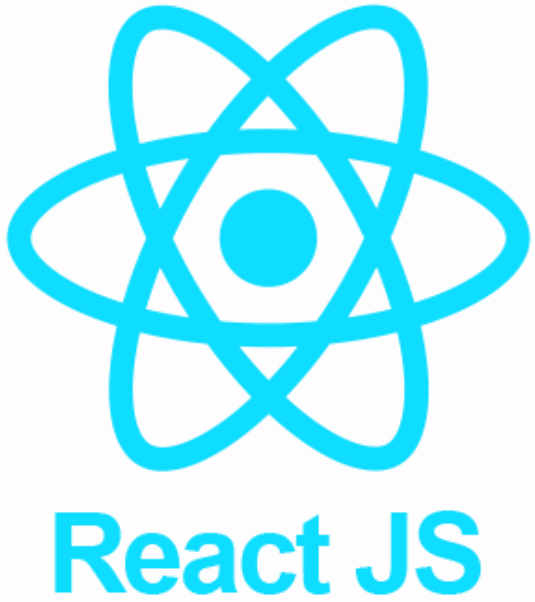
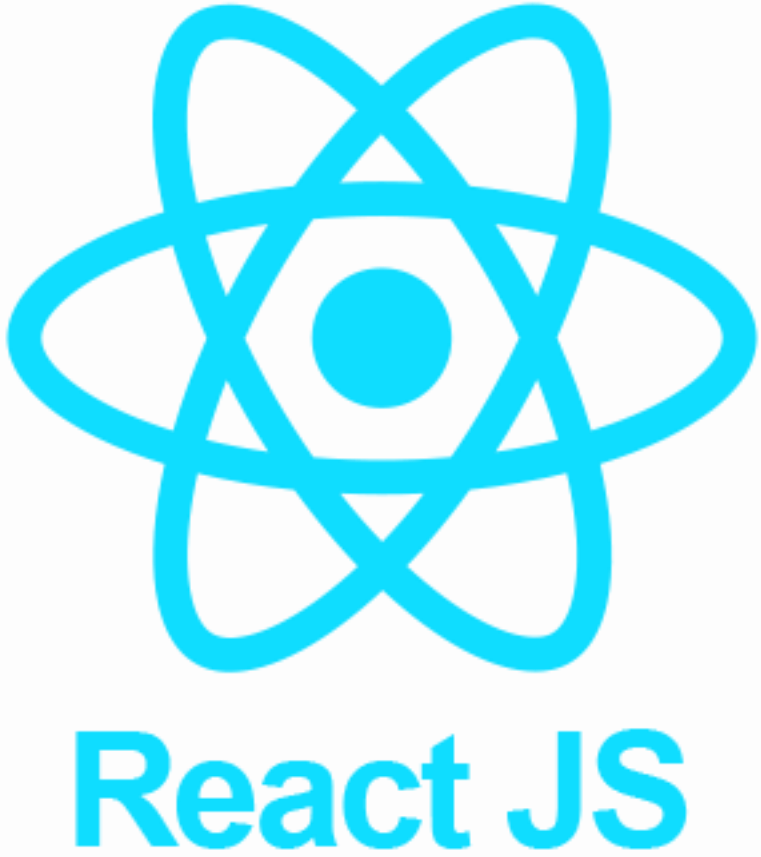
sourcemind

REACT JS

INTRODUCTION



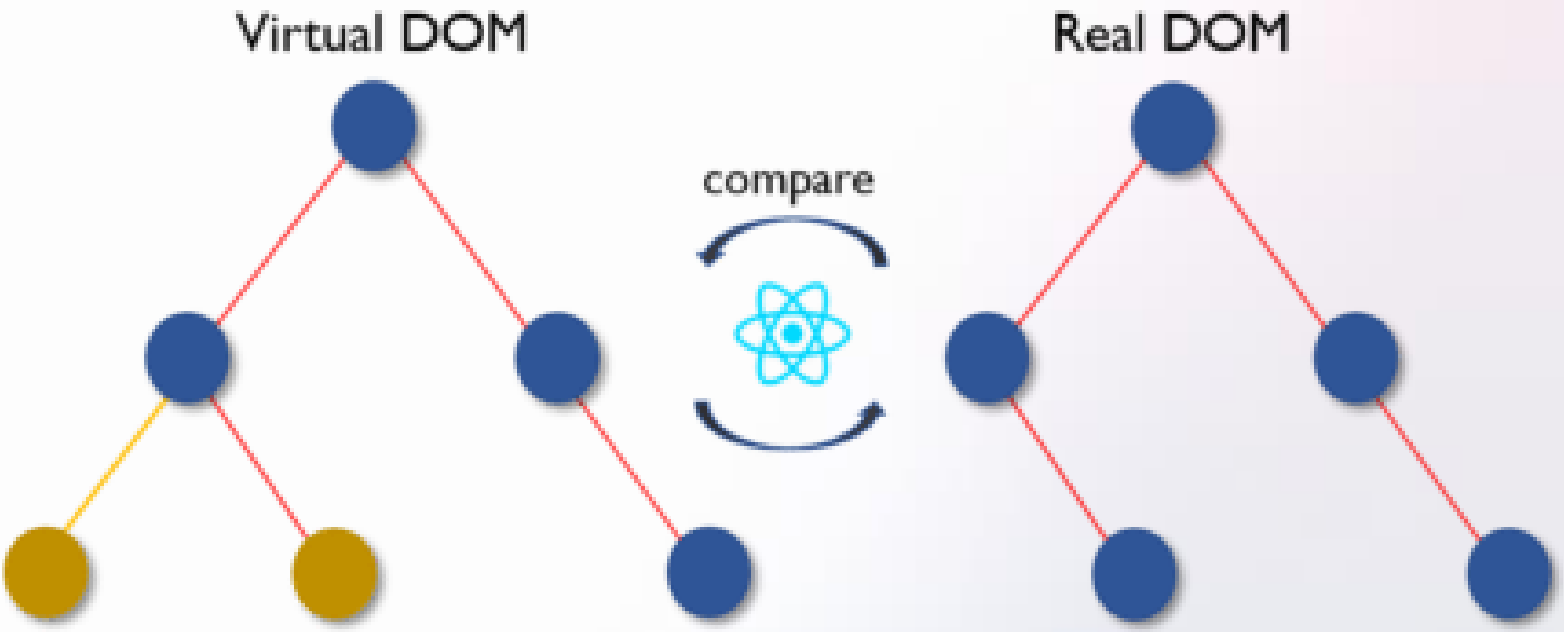


A JavaScript library for building user interfaces

React makes it painless to create interactive UIs.
Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

sourcemind

Virtual DOM vs Real DOM



Installing React

```
npx create-react-app my-app
cd my-app
npm start
```

Component

sourcemind

React DOM

What is the DOM?

The DOM (Document Object Model) represents the web page as a tree structure. Any piece of HTML that we write is added as a node, to this tree.

What is ReactDOM?

ReactDOM is **a package that provides DOM specific methods that can be used at the top level of a web app to enable an efficient way of managing DOM elements of the web page**. ReactDOM provides the developers with an API containing the following methods and a few more.07 հնվ, 2021 թ.

sourcemind

JSX

```
const element = <h1>Hello, world!</h1>;
```

It is called JSX, and it is a syntax extension to JavaScript. We recommend using it with React to describe what the UI should look like. JSX may remind you of a template language, but it comes with the full power of JavaScript.

First Component

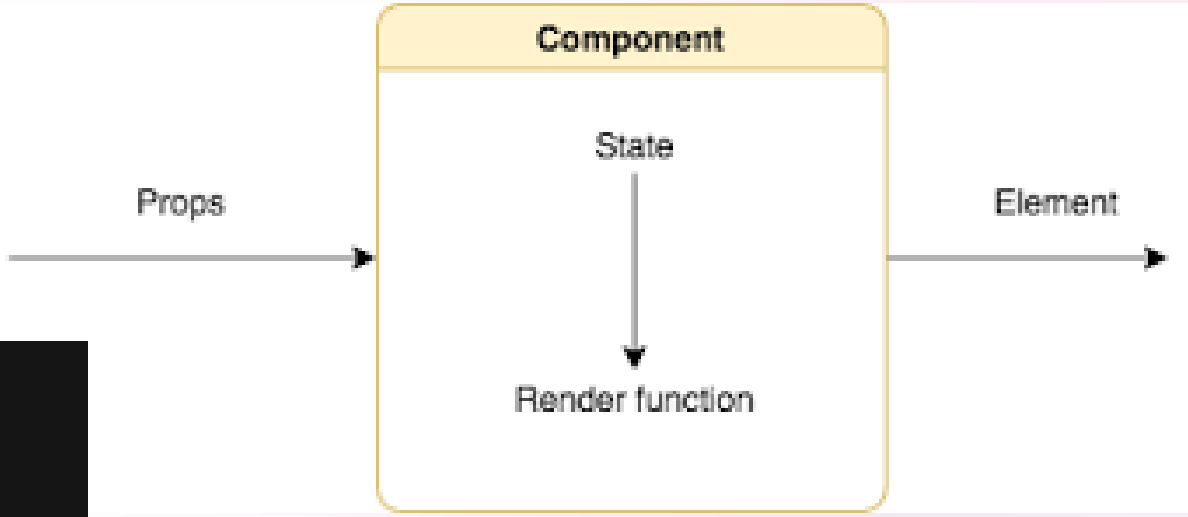
```jsx
import React from "react";

class MyFirstComponent extends React.Component {
  render() {
    return <h1>Sourcemind React Course </h1>;
  }
}

export default MyFirstComponent;
```

```jsx
<MyFirstComponent />
```

Sourcemind React Course

sourcemind

State, Props



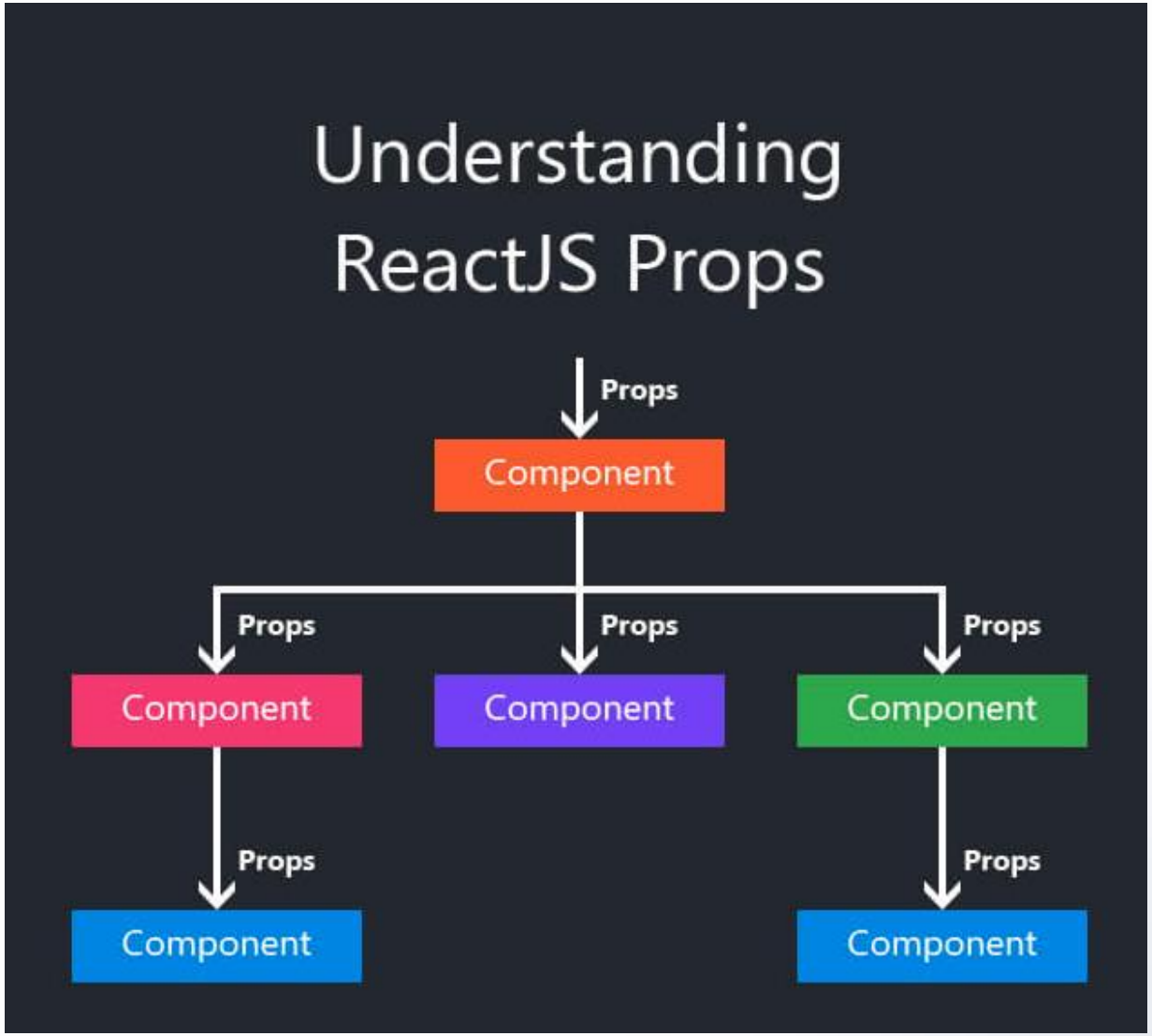

```
import React from "react";

class MyFirstComponent extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 1
    };
  }
  render() {
    return <h1>Sourcemind React Course {this.state.count} </h1>;
  }
}

export default MyFirstComponent;
```

```
constructor(props) {
  super(props);
  this.state = {
    count: 1
  };
}
```

```
<div>
  <h1>Sourcemind React Course {this.state.count} </h1>
  <button
    onClick={() => {
      this.setState({ count: this.state.count + 1 });
    }}
  >
    Next
  </button>
</div>
```

sourcemind

State, Props




```
<MyFirstComponent count={courceCount} />
```

```
const count = this.props.count;
```

sourcemind

TASKS

1. Implement same functional using props-state
2. Create Counter Page 

Render Array Elements

```
constructor(props) {
  super(props);
  this.state = {
    users: [
      { name: "User1", id: 1 },
      { name: "User2", id: 2 }
    ]
  };
}
render() {
  return (
    <div>
      User List
      <ol>
        {this.state.users.map((user) => {
          return <li key={user.id}>{user.name}</li>;
        })}
      </ol>
    </div>
```

User List

1. User1
2. User2

TASKS

1. Create a User List where user can
   1.1 Add user
   1.2 Remove user