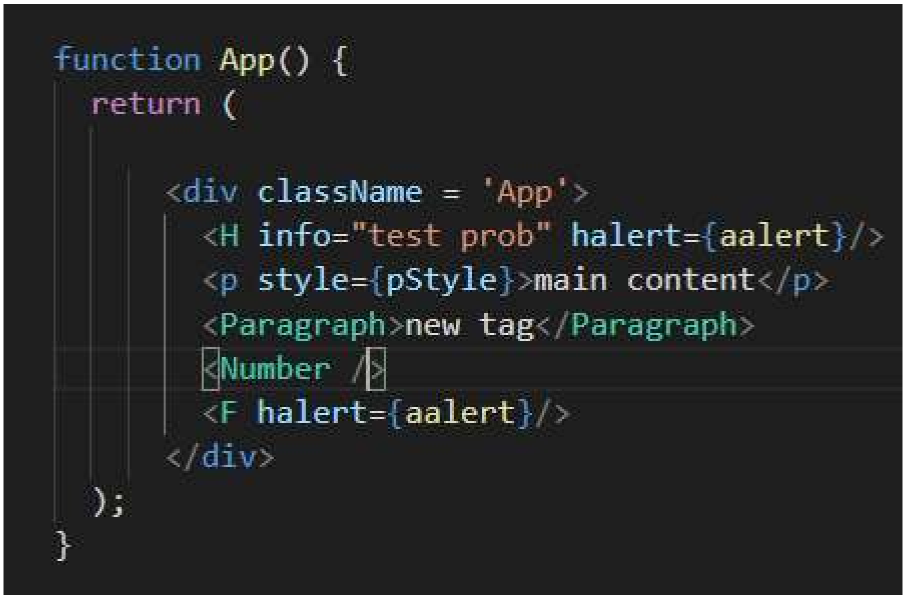


```
import styled from 'styled-components'


function aalert(){
  alert('hi app')
}
const pStyle ={
  fontSize: '2em',
  color: 'red'
}


const Paragraph = styled.p`
  font-size: 3em;
  color: green;
`;


function App() {
  return (

      <div className = 'App'>
        <H info="test prob" halert={aalert}/>
        <p style={pStyle}>main content</p>
        <Paragraph>new tag</Paragraph>
        <F halert={aalert}/>
      </div>
  );
```

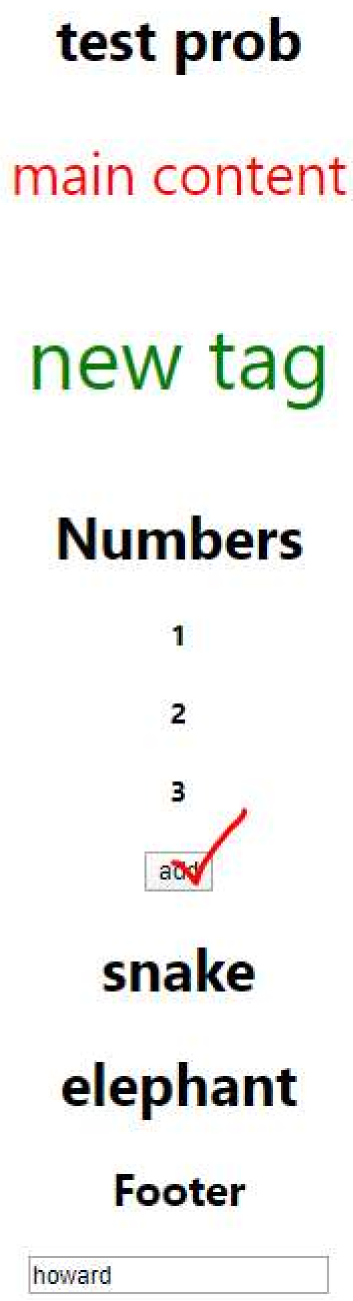
Hooks state



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

const Numbers = () =>{
    const [num,setNum] = useState(['1','2','3']);
    const add =()=>{
        setNum([...num,'4'])
    }

    return (
        <div>
            <h1>Numbers</h1>
            {num.map(n=>{
                return <h4>{n}</h4>
            })}
            <button onClick = {add}>add</button>
        </div>

    )

}

export default Numbers;
```



```
function App() {
  return (

    <div className = 'App'>
      <H info="test prob" halert={aalert}/>
      <p style={pStyle}>main content</p>
      <Paragraph>new tag</Paragraph>
      <Number />
      <F halert={aalert}/>
    </div>
  );
}
```



test prob

main content

new tag

Numbers

1

2

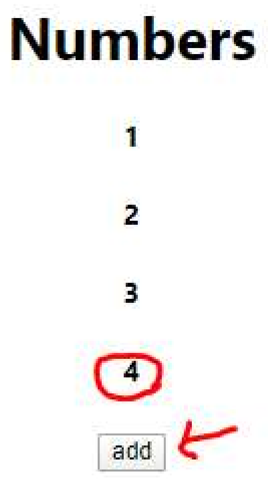
3

snake

elephant

Footer

howard



Numbers

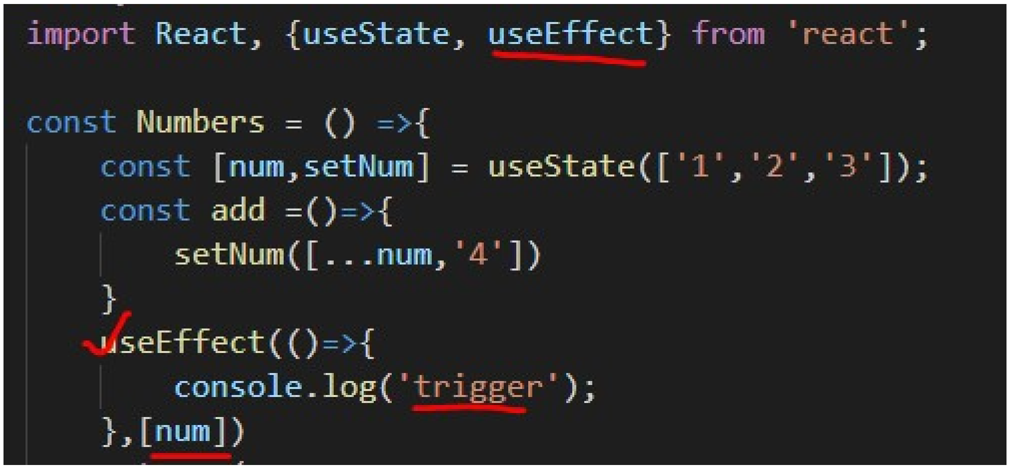
1

2

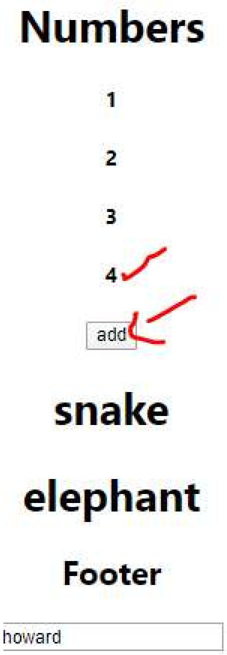
3

4

Hook effect



```
import React, {useState, useEffect} from 'react';

const Numbers = () =>{
    const [num,setNum] = useState(['1','2','3']);
    const add =()=>{
        setNum([...num,'4'])
    }
    useEffect(()=>{
        console.log('trigger');
    },[num])
```



Numbers

1

2

3

4

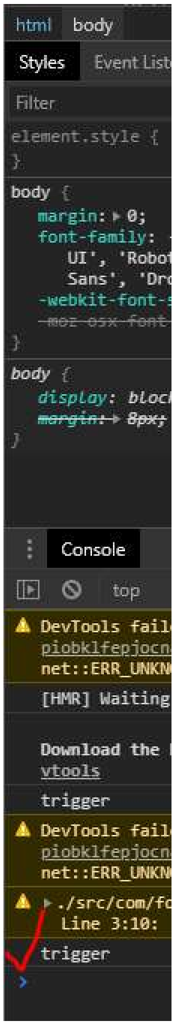
add

snake

elephant

Footer

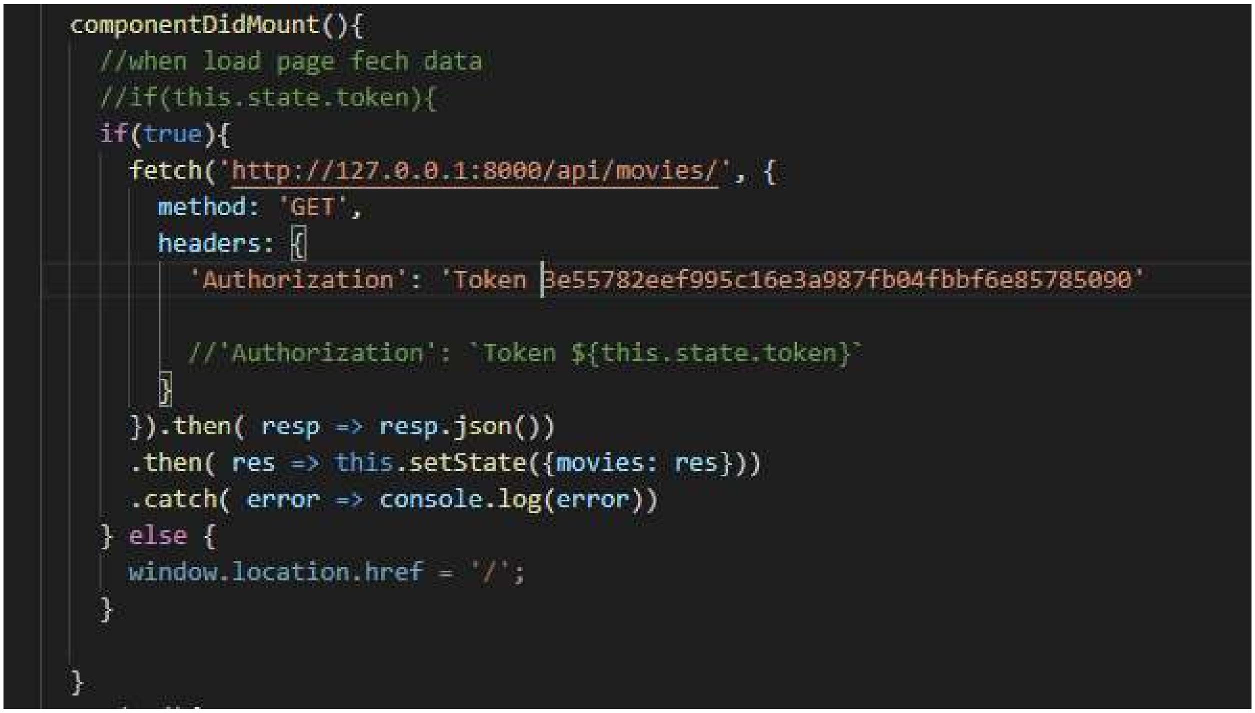
howard



```
element.style {
}
body {
  margin: ▸ 0;
  font-family: -
    UI', 'Robot
    Sans', 'Dro
  -webkit-font-s
  ~~-moz-osx-font-~~
}
body {
  display: block
  ~~margin: ▸ 8px;~~
}
```

```
⚠ DevTools fail
  piobklfepjocn
  net::ERR_UNKN
[HMR] Waiting

Download the
vtools
trigger
⚠ DevTools fail
  piobklfepjocn
  net::ERR_UNKN
⚠ ▸ ./src/com/fo
    Line 3:10:
trigger
>
```

Get data from api with Token



```
componentDidMount(){
  //when load page fech data
  //if(this.state.token){
  if(true){
    fetch('http://127.0.0.1:8000/api/movies/', {
      method: 'GET',
      headers: {
        'Authorization': 'Token 3e55782eef995c16e3a987fb04fbbf6e85785090'

        //'Authorization': `Token ${this.state.token}`
      }
    }).then( resp => resp.json())
    .then( res => this.setState({movies: res}))
    .catch( error => console.log(error))
  } else {
    window.location.href = '/';
  }

}
```