Introduction to Reactjs

By Joey Chua

About Me

- Joey Chua
- Data Scientist @ Vertex
- https://www.linkedin.com/in/joeychuasm/
- Course Materials @ https://github.com/joeychuavertex/techladies-react-intro

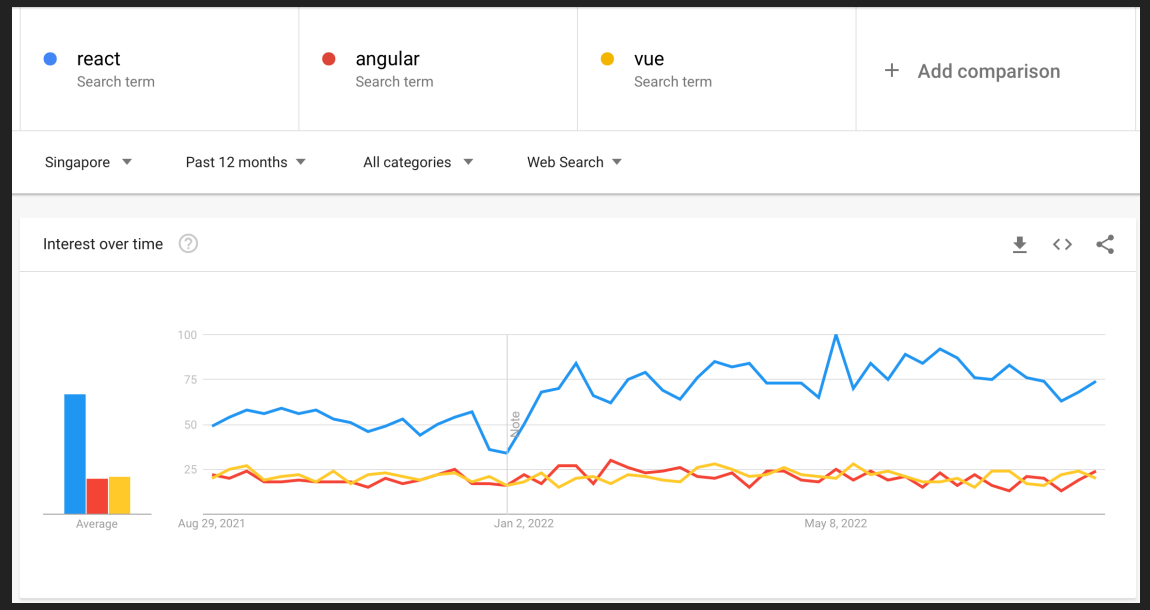
Overview of React

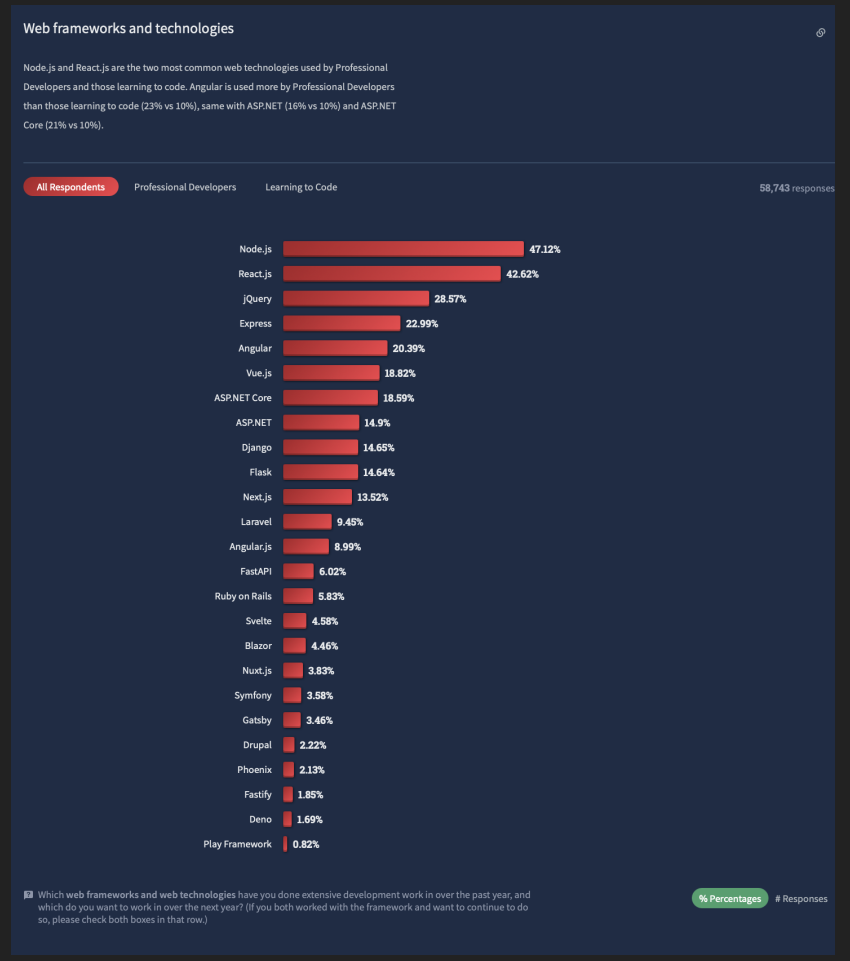
- What is React
- Why React
- Using React
- Create a simple app

Why React?

It is easy to understand, has a large community, and is one of the most popular library

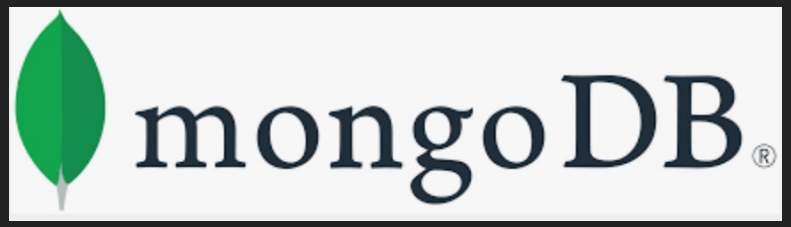
Trends


react
Search term
angular
Search term
vue
Search term
Add comparison
Singapore
Past 12 months
All categories
Web Search
Interest over time
100
75
50
25
Note
Average
Aug 29, 2021
Jan 2, 2022
May 8, 2022


Web frameworks and technologies
Node.js and React.js are the two most common web technologies used by Professional Developers and those learning to code. Angular is used more by Professional Developers than those learning to code (23% vs 10%), same with ASP.NET (16% vs 10%) and ASP.NET Core (21% vs 10%).
All Respondents
Professional Developers
Learning to Code
58,743 responses
Node.js 47.12%
React.js 42.62%
jQuery 28.57%
Express 22.99%
Angular 20.39%
Vue.js 18.82%
ASP.NET Core 18.59%
ASP.NET 14.9%
Django 14.65%
Flask 14.64%
Next.js 13.52%
Laravel 9.45%
Angular.js 8.99%
FastAPI 6.02%
Ruby on Rails 5.83%
Svelte 4.58%
Blazor 4.46%
Nuxt.js 3.83%
Symfony 3.58%
Gatsby 3.46%
Drupal 2.22%
Phoenix 2.13%
Fastify 1.85%
Deno 1.69%
Play Framework 0.82%
Which web frameworks and web technologies have you done extensive development work in over the past year, and which do you want to work in over the next year? (If you both worked with the framework and want to continue to do so, please check both boxes in that row.)
% Percentages
# Responses

Stack

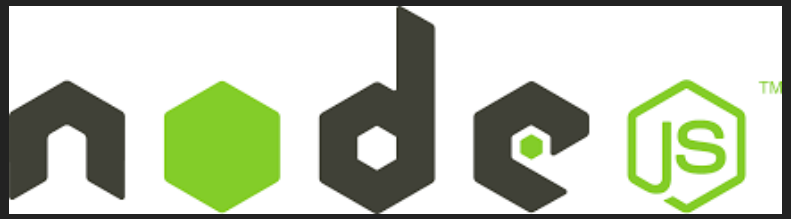
M



E

R

N



What is React?

React is a JavaScript library for building user interfaces

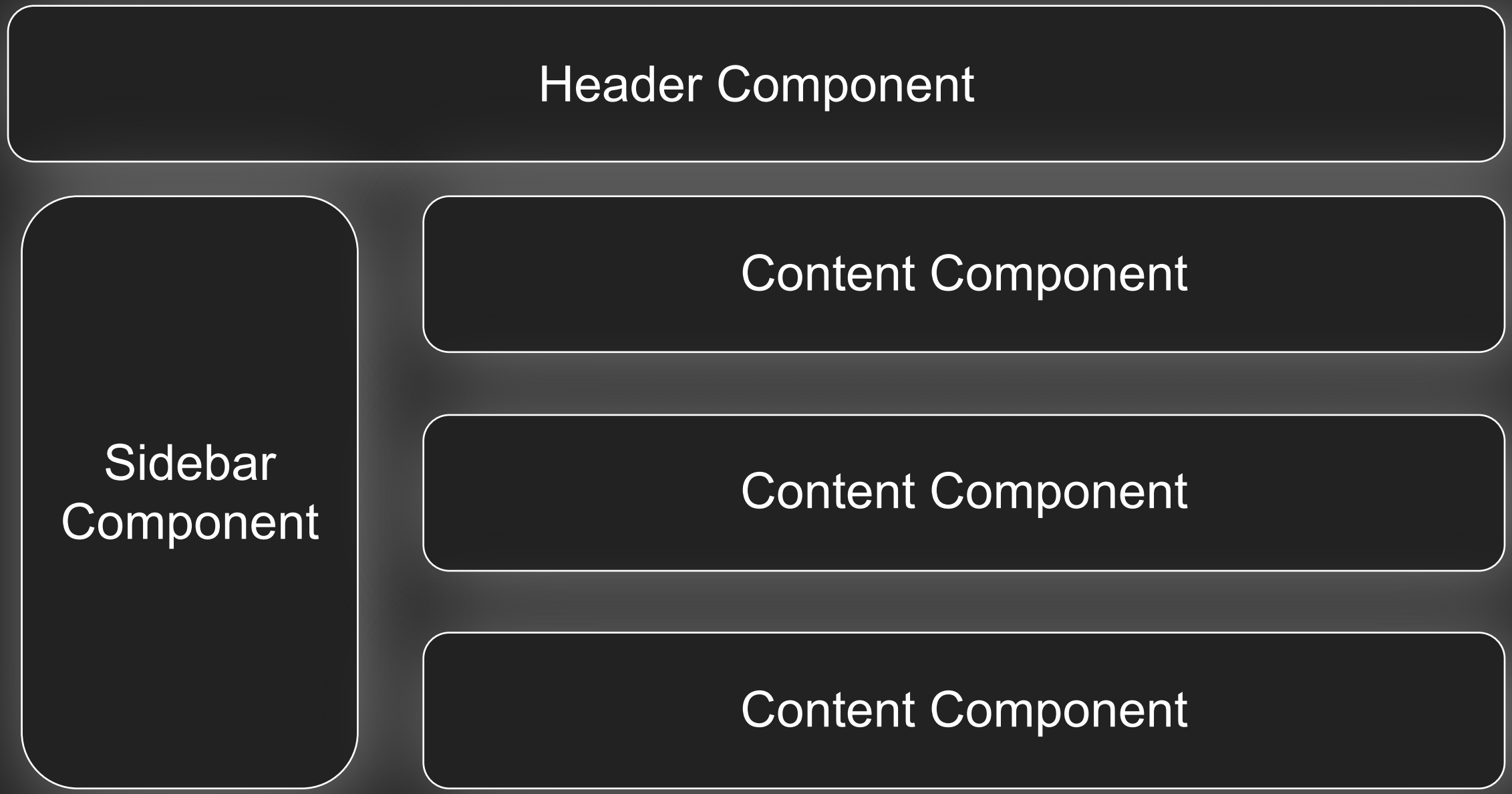
React is developed by Facebook

Declarative

Each line of code declares what each element of the app is.

Component-based

Build reusable UI elements and Javascript functions that accepts inputs


Header Component
Sidebar Component
Content Component
Content Component
Content Component

JSX is used to describe how the interface should look like

javascript.js

```
1  function Item(){
2      return <p></p>
3  }
```

Two Types of Components

1. Class-based
2. Functional

Class-based

class-based-component.js

```
1  import react, { Component } from 'react'
2
3  class Example extends Component {
4      render(){
5          return <div>Hello World!</div>
6      }
7  }
```

Functional

functional-component.js

```
1  import React from 'react';
2
3  const Example = () => {
4      return <div>Hello World!</div>
5  }
```

How to use React?

React Environment Setup

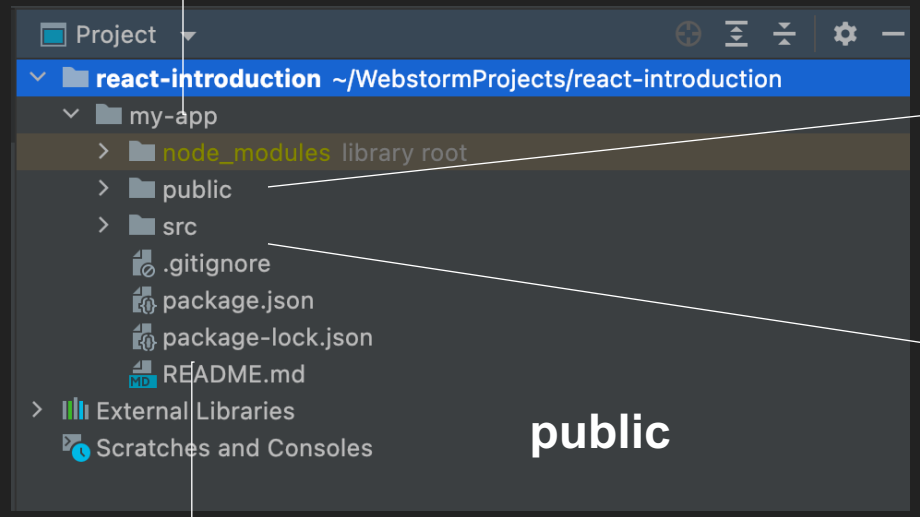
1. Manually set up using webpack & babel
2. Use create-react-app command

```
Terminal:  Local  +
joeychua@JoeyChuas-MacBook-Pro my-app % npx create-react-app my-app
```

Create-react-app is a simple tool or command that generates required **files & folders** to start the application

Node is a Javascript runtime that allows us to **execute JS code** on the server

Files and Folders



node_modules: Source code for dependencies. Do not touch.

public: The directory where your static files are stored

src: index.js, app.js etc.

package.json: The main file that defines the dependencies and other settings for the project

Index.html

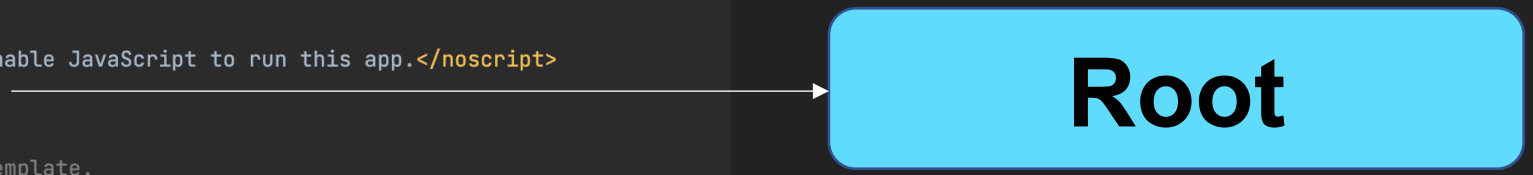
React app is mounted here

index.html

```
    Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
    work correctly both with client-side routing and a non-root public URL.
    Learn how to configure a non-root public URL by running `npm run build`.
  -->
  <title>React App</title>
</head>
<body>
  <noscript>You need to enable JavaScript to run this app.</noscript>
  <div id="root"></div>
  <!--
    This HTML file is a template.
    If you open it directly in the browser, you will see an empty page.

    You can add webfonts, meta tags, or analytics to this file.
    The build step will place the bundled scripts into the <body> tag.

    To begin the development, run `npm start` or `yarn start`.
    To create a production bundle, use `npm run build` or `yarn build`.
  -->
</body>
</html>
```



Index.js

We have ReactDOM to render the application with ID root

index.js

```
1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import './index.css';
4  import App from './App';
5  import reportWebVitals from './reportWebVitals';
6
7  const root = ReactDOM.createRoot(document.getElementById( elementId: 'root'));
8  root.render(
9    <React.StrictMode>
10     <App />
11   </React.StrictMode>
12 );
13
14 // If you want to start measuring performance in your app, pass a function
15 // to log results (for example: reportWebVitals(console.log))
16 // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
17 reportWebVitals();
```

App.js

In HTML, we write class but in JSX we will write className

App.js

```
1  import React from "react";
2
3  const App = () => {
4
5      return (
6          <>
7          </>
8      )
9  };
10
11 export default App;
```

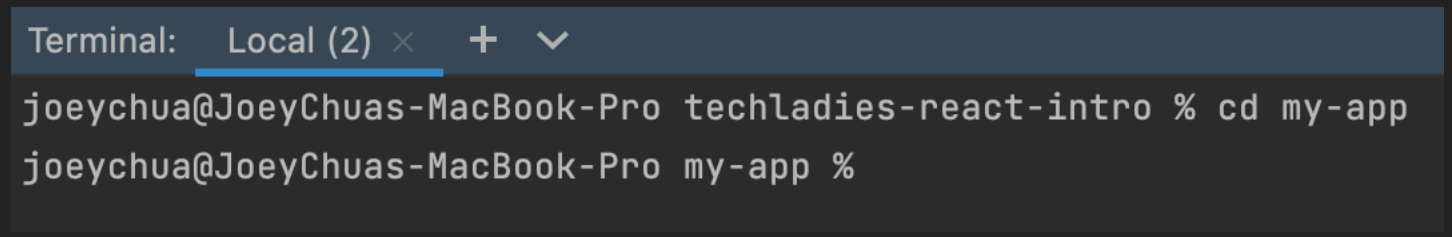
Function

Note! You only export 1 component from each file

Export

Terminal

Change Directory

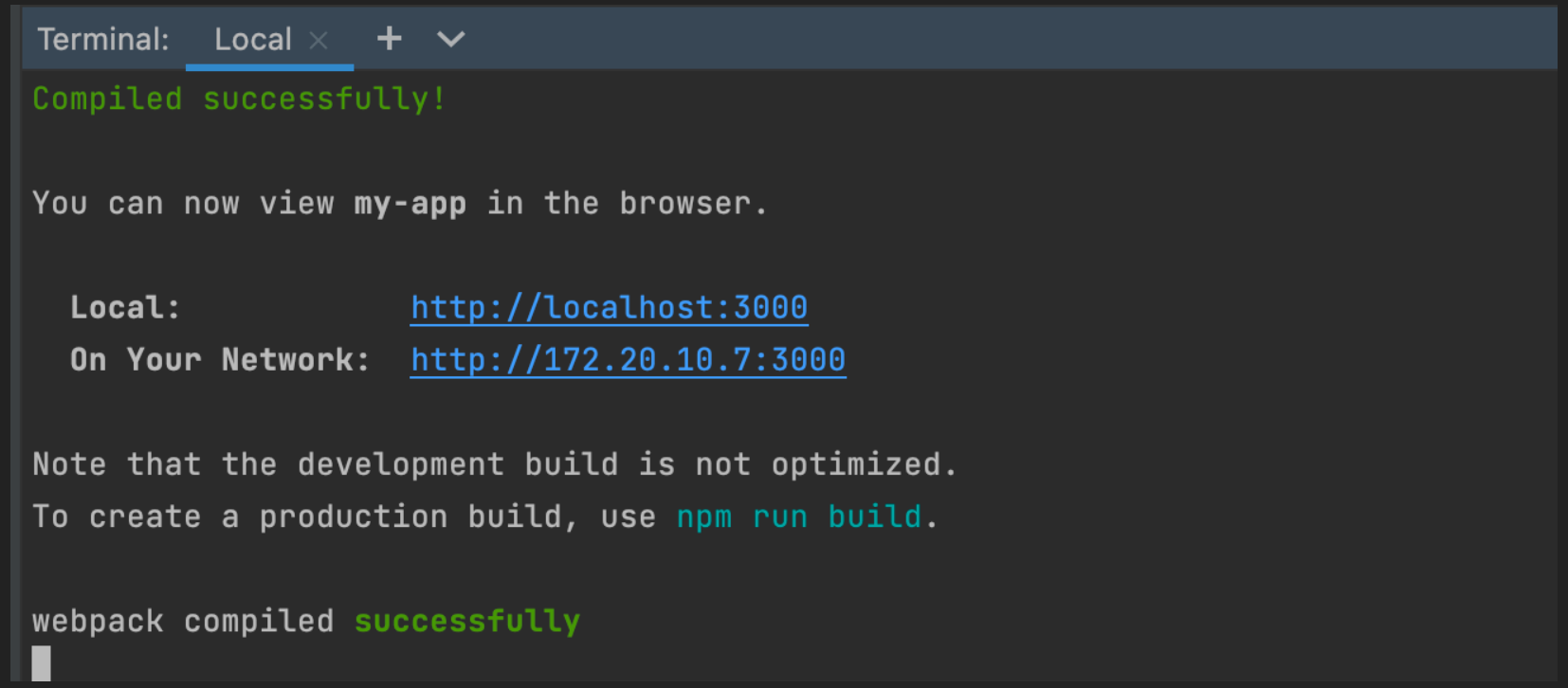


Start Server

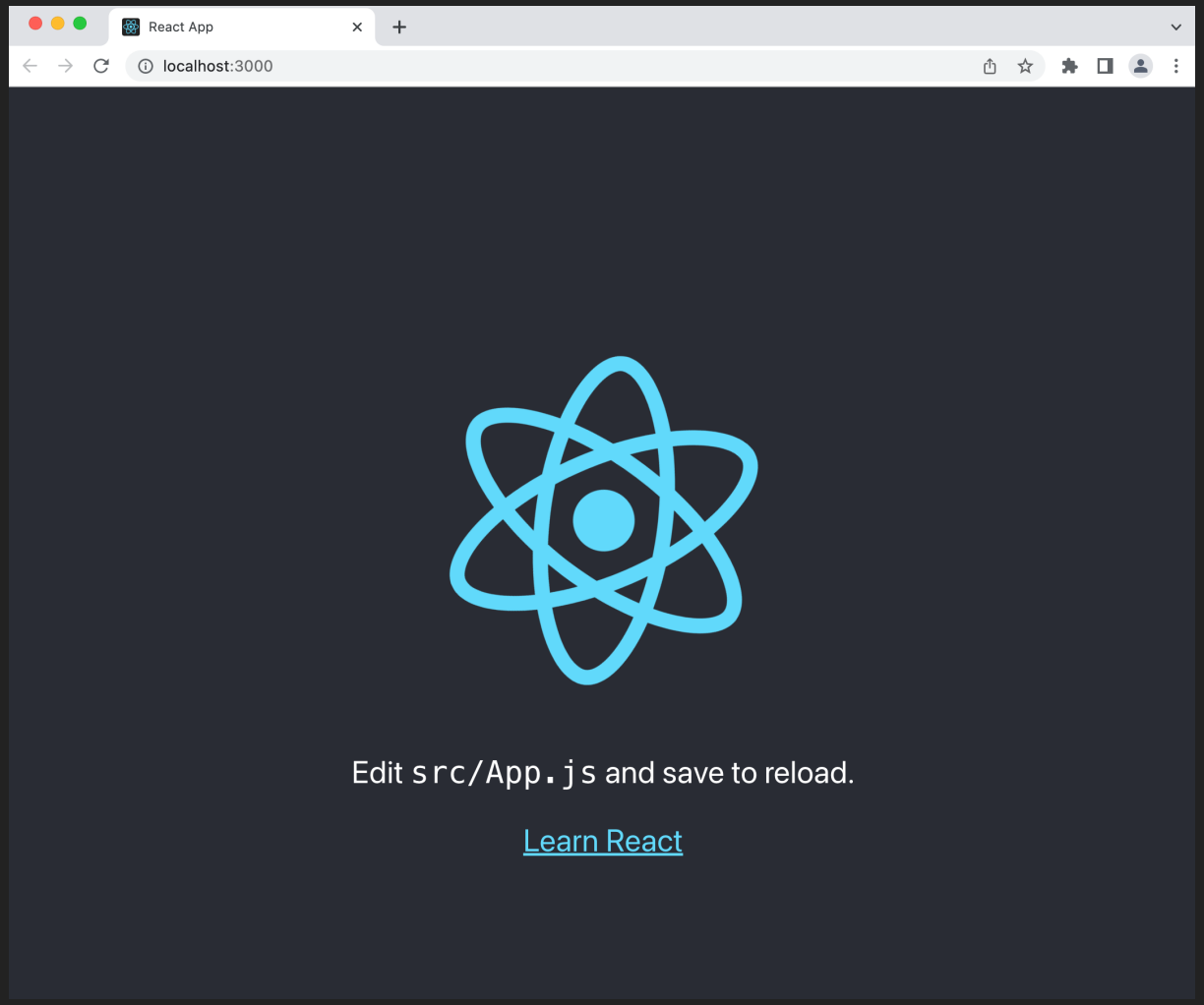
```
joeychua@JoeyChuas-MacBook-Pro my-app % npm start

> my-app@0.1.0 start
> react-scripts start
```

Terminal

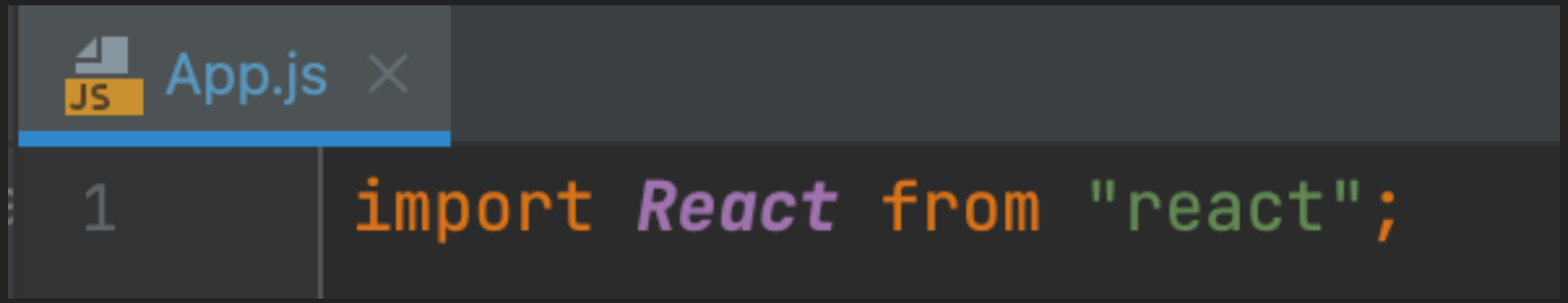

Terminal: Local
Compiled successfully!
You can now view my-app in the browser.
Local: http://localhost:3000
On Your Network: http://172.20.10.7:3000
Note that the development build is not optimized.
To create a production build, use npm run build.
webpack compiled successfully

LocalHost


React App
localhost:3000
Edit src/App.js and save to reload.
Learn React

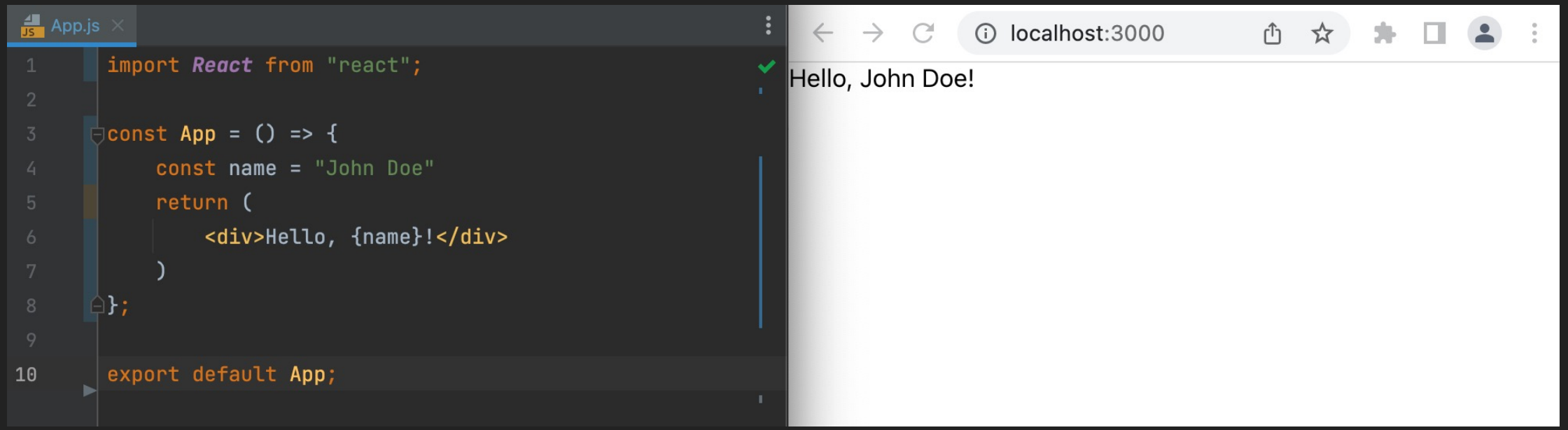
App.js

To use the library, we need to import React



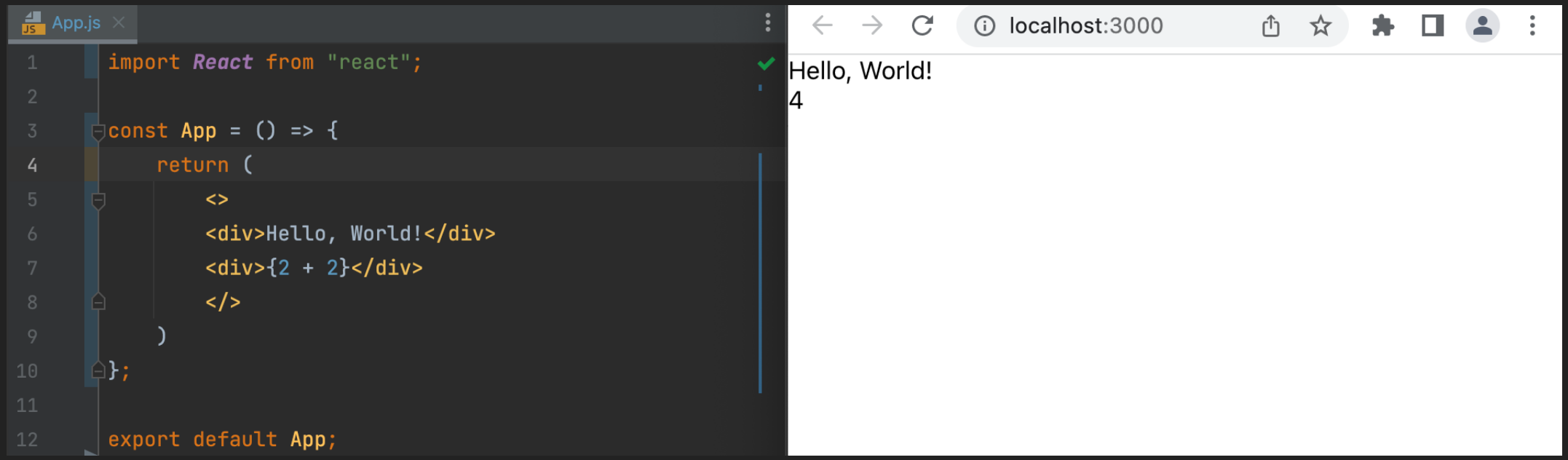
App.js

Delete function and change to const



App.js

Dynamic



App.js

Props Destructuring

App.js

```
1  import React from "react";
2
3  const Welcome = () => {
4      const greeting = 'Welcome to React';
5      return <div>{greeting}</div>;
6  };
7
8  const App = () => {
9      const name = "John Doe"
10
11     return (
12         <>
13             <div>Hello World!</div>
14             <div>{ name }</div>
15             <div>{ 2 + 2 }</div>
16             <Welcome/>
17         </>
18     )
19 };
20
21 export default App;
```

localhost:3000

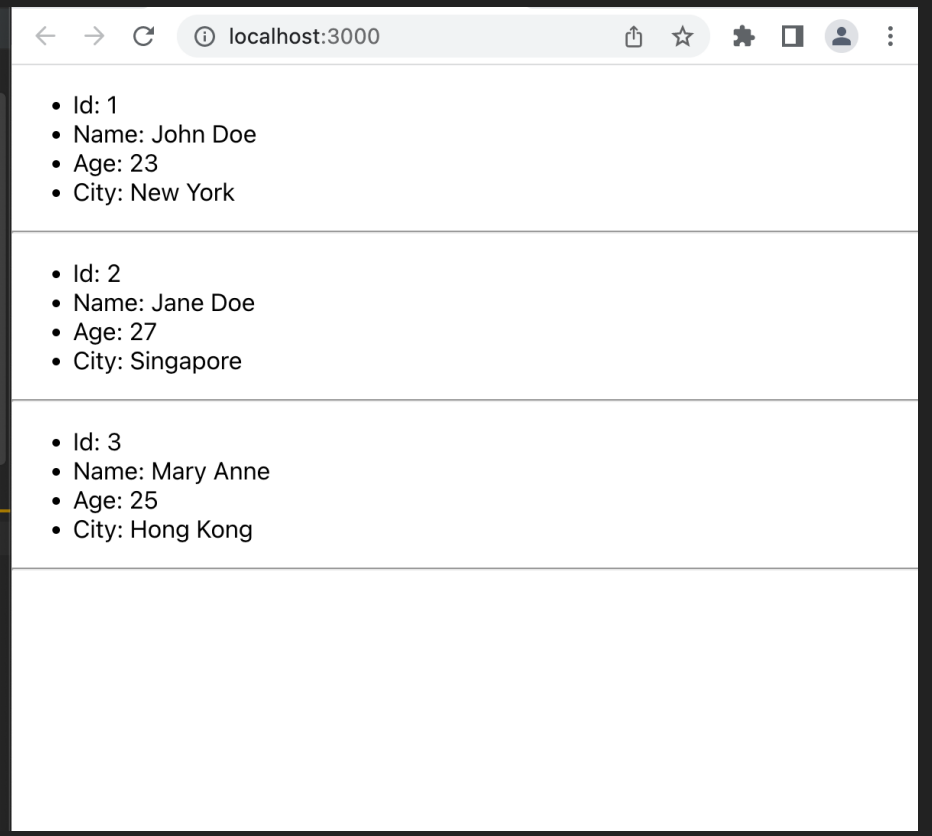
Hello World!
John Doe
4
Welcome to React

App.js

List and Keys

App.js

```
1  import React from "react";
2
3  const App = () => {
4      let employees = [
5          {
6              id: 1,
7              name: "John Doe",
8              age: 23,
9              city: "New York",
10         },
11         {
12             id: 2,
13             name: "Jane Doe",
14             age: 27,
15             city: "Singapore",
16         },
17         {
18             id: 3,
19             name: "Mary Anne",
20             age: 25,
21             city: "Hong Kong",
22         },
23     ];
```



App.js

List and Keys

App.js

```
24
25      return (
26          <div>
27              {employees.map((employee :{…} ) => {
28                  const list = (
29                      <>
30                          <ul>
31                              <li>Id: {employee.id}</li>
32                              <li>Name: {employee.name}</li>
33                              <li>Age: {employee.age}</li>
34                              <li>City: {employee.city}</li>
35                          </ul>
36                          <hr />
37                      </>
38                  );
39                  return list;
40              })}
41          </div>
42      )
43  };
44
45  export default App;
```

localhost:3000

- Id: 1
- Name: John Doe
- Age: 23
- City: New York

- Id: 2
- Name: Jane Doe
- Age: 27
- City: Singapore

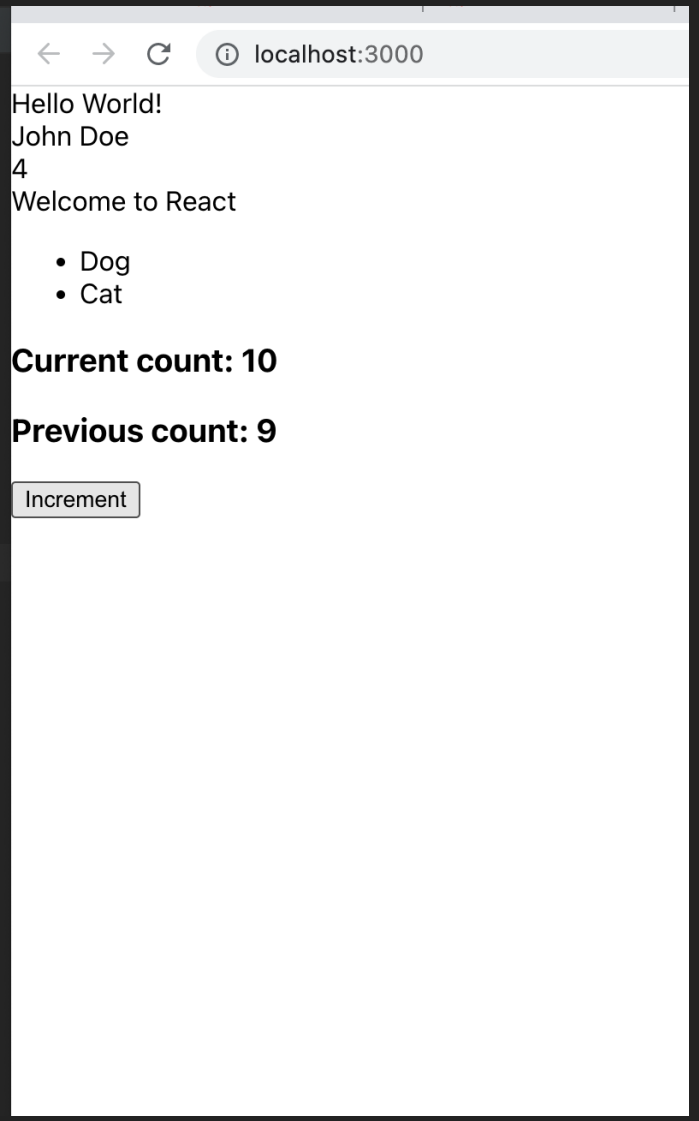
- Id: 3
- Name: Mary Anne
- Age: 25
- City: Hong Kong

App.js

State

App.js

```
24  const App = () => {
25      const name = "John Doe"
26
27      const [count, setCount] = useState( initialState: 0);
28      const [prevCount, setPrevCount] = useState( initialState: 0);
29
30      const handleClick = () => {
31          setCount( value: (prev : number ) => {
32              setPrevCount(prev);
33          });
34          setCount( value: count + 1);
35      };
36
37      return (
38          <>
39              <div>Hello World!</div>
40              <div>{ name }</div>
41              <div>{ 2 + 2 }</div>
42              <Welcome/>
43              <Data/>
44              <h3>Current count: {count}</h3>
45              <h3>Previous count: {prevCount}</h3>
46              <button onClick={handleClick}>Increment</button>
47          </>
48      )
49  };
50
51  export default App;
```

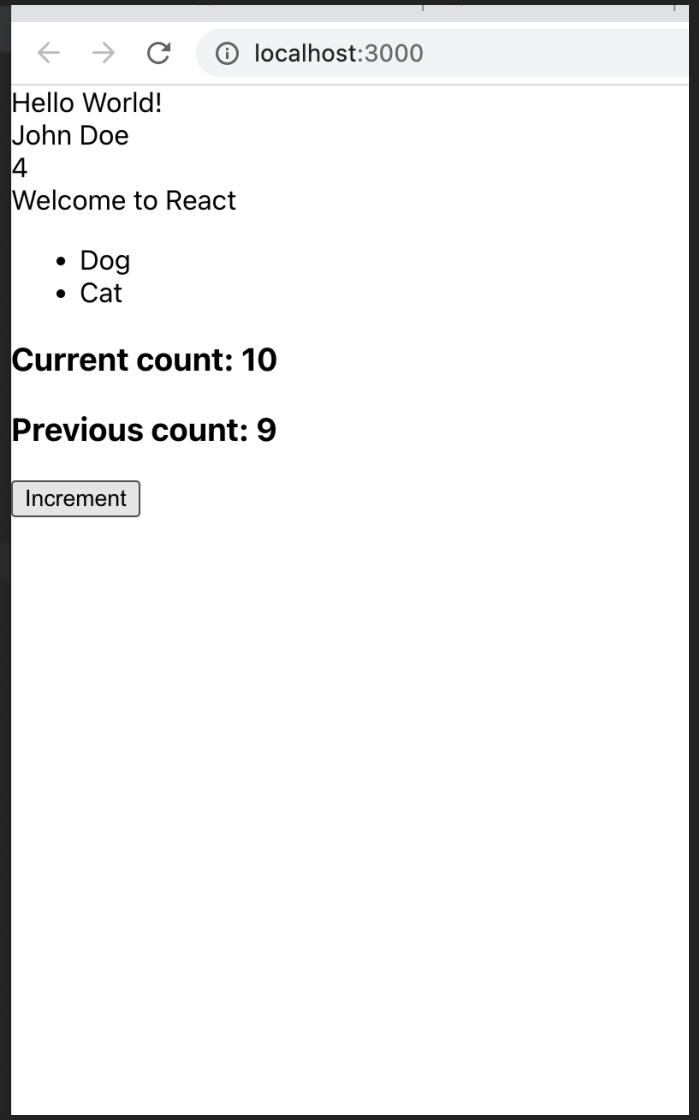


App.js

State Hook

App.js

```
24  const App = () => {
25      const name = "John Doe"
26
27      const [count, setCount] = useState( initialState: 0);
28      const [prevCount, setPrevCount] = useState( initialState: 0);
29
30      const handleClick = () => {
31          setCount( value: (prev : number ) => {
32              setPrevCount(prev);
33          });
34          setCount( value: count + 1);
35      };
36
37      return (
38          <>
39              <div>Hello World!</div>
40              <div>{ name }</div>
41              <div>{ 2 + 2 }</div>
42              <Welcome/>
43              <Data/>
44              <h3>Current count: {count}</h3>
45              <h3>Previous count: {prevCount}</h3>
46              <button onClick={handleClick}>Increment</button>
47          </>
48      )
49  };
50
51  export default App;
```



App.js

Effect Hook

App.js

```
22
23  const Countdown = ({ hr, min, sec }) => {
24      const [over, setOver] = useState( initialState: false);
25      const [paused, setPaused] = useState( initialState: true);
26      const [[h, m, s], setTime] = useState( initialState: [hr, min, sec]);
27
28      const tick = () => {
29          if (paused || over) {
30              return;
31          }
32          if (h === 0 && m === 0 && s === 0) {
33              setOver( value: true);
34          } else if (m === 0 && s === 0) {
35              setTime( value: [h - 1, 59, 59]);
36          } else if (s === 0) {
37              setTime( value: [h, m - 1, 59]);
38          } else {
39              setTime( value: [h, m, s - 1]);
40          }
41      };
42
43      const handleReset = () => {
44          setTime( value: [hr, min, sec]);
45          setPaused( value: true);
46          setOver( value: false);
47      };
48
49      const handlePause = () => setPaused(!paused);
50
51      const fmt = (val) => val.toString().padStart(2, '0');
52
53      useEffect( effect: () => {
54          let ticker = setInterval( handler: () => tick(),  timeout: 1000);
55          return () => {
56              clearInterval(ticker);
57          };
58      });
```

Hands-on

https://pokeapi.co

API

Try it now!

https://pokeapi.co/api/v2/ pokemon? Submit

Need a hint? Try pokemon/ditto, pokemon-species/aegislash, type/3, ability/battle-armor, or pokemon?limit=100000&offset=0.

Direct link to results: https://pokeapi.co/api/v2/pokemon?

Resource for pokemon?

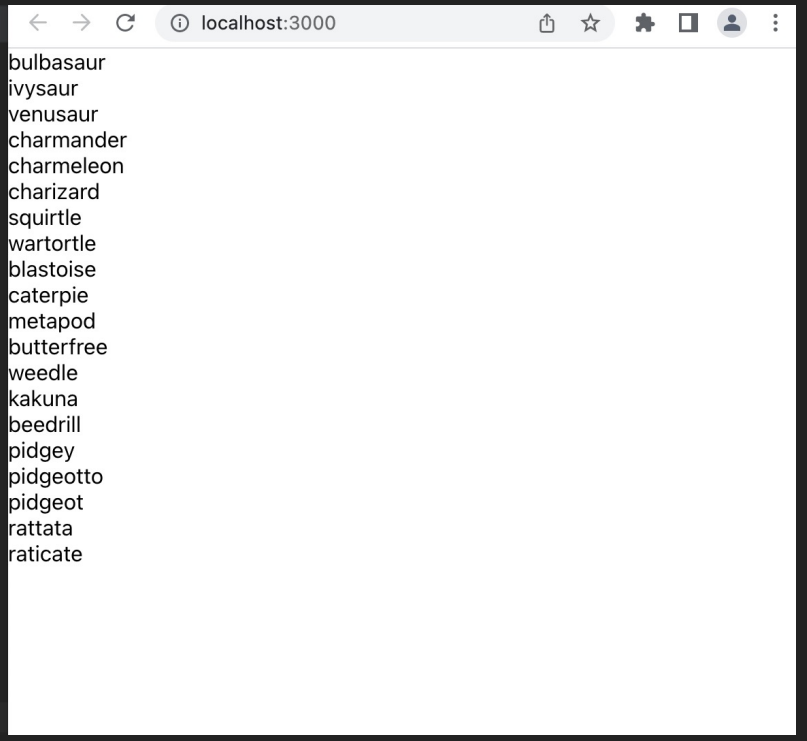

```
count: 1154
next: "https://pokeapi.co/api/v2/pokemon?offset=20&limit=20"
previous: null
▾ results: ☐ 20 items
  ▾ 0: {} 2 keys
      name: "bulbasaur"
      url: "https://pokeapi.co/api/v2/pokemon/1/"
  ▾ 1: {} 2 keys
      name: "ivysaur"
      url: "https://pokeapi.co/api/v2/pokemon/2/"
  ▾ 2: {} 2 keys
      name: "venusaur"
      url: "https://pokeapi.co/api/v2/pokemon/3/"
  ▾ 3: {} 2 keys
      name: "charmander"
      url: "https://pokeapi.co/api/v2/pokemon/4/"
  ▾ 4: {} 2 keys
      name: "charmeleon"
      url: "https://pokeapi.co/api/v2/pokemon/5/"
  ▾ 5: {} 2 keys
      name: "charizard"
      url: "https://pokeapi.co/api/v2/pokemon/6/"
  ▾ 6: {} 2 keys
      name: "squirtle"
      url: "https://pokeapi.co/api/v2/pokemon/7/"
```


View raw JSON (1.956 kB, 87 lines)

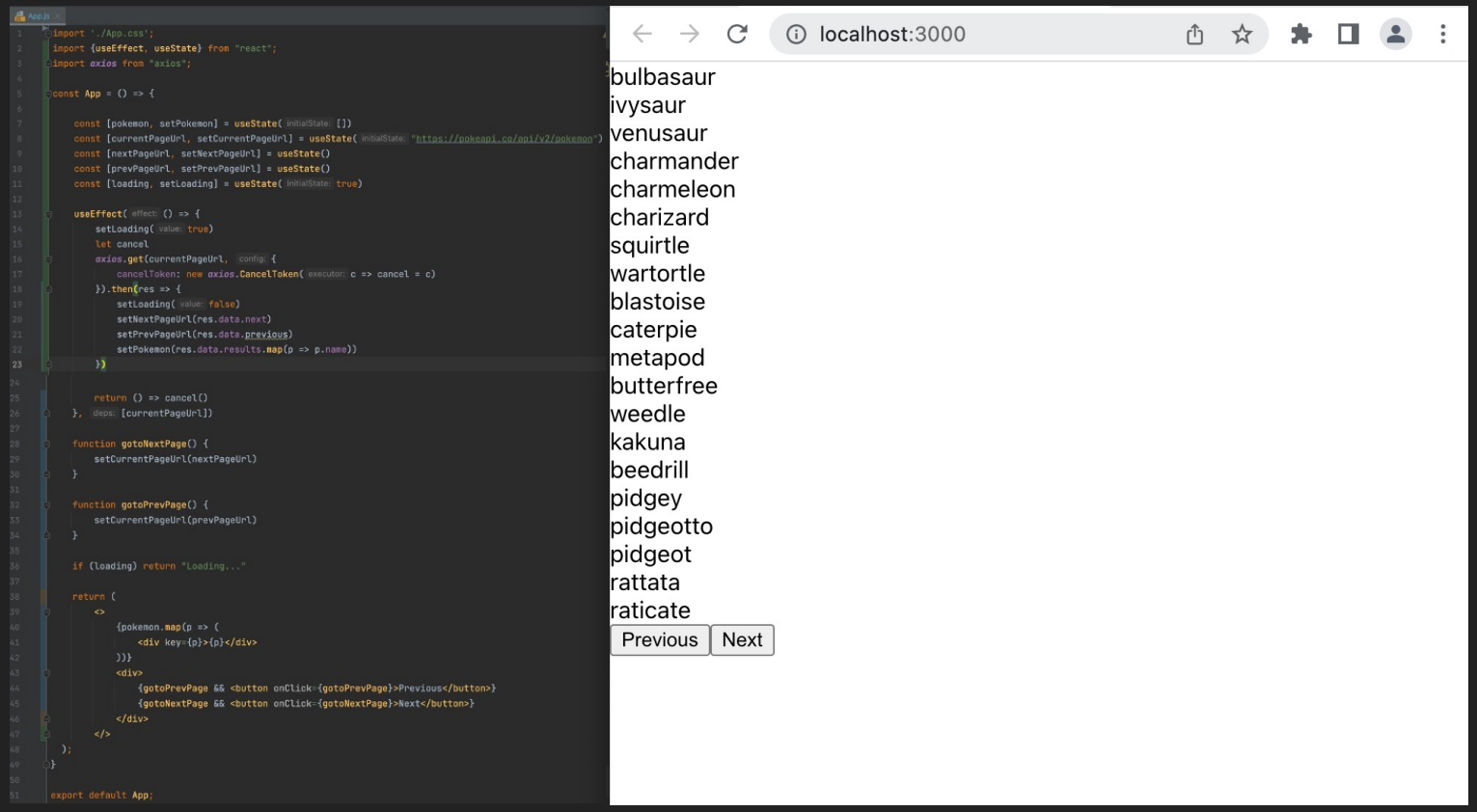
App.js
Axios, useEffect, useState, map

App.js

```
1  import './App.css';
2  import {useEffect, useState} from "react";
3  import axios from "axios";
4
5  const App = () => {
6
7      const [pokemon, setPokemon] = useState( initialState: [])
8
9      useEffect( effect: () => {
10     }, deps: [])
11
12     axios.get( url: "https://pokeapi.co/api/v2/pokemon").then(res => {
13         setPokemon((res.data.results.map(p => p.name )))
14     })
15
16     return (
17         <>
18             {pokemon.map(p => (
19                 <div key={p}>{p}</div>
20             ))}
21         </>
22 );
23 }
```



App.js Pagination



```
import './App.css';
import {useEffect, useState} from "react";
import axios from "axios";

const App = () => {

    const [pokemon, setPokemon] = useState( initialState: [])
    const [currentPageUrl, setCurrentPageUrl] = useState( initialState: "https://pokeapi.co/api/v2/pokemon")
    const [nextPageUrl, setNextPageUrl] = useState()
    const [prevPageUrl, setPrevPageUrl] = useState()
    const [loading, setLoading] = useState( initialState: true)

    useEffect( effect: () => {
        setLoading( value: true)
        let cancel
        axios.get(currentPageUrl, config: {
            cancelToken: new axios.CancelToken( executor: c => cancel = c)
        }).then(res => {
            setLoading( value: false)
            setNextPageUrl(res.data.next)
            setPrevPageUrl(res.data.previous)
            setPokemon(res.data.results.map(p => p.name))
        })

        return () => cancel()
    }, deps: [currentPageUrl])

    function gotoNextPage() {
        setCurrentPageUrl(nextPageUrl)
    }

    function gotoPrevPage() {
        setCurrentPageUrl(prevPageUrl)
    }

    if (loading) return "Loading..."

    return (
        <>
            {pokemon.map(p => (
                <div key={p}>{p}</div>
            ))}
            <div>
                {gotoPrevPage && <button onClick={gotoPrevPage}>Previous</button>}
                {gotoNextPage && <button onClick={gotoNextPage}>Next</button>}
            </div>
        </>
    );
}

export default App;
```

localhost:3000

bulbasaur
ivysaur
venusaur
charmander
charmeleon
charizard
squirtle
wartortle
blastoise
caterpie
metapod
butterfree
weedle
kakuna
beedrill
pidgey
pidgeotto
pidgeot
rattata
raticate

Previous Next