React JS

ADAL DÍAZ FARIÑA
JORGE QUINTANA GARCÍA

01.

INTRODUCTION

➔ Javascript Frameworks

➔ Most used Frameworks

- ✓ Flexible
- ✓ High performance
- ✓ For small application development
- ✓ Easy to learn
- ✓ Light

- ✗ Poor community
- ✗ Does not work properly in large applications

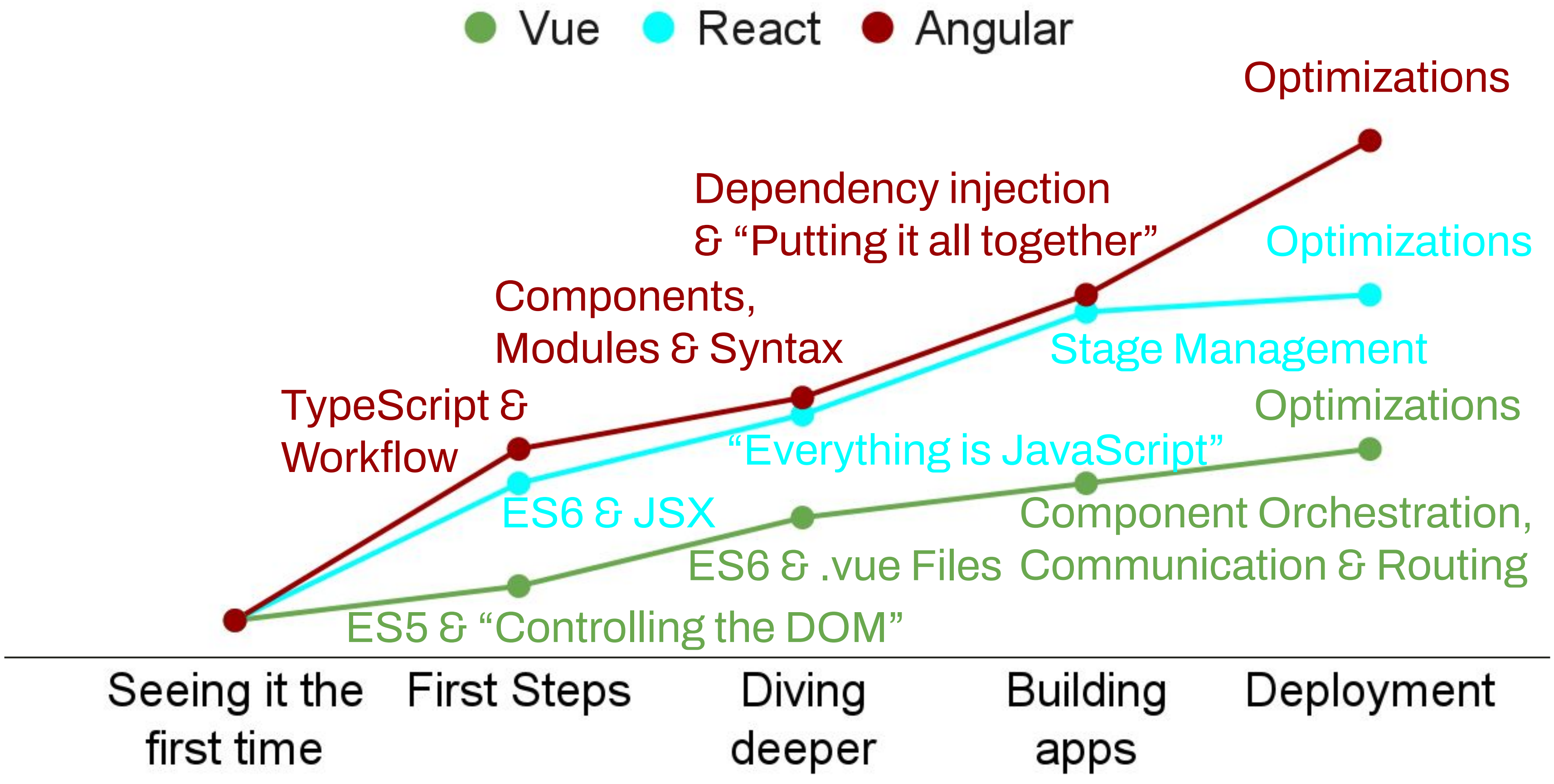
- ✓ Flexible
- ✓ High performance
- ✓ For small and large application development
- ✓ Easy to learn
- ✓ Great community supported by Facebook

- ✗ It has no fixed directory structure so projects can be organised in different ways and can be chaotic.

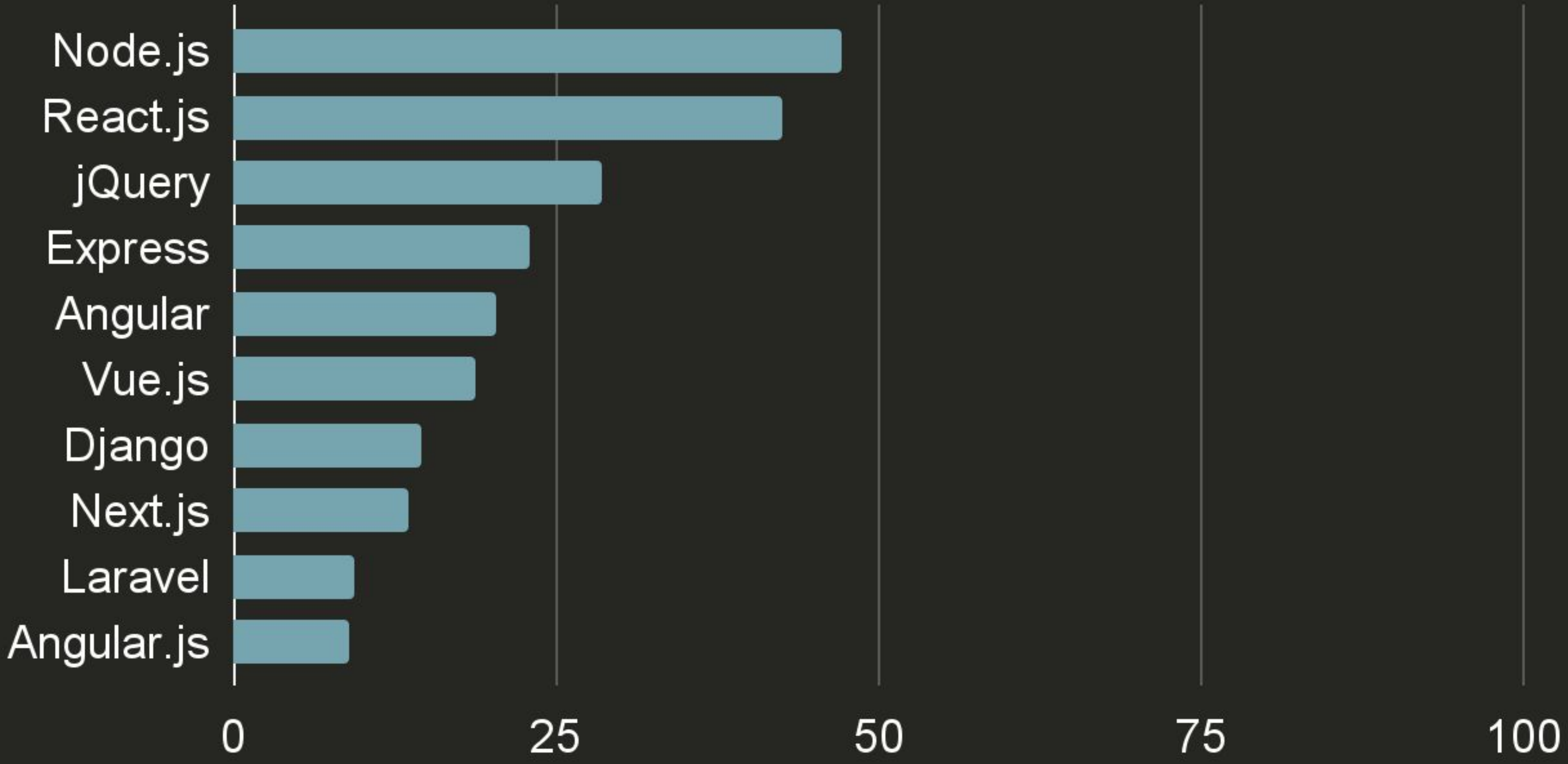
- ✓ For large application development
- ✓ Great community supported by Google
- ✓ Organised projects
- ✓ A lot of tools and template including

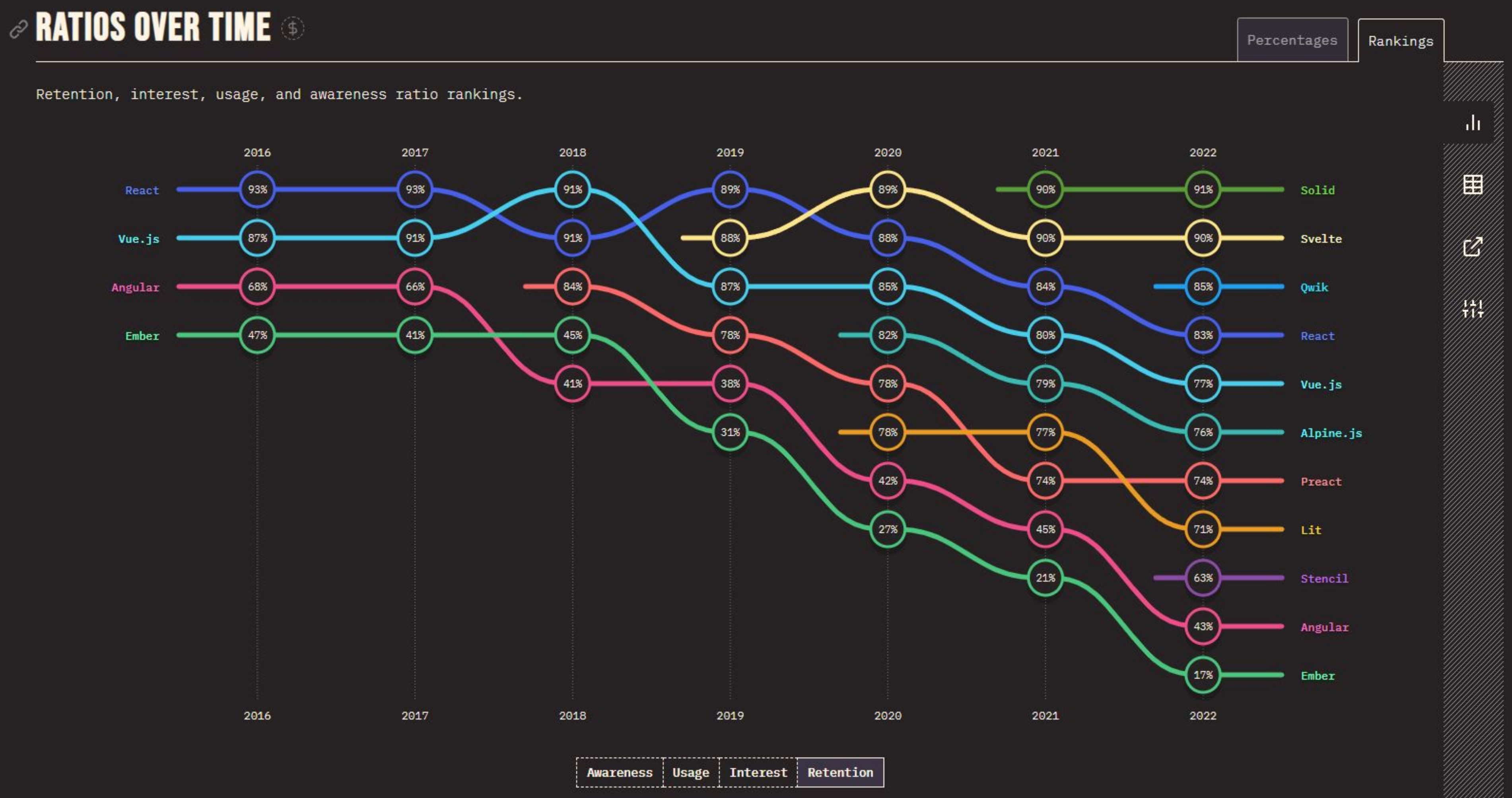
- ✗ Heavy
- ✗ Difficult to learn
- ✗ Rigid

(Possible) Learning Curve



Most common web technologies used by Professional





RATIOS OVER TIME

Retention, interest, usage, and awareness ratio rankings.

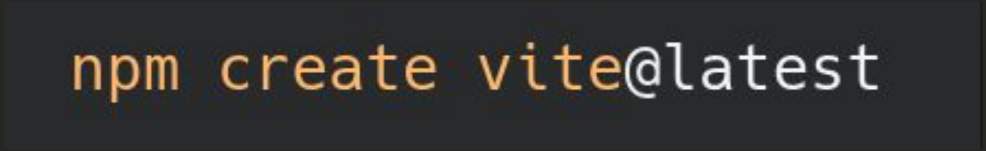
02.

BUILD TOOLS

→ Build tools for a React Project

VITE

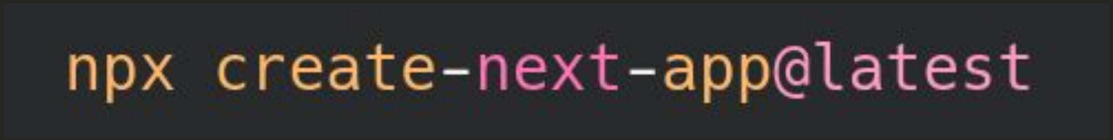
Vite is a build tool that aims to provide a faster and leaner development experience for modern web projects.



```
npm create vite@latest
```

NEXT JS

Next.js enables you to create full-stack web applications



```
npx create-next-app@latest
```

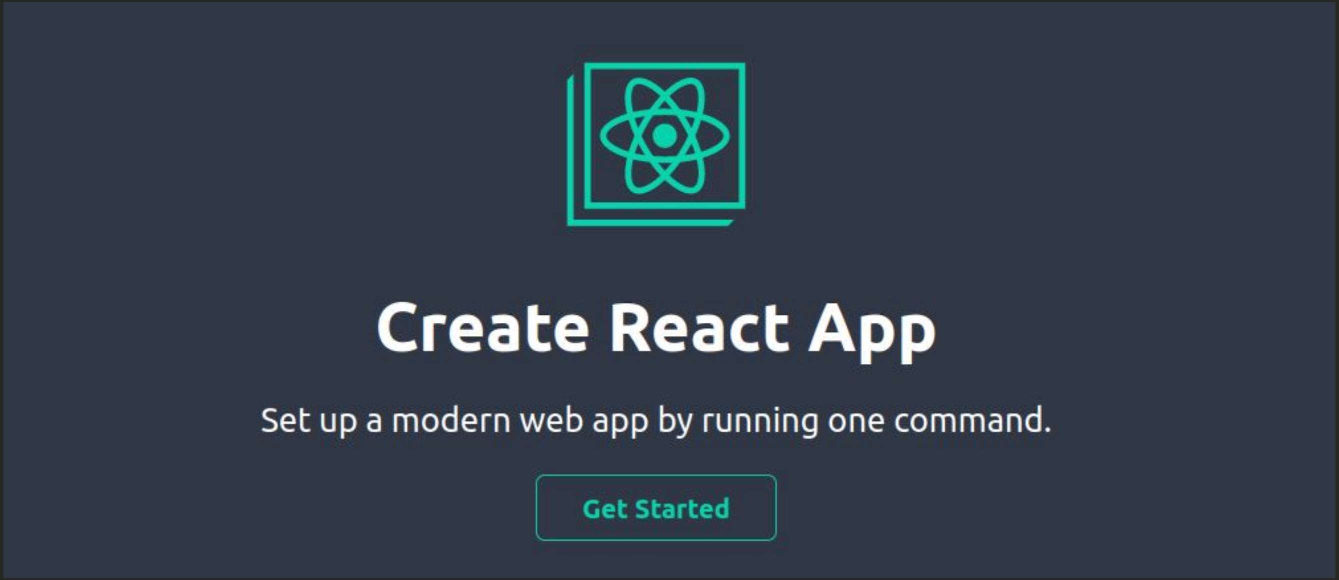
NX

REMIX

ASTRO

GATSBY

❌ ¡create react app is deprecated!



Create React App

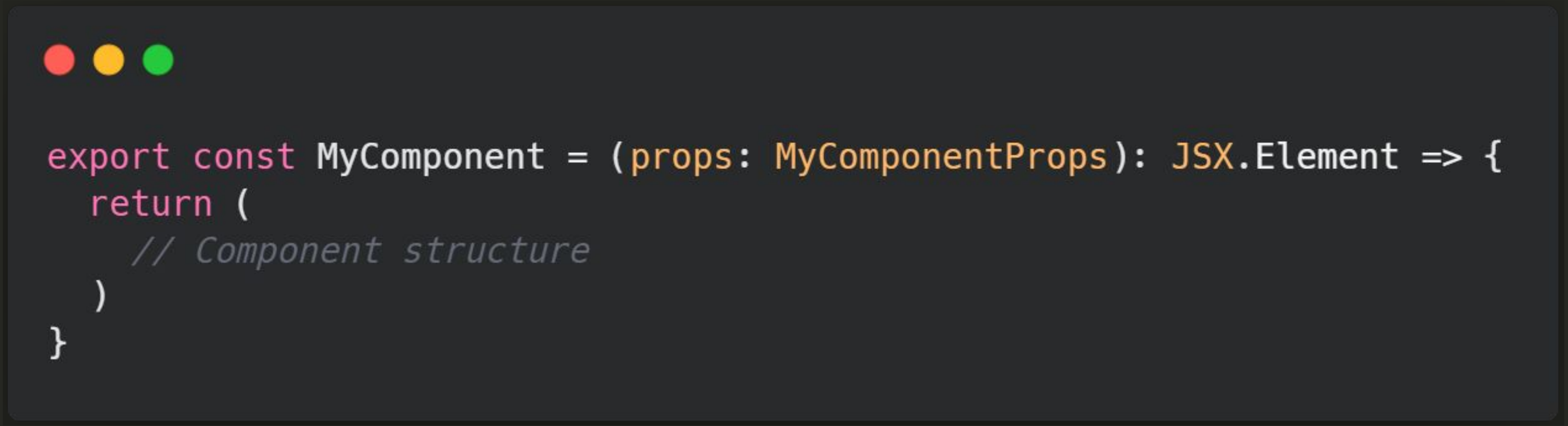
Set up a modern web app by running one command.

03.

COMPONENTS

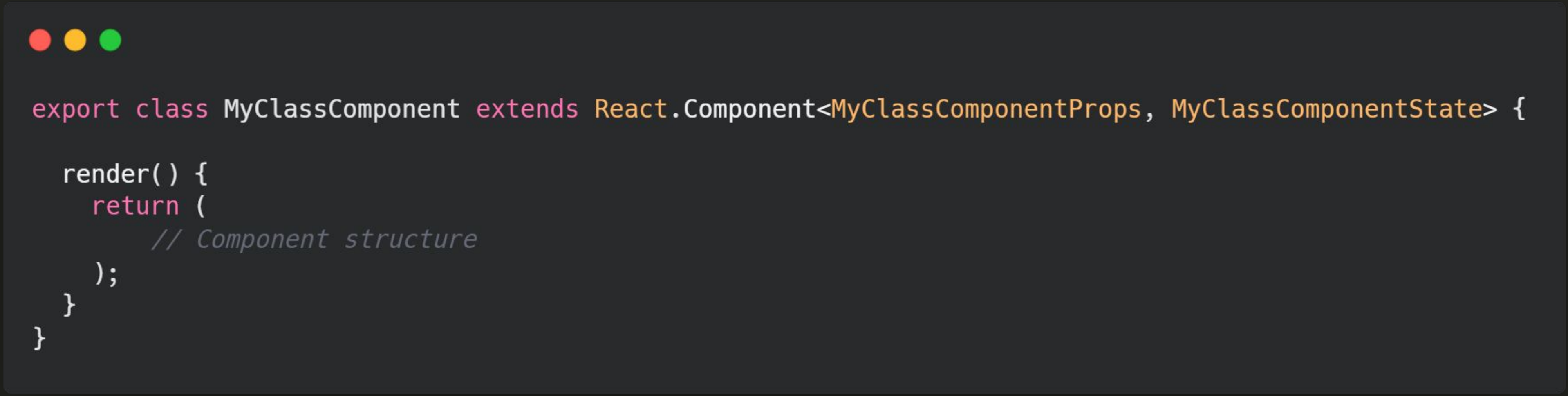
- → Functional component
- → Class component
- → How to create a component

REACT FUNCTIONAL COMPONENT



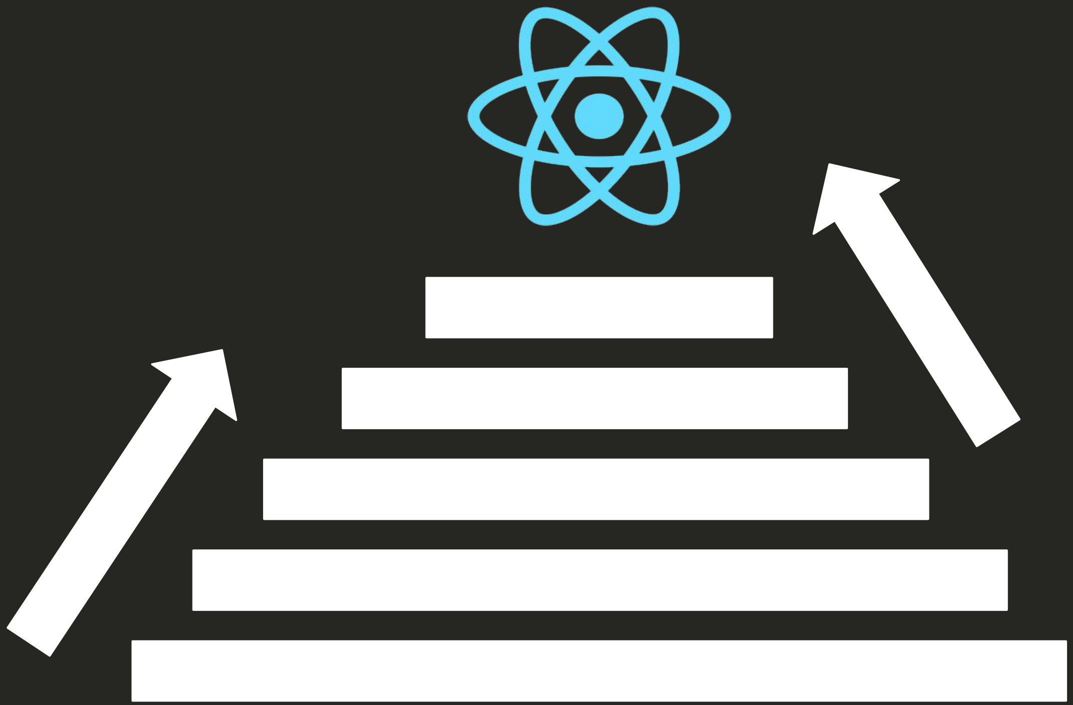
```
export const MyComponent = (props: MyComponentProps): JSX.Element => {
  return (
    // Component structure
  )
}
```

REACT CLASS COMPONENT



```
export class MyClassComponent extends React.Component<MyClassComponentProps, MyClassComponentState> {

  render() {
    return (
        // Component structure
    );
  }
}
```

HELLO WORLD

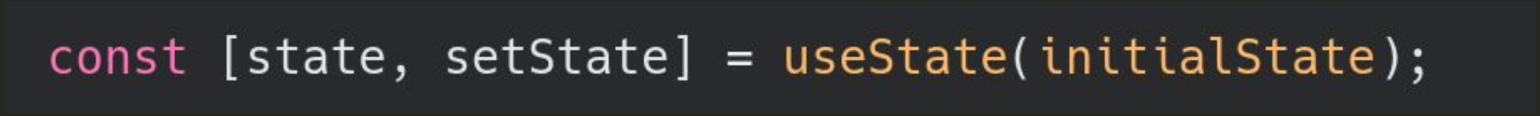


04.

HOOKS

- → useState
- → useEffect
- → useContext
- → Custom hook

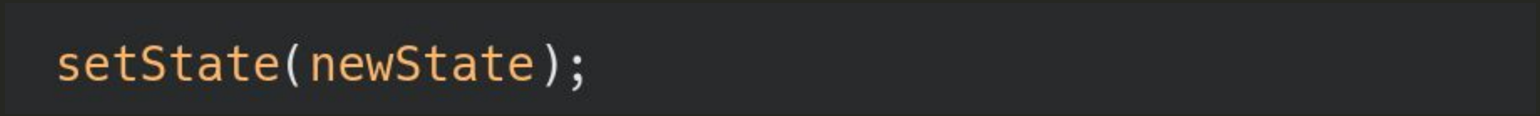
useState



```
const [state, setState] = useState(initialState);
```

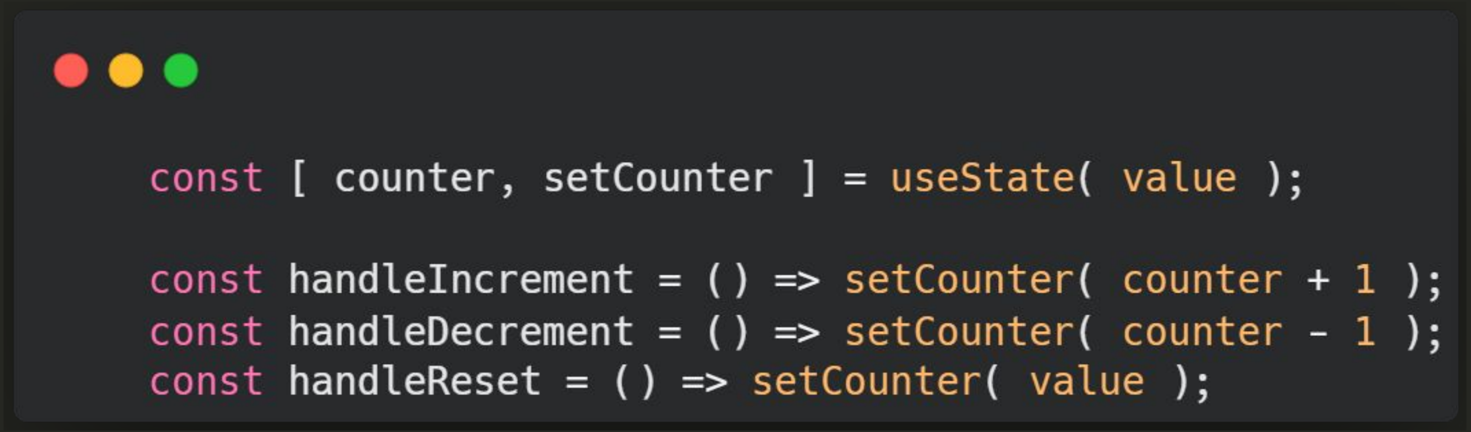
Returns a value with state and a function to update it.

The setState function is used to update the state. It accepts a new state value and queues a new rendering of the component.



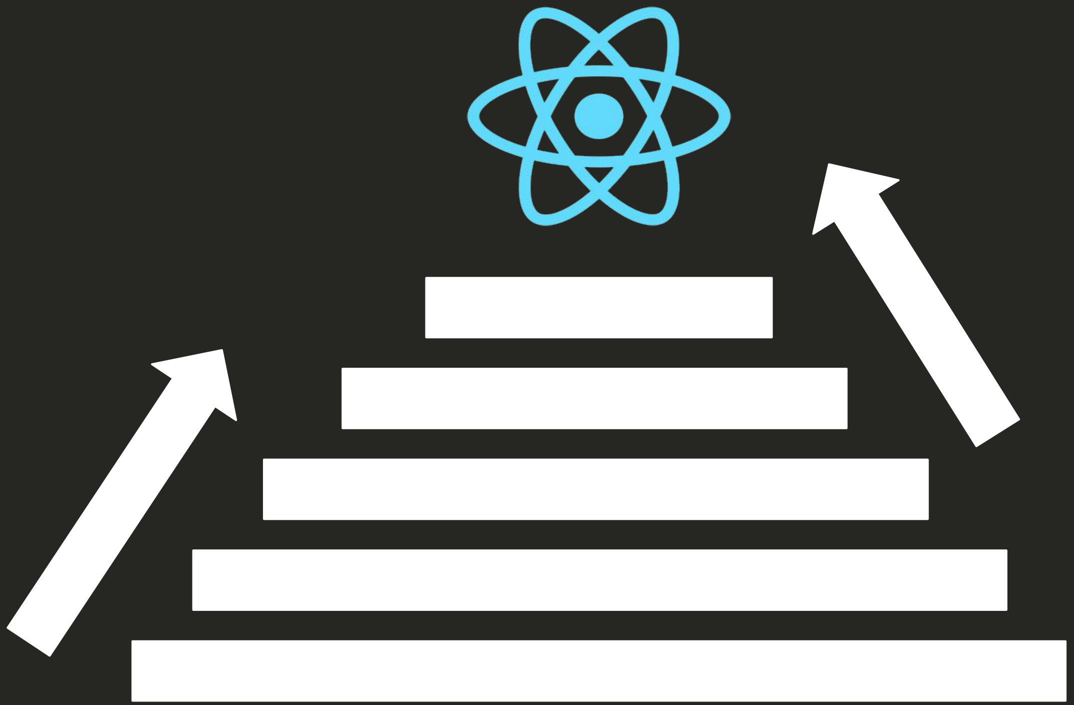
```
setState(newState);
```

Counter example



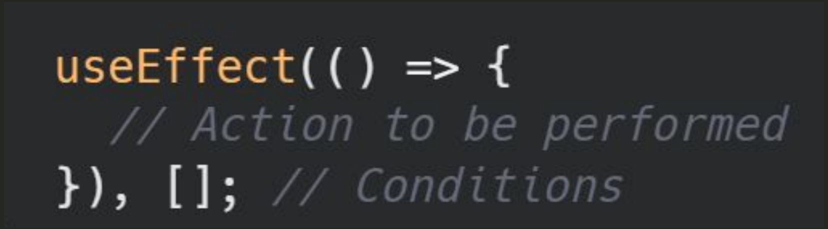
```
const [ counter, setCounter ] = useState( value );

const handleIncrement = () => setCounter( counter + 1 );
const handleDecrement = () => setCounter( counter - 1 );
const handleReset = () => setCounter( value );
```

TIC TAC TOE EXAMPLES



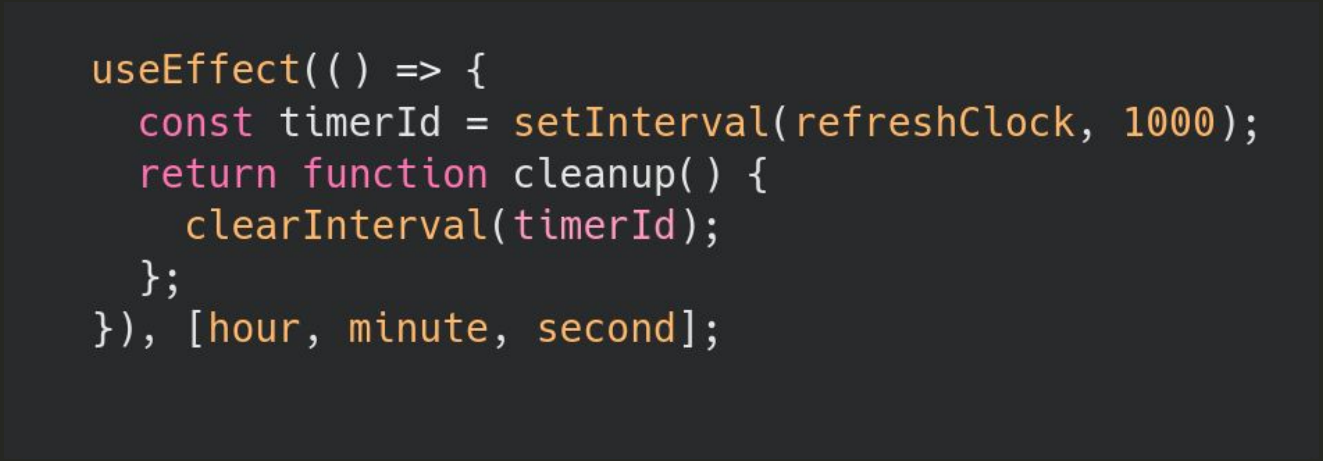
useEffect

useEffect allows us to make a number of changes to an already rendered component. By default the action is performed after each render but, choose to execute it only when certain values change.



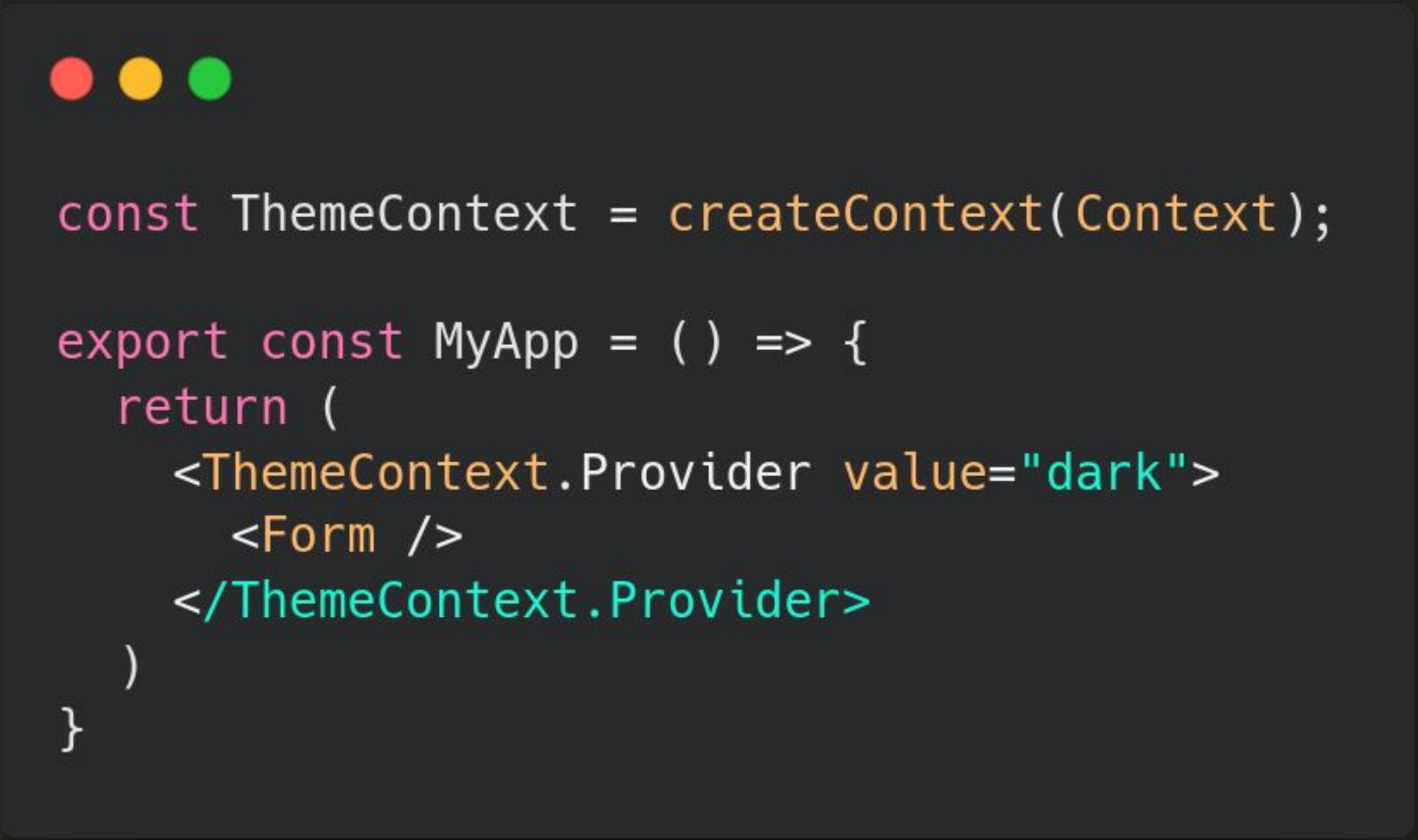
```
useEffect(() => {
  // Action to be performed
}), []; // Conditions
```

Clock example

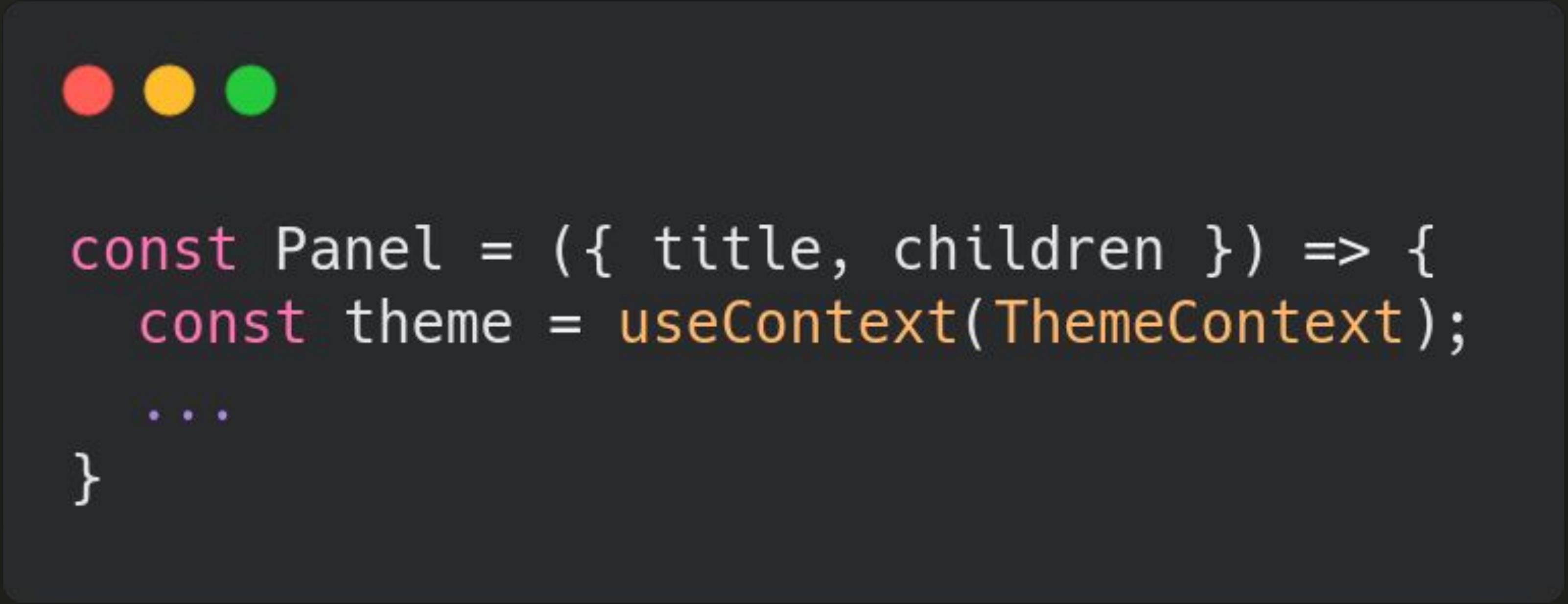


```
useEffect(() => {
  const timerId = setInterval(refreshClock, 1000);
  return function cleanup() {
    clearInterval(timerId);
  };
}), [hour, minute, second];
```

useContext



```
const ThemeContext = createContext(Context);

export const MyApp = () => {
  return (
    <ThemeContext.Provider value="dark">
      <Form />
    </ThemeContext.Provider>
  )
}
```



```
const Panel = ({ title, children }) => {
  const theme = useContext(ThemeContext);
  ...
}
```

Other hooks

01

useReducer

is a React Hook that lets you add a reducer to your component.

02

useCallback

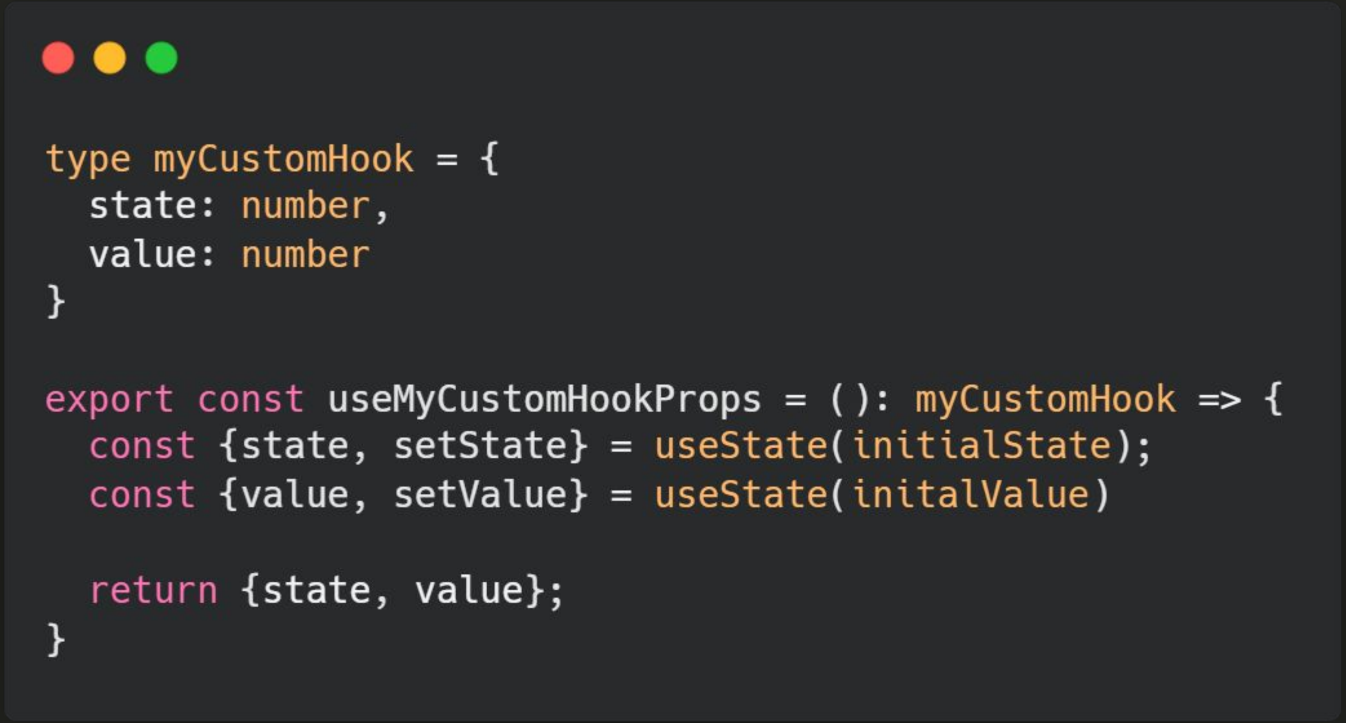
is a React Hook that lets you cache a function definition between re-renders.

03

useTransition

is a React Hook that lets you update the state without blocking the UI.

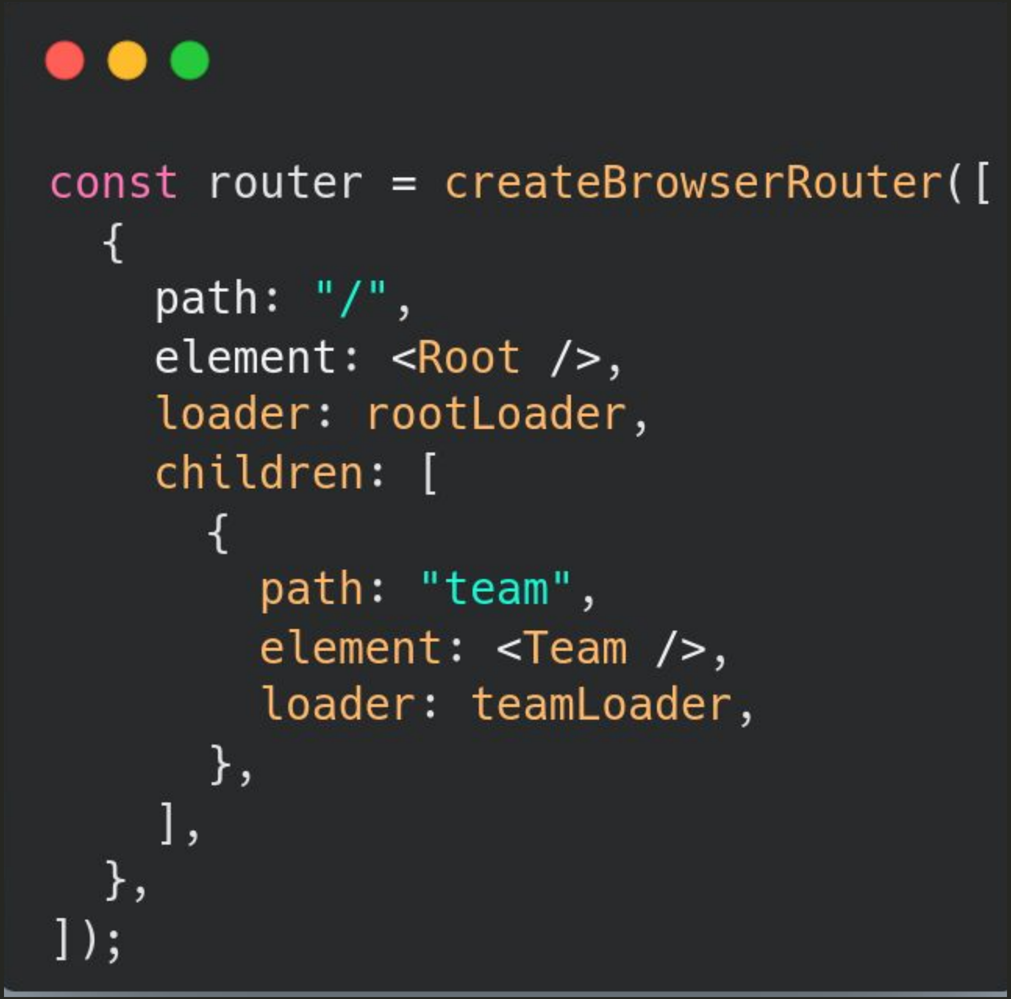
Custom Hooks



```
type myCustomHook = {
  state: number,
  value: number
}

export const useMyCustomHookProps = (): myCustomHook => {
  const {state, setState} = useState(initialState);
  const {value, setValue} = useState(initalValue)

  return {state, value};
}
```

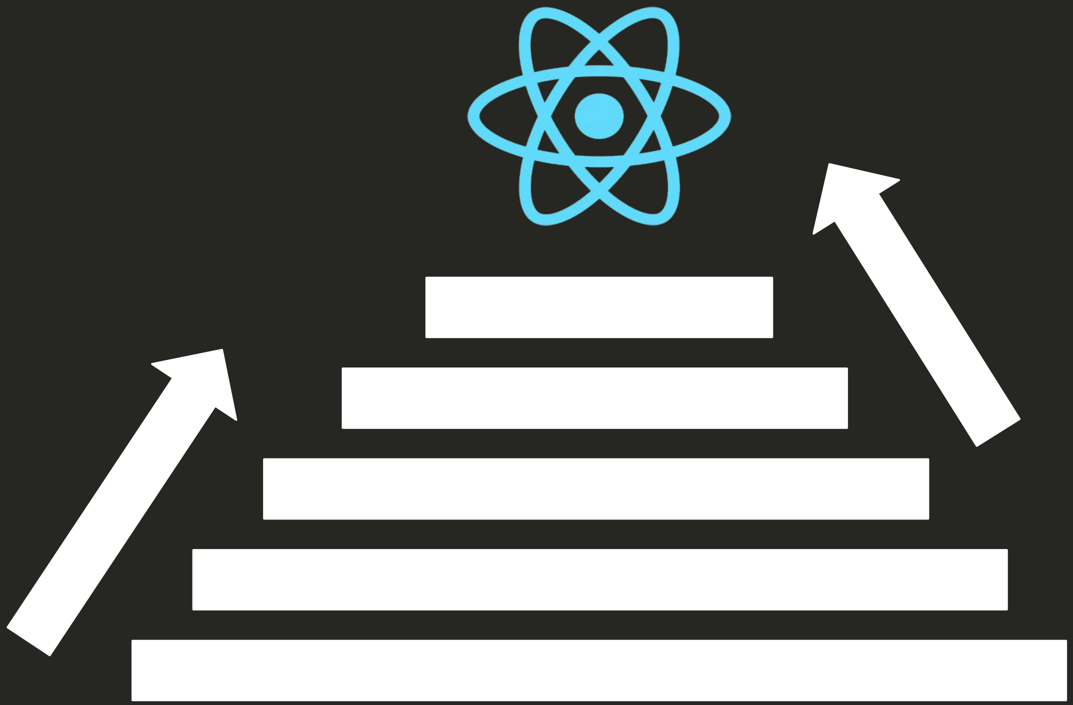
05.

REACT ROUTER



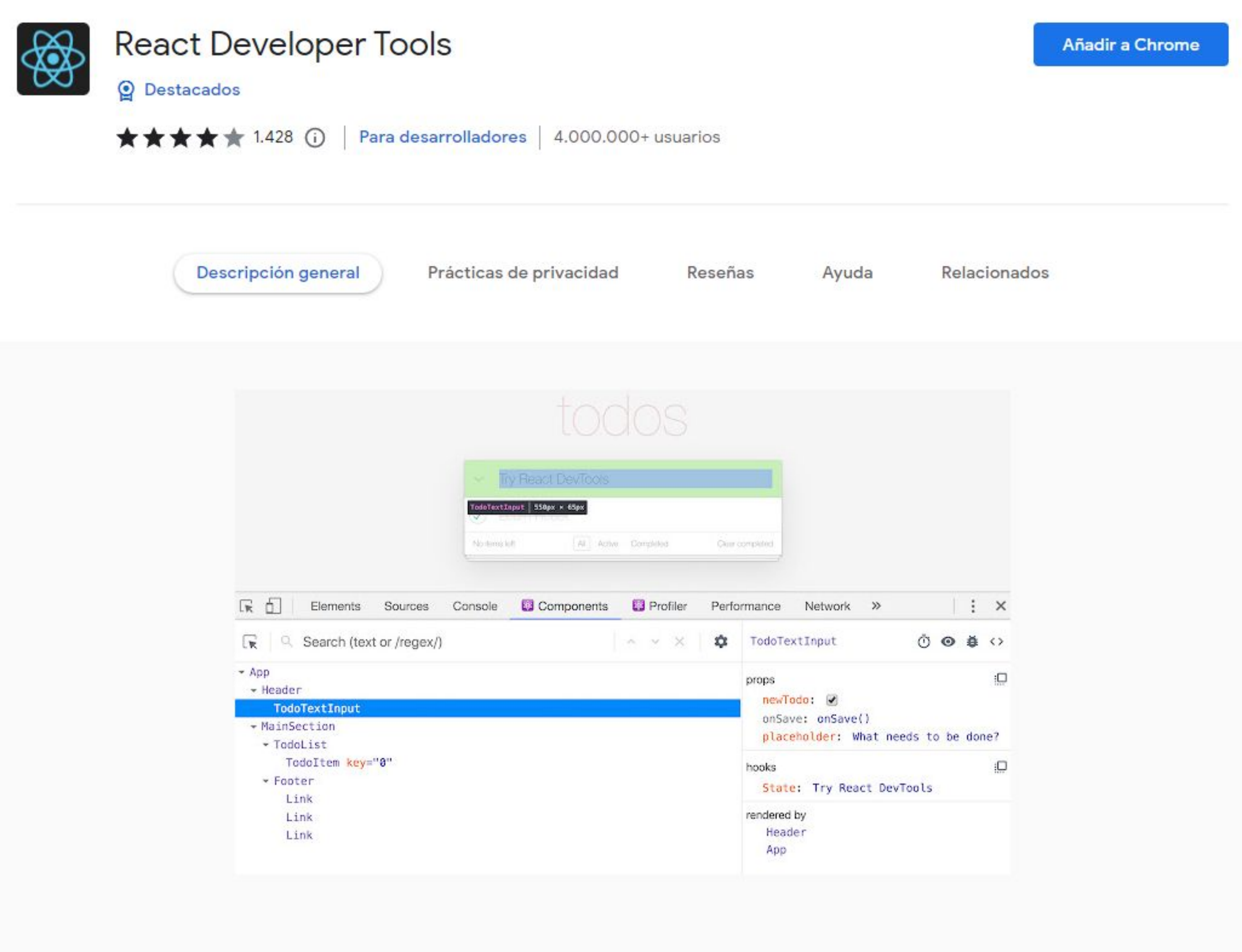
```
const router = createBrowserRouter([
  {
    path: "/",
    element: <Root />,
    loader: rootLoader,
    children: [
      {
        path: "team",
        element: <Team />,
        loader: teamLoader,
      },
    ],
  },
]);
```

REACT ROUTER EXAMPLE



06.

TOOLS



React Developer Tools

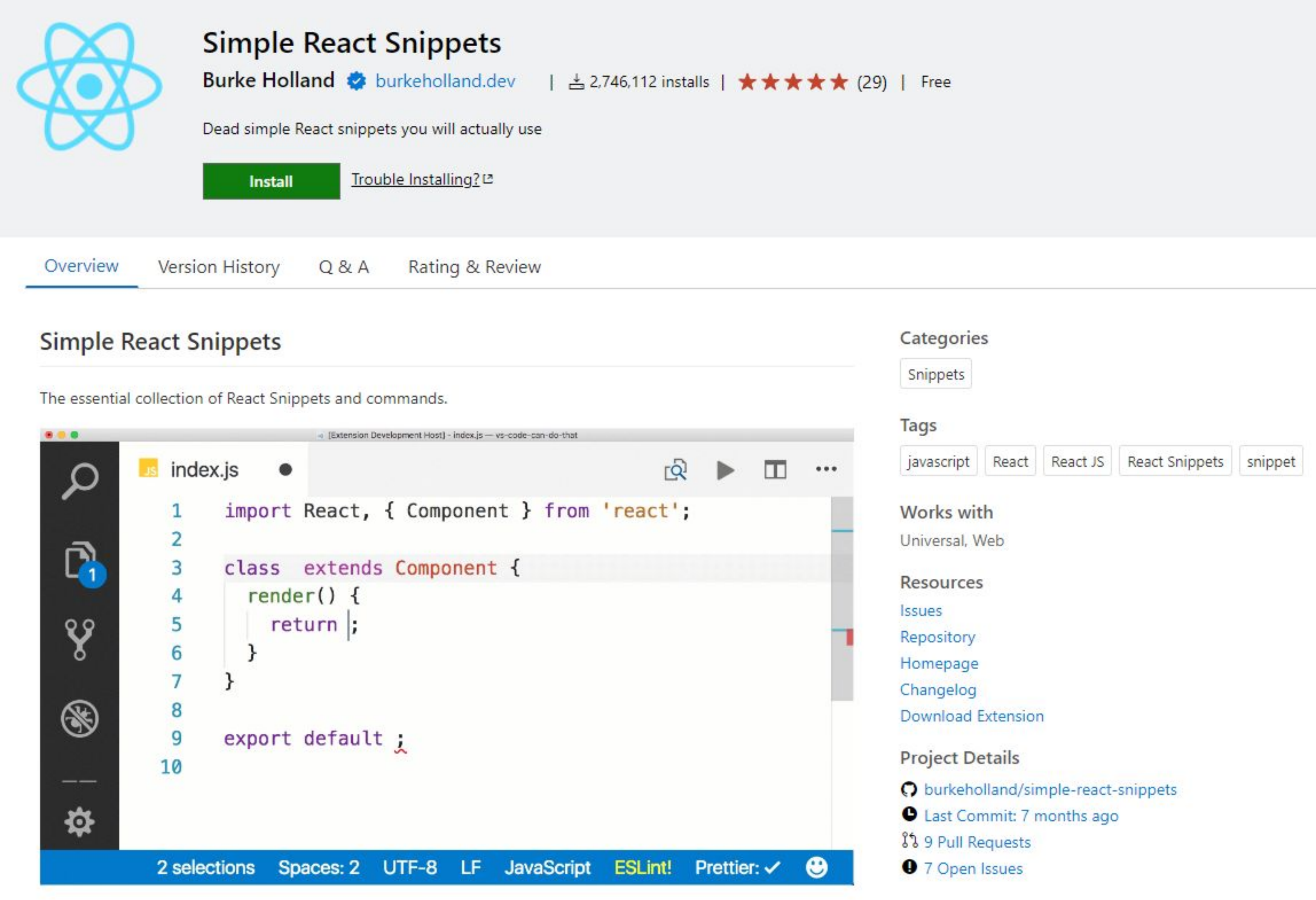
Destacados

★★★★★ 1.428 ⓘ | Para desarrolladores | 4.000.000+ usuarios

Añadir a Chrome

```
▾ App
  ▾ Header
      TodoTextInput
  ▾ MainSection
    ▾ TodoList
        TodoItem key="0"
    ▾ Footer
        Link
        Link
        Link
```

```
TodoTextInput
props
  newTodo: ☑
  onSave: onSave()
  placeholder: What needs to be done?
hooks
  State: Try React DevTools
rendered by
  Header
  App
```



Simple React Snippets

Burke Holland burkeholland.dev | 2,746,112 installs | ★★★★★ (29) | Free

Dead simple React snippets you will actually use

Simple React Snippets

The essential collection of React Snippets and commands.

[Extension Development Host] - index.js — vs-code-can-do-that

index.js

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

class  extends Component {
  render() {
    return ;
  }
}

export default ;
```

2 selections Spaces: 2 UTF-8 LF JavaScript ESLint! Prettier: ✓

Categories

Snippets

Tags

javascript React React JS React Snippets snippet

Works with

Universal, Web

Resources

- Issues
- Repository
- Homepage
- Changelog
- Download Extension

Project Details

- burkeholland/simple-react-snippets
- Last Commit: 7 months ago
- 9 Pull Requests
- 7 Open Issues

THANKS

DO YOU HAVE ANY QUESTIONS?

Adal Díaz Fariña
alu0101112251@ull.edu.es
Jorge Quintana Rodriguez
alu0101123547@ull.edu.es

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

Please keep this slide for attribution