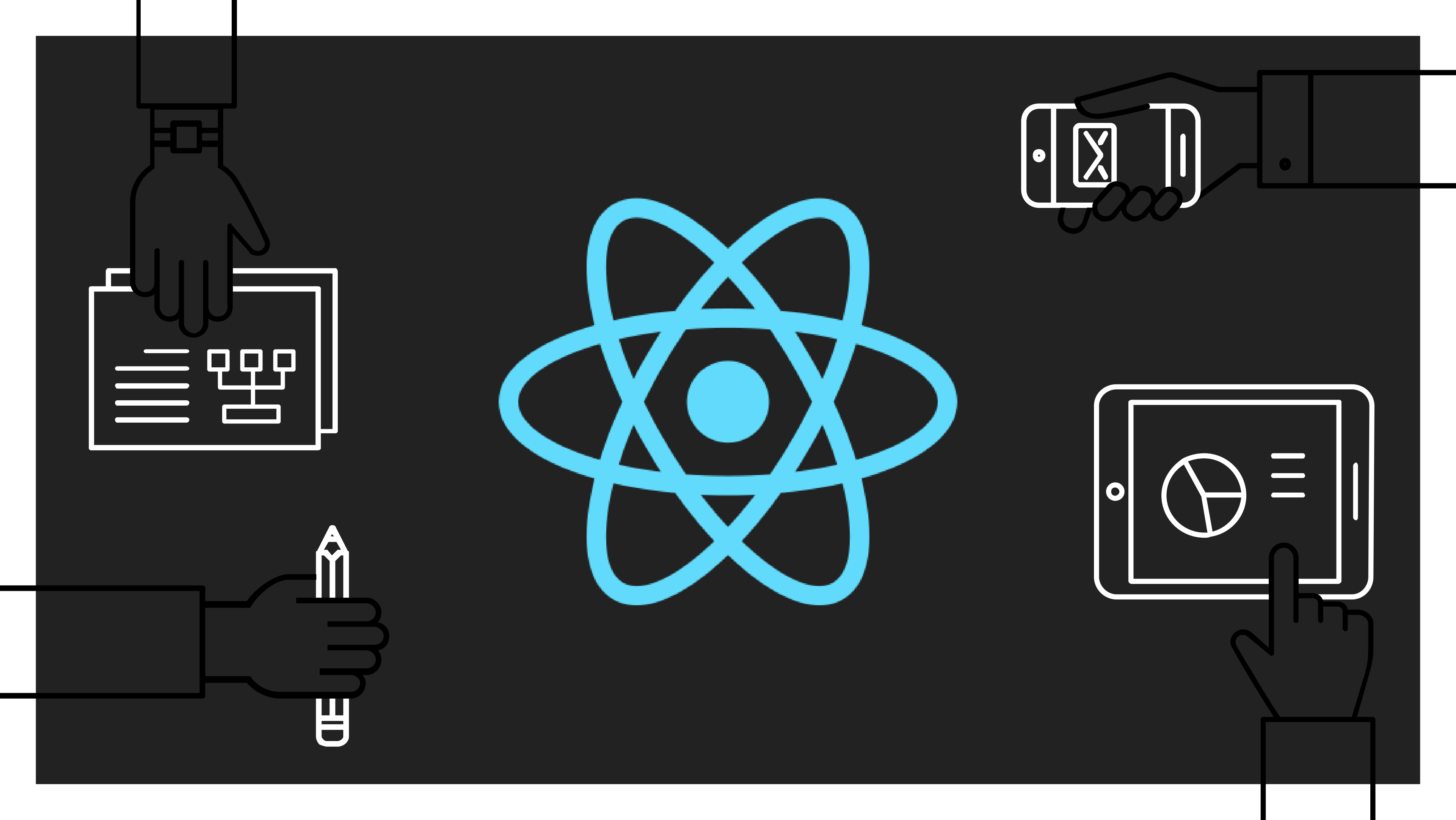
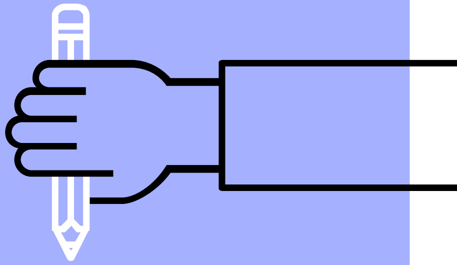
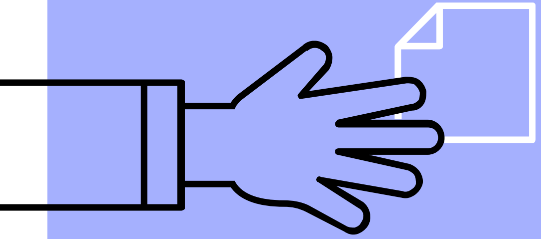
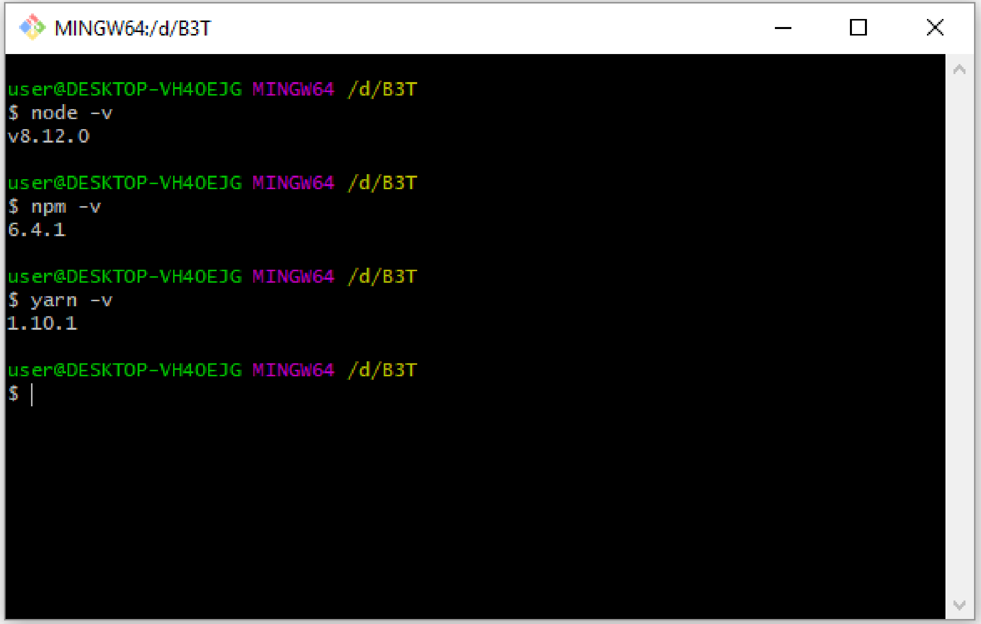




Setup

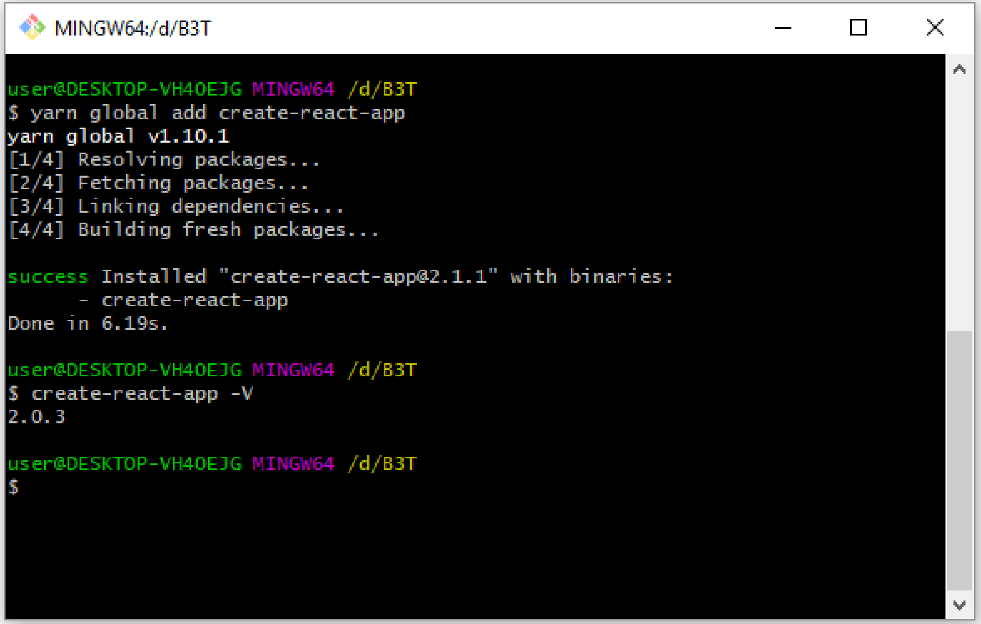


Require package manager

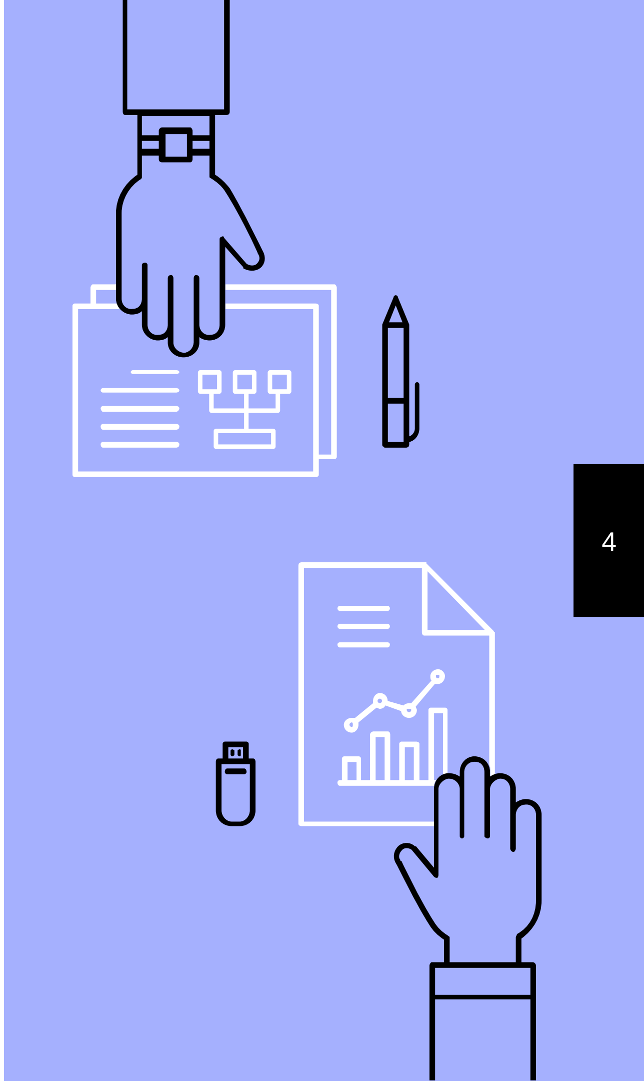


```
user@DESKTOP-VH40EJG MINGW64 /d/B3T
$ node -v
v8.12.0

user@DESKTOP-VH40EJG MINGW64 /d/B3T
$ npm -v
6.4.1

user@DESKTOP-VH40EJG MINGW64 /d/B3T
$ yarn -v
1.10.1

user@DESKTOP-VH40EJG MINGW64 /d/B3T
$ |
```

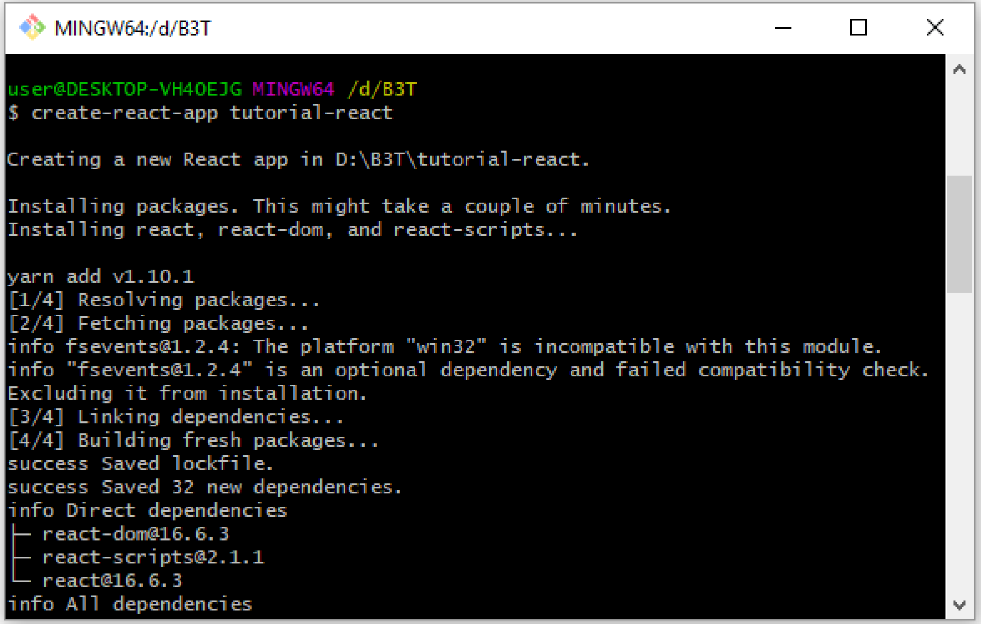
Install create-react-app



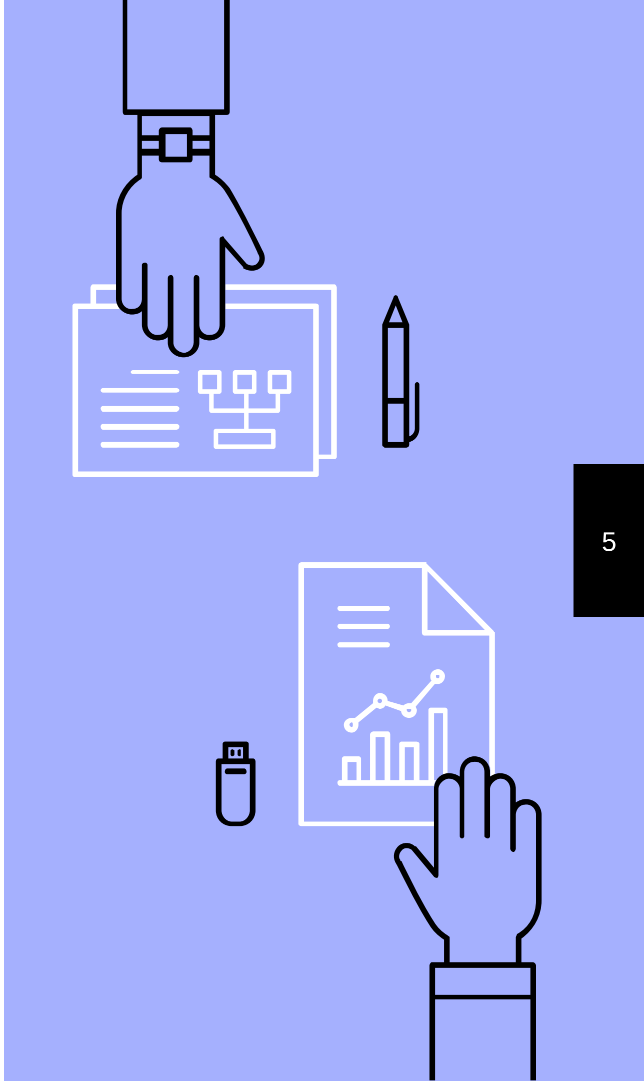
```
user@DESKTOP-VH40EJG MINGW64 /d/B3T
$ yarn global add create-react-app
yarn global v1.10.1
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Installed "create-react-app@2.1.1" with binaries:
      - create-react-app
Done in 6.19s.

user@DESKTOP-VH40EJG MINGW64 /d/B3T
$ create-react-app -V
2.0.3

user@DESKTOP-VH40EJG MINGW64 /d/B3T
$
```



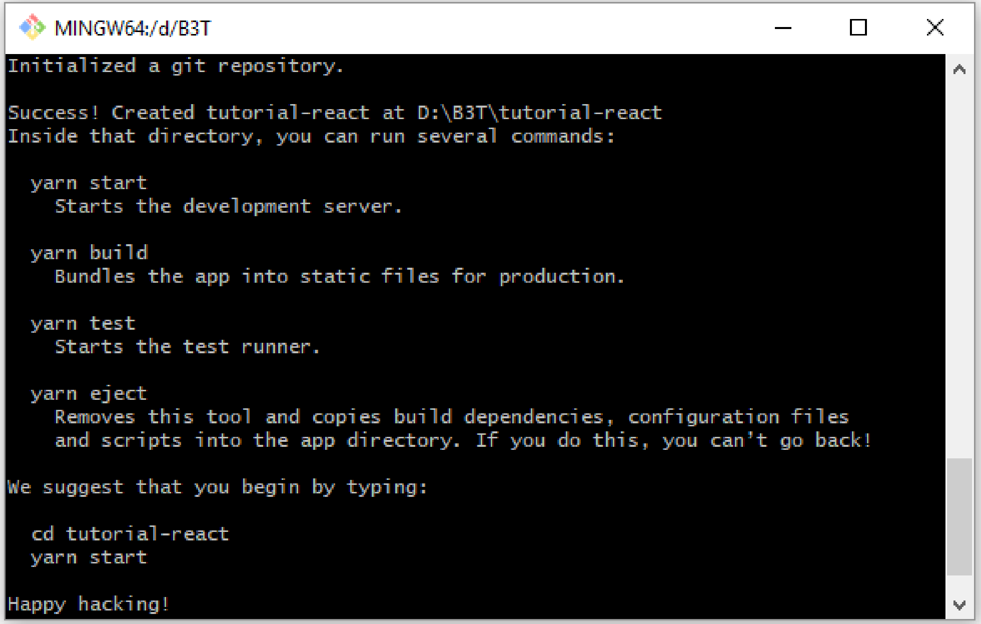
Create project



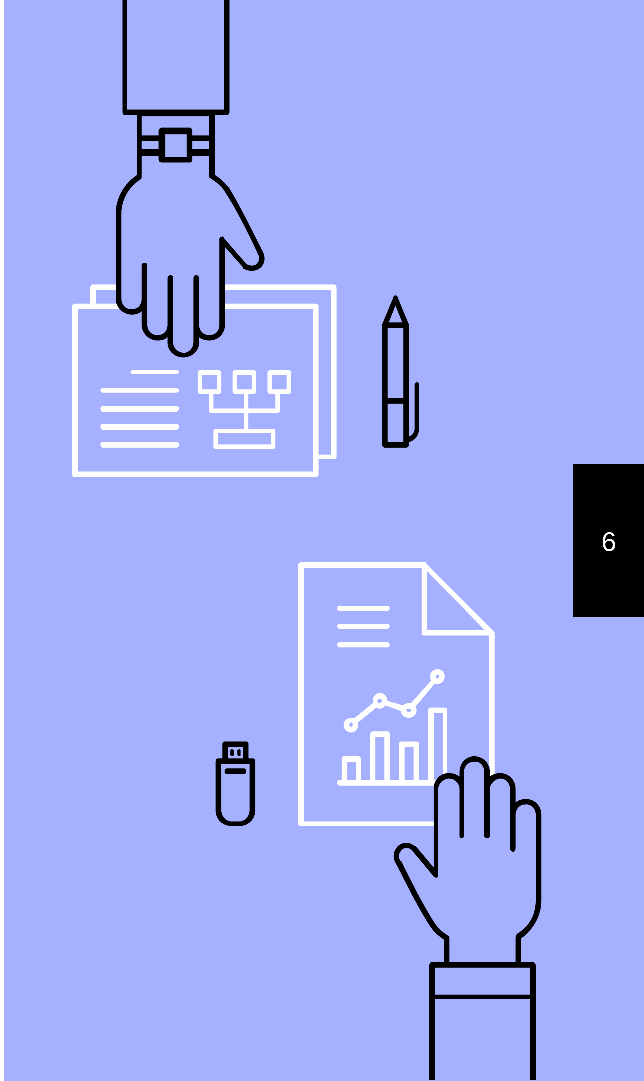
```
user@DESKTOP-VH40EJG MINGW64 /d/B3T
$ create-react-app tutorial-react

Creating a new React app in D:\B3T\tutorial-react.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts...

yarn add v1.10.1
[1/4] Resolving packages...
[2/4] Fetching packages...
info fsevents@1.2.4: The platform "win32" is incompatible with this module.
info "fsevents@1.2.4" is an optional dependency and failed compatibility check.
Excluding it from installation.
[3/4] Linking dependencies...
[4/4] Building fresh packages...
success Saved lockfile.
success Saved 32 new dependencies.
info Direct dependencies
├─ react-dom@16.6.3
├─ react-scripts@2.1.1
└─ react@16.6.3
info All dependencies
```



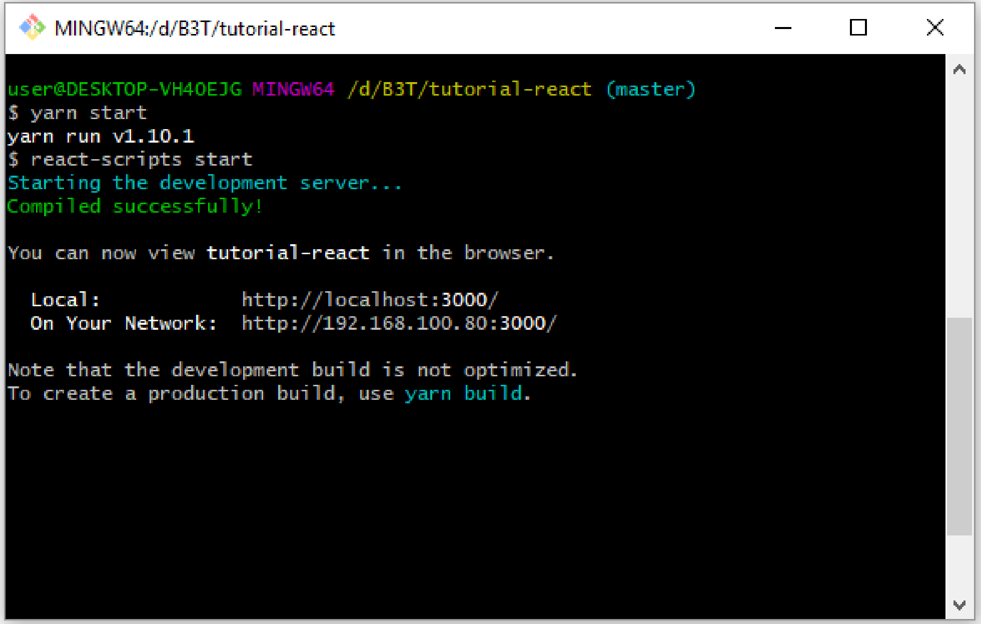
Done



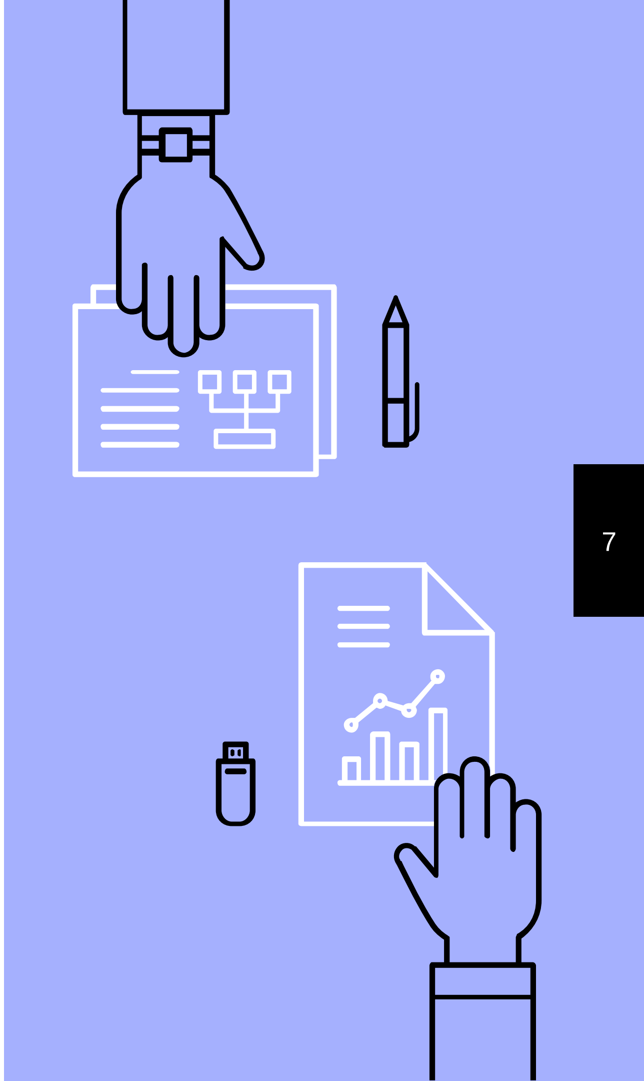
```
Initialized a git repository.

Success! Created tutorial-react at D:\B3T\tutorial-react
Inside that directory, you can run several commands:

  yarn start
    Starts the development server.

  yarn build
    Bundles the app into static files for production.

  yarn test
    Starts the test runner.

  yarn eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd tutorial-react
  yarn start

Happy hacking!
```

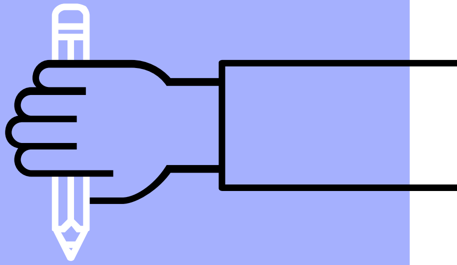


Run dev server

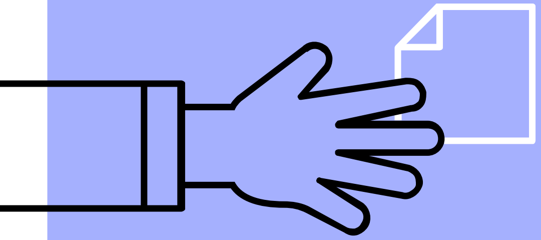


```
user@DESKTOP-VH40EJG MINGW64 /d/B3T/tutorial-react (master)
$ yarn start
yarn run v1.10.1
$ react-scripts start
Starting the development server...
Compiled successfully!

You can now view tutorial-react in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://192.168.100.80:3000/

Note that the development build is not optimized.
To create a production build, use yarn build.
```

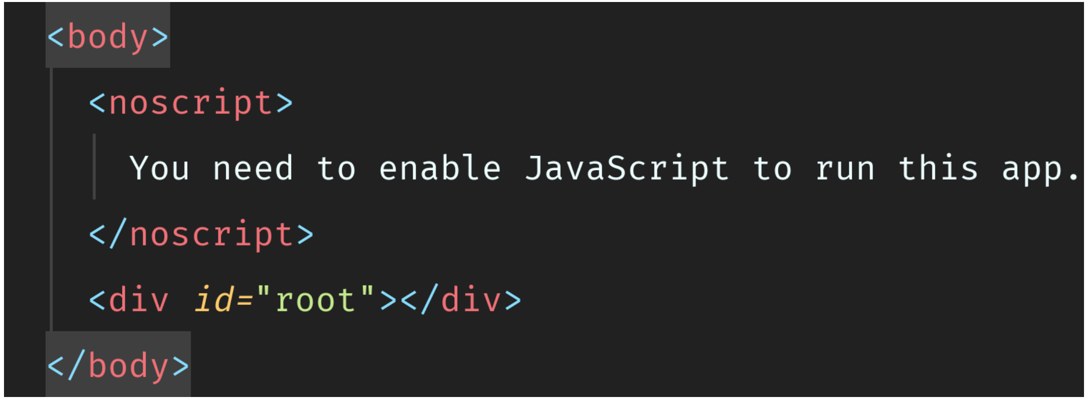




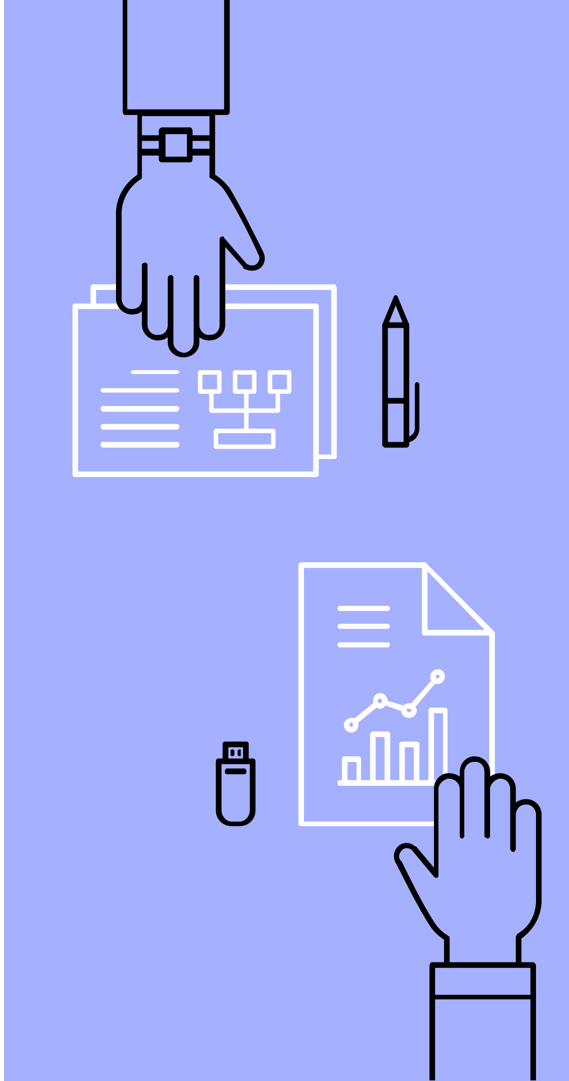
Hello World Page



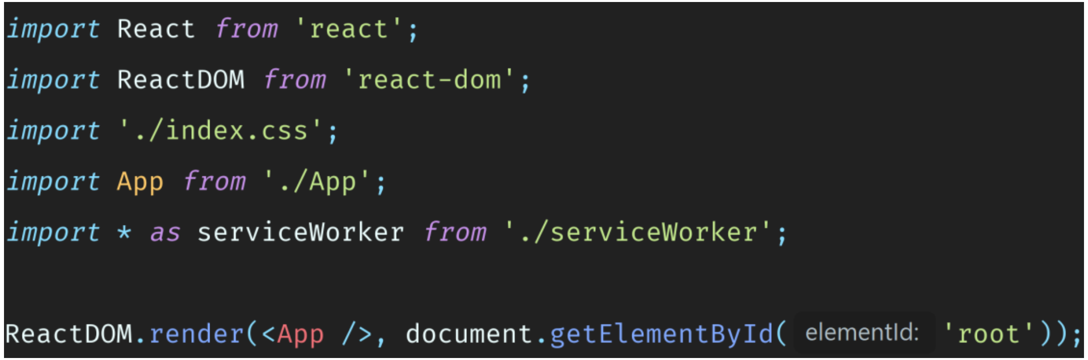
public → index.html



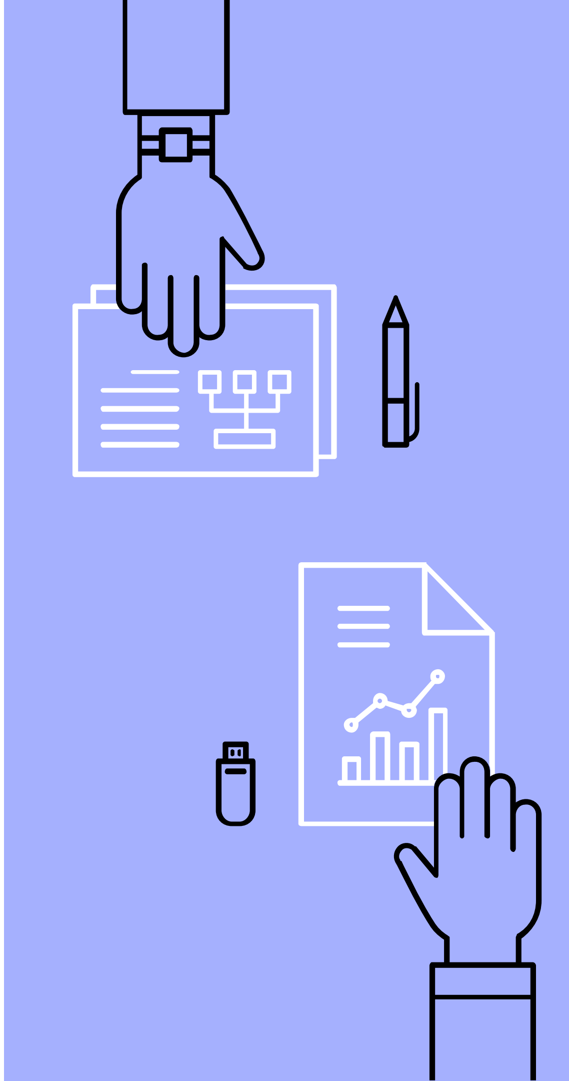
```
<body>
  <noscript>
    You need to enable JavaScript to run this app.
  </noscript>
  <div id="root"></div>
</body>
```



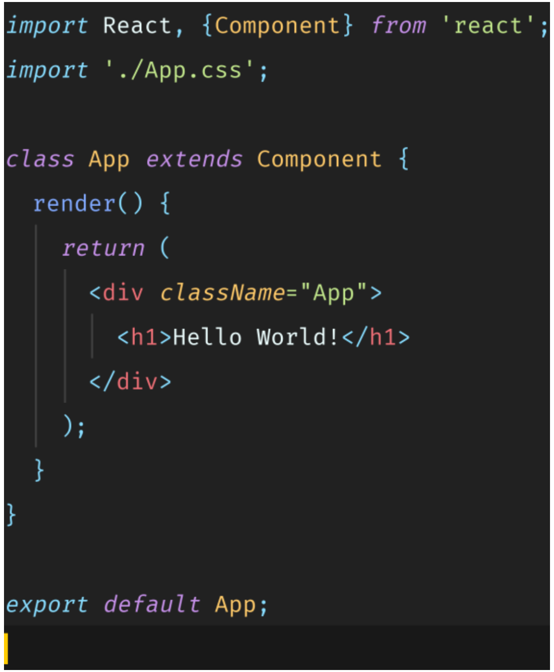
src → index.js



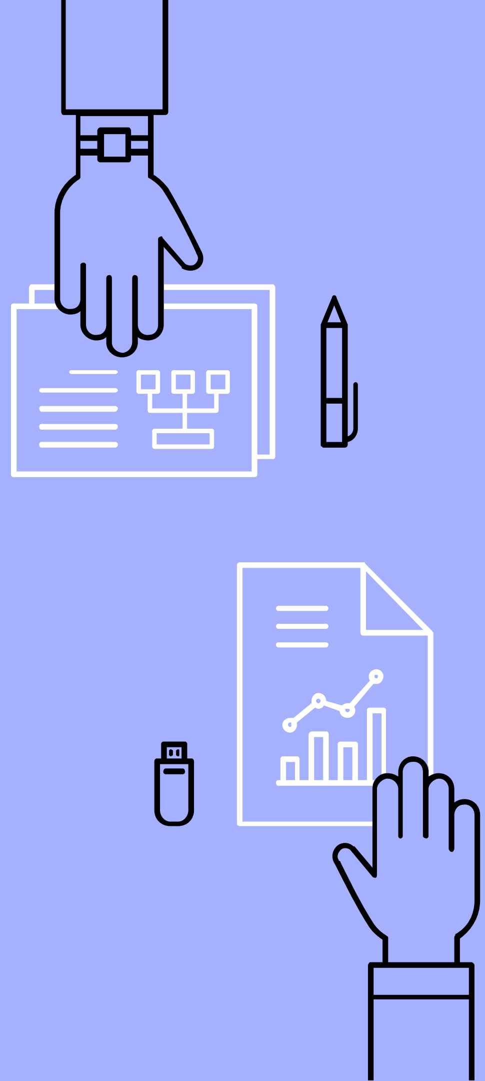
```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import * as serviceWorker from './serviceWorker';

ReactDOM.render(<App />, document.getElementById( elementId: 'root'));
```



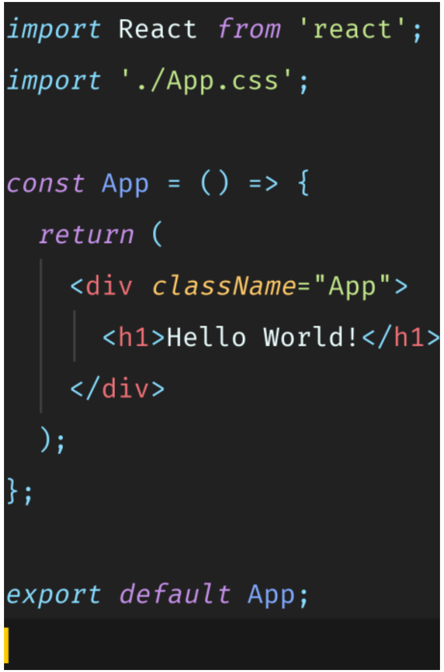
src → App.js



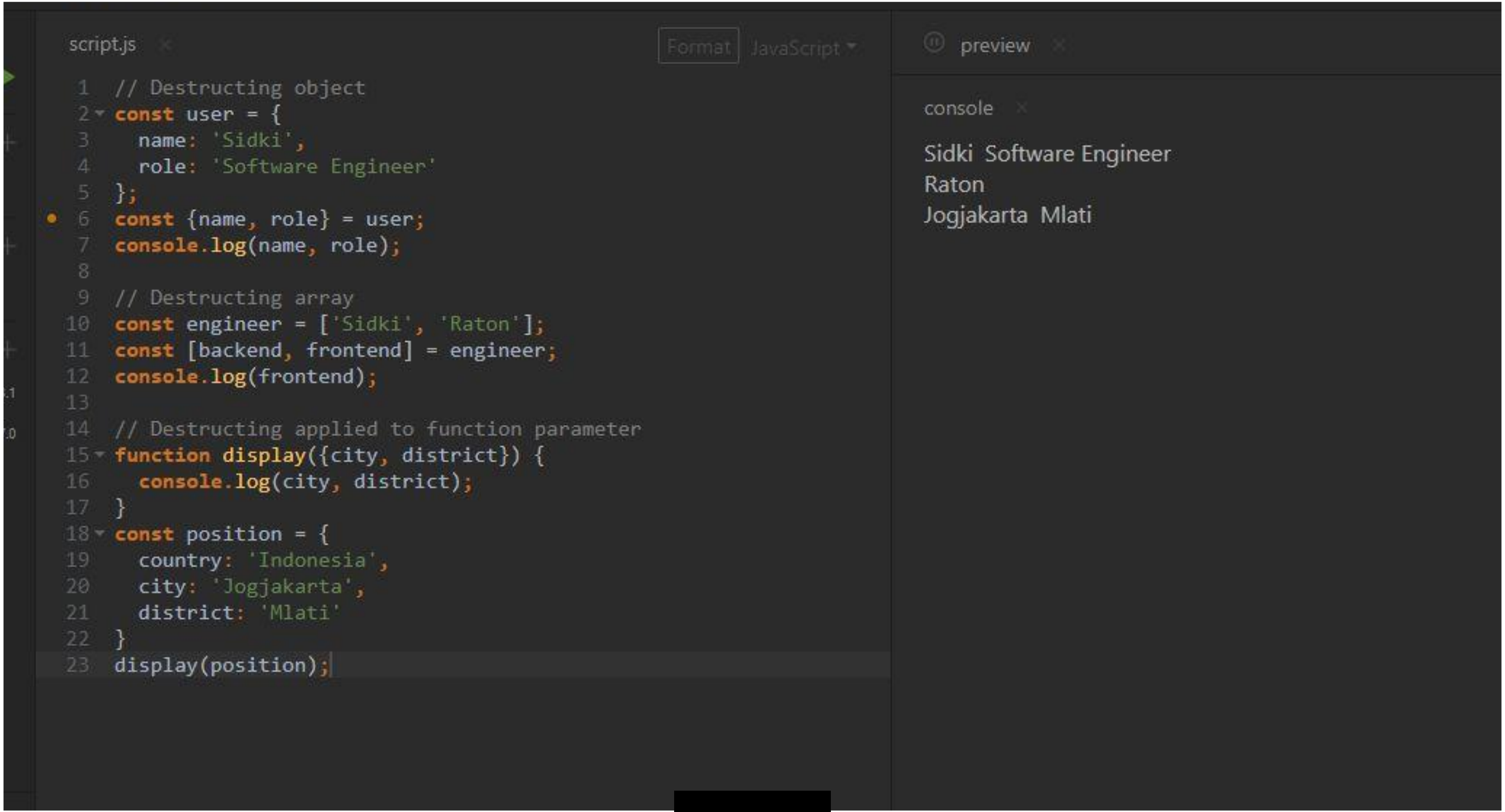
```
import React, {Component} from 'react';
import './App.css';

class App extends Component {
  render() {
    return (
      <div className="App">
        <h1>Hello World!</h1>
      </div>
    );
  }
}

export default App;
```



Stateless component

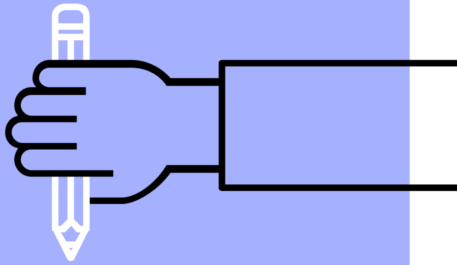


```
import React from 'react';
import './App.css';

const App = () => {
  return (
    <div className="App">
      <h1>Hello World!</h1>
    </div>
  );
};

export default App;
```

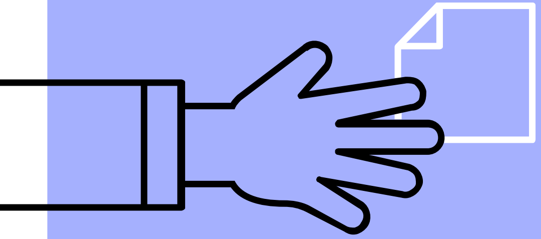


```javascript
// Destructing object
const user = {
  name: 'Sidki',
  role: 'Software Engineer'
};
const {name, role} = user;
console.log(name, role);

// Destructing array
const engineer = ['Sidki', 'Raton'];
const [backend, frontend] = engineer;
console.log(frontend);

// Destructing applied to function parameter
function display({city, district}) {
  console.log(city, district);
}
const position = {
  country: 'Indonesia',
  city: 'Jogjakarta',
  district: 'Mlati'
}
display(position);
```

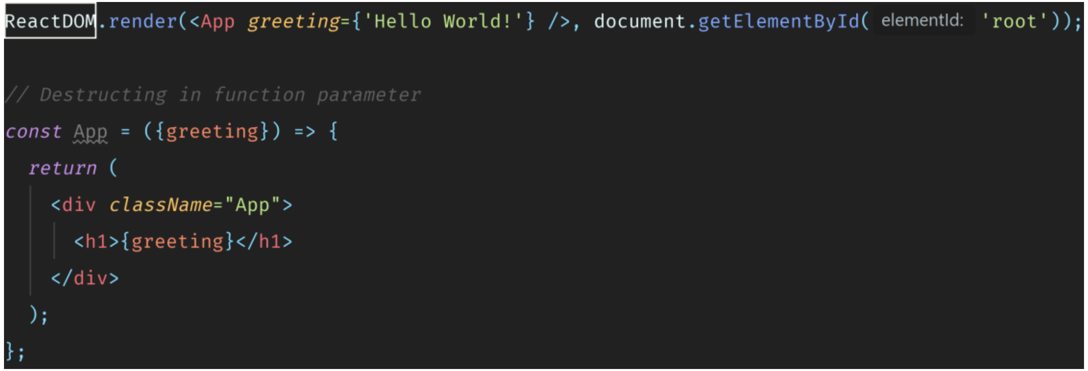
```
Sidki  Software Engineer
Raton
Jogjakarta  Mlati
```



Data Model

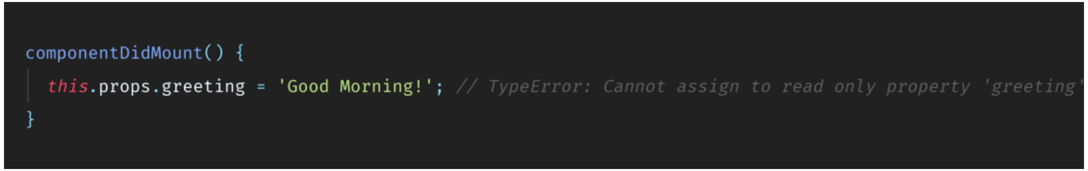


Props



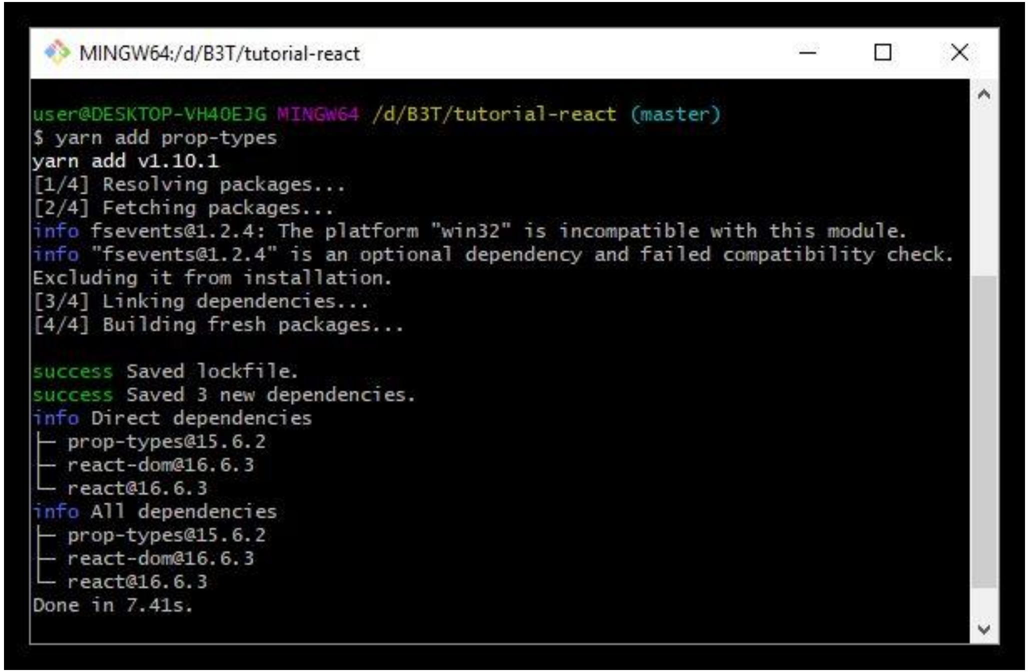
```
ReactDOM.render(<App greeting={'Hello World!'} />, document.getElementById( elementId: 'root'));

// Destructing in function parameter
const App = ({greeting}) => {
  return (
    <div className="App">
      <h1>{greeting}</h1>
    </div>
  );
};
```

Props are immutable



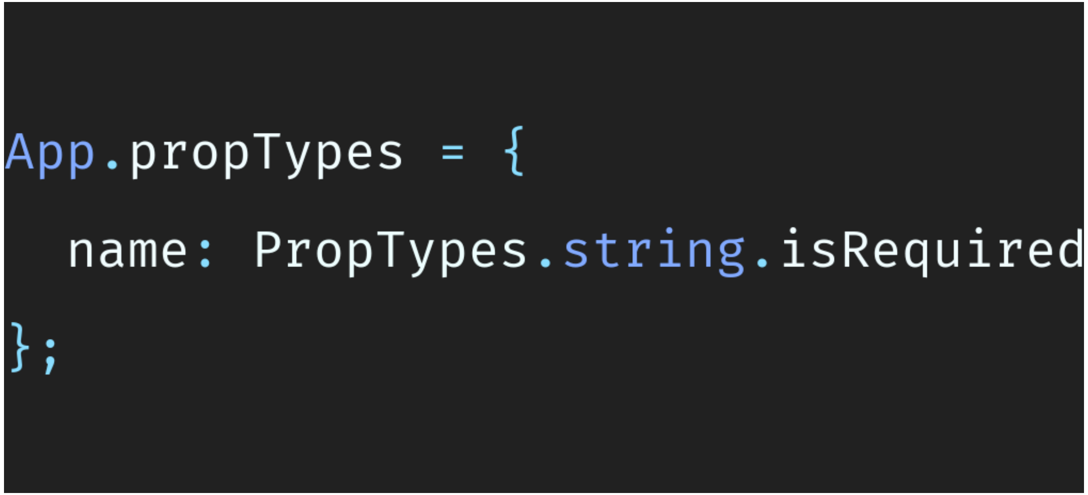
```
componentDidMount() {
  this.props.greeting = 'Good Morning!'; // TypeError: Cannot assign to read only property 'greeting'
}
```

Prop types

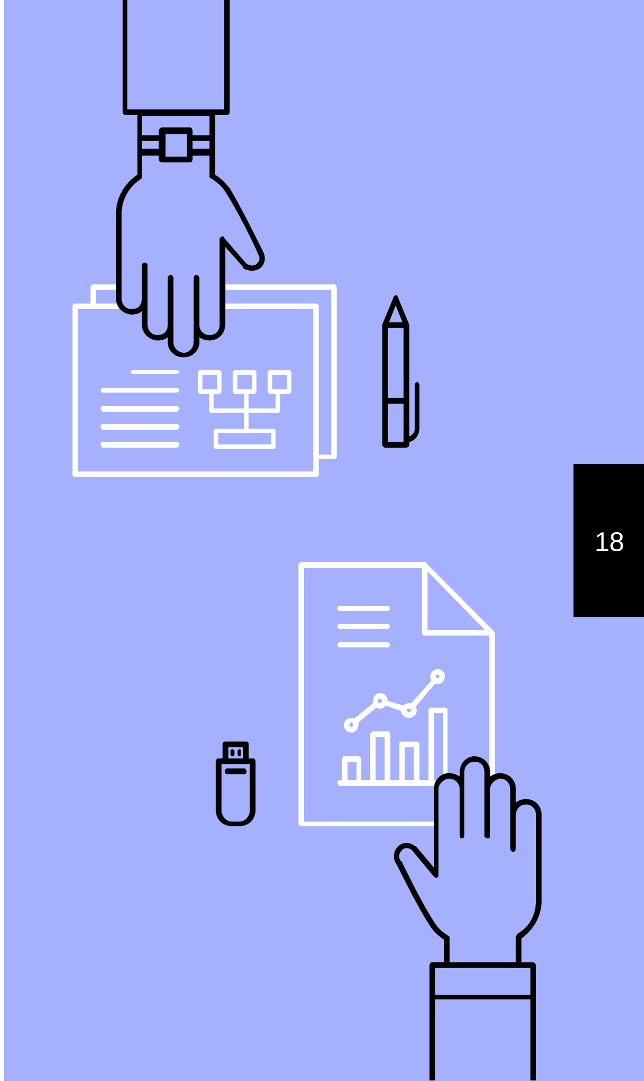


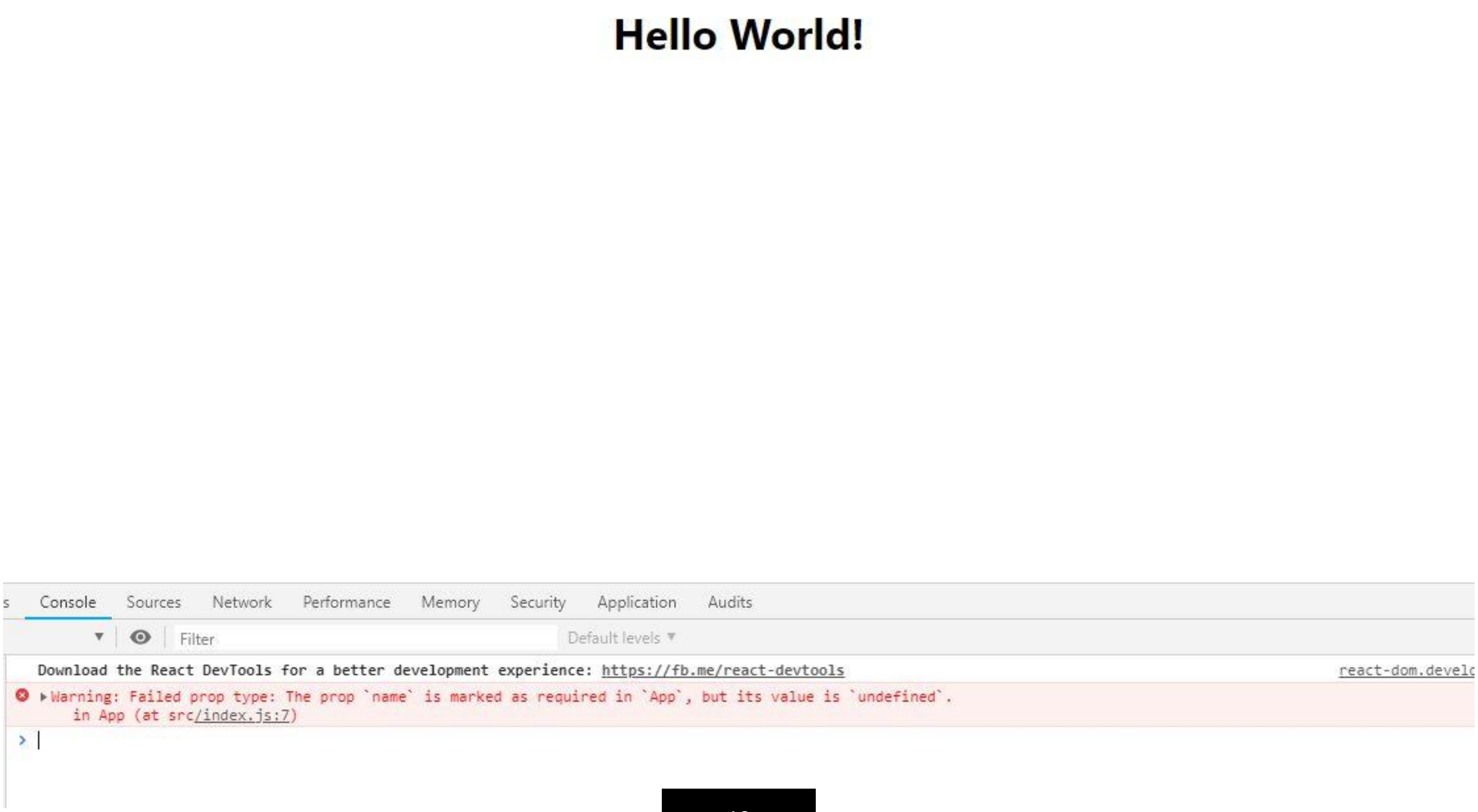
```
user@DESKTOP-VH40EJG MINGW64 /d/B3T/tutorial-react (master)
$ yarn add prop-types
yarn add v1.10.1
[1/4] Resolving packages...
[2/4] Fetching packages...
info fsevents@1.2.4: The platform "win32" is incompatible with this module.
info "fsevents@1.2.4" is an optional dependency and failed compatibility check.
Excluding it from installation.
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Saved lockfile.
success Saved 3 new dependencies.
info Direct dependencies
├─ prop-types@15.6.2
├─ react-dom@16.6.3
└─ react@16.6.3
info All dependencies
├─ prop-types@15.6.2
├─ react-dom@16.6.3
└─ react@16.6.3
Done in 7.41s.
```

Prop types



```
App.propTypes = {
  name: PropTypes.string.isRequired
};
```

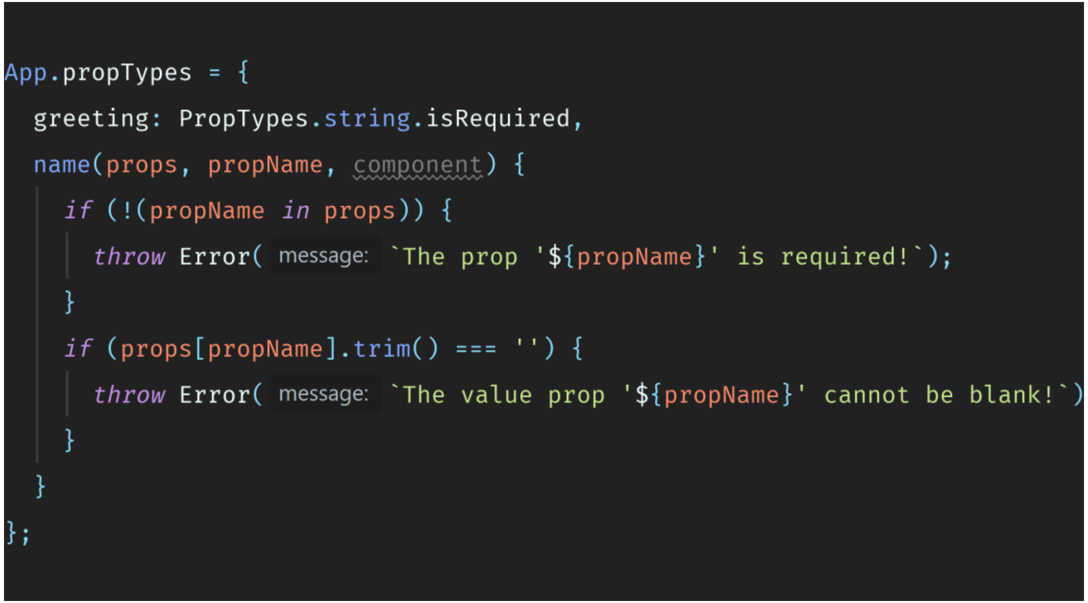




Hello World!

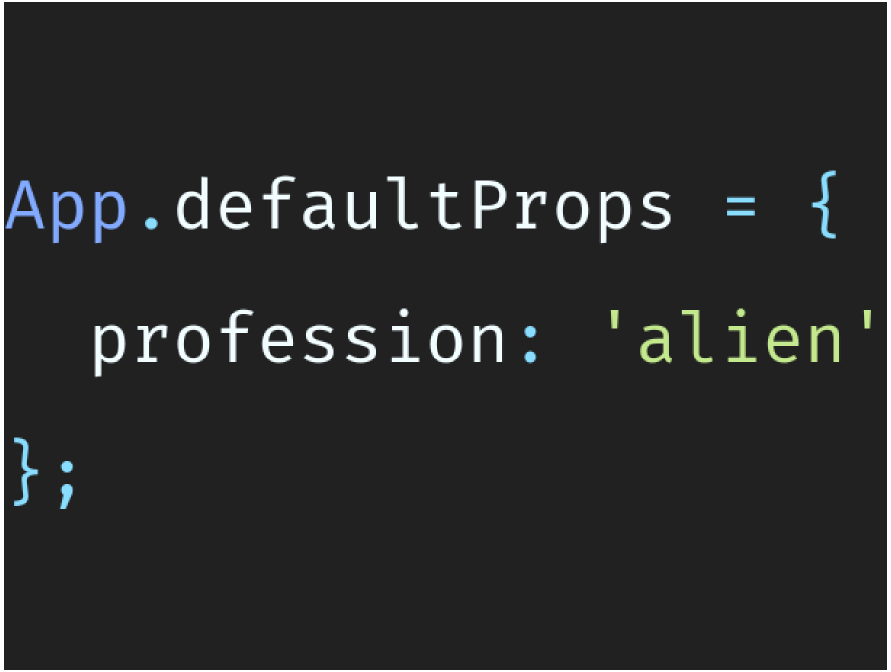
```
Download the React DevTools for a better development experience: https://fb.me/react-devtools     react-dom.develc
▸Warning: Failed prop type: The prop `name` is marked as required in `App`, but its value is `undefined`.
    in App (at src/index.js:7)
> |
```

Custom prop types validation



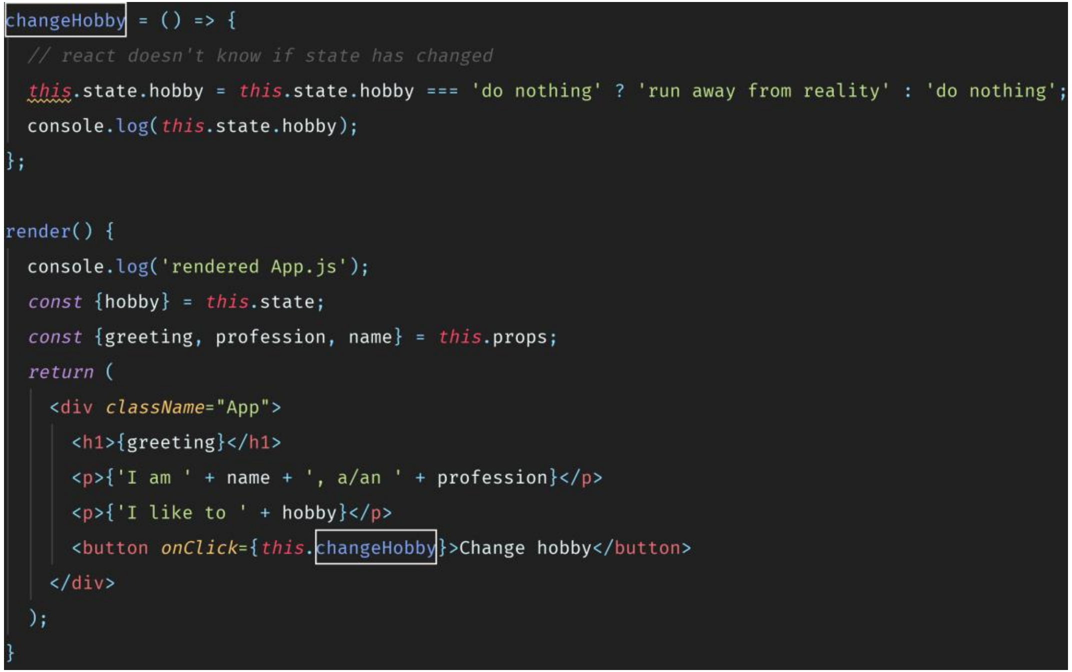
```
App.propTypes = {
  greeting: PropTypes.string.isRequired,
  name(props, propName, component) {
    if (!(propName in props)) {
      throw Error( message: `The prop '${propName}' is required!`);
    }
    if (props[propName].trim() === '') {
      throw Error( message: `The value prop '${propName}' cannot be blank!`)
    }
  }
};
```

Default props



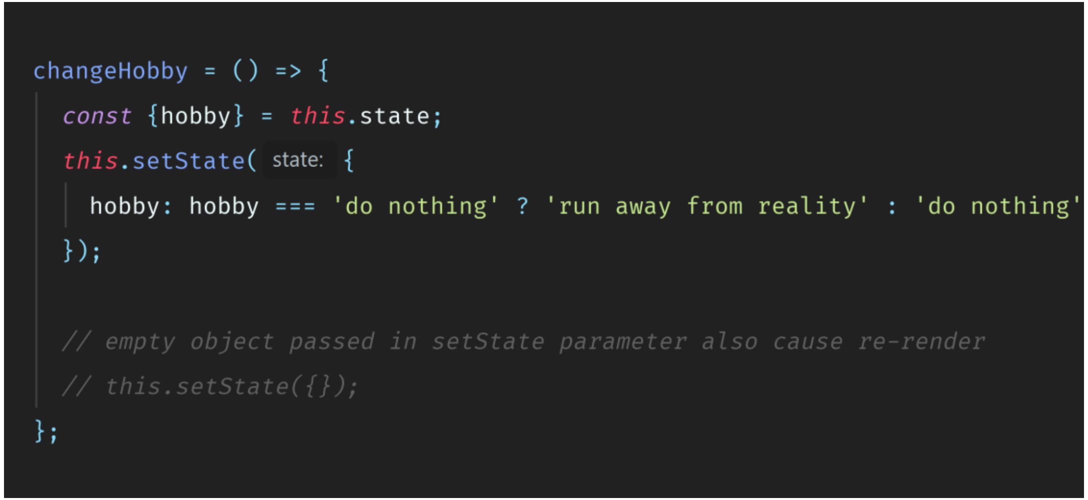
```
App.defaultProps = {
  profession: 'alien'
};
```

State

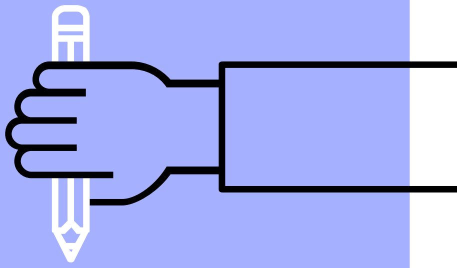


```
changeHobby = () => {
  // react doesn't know if state has changed
  this.state.hobby = this.state.hobby === 'do nothing' ? 'run away from reality' : 'do nothing';
  console.log(this.state.hobby);
};

render() {
  console.log('rendered App.js');
  const {hobby} = this.state;
  const {greeting, profession, name} = this.props;
  return (
    <div className="App">
      <h1>{greeting}</h1>
      <p>{'I am ' + name + ', a/an ' + profession}</p>
      <p>{'I like to ' + hobby}</p>
      <button onClick={this.changeHobby}>Change hobby</button>
    </div>
  );
}
```

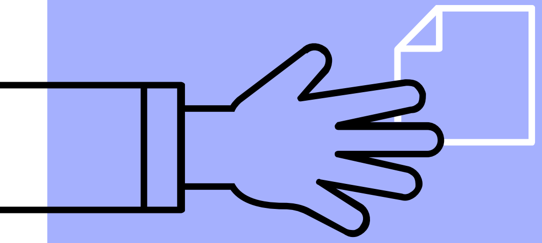
Use setState to change state



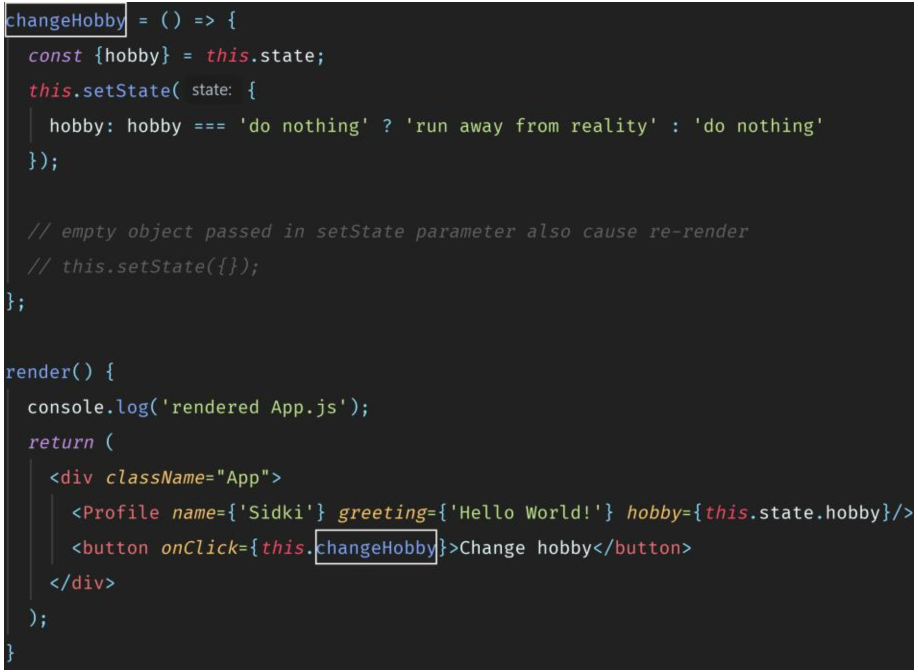
```
changeHobby = () => {
  const {hobby} = this.state;
  this.setState( state: {
    hobby: hobby === 'do nothing' ? 'run away from reality' : 'do nothing'
  });

  // empty object passed in setState parameter also cause re-render
  // this.setState({});
};
```



Component

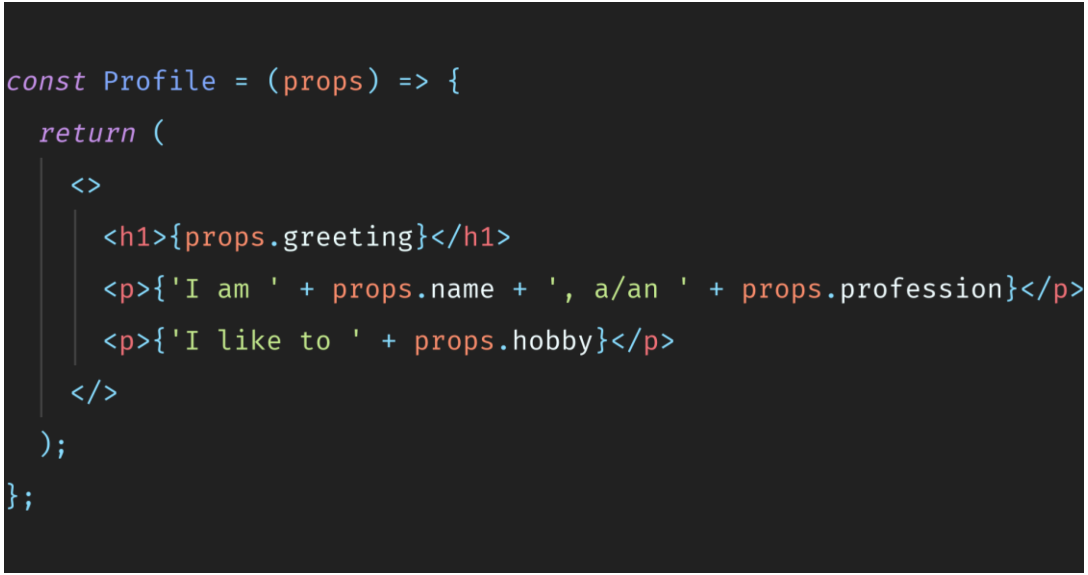


Reusable component



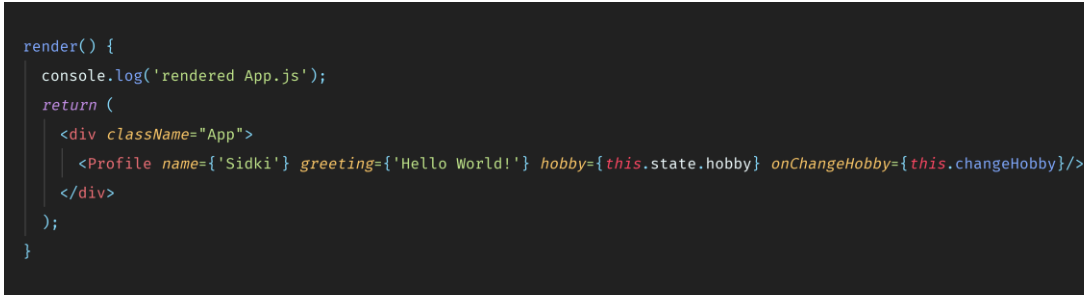
```
changeHobby = () => {
  const {hobby} = this.state;
  this.setState( state: {
    hobby: hobby === 'do nothing' ? 'run away from reality' : 'do nothing'
  });

  // empty object passed in setState parameter also cause re-render
  // this.setState({});
};

render() {
  console.log('rendered App.js');
  return (
    <div className="App">
      <Profile name={'Sidki'} greeting={'Hello World!'} hobby={this.state.hobby}/>
      <button onClick={this.changeHobby}>Change hobby</button>
    </div>
  );
}
```

Reusable component



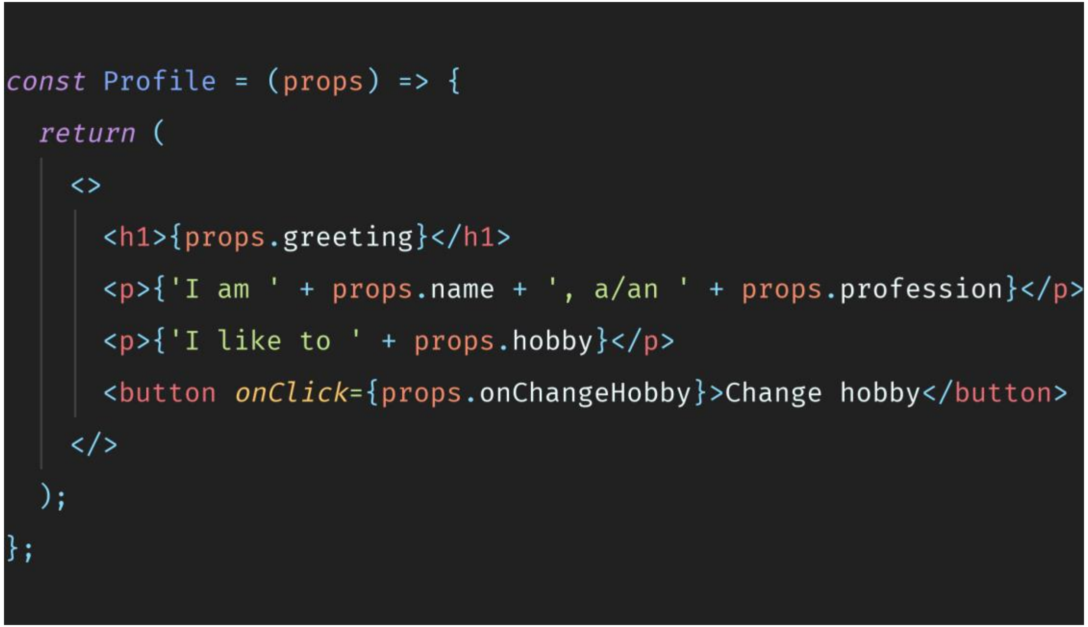
```
const Profile = (props) => {
  return (
    <>
      <h1>{props.greeting}</h1>
      <p>{'I am ' + props.name + ', a/an ' + props.profession}</p>
      <p>{'I like to ' + props.hobby}</p>
    </>
  );
};
```

Pass event to children



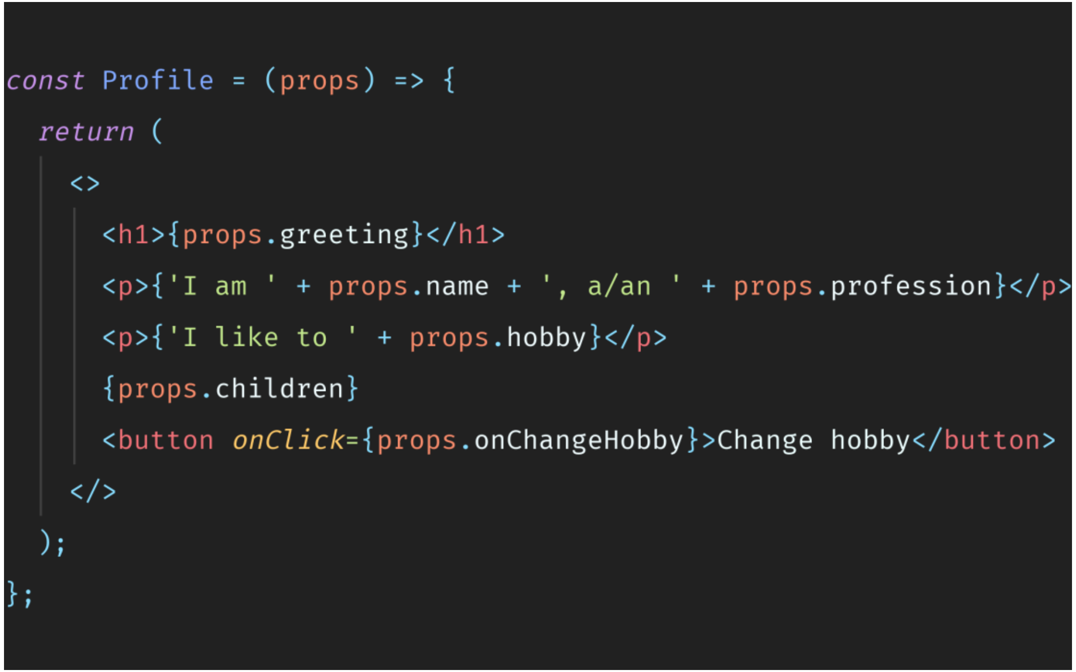
```
render() {
  console.log('rendered App.js');
  return (
    <div className="App">
      <Profile name={'Sidki'} greeting={'Hello World!'} hobby={this.state.hobby} onChangeHobby={this.changeHobby}/>
    </div>
  );
}
```

Pass event to children

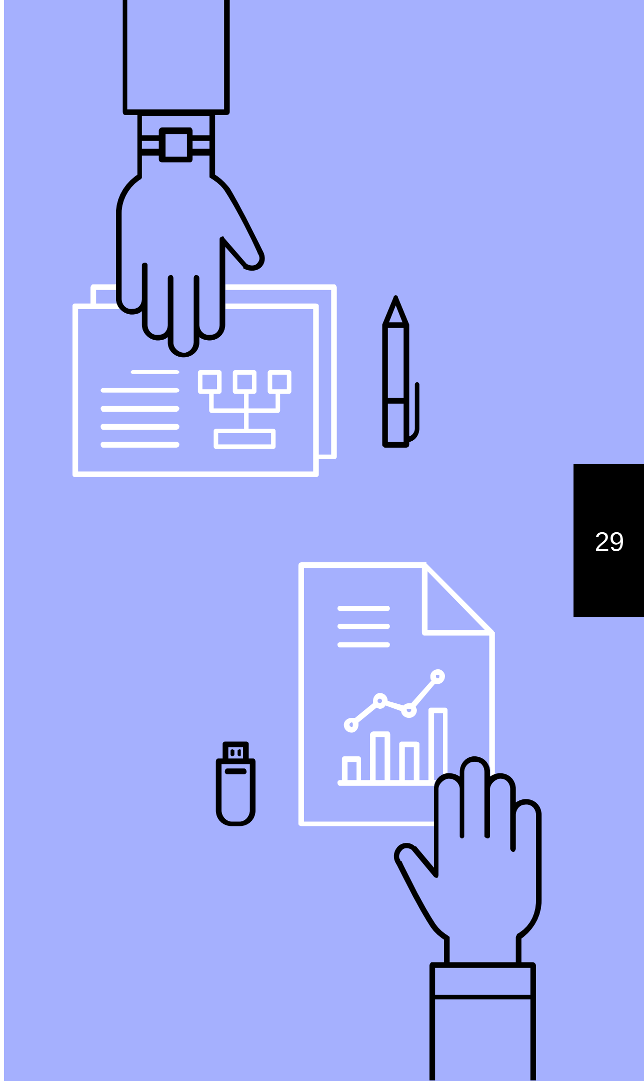


```
const Profile = (props) => {
  return (
    <>
      <h1>{props.greeting}</h1>
      <p>{'I am ' + props.name + ', a/an ' + props.profession}</p>
      <p>{'I like to ' + props.hobby}</p>
      <button onClick={props.onChangeHobby}>Change hobby</button>
    </>
  );
};
```

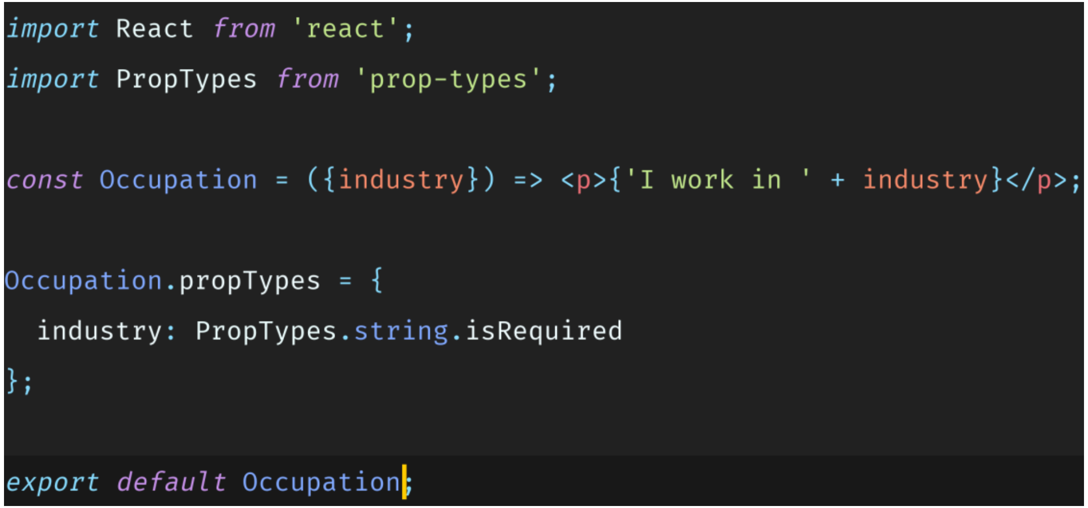
Nested component



```
const Profile = (props) => {
  return (
    <>
      <h1>{props.greeting}</h1>
      <p>{'I am ' + props.name + ', a/an ' + props.profession}</p>
      <p>{'I like to ' + props.hobby}</p>
      {props.children}
      <button onClick={props.onChangeHobby}>Change hobby</button>
    </>
  );
};
```

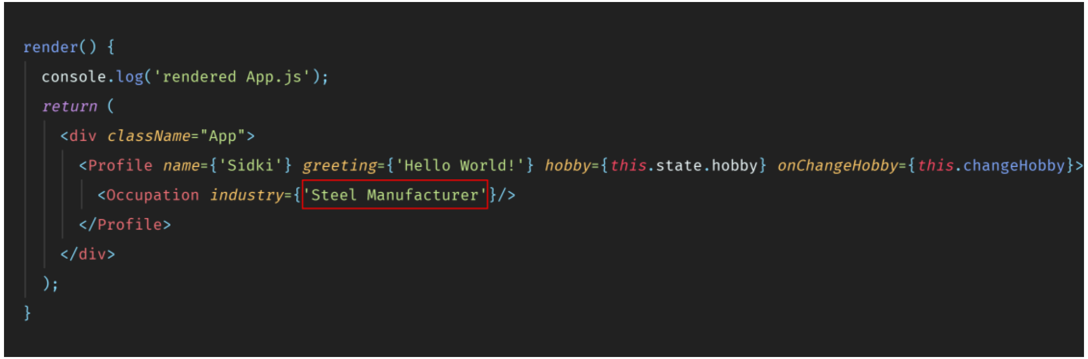


Nested component



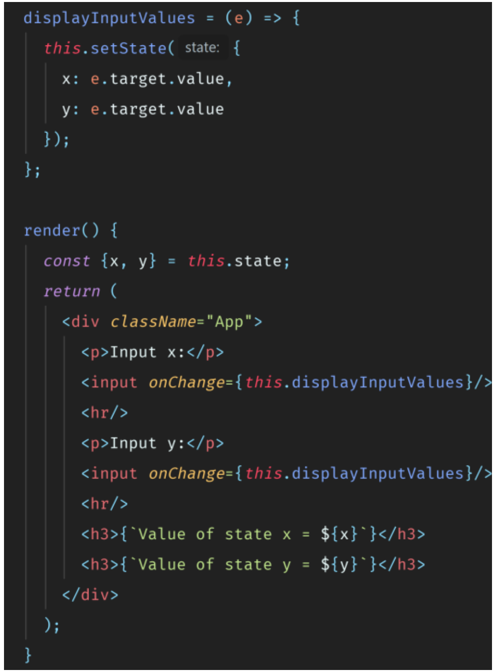
```
import React from 'react';
import PropTypes from 'prop-types';

const Occupation = ({industry}) => <p>{'I work in ' + industry}</p>;

Occupation.propTypes = {
  industry: PropTypes.string.isRequired
};

export default Occupation;
```

Nested component



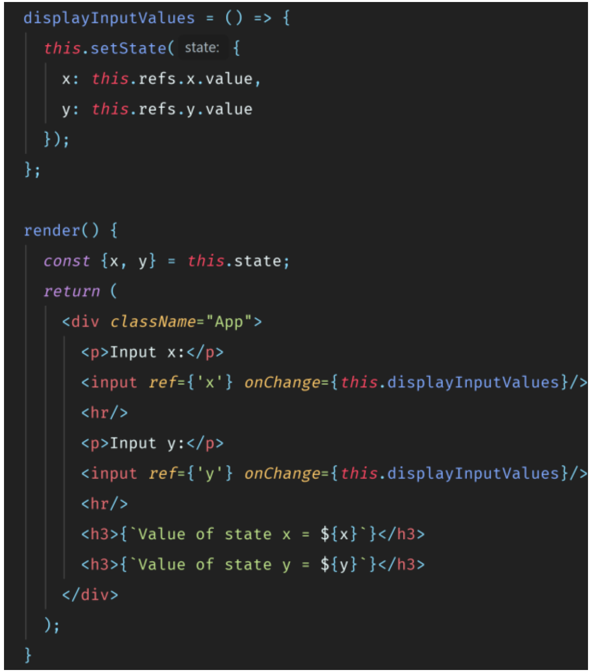
```
render() {
  console.log('rendered App.js');
  return (
    <div className="App">
      <Profile name={'Sidki'} greeting={'Hello World!'} hobby={this.state.hobby} onChangeHobby={this.changeHobby}>
        <Occupation industry={'Steel Manufacturer'}/>
      </Profile>
    </div>
  );
}
```

Referencing DOM node

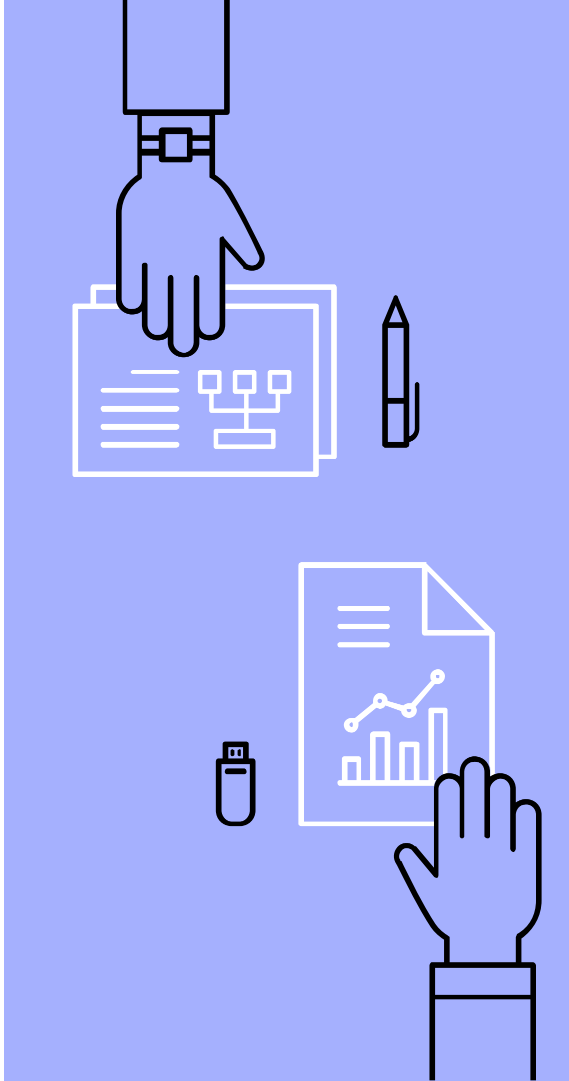


```
displayInputValues = (e) => {
  this.setState( state: {
    x: e.target.value,
    y: e.target.value
  });
};

render() {
  const {x, y} = this.state;
  return (
    <div className="App">
      <p>Input x:</p>
      <input onChange={this.displayInputValues}/>
      <hr/>
      <p>Input y:</p>
      <input onChange={this.displayInputValues}/>
      <hr/>
      <h3>{`Value of state x = ${x}`}</h3>
      <h3>{`Value of state y = ${y}`}</h3>
    </div>
  );
}
```

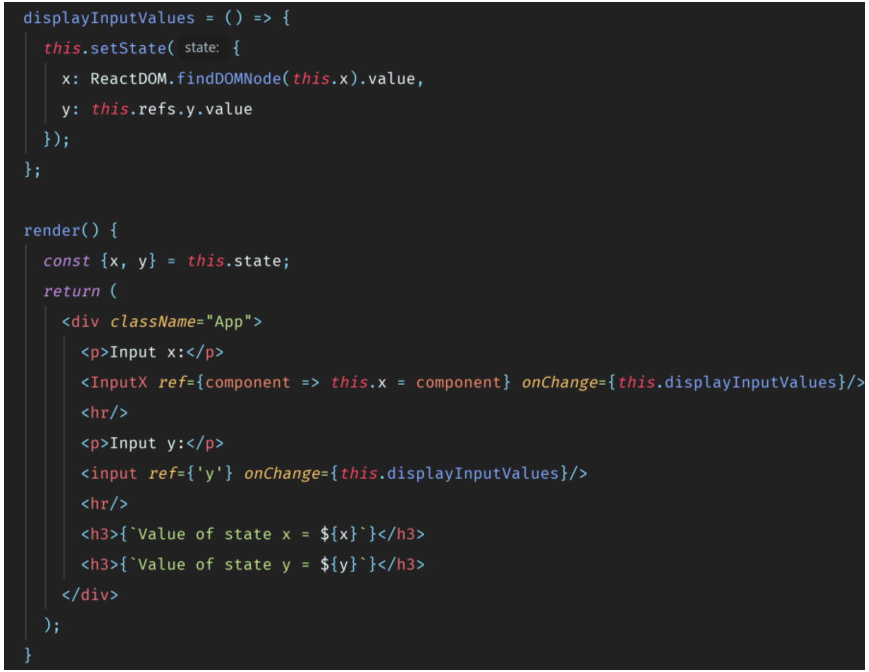
Referencing DOM node



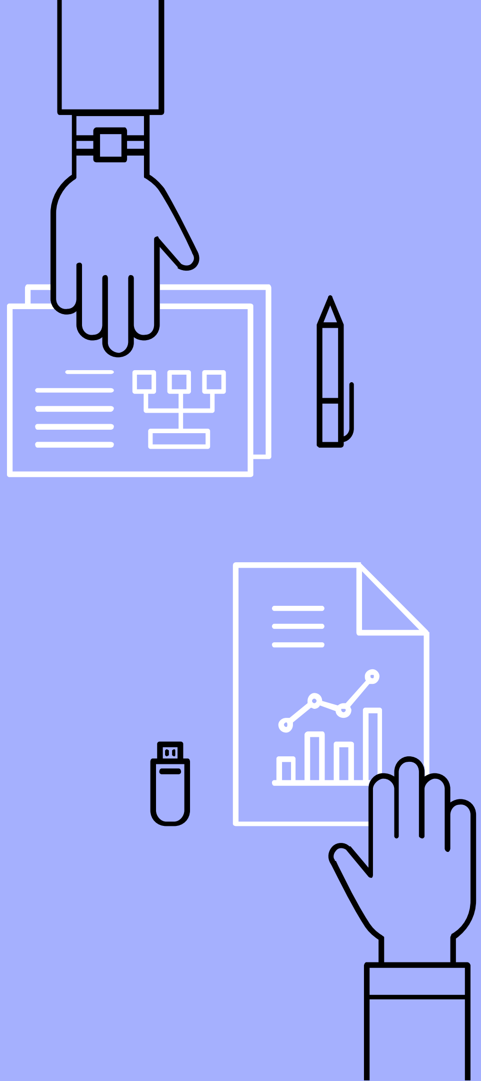
```
displayInputValues = () => {
  this.setState( state: {
    x: this.refs.x.value,
    y: this.refs.y.value
  });
};

render() {
  const {x, y} = this.state;
  return (
    <div className="App">
      <p>Input x:</p>
      <input ref={'x'} onChange={this.displayInputValues}/>
      <hr/>
      <p>Input y:</p>
      <input ref={'y'} onChange={this.displayInputValues}/>
      <hr/>
      <h3>{`Value of state x = ${x}`}</h3>
      <h3>{`Value of state y = ${y}`}</h3>
    </div>
  );
}
```



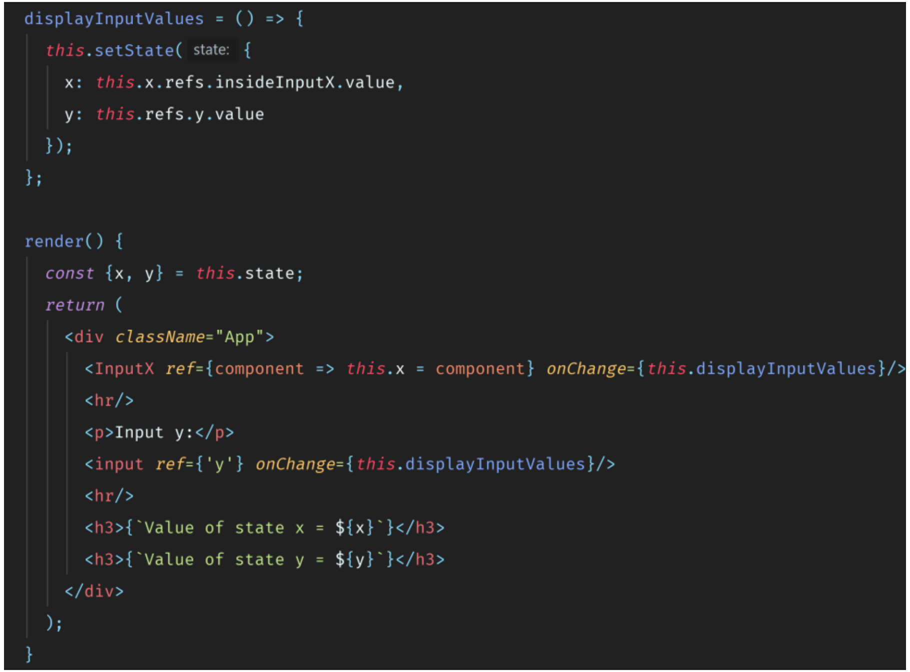
Referencing component



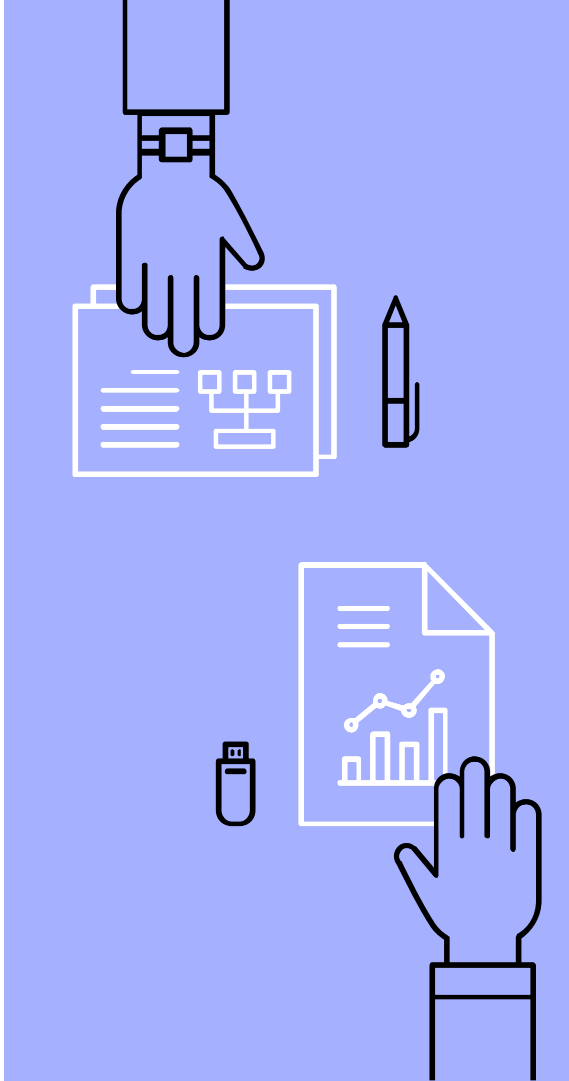
```
displayInputValues = () => {
  this.setState( state: {
    x: ReactDOM.findDOMNode(this.x).value,
    y: this.refs.y.value
  });
};

render() {
  const {x, y} = this.state;
  return (
    <div className="App">
      <p>Input x:</p>
      <InputX ref={component => this.x = component} onChange={this.displayInputValues}/>
      <hr/>
      <p>Input y:</p>
      <input ref={'y'} onChange={this.displayInputValues}/>
      <hr/>
      <h3>{`Value of state x = ${x}`}</h3>
      <h3>{`Value of state y = ${y}`}</h3>
    </div>
  );
}
```



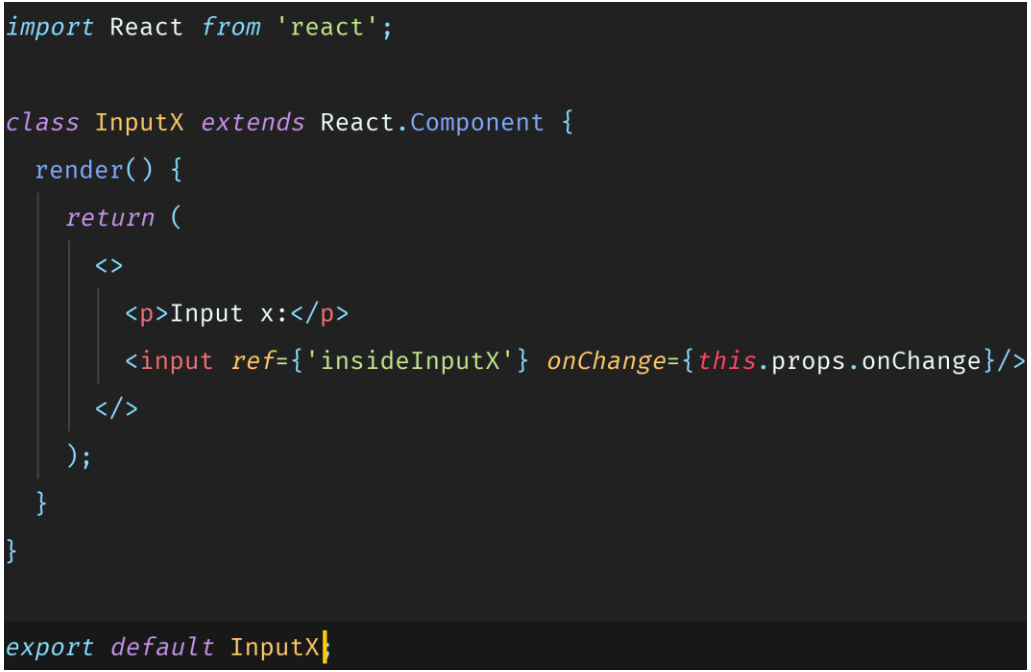
Refs component node



```
displayInputValues = () => {
  this.setState( state: {
    x: this.x.refs.insideInputX.value,
    y: this.refs.y.value
  });
};

render() {
  const {x, y} = this.state;
  return (
    <div className="App">
      <InputX ref={component => this.x = component} onChange={this.displayInputValues}/>
      <hr/>
      <p>Input y:</p>
      <input ref={'y'} onChange={this.displayInputValues}/>
      <hr/>
      <h3>{`Value of state x = ${x}`}</h3>
      <h3>{`Value of state y = ${y}`}</h3>
    </div>
  );
}
```



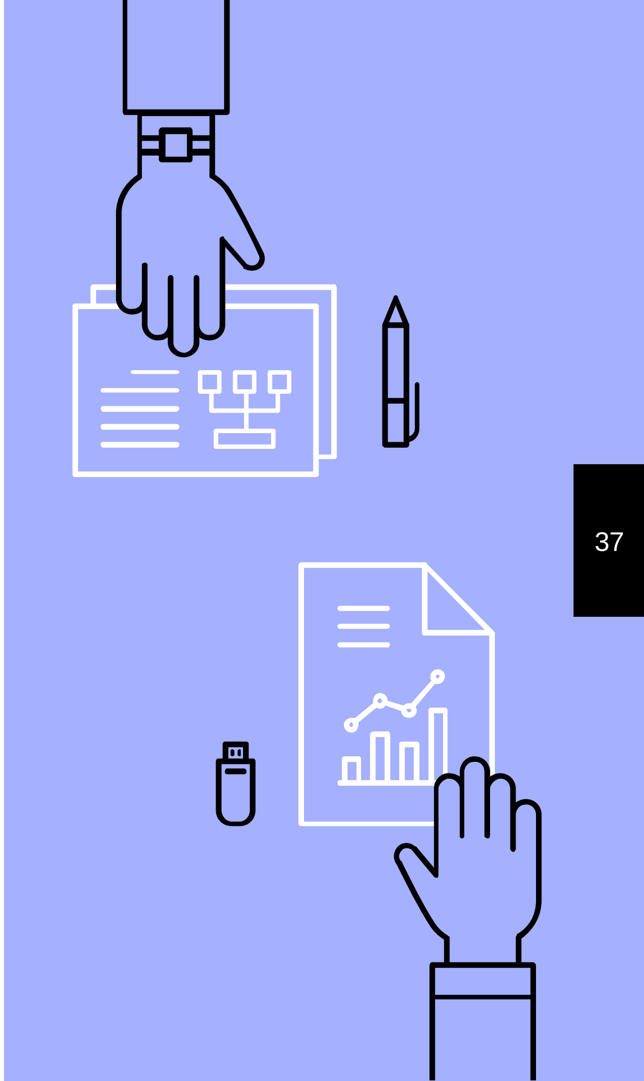
Refs component node

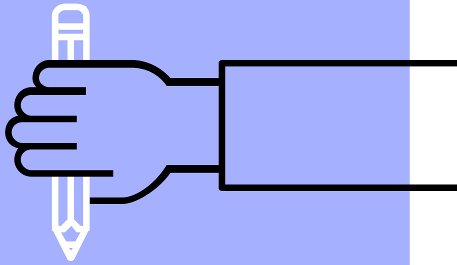


```
import React from 'react';

class InputX extends React.Component {
  render() {
    return (
      <>
        <p>Input x:</p>
        <input ref={'insideInputX'} onChange={this.props.onChange}/>
      </>
    );
  }
}

export default InputX
```

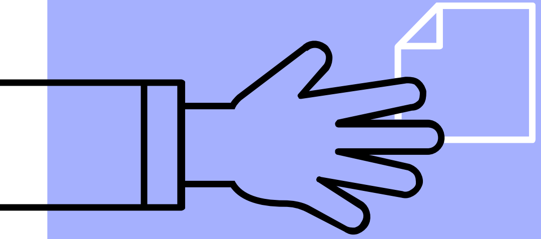
Warn!

- Do not overuse refs!
- See: https://reactjs.org/docs/refs-and-the-dom.html#dont-overuse-refs



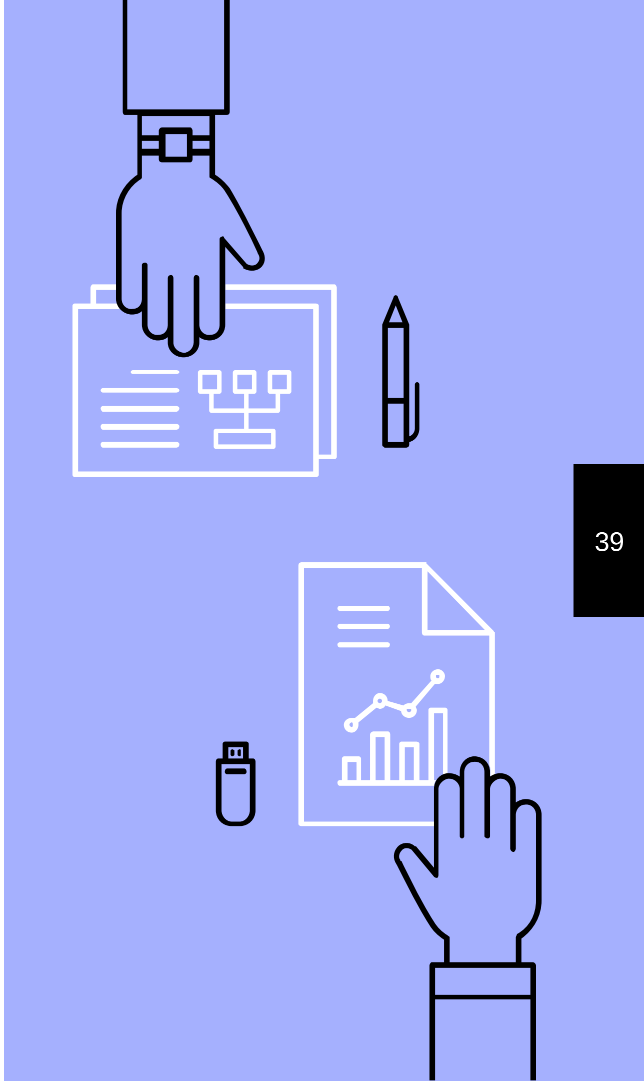


Life Cycle



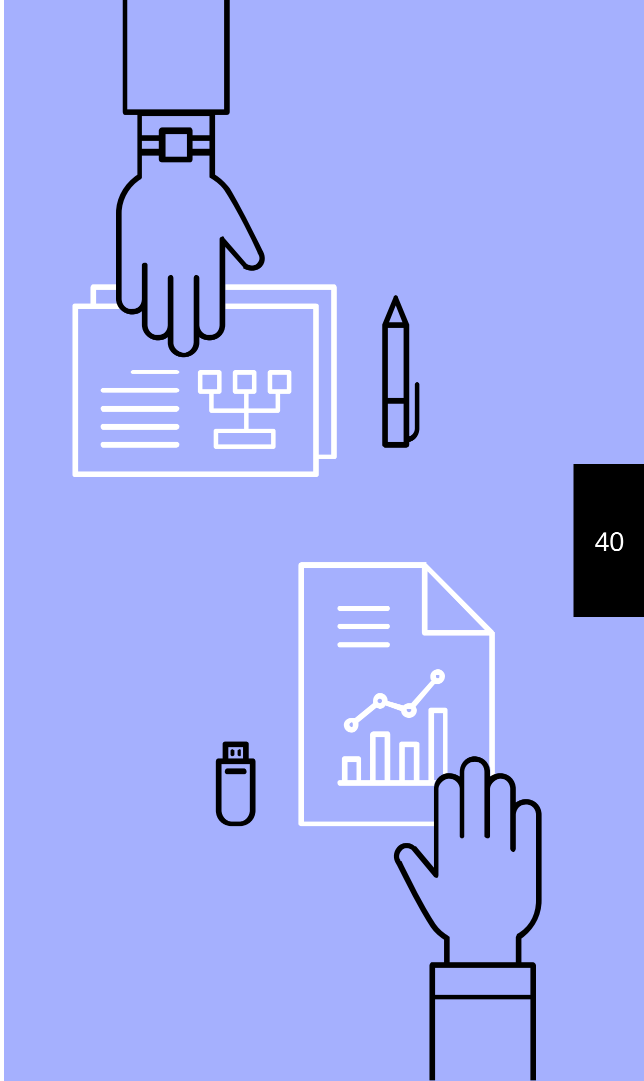
Commonly used

- constructor()
- ~~componentWillMount()~~
- render()
- componentDidMount()
- componentWillUnmount()

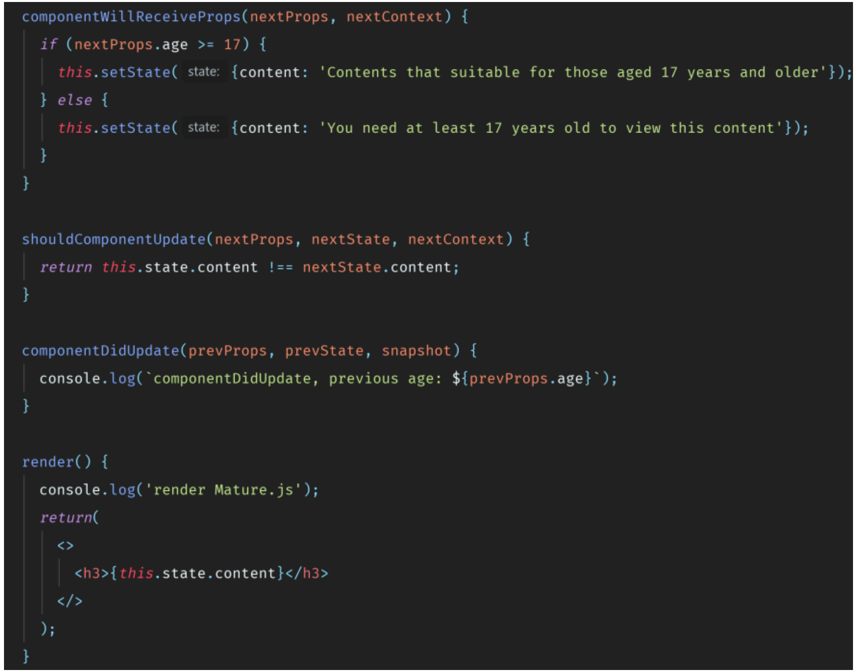


Rarely used

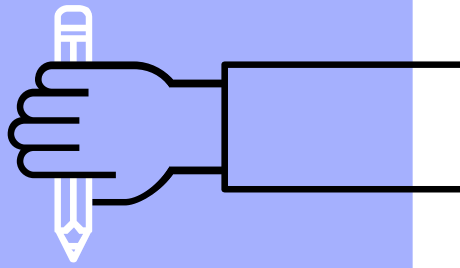
- componentWillReceiveProps()
- shouldComponentUpdate()
- componentDidUpdate()



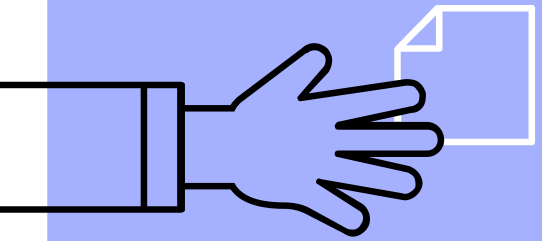
Other cycle



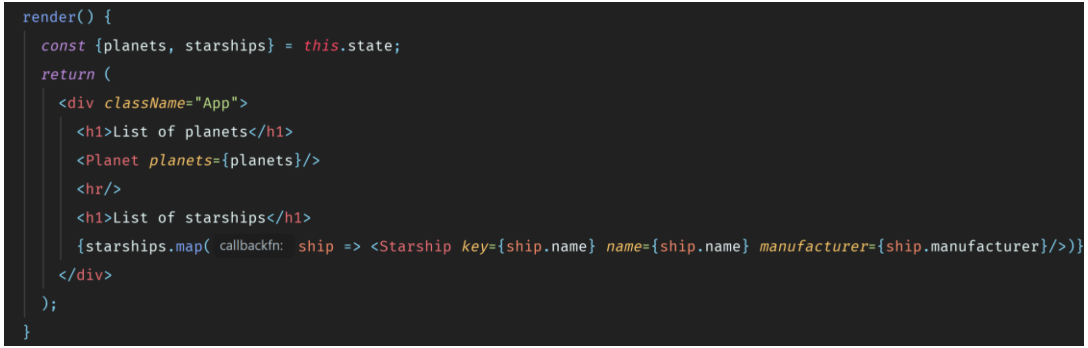
```
componentWillReceiveProps(nextProps, nextContext) {
  if (nextProps.age >= 17) {
    this.setState( state: {content: 'Contents that suitable for those aged 17 years and older'});
  } else {
    this.setState( state: {content: 'You need at least 17 years old to view this content'});
  }
}

shouldComponentUpdate(nextProps, nextState, nextContext) {
  return this.state.content !== nextState.content;
}

componentDidUpdate(prevProps, prevState, snapshot) {
  console.log(`componentDidUpdate, previous age: ${prevProps.age}`);
}

render() {
  console.log('render Mature.js');
  return(
    <>
      <h3>{this.state.content}</h3>
    </>
  );
}
```



Iterate Component

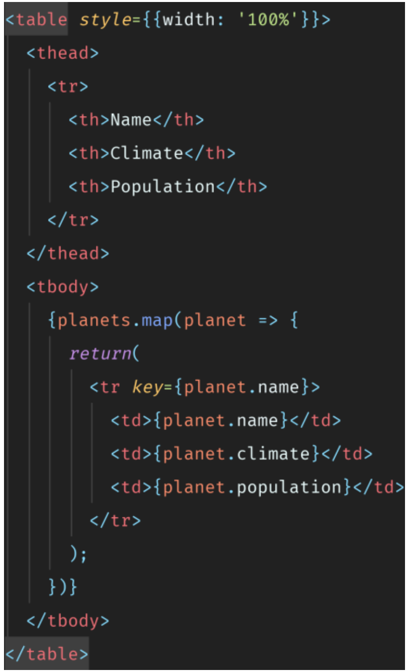


Use array.map()

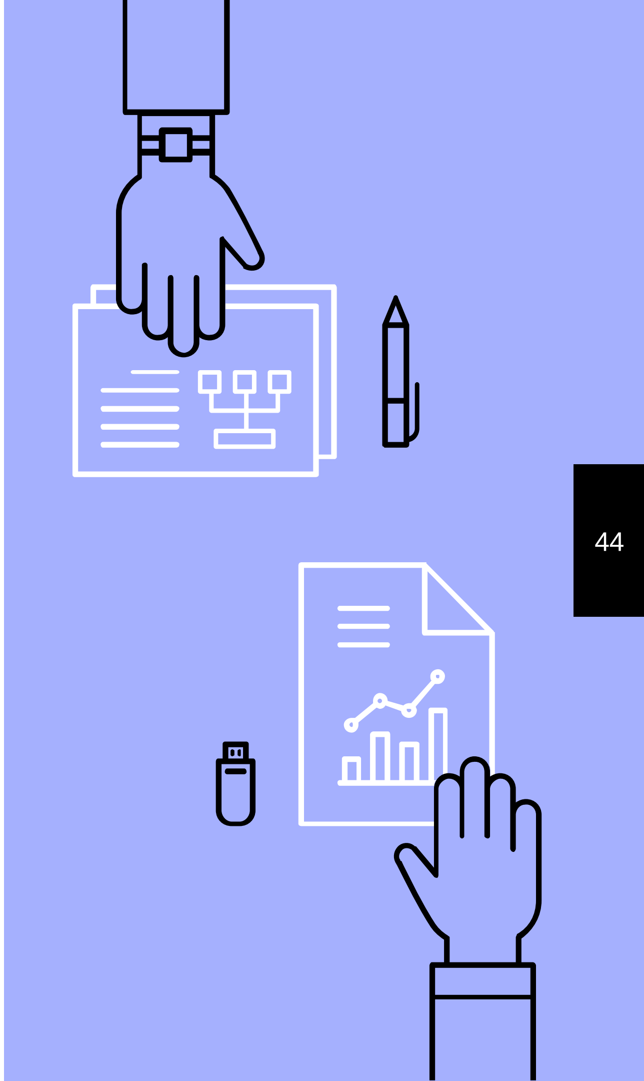


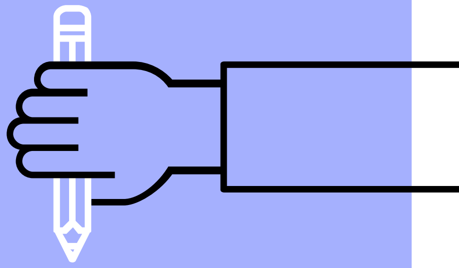
```
render() {
  const {planets, starships} = this.state;
  return (
    <div className="App">
      <h1>List of planets</h1>
      <Planet planets={planets}/>
      <hr/>
      <h1>List of starships</h1>
      {starships.map( callbackfn: ship => <Starship key={ship.name} name={ship.name} manufacturer={ship.manufacturer}/>)}
    </div>
  );
}
```

Planet.js

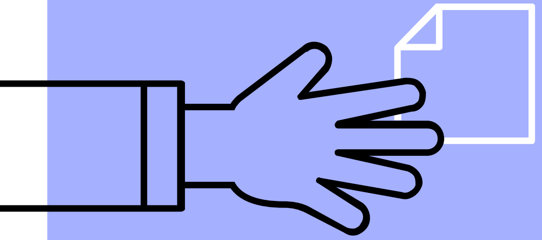


```
<table style={{width: '100%'}}>
  <thead>
    <tr>
      <th>Name</th>
      <th>Climate</th>
      <th>Population</th>
    </tr>
  </thead>
  <tbody>
    {planets.map(planet => {
      return(
        <tr key={planet.name}>
          <td>{planet.name}</td>
          <td>{planet.climate}</td>
          <td>{planet.population}</td>
        </tr>
      );
    })}
  </tbody>
</table>
```

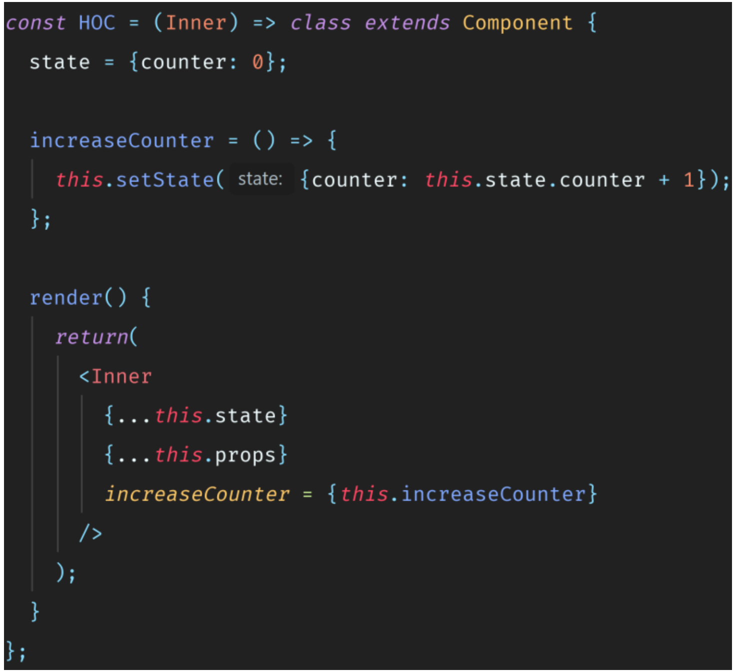




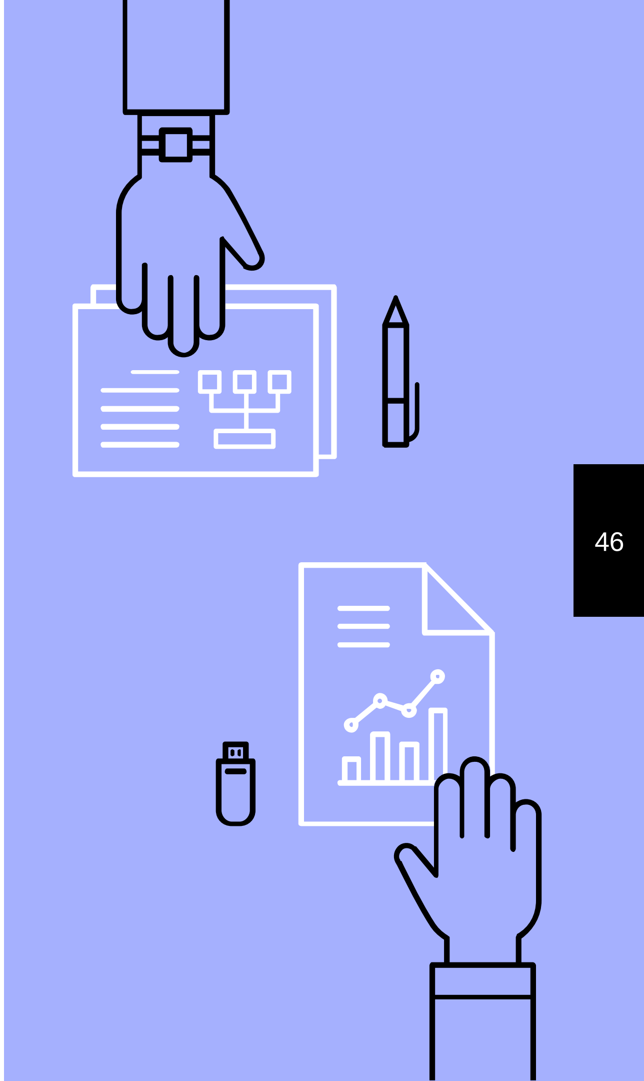
Higher Order Component



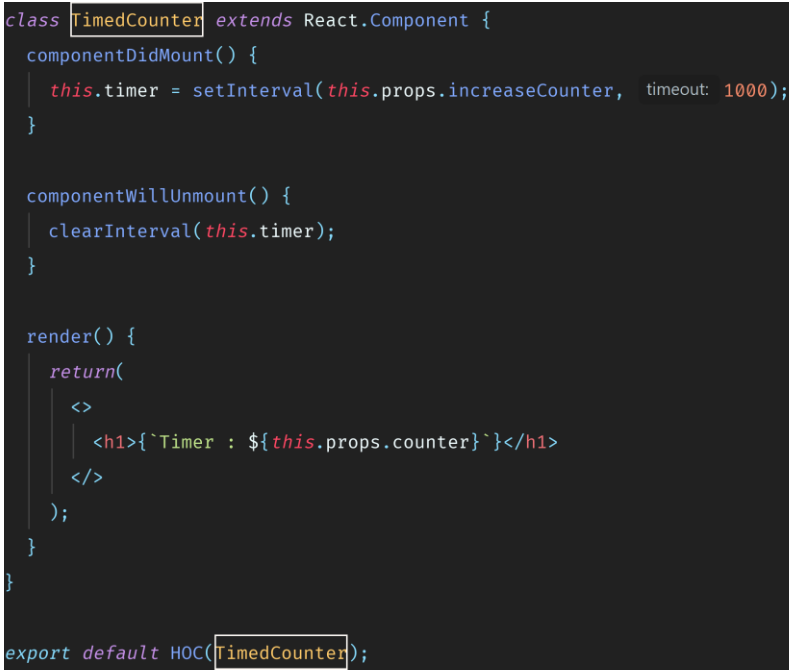
HOC



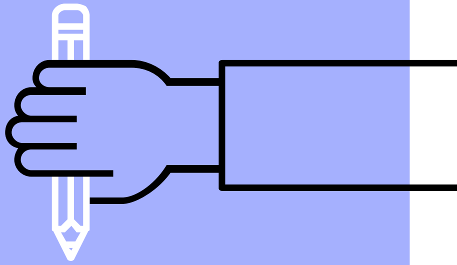
```
const HOC = (Inner) => class extends Component {
  state = {counter: 0};

  increaseCounter = () => {
    this.setState( state: {counter: this.state.counter + 1});
  };

  render() {
    return(
      <Inner
        {...this.state}
        {...this.props}
        increaseCounter = {this.increaseCounter}
      />
    );
  }
};
```



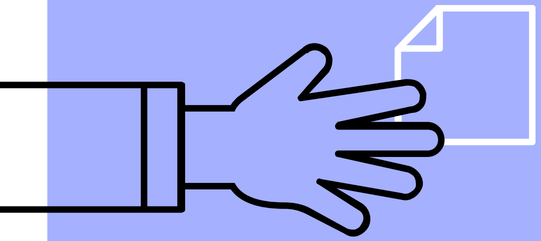
HOC

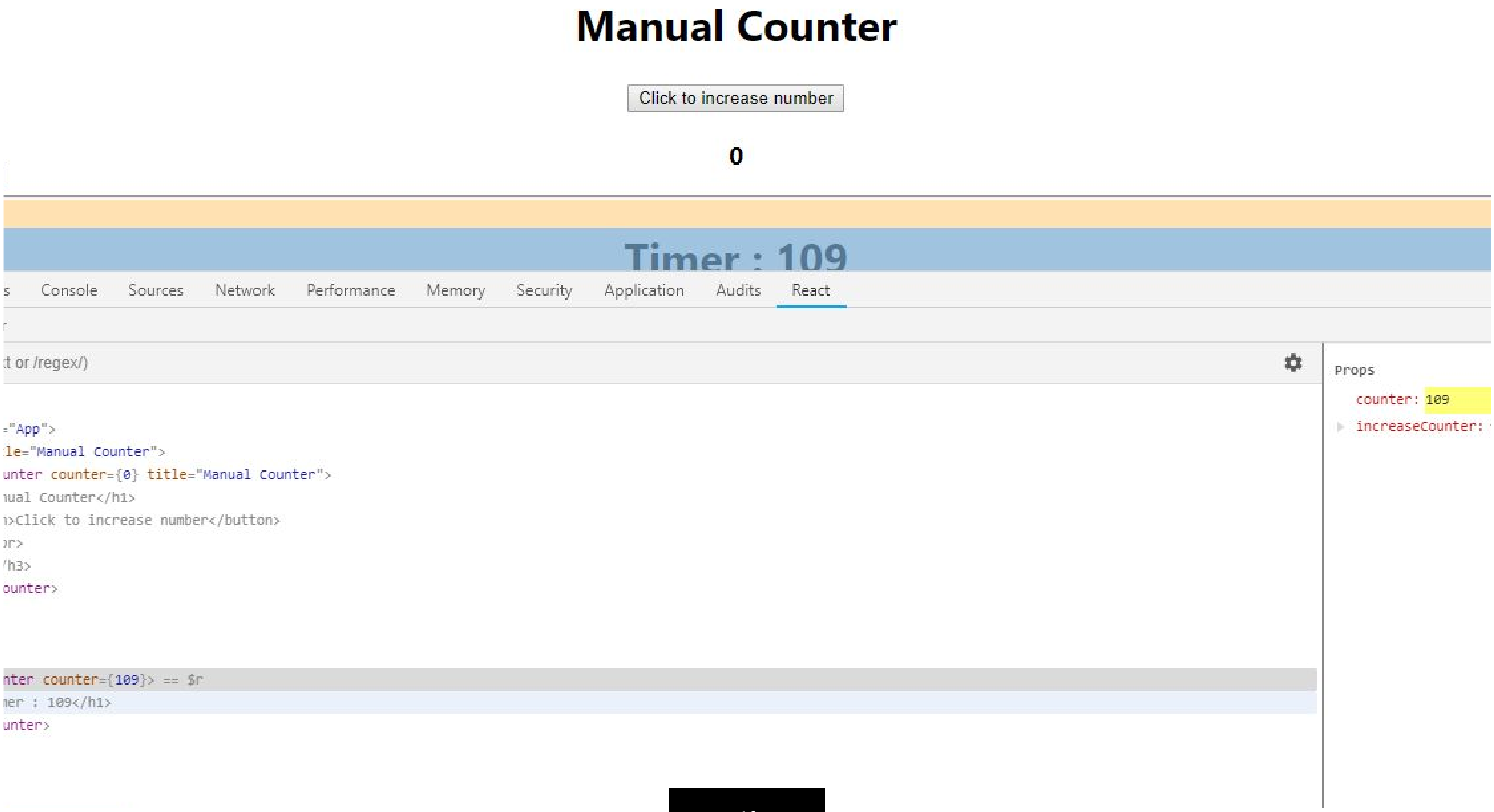


```
class TimedCounter extends React.Component {
  componentDidMount() {
    this.timer = setInterval(this.props.increaseCounter,  timeout: 1000);
  }

  componentWillUnmount() {
    clearInterval(this.timer);
  }

  render() {
    return(
      <>
        <h1>{`Timer : ${this.props.counter}`}</h1>
      </>
    );
  }
}

export default HOC(TimedCounter);
```



Dev Tools





Manual Counter

```
="App">
le="Manual Counter">
unter counter={0} title="Manual Counter">
ual Counter</h1>
>Click to increase number</button>
r>
/h3>
ounter>


nter counter={109}> == $r
ner : 109</h1>
unter>
```

```
counter: 109
increaseCounter:
```

Thank You