

create react project

```
DaChingLo@MacBook-xiaoqing ~ npx create-react-app my-app
npx: 97 安裝成功，花費 10.307 秒
Creating a new React app in /Users/DaChingLo/my-app.

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

yarn add v1.22.5
[1/4] 🕵️ Resolving packages...
[2/4] 🚀 Fetching packages...
[3/4] 🛡️ Linking dependencies...
warning "react-scripts > @typescript-eslint/eslint-plugin > tsutils@3.17.1" has unmet peer dependency "typescript@>=2.8.0 || >= 3.0.0".
[4/4] 🎯  Building fresh packages.

=> Create a react app for robofriends
$ npx create-react-app robofriends
$ cd robofriends
$ code .

=> VS Code
$ npm start

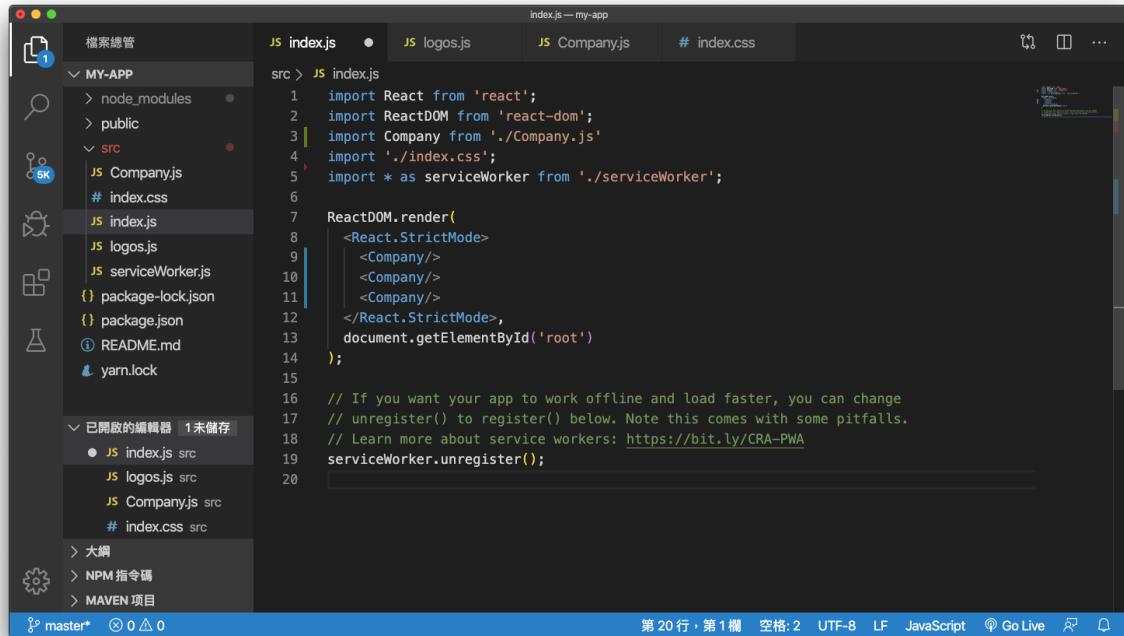
-> delete unnecessary files, leave only index.js, index.css
```

定義 Company Component (class component)

```
Company.js — my-app
JS index.js JS logos.js JS Company.js ● # index.css

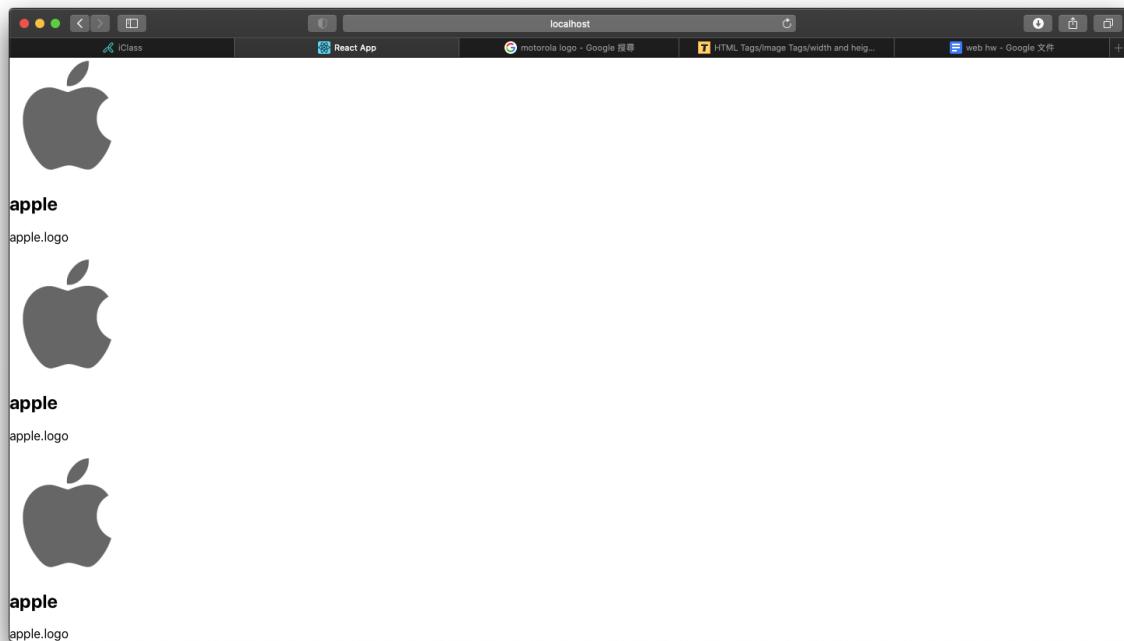
src > JS Company.js > Company > render
1 import React from 'react';
2
3 class Company extends React.Component {
4   render() {
5     return(
6       <div>
7         <img src = 'apple.png' width = "150"/>
8         <h2>apple</h2>
9         <p><img alt="apple logo" /></p>
10      );
11    }
12  }
13
14
15  export default Company;
```

在index中插入三個company的標籤

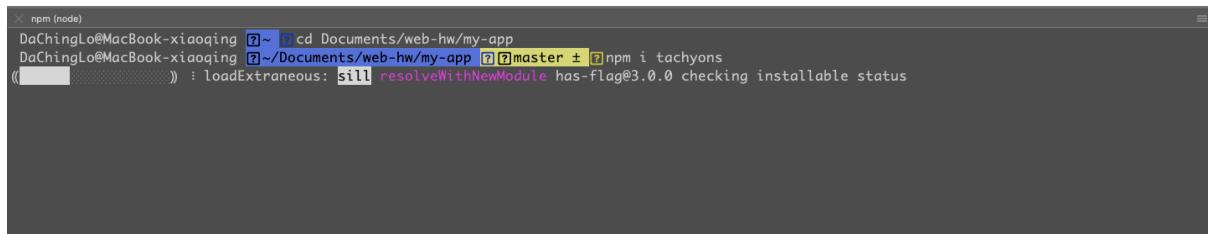


```
index.js — my-app
src > JS index.js
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import Company from './Company.js'
4 import './index.css';
5 import * as serviceWorker from './serviceWorker';
6
7 ReactDOM.render(
8   <React.StrictMode>
9     <Company/>
10    <Company/>
11    <Company/>
12  </React.StrictMode>,
13  document.getElementById('root')
14 );
15
16 // If you want your app to work offline and load faster, you can change
17 // unregister() to register() below. Note this comes with some pitfalls.
18 // Learn more about service workers: https://bit.ly/CRA-PWA
19 serviceWorker.unregister();
20
```

成功



下載tachyons套件



```
npm (node)
DoChingLo@MacBook-xiaoqing:~/Documents/web-hw/my-app
DoChingLo@MacBook-xiaoqing:~/Documents/web-hw/my-app:master ± npm i tachyons
( ) : loadExtraneous: sill resolveWithNewModule has-flag@3.0.0 checking installable status
```

import tachyons

A screenshot of a code editor showing the `index.js` file from a React application. The code includes an import statement for `tachyons`. The code editor interface shows tabs for `index.js`, `logos.js`, `Company.js`, and `index.css`. The sidebar on the left shows the project structure with files like `node_modules`, `public`, and `src`. A status bar at the bottom indicates the file is on line 5, has 18 lines selected, and is in JavaScript mode.

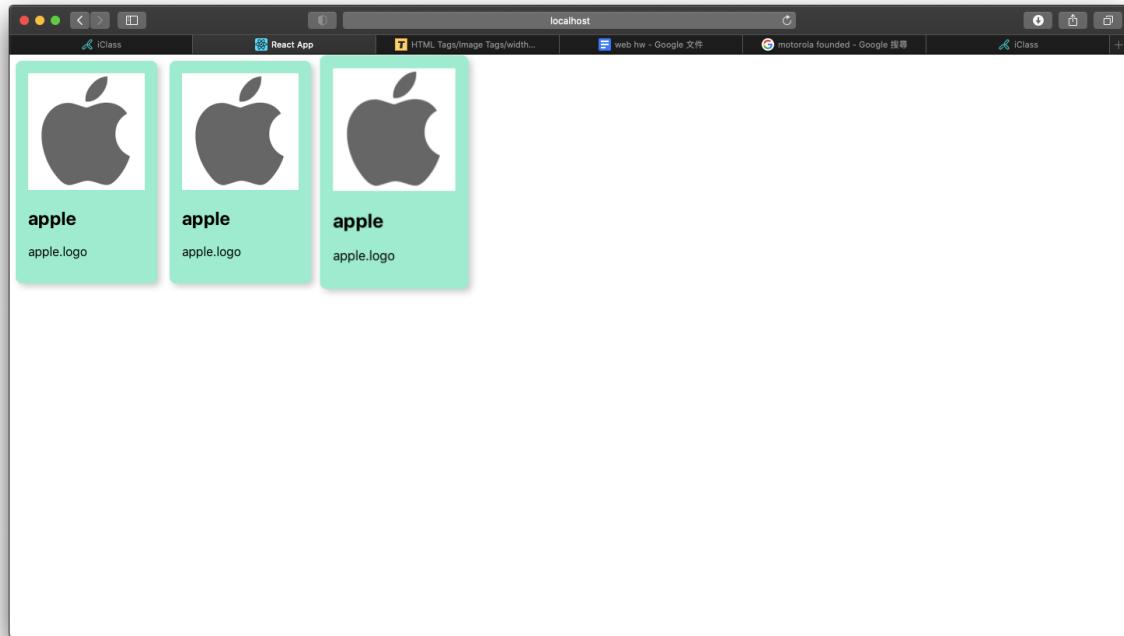
```
index.js — my-app
src > JS index.js
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import Company from './Company.js'
4 import './index.css';
5 import 'tachyons';
6 import * as serviceWorker from './serviceWorker';
7
8 ReactDOM.render(
9   <React.StrictMode>
10    <Company/>
11    <Company/>
12    <Company/>
13  </React.StrictMode>,
14  document.getElementById('root')
15 );
16
17 // If you want your app to work offline and load faster, you can change
18 // unregister() to register() below. Note this comes with some pitfalls.
19 // Learn more about service workers: https://bit.ly/CRA-PWA
20 serviceWorker.unregister();
21
```

在company 的 div 中加入 className

A screenshot of a code editor showing the `Company.js` file. The code defines a `Company` component that renders a single `div` element with a specific class name. The code editor interface shows tabs for `index.js`, `logos.js`, `Company.js`, and `index.css`. The sidebar on the left shows the project structure. A status bar at the bottom indicates the file is on line 6, has 60 lines selected, and is in JavaScript mode.

```
Company.js — my-app
src > JS Company.js > Company > render
1 import React from 'react';
2
3 class Company extends React.Component {
4   render() {
5     return(
6       <div className='bg-light-green dib br3 pa3 ma2 grow bw2 shadow-5'>
7         <img src = 'apple.png' width = "150"/>
8         <h2>apple</h2>
9         <p>apple.logo</p>
10      </div>
11    );
12  }
13}
14
15 export default Company;
```

成功

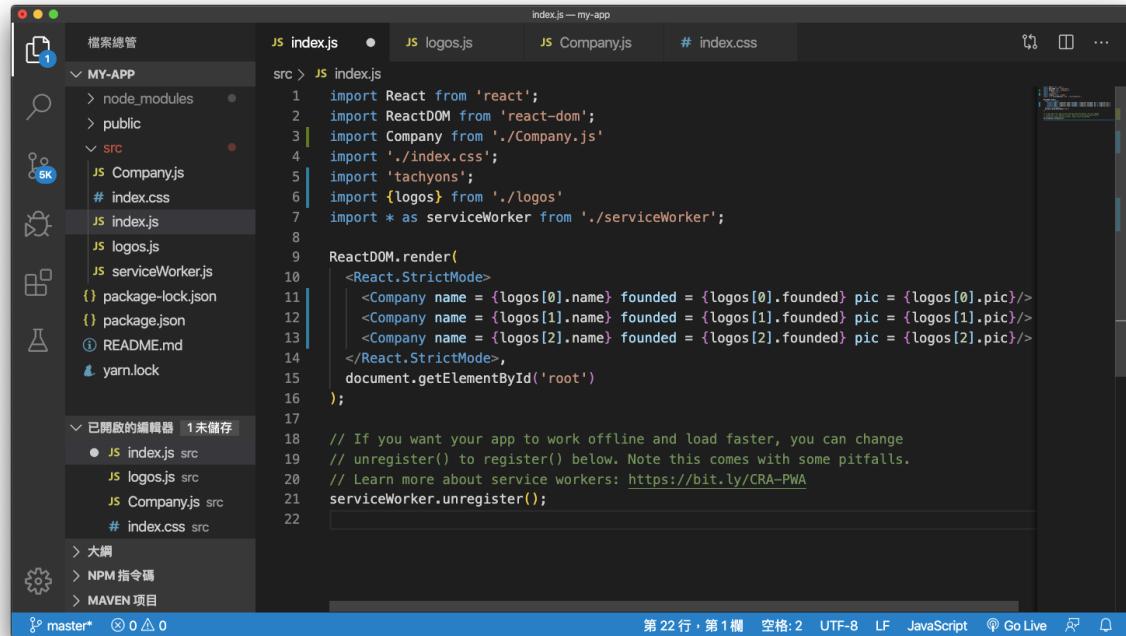


建立10個logo

```
logos.js — my-app
src > JS logos.js > logos > id
id: 7,
name: 'Apple',
founded: 1976,
pic: 'apple.png'
},
{
id: 8,
name: 'Samsung',
founded: 1938,
pic: 'samsung.png'
},
{
id: 9,
name: 'Apple',
founded: 1928,
pic: 'motorola.jpg'
},
[
id: 10,
name: 'Apple',
founded: 1928,
pic: 'motorola.jpg'
]
```

The screenshot shows a code editor interface with the file 'logos.js' open. The file contains an array of objects representing different companies. Each object has properties for 'id', 'name', 'founded', and 'pic'. There are two entries for 'Apple' (id 7 and id 10) and one entry for 'Samsung' (id 8). The 'pic' field for each entry points to a specific image file name like 'apple.png' or 'samsung.png'. The code editor also shows the project structure on the left, including files like index.js, Company.js, and index.css.

用props將logos的資訊傳入Company。



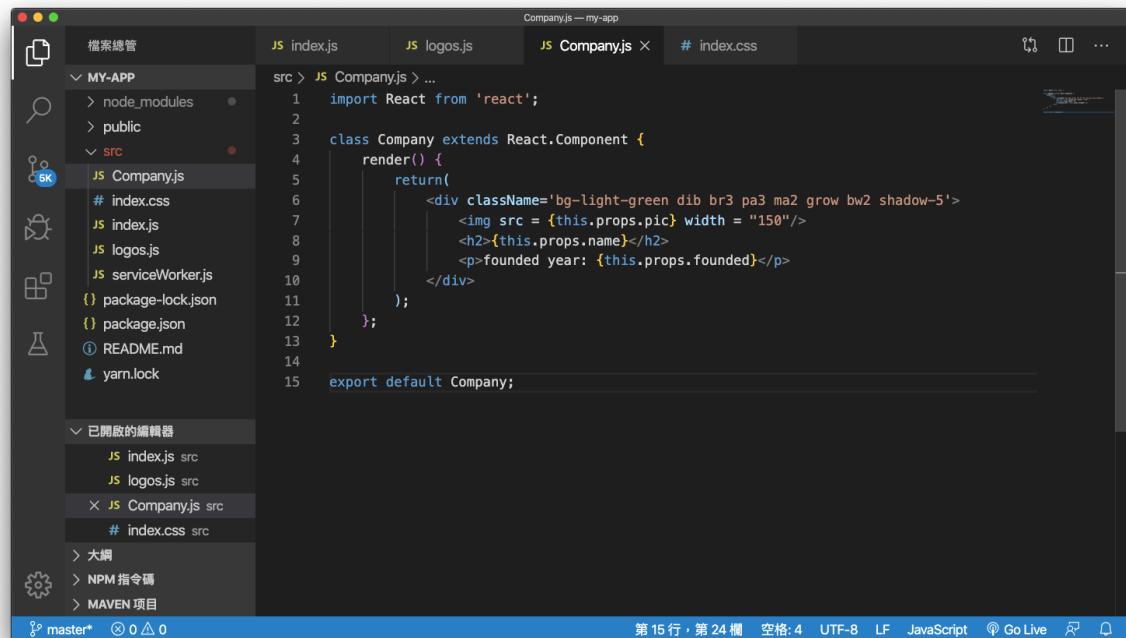
The screenshot shows a code editor with the file `index.js` open. The code imports `React`, `ReactDOM`, and `Company` from `./Company.js`. It then imports `logos` from `./logos` and `serviceWorker` from `./serviceWorker`. Inside the `ReactDOM.render` function, it loops through three companies in the `logos` array and creates a `<Company>` component for each, passing the `name`, `founded`, and `pic` properties.

```
import React from 'react';
import ReactDOM from 'react-dom';
import Company from './Company.js'
import './index.css';
import {logos} from './logos';
import * as serviceWorker from './serviceWorker';

ReactDOM.render(
  <React.StrictMode>
    <Company name={logos[0].name} founded={logos[0].founded} pic={logos[0].pic}/>
    <Company name={logos[1].name} founded={logos[1].founded} pic={logos[1].pic}/>
    <Company name={logos[2].name} founded={logos[2].founded} pic={logos[2].pic}/>
  </React.StrictMode>,
  document.getElementById('root')
);

// If you want your app to work offline and load faster, you can change
// unregister() to register() below. Note this comes with some pitfalls.
// Learn more about service workers: https://bit.ly/CRA-PWA
serviceWorker.unregister();
```

用props接收由index傳入的資訊。

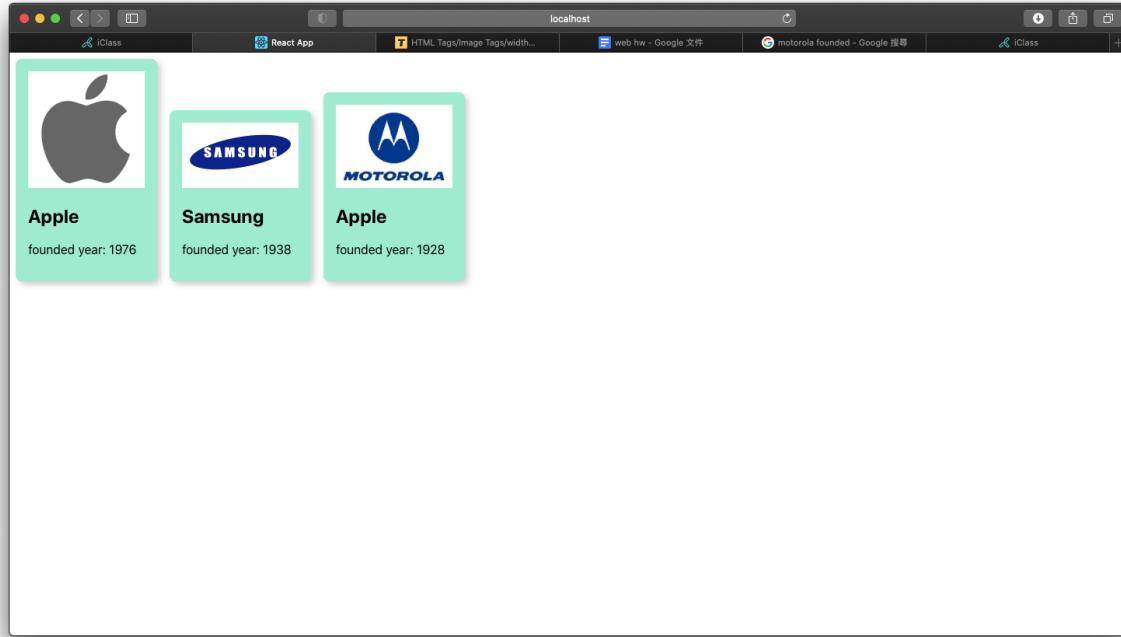


The screenshot shows a code editor with the file `Company.js` open. The code defines a class `Company` that extends `React.Component`. In its `render` method, it returns a `<div>` element with a `bg-light-green` class, containing an `img` with the `src` prop set to `this.props.pic`, a `<h2>` with the `name` prop, and a `<p>` with the `founded` prop.

```
class Company extends React.Component {
  render() {
    return(
      <div className='bg-light-green dib br3 pa3 ma2 grow bw2 shadow-5'>
        <img src={this.props.pic} width="150"/>
        <h2>{this.props.name}</h2>
        <p>founded year: {this.props.founded}</p>
      </div>
    );
  }
}

export default Company;
```

成功

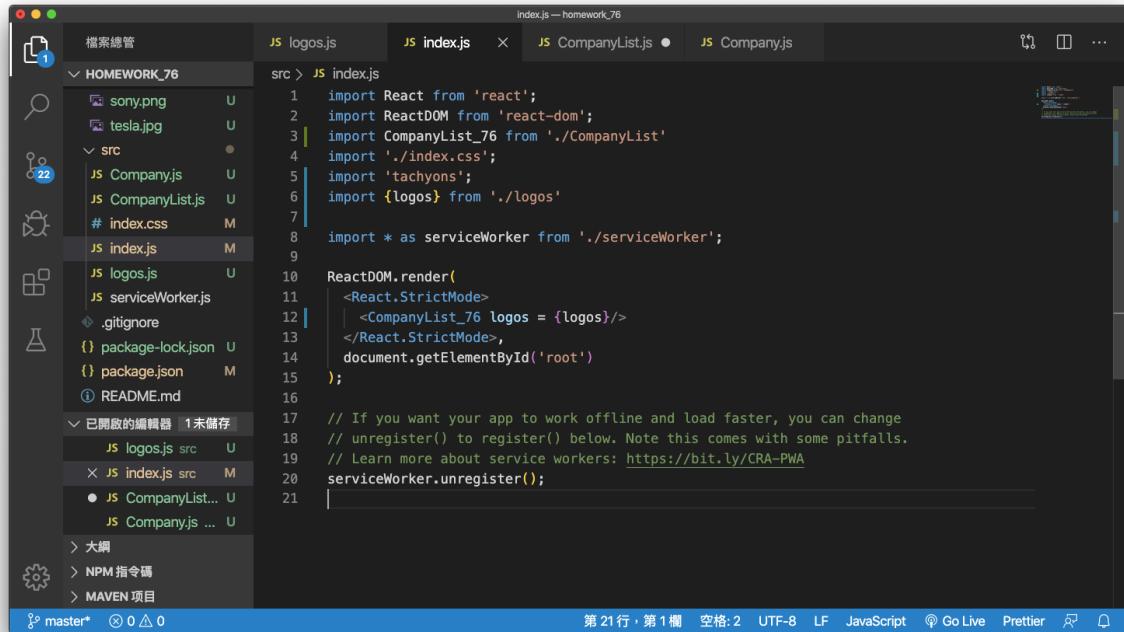


建立 companyList_76 component，利用 map 函式 render 多個 company_76，並把 company.id 設為 company 的 key。

```
CompanyList.js — homework_76
src > JS CompanyList.js > ...
1 import Company_76 from './Company'
2 import React from 'react'
3
4 class CompanyList_76 extends React.Component {
5   render() {
6     const CompanyComponents = this.props.logos.map((company) => {
7       return (
8         <Company_76
9           key = {company.id}
10          name = {company.name}
11          founded = {company.founded}
12          pic = {company.pic}
13        />
14      );
15    });
16    return (
17      <div>{CompanyComponents}</div>
18    )
19  }
20
21 export default CompanyList_76;
22
```

The code editor shows the implementation of the `CompanyList_76` component. It imports the `Company_76` component and the `React` library. The component uses the `map` function to render multiple `Company_76` components, each with its own unique key (`company.id`), name (`company.name`), founding year (`company.founded`), and logo (`company.pic`). The resulting components are then returned within a single `div` element.

在 index 建立 companyList_76, 把 logos 傳進去。



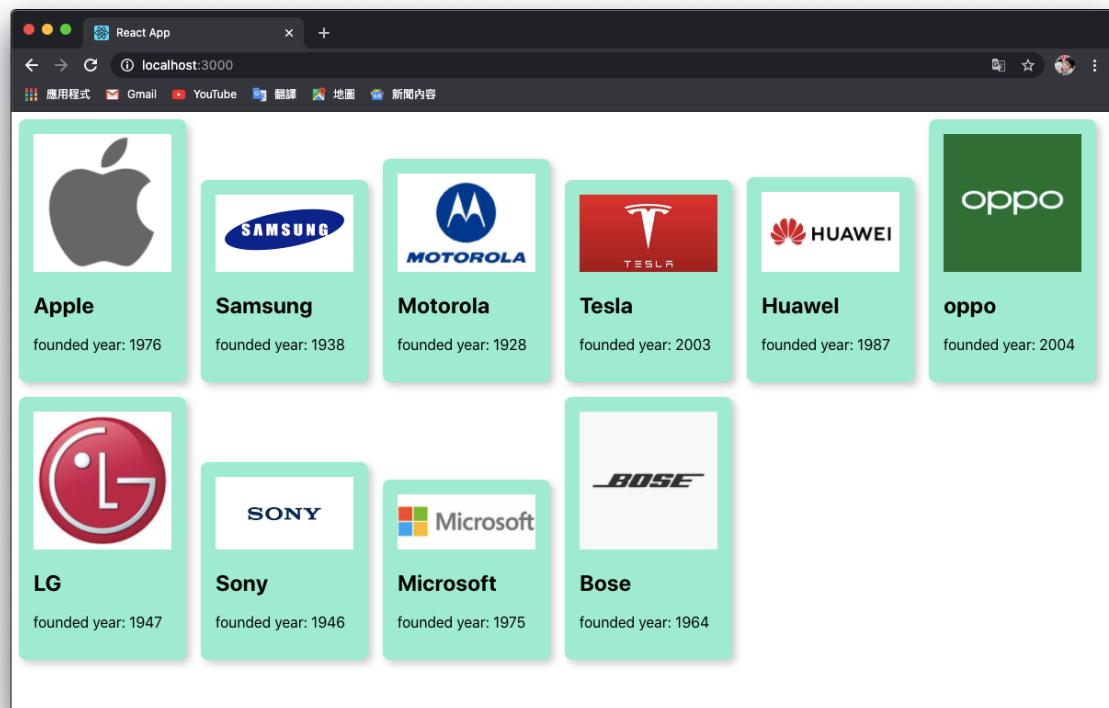
The screenshot shows a code editor with the following file structure:

- index.js
- logos.js
- CompanyList.js
- Company.js
- index.css
- index.js (selected)
- logos.js
- serviceWorker.js
- .gitignore
- package-lock.json
- package.json
- README.md

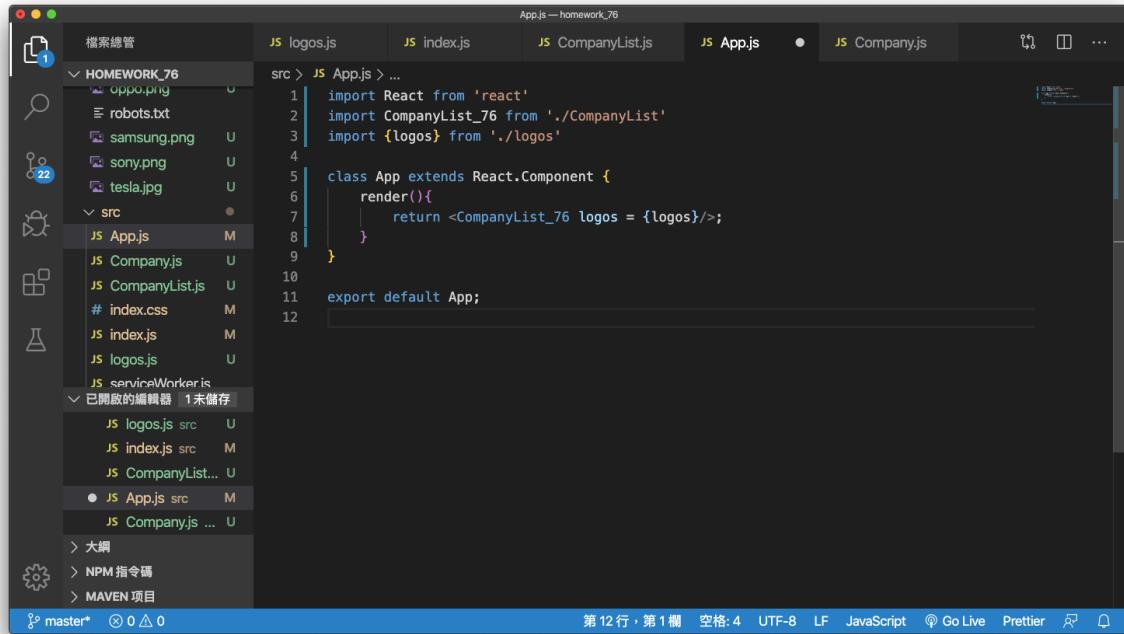
The code in index.js imports CompanyList_76 from './CompanyList'. It then imports logos from './logos' and renders them using ReactDOM.render. The logs section indicates that the app can work offline and faster by changing serviceWorker.unregister() to register().

```
index.js — homework_76
src > JS index.js
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import CompanyList_76 from './CompanyList';
4 import './index.css';
5 import 'tachyons';
6 import {logos} from './logos';
7
8 import * as serviceWorker from './serviceWorker';
9
10 ReactDOM.render(
11   <React.StrictMode>
12     | <CompanyList_76 logos={logos}/>
13   </React.StrictMode>,
14   document.getElementById('root')
15 );
16
17 // If you want your app to work offline and load faster, you can change
18 // unregister() to register() below. Note this comes with some pitfalls.
19 // Learn more about service workers: https://bit.ly/CRA-PWA
20 serviceWorker.unregister();
21
```

成功



把建立 companyList_76 的工作交給 App component



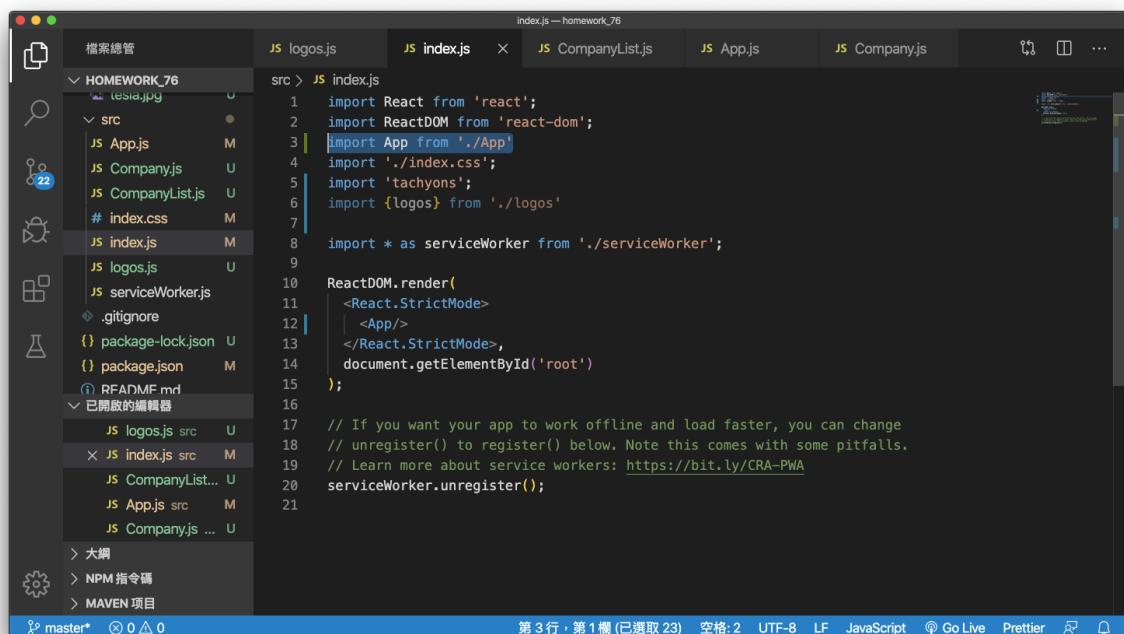
The screenshot shows a code editor with the following file structure in the sidebar:

- HOMEWORK_76
- oppo.png
- robots.txt
- samsung.png
- sony.png
- tesla.jpg
- src
- JS App.js
- JS Company.js
- JS CompanyList.js
- # index.css
- JS index.js
- JS logos.js
- JS serviceWorker.js

The JS App.js tab is active, displaying the following code:

```
src > JS App.js > ...
1 import React from 'react'
2 import CompanyList_76 from './CompanyList'
3 import {logos} from './logos'
4
5 class App extends React.Component {
6   render(){
7     return <CompanyList_76 logos = {logos}/>;
8   }
9 }
10
11 export default App;
```

index 只需要用 App 即可



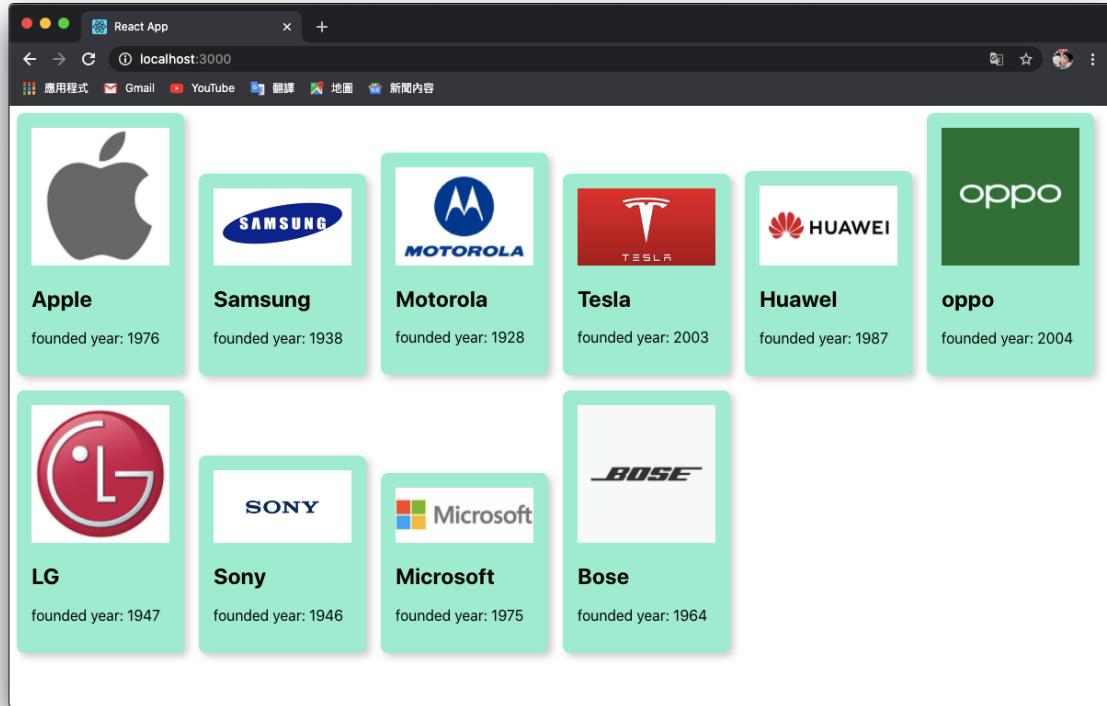
The screenshot shows a code editor with the following file structure in the sidebar:

- HOMEWORK_76
- tesla.jpg
- src
- JS App.js
- JS Company.js
- JS CompanyList.js
- # index.css
- JS index.js
- JS logos.js
- JS serviceWorker.js
- .gitignore
- { package-lock.json
- { package.json
- RFADMF.mrd
- JS logos.js src
- X JS index.js src
- JS CompanyList... U
- JS App.js src
- JS Company.js ... U

The JS index.js tab is active, displaying the following code:

```
src > JS index.js > ...
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import App from './App'
4 import './index.css';
5 import 'tachyons';
6 import {logos} from './logos'
7
8 import * as serviceWorker from './serviceWorker';
9
10 ReactDOM.render(
11   <React.StrictMode>
12     <App/>
13   </React.StrictMode>,
14   document.getElementById('root')
15 );
16
17 // If you want your app to work offline and load faster, you can change
18 // unregister() to register() below. Note this comes with some pitfalls.
19 // Learn more about service workers: https://bit.ly/CRA-PWA
20 serviceWorker.unregister();
21
```

測試成功



定義 SearchBox_76 component

```
src > JS SearchBox.js > ...
1 import React from 'react'
2
3 class SearchBox_76 extends React.Component {
4   render() {
5     return(
6       <div className = 'pa2'>
7         <input
8           className = 'pa3 ba b--green bg-lightest-blue'
9             type = 'search'
10            placeholder = 'search companies'
11          />
12        </div>
13      )
14    }
15  }
16
17 export default SearchBox_76;
```

在 App 中多 render SearchBox_76

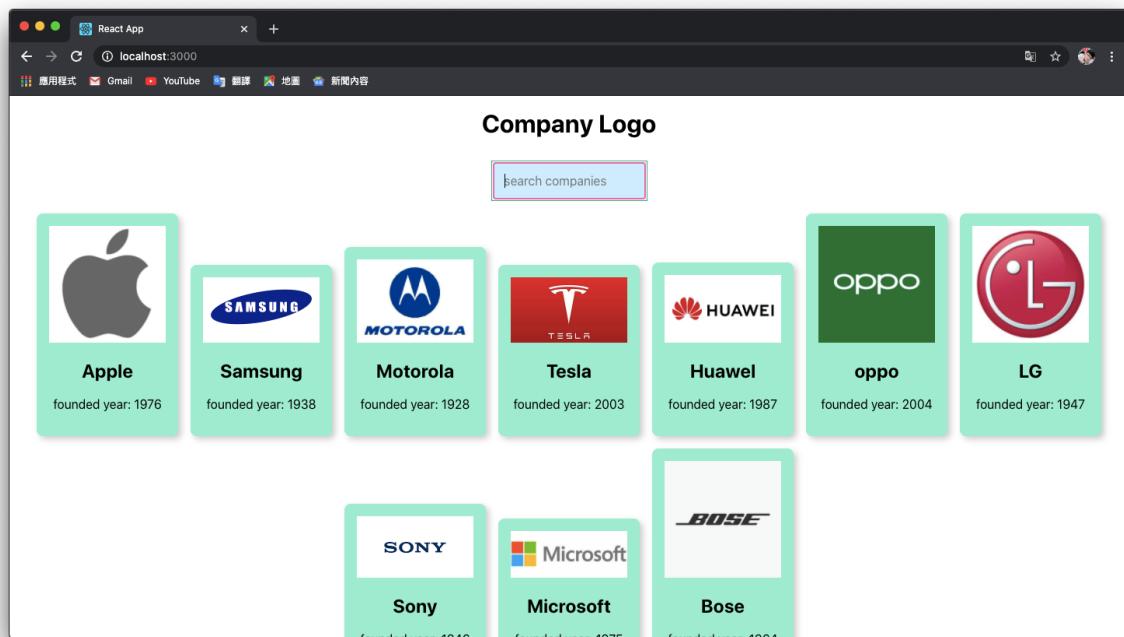
The screenshot shows a code editor with the following file structure:

- File Explorer: HOMEWORK_76 folder containing robots.txt, samsung.png, sony.png, tesla.jpg, and a src folder.
- src folder contains JS files: App.js, Company.js, CompanyList.js, index.css, index.js, and logos.js.
- Editor tab: App.js (highlighted)
- Code content:

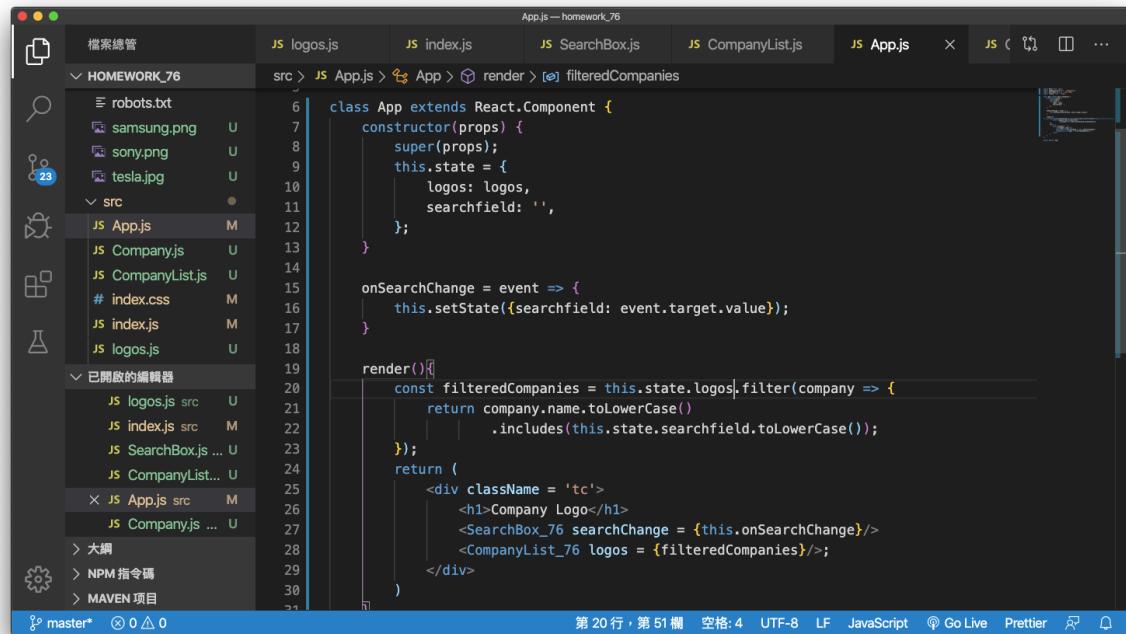
```
App.js — homework_76
src > JS App.js > ...
2 import CompanyList_76 from './CompanyList'
3 import SearchBox_76 from './SearchBox'
4 import {logos} from './logos'
5
6 class App extends React.Component {
7   render() {
8     return (
9       <div className = 'tc'>
10      <h1>Company Logo</h1>
11      <SearchBox_76/>
12      <CompanyList_76 logos = {logos}/>;
13    </div>
14  }
15}
16
17 export default App;
```

- Bottom status bar: 第 19 行，第 1 欄 空格:4 UTF-8 LF JavaScript Go Live Prettier

成功



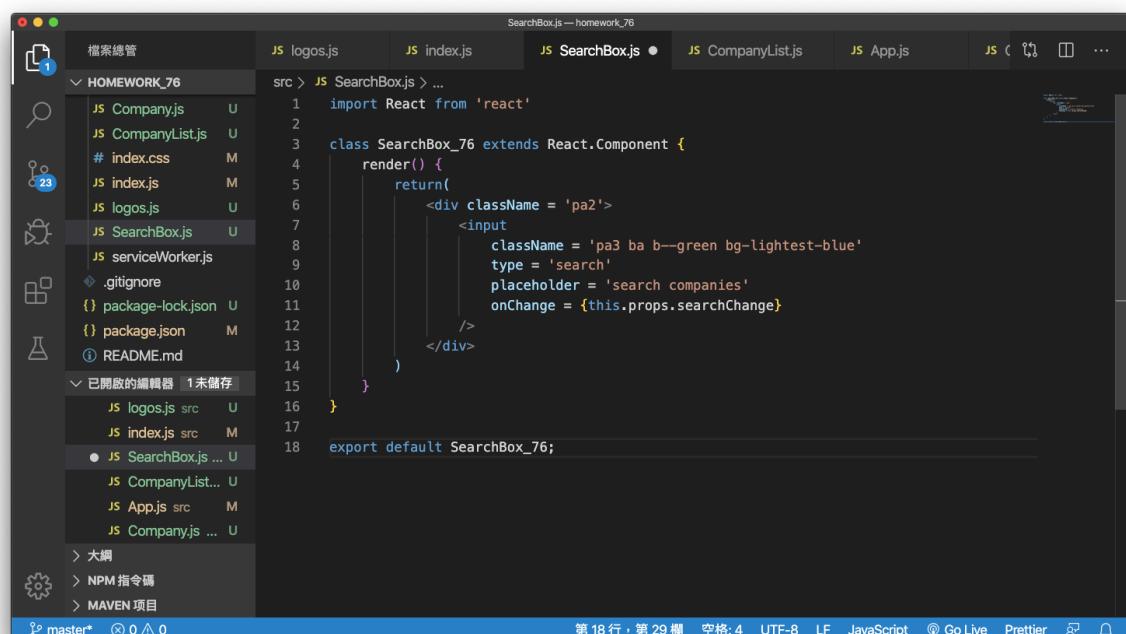
實作搜尋的邏輯：把 logos 跟搜尋的目標(searchfield)存進 App 的 state 中，利用 filter 找出 name 包含 searchfield 的 company，將這些 company 傳入 CompanyList_76 中。



```
App.js — homework_76
src > JS App.js > App > render [e] filteredCompanies
6   class App extends React.Component {
7     constructor(props) {
8       super(props);
9       this.state = {
10         logos: logos,
11         searchfield: ''
12     };
13   }
14
15   onSearchChange = event => {
16     this.setState({searchfield: event.target.value});
17   }
18
19   render() {
20     const filteredCompanies = this.state.logos.filter(company => {
21       return company.name.toLowerCase()
22         .includes(this.state.searchfield.toLowerCase());
23     });
24     return (
25       <div className = 'tc'>
26         <h1>Company Logo</h1>
27         <SearchBox_76 searchChange = {this.onSearchChange}>/
28         <CompanyList_76 logos = {filteredCompanies}>/
29       </div>
30     );
31   }
32 }

第 20 行，第 51 欄 空格:4 UTF-8 LF JavaScript ⚡ Go Live Prettier ⚡ 🔍
```

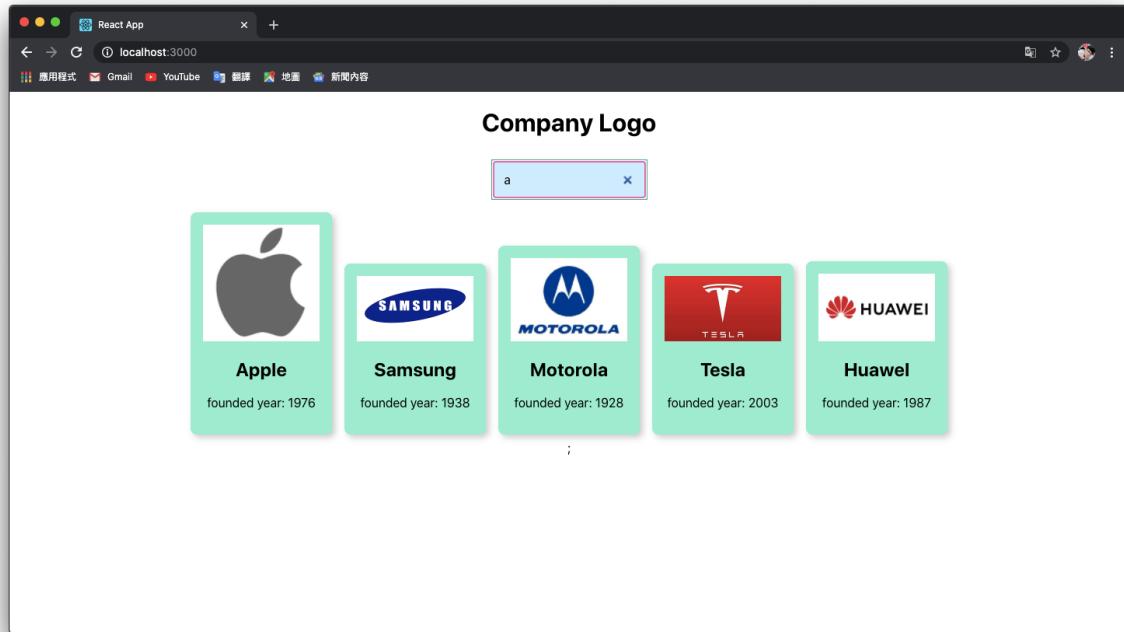
設定 onChange 為 App 傳入的 seachChange



```
SearchBox.js — homework_76
src > JS SearchBox.js > ...
1   import React from 'react'
2
3   class SearchBox_76 extends React.Component {
4     render() {
5       return(
6         <div className = 'pa2'>
7           <input
8             className = 'pa3 ba b--green bg-lightest-blue'
9               type = 'search'
10              placeholder = 'search companies'
11              onChange = {this.props.searchChange}
12            />
13         </div>
14       )
15     }
16   }
17
18   export default SearchBox_76;

第 18 行，第 29 欄 空格:4 UTF-8 LF JavaScript ⚡ Go Live Prettier ⚡ 🔍
```

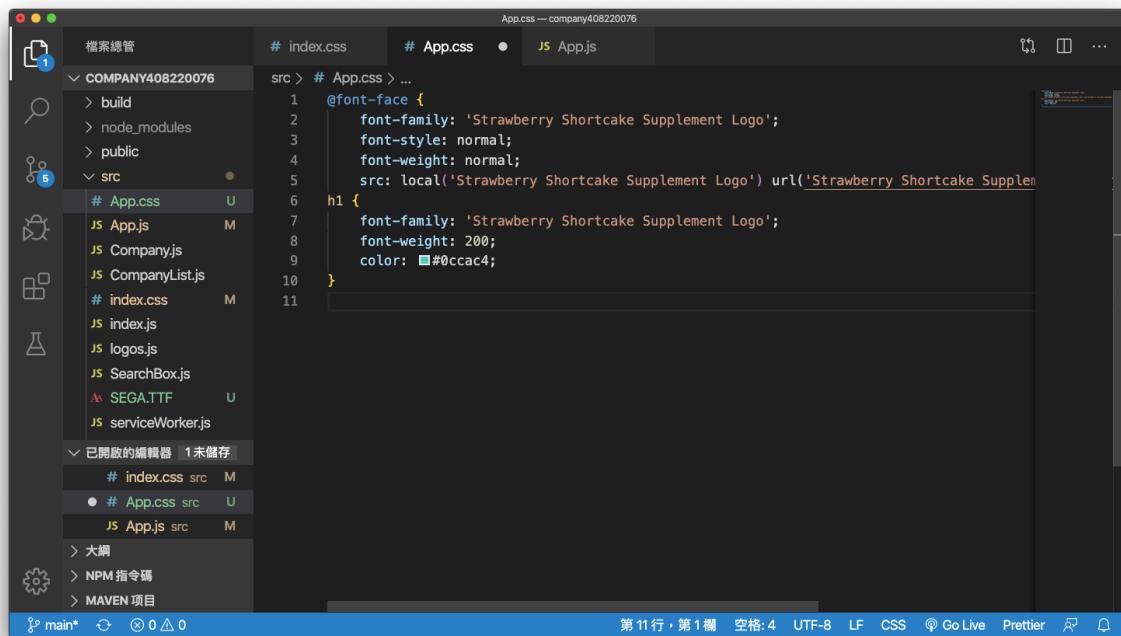
成功



修改背景

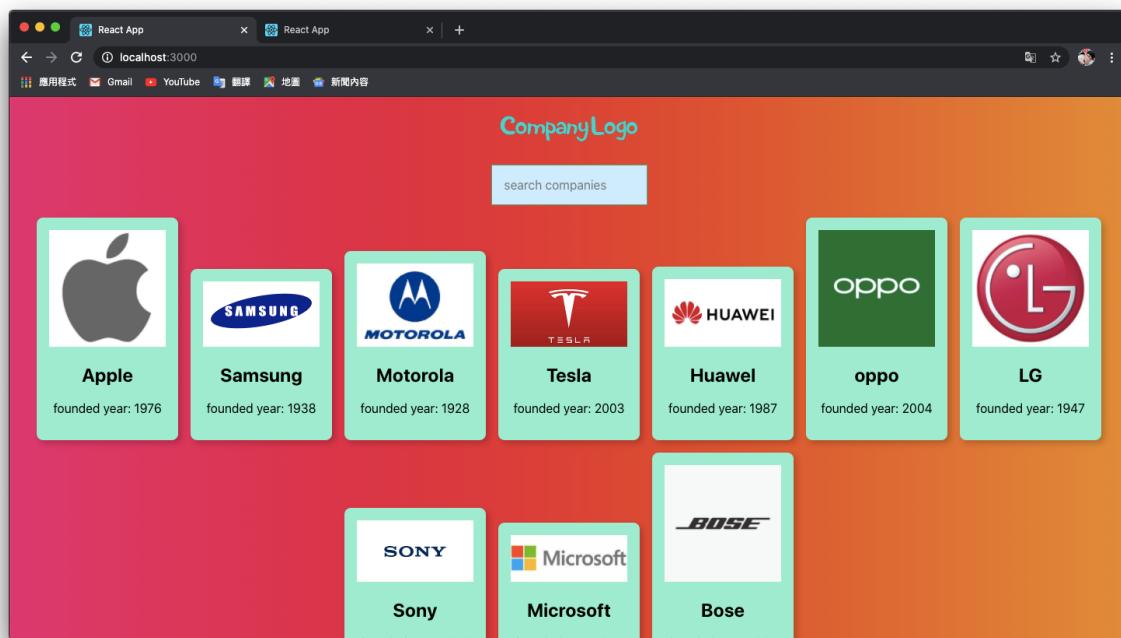
A screenshot of a code editor showing the file "index.css" with the title "index.css — company408220076". The code defines a gradient background for the body element and a monospace font for the code block. The code editor interface includes a sidebar with project files like "App.js", "Company.js", and "CompanyList.js", and a bottom status bar with file information and icons.

修改字體

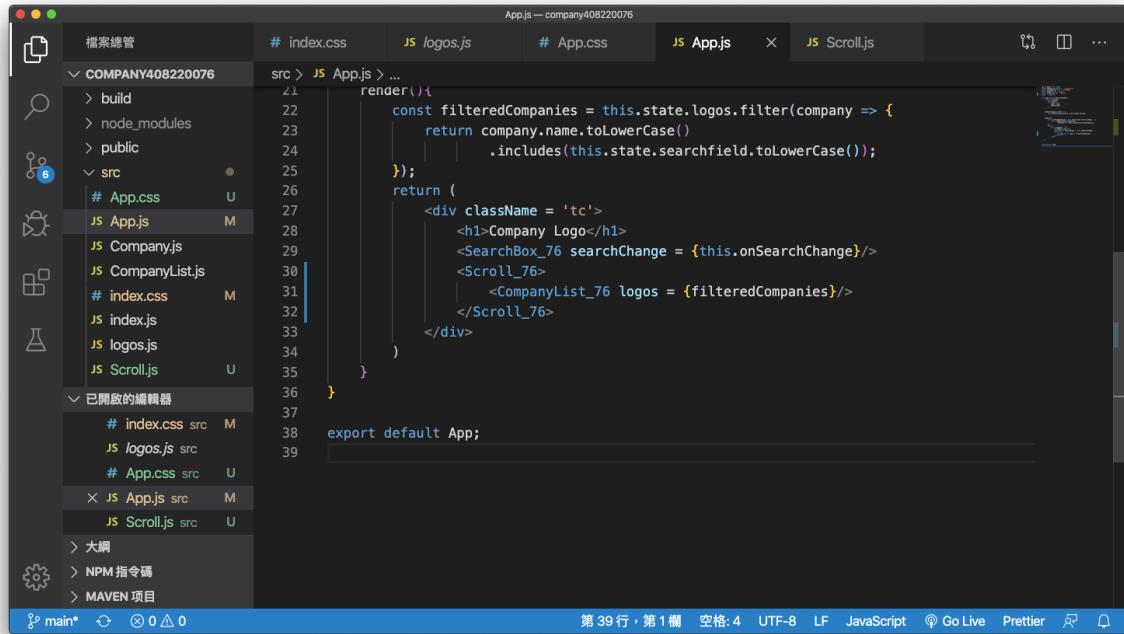


```
App.css — company408220076
# index.css # App.css ● JS App.js
src > # App.css > ...
1 @font-face {
2   font-family: 'Strawberry Shortcake Supplement Logo';
3   font-style: normal;
4   font-weight: normal;
5   src: local('Strawberry Shortcake Supplement Logo') url('Strawberry Shortcake Supplement Logo.woff2') format('woff2'), local('Strawberry Shortcake Supplement Logo') url('Strawberry Shortcake Supplement Logo.woff') format('woff');
6 h1 {
7   font-family: 'Strawberry Shortcake Supplement Logo';
8   font-weight: 200;
9   color: #0ccac4;
10 }
11 
```

成功

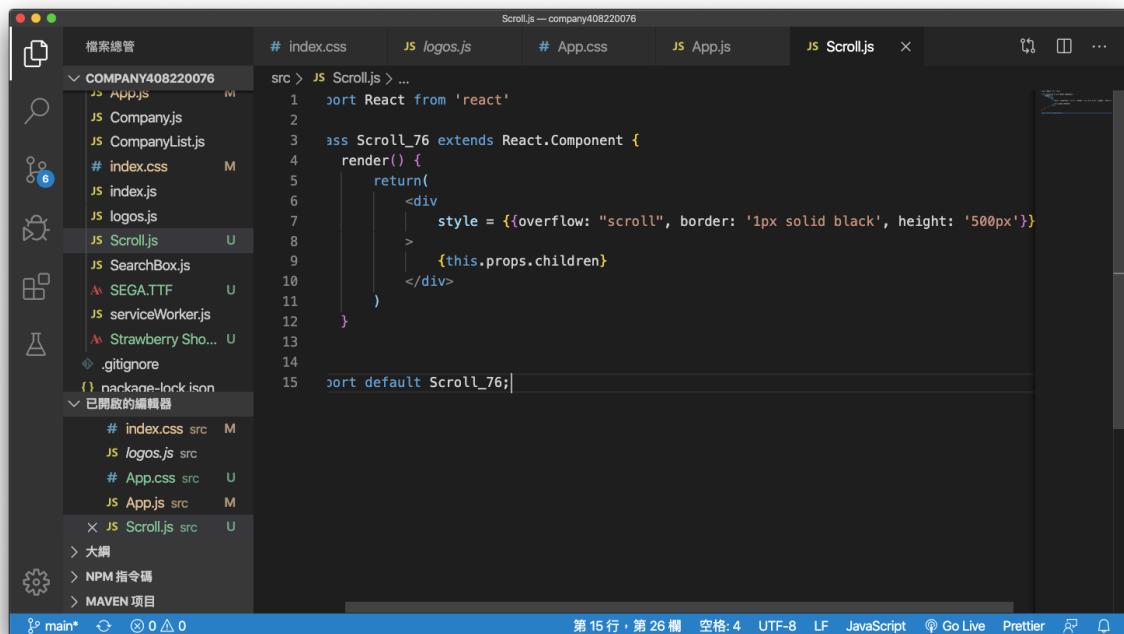


把 CompanyList_76 包進 Scroll_76



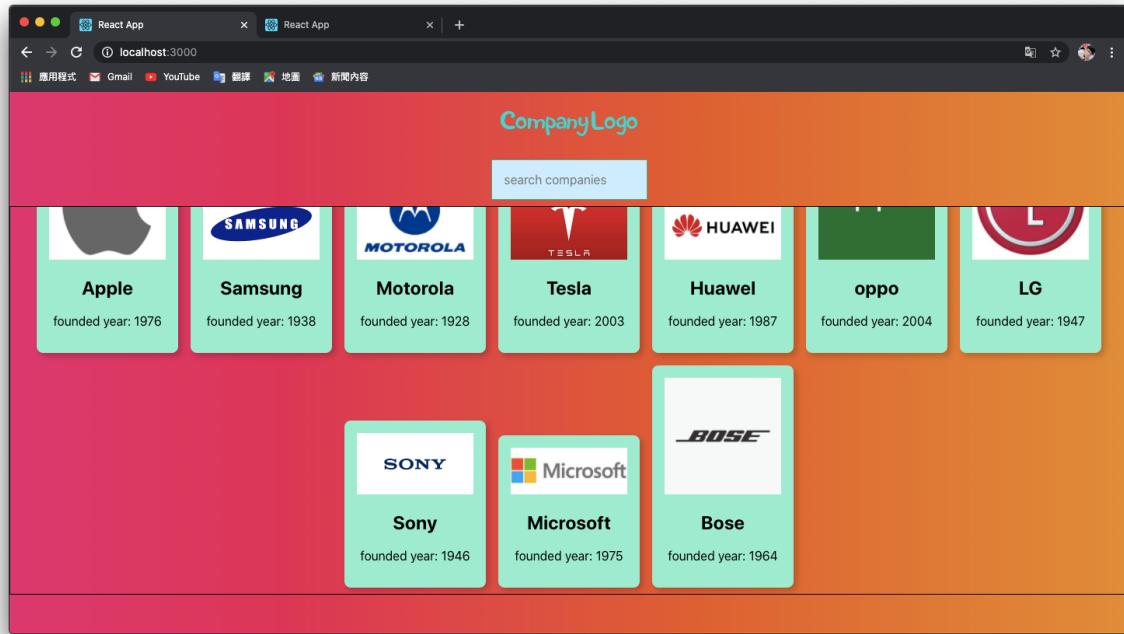
```
App.js — company408220076
src > JS App.js > ...
  render() {
    const filteredCompanies = this.state.logos.filter(company => {
      return company.name.toLowerCase()
        .includes(this.state.searchfield.toLowerCase());
    });
    return (
      <div className = 'tc'>
        <h1>Company Logo</h1>
        <SearchBox_76 searchChange = {this.onSearchChange}>/>
        <Scroll_76>
          <CompanyList_76 logos = {filteredCompanies}>/>
        </Scroll_76>
      </div>
    );
  }
  export default App;
```

在 Scroll_76 實作滾輪效果



```
Scroll.js — company408220076
src > JS Scroll.js > ...
  import React from 'react';
  class Scroll_76 extends React.Component {
    render() {
      return(
        <div
          style = {{overflow: "scroll", border: '1px solid black', height: '500px'}}
        >
          {this.props.children}
        </div>
      );
    }
  }
  export default Scroll_76;
```

成功



上傳 github

```
Published
DaChingLo@MacBook-xiaoqing ▶ ~/company408220076 ▶ main ± git add -A
DaChingLo@MacBook-xiaoqing ▶ ~/company408220076 ▶ main + git commit -m "finish"

DaChingLo@MacBook-xiaoqing ▶ ~/company408220076 ▶ main ➔ git push
枚舉物件: 35, 完成.
物件計數中: 100% (35/35), 完成.
使用 4 個執行緒進行壓縮
壓縮物件中: 100% (24/24), 完成.
寫入物件中: 100% (24/24), 169.92 KiB | 4.59 MiB/s, 完成.
總共 24 (差異 7), 復用 0 (差異 0), 重用包 0
remote: Resolving deltas: 100% (7/7), completed with 7 local objects.
To https://github.com/lodaching/company408220076.git
  a53a9db..13dec2c main -> main
DaChingLo@MacBook-xiaoqing ▶ ~/company408220076 ▶ main ➔
```