React from scratch

2016010873 박정욱

목차

- React란?
- 툴체인

React란?

- 사용자 인터페이스를 만들기 위한 JavaScript **라이브러리**
- 프레임워크(뼈대)가 아닌, 라이브러리(도구)임에 주의!

React란?

툴체인

- 패키지 매니저
- 번들러
- 트랜스파일러

패키지 매니저

- 설치/최신화/설정/제거 등의 자동화
- 자바스크립트의 경우 **npm**, yarn, pnpm 등이 존재함

패키지 매니저


package.json
package.json > ...
Debug
{
"name": "react-from-scratch",
"version": "1.0.0",
"description": "",
"main": "index.js",
"scripts": {
"test": "echo \"Error: no test specified\" && exit 1",
"start": "webpack serve",
"build": "webpack --mode development"
},
"author": "",
"license": "ISC",
"devDependencies": {
"@babel/core": "^7.13.1",
"@babel/preset-env": "^7.13.5",
"@babel/preset-react": "^7.12.13",
"babel-loader": "^8.2.2",
"css-loader": "^5.0.2",
"style-loader": "^2.0.0",
"webpack": "^5.24.1",
"webpack-cli": "^4.5.0",
"webpack-dev-server": "^3.11.2"
},
"dependencies": {
"react": "^17.0.1",
"react-dom": "^17.0.1",
"react-hot-loader": "^4.13.0"
}
}

번들러

- 웹의 발전으로 인해, '모듈'의 필요성이 대두됨
- **Webpack**, Rollup, Parcel 등이 존재함

파일 개수를 줄여서 네트워크 요청을 줄이자!	유지보수를 위해 큰 파일을 분할하여 가독성을 늘리자!
한 파일에 엄청난 양의 코드가 있다면 가독성이 저하되어 유지보수가 어렵다!	요청에 응답하는 시간이 길어지면 사용자 경험에 나쁜 영향을 미친다!

번들러

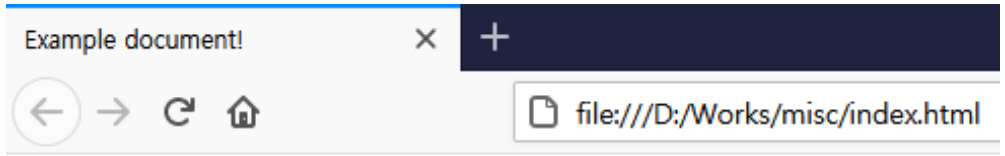
foo.js

```
var foo = 13;
```

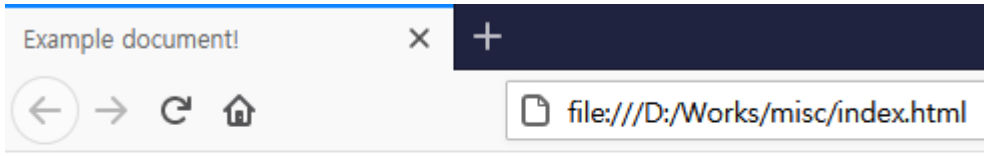
bar.js

```
var bar = foo + 42;
```

```
<body>
    <h1 id="h1-value"></h1>
    <script src="foo.js"></script>
    <script src="bar.js"></script>
    <script>document.getElementById("h1-value").innerText = bar;</script>
</body>
```




```
<body>
    <h1 id="h1-value"></h1>
    <script src="bar.js"></script>
    <script src="foo.js"></script>
    <script>document.getElementById("h1-value").innerText = bar;</script>
</body>
```



번들러

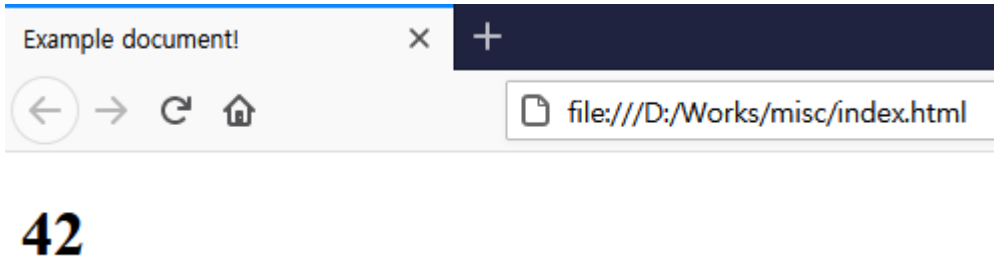
foo.js

```
var value = 13;
```

bar.js

```
var value = 42;
```

```
<body>
    <h1 id="h1-value"></h1>
    <script src="foo.js"></script>
    <script src="bar.js"></script>
    <script>document.getElementById("h1-value").innerText = value;</script>
</body>
```



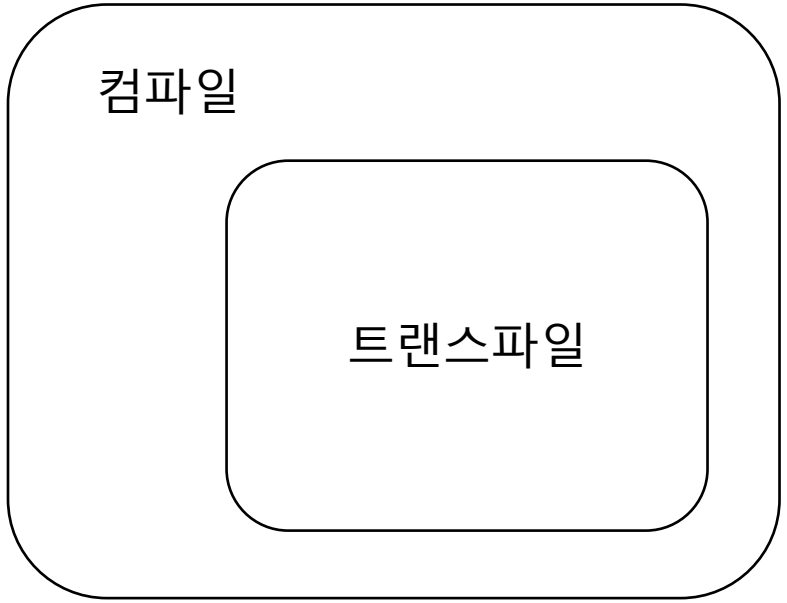
번들러

webpack.config.js

webpack.config.js > <unknown> > module > rules

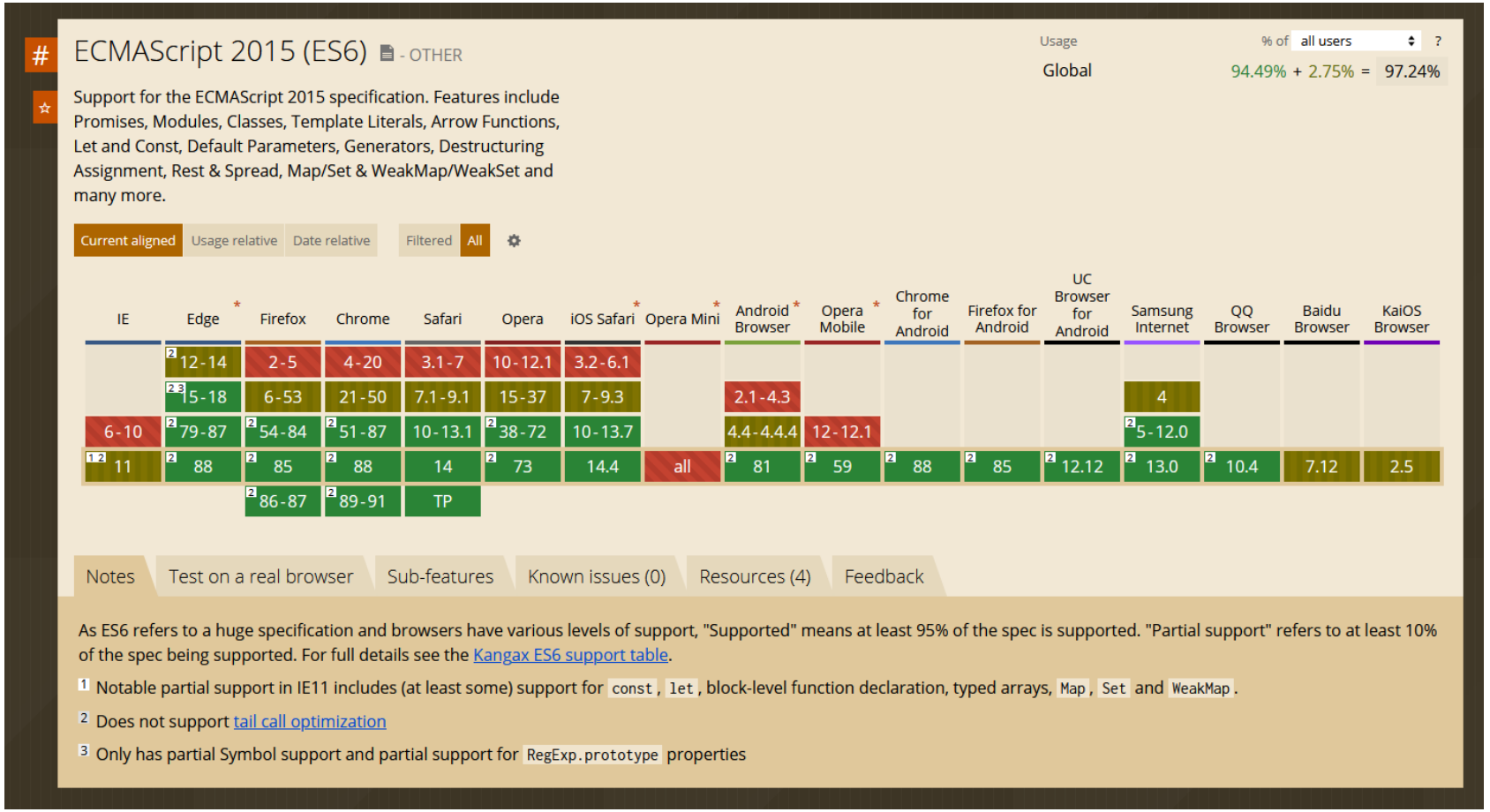
```
const path = require("path");
const webpack = require("webpack");

module.exports = {
    entry: "./src/index.js",
    mode: "development",
    module: {
        rules: [
            {
                test: /\.(js|jsx)$/,
                exclude: /(node_modules|bower_components)/,
                loader: "babel-loader",
                options: { presets: ["@babel/env"] }
            },
            {
                test: /\.css$/,
                use: ["style-loader", "css-loader"]
            }
        ]
    },
    resolve: { extensions: ["*", ".js", ".jsx"] },
    output: {
        path: path.resolve(__dirname, "dist/"),
        publicPath: "/dist/",
        filename: "bundle.js"
    },
    devServer: {
        contentBase: path.join(__dirname, "public/"),
        port: 3000,
        publicPath: "http://localhost:3000/dist/",
        hotOnly: true
    },
    plugins: [new webpack.HotModuleReplacementPlugin()]
};
```

트랜스파일러



컴파일	트랜스파일
고급 언어 → 저급 언어	고급 언어 → 고급 언어

트랜스파일러



ECMAScript 2015 (ES6) - OTHER

Support for the ECMAScript 2015 specification. Features include Promises, Modules, Classes, Template Literals, Arrow Functions, Let and Const, Default Parameters, Generators, Destructuring Assignment, Rest & Spread, Map/Set & WeakMap/WeakSet and many more.

Usage: Global — % of all users: 94.49% + 2.75% = 97.24%

Current aligned | Usage relative | Date relative | Filtered | All

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser
	12-14	2-5	4-20	3.1-7	10-12.1	3.2-6.1										
	15-18	6-53	21-50	7.1-9.1	15-37	7-9.3		2.1-4.3					4			
6-10	79-87	54-84	51-87	10-13.1	38-72	10-13.7		4.4-4.4.4	12-12.1				5-12.0			
11	88	85	88	14	73	14.4	all	81	59	88	85	12.12	13.0	10.4	7.12	2.5
		86-87	89-91	TP												

Notes | Test on a real browser | Sub-features | Known issues (0) | Resources (4) | Feedback

As ES6 refers to a huge specification and browsers have various levels of support, "Supported" means at least 95% of the spec is supported. "Partial support" refers to at least 10% of the spec being supported. For full details see the Kangax ES6 support table.

1 Notable partial support in IE11 includes (at least some) support for `const`, `let`, block-level function declaration, typed arrays, `Map`, `Set` and `WeakMap`.

2 Does not support tail call optimization

3 Only has partial Symbol support and partial support for `RegExp.prototype` properties

트랜스파일러


.babelrc
.babelrc > ...
1 {
2 "presets": ["@babel/preset-env", "@babel/preset-react"]
3 }

예제

index.html

public > index.html > ...

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <div id="root"></div>
    <script src="../dist/bundle.js"></script>
</body>
</html>
```

예제

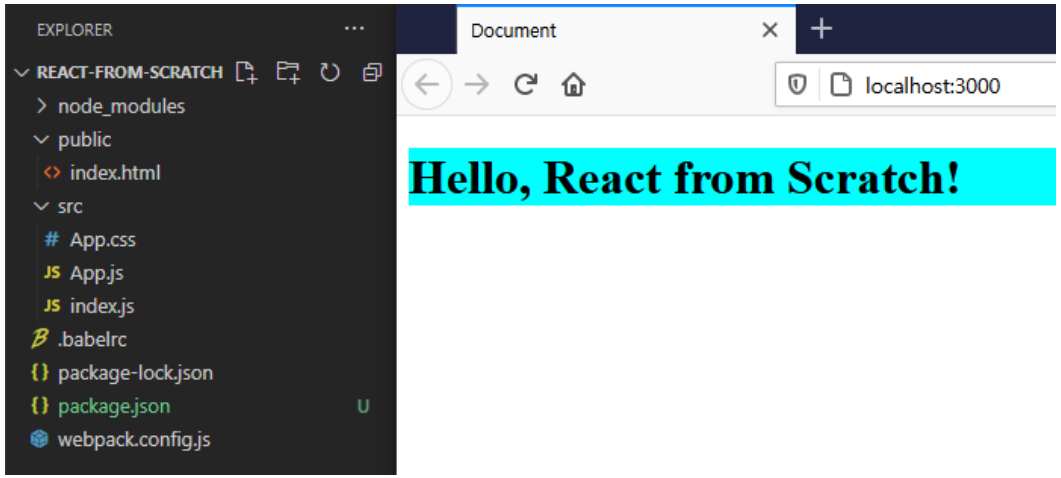
index.js

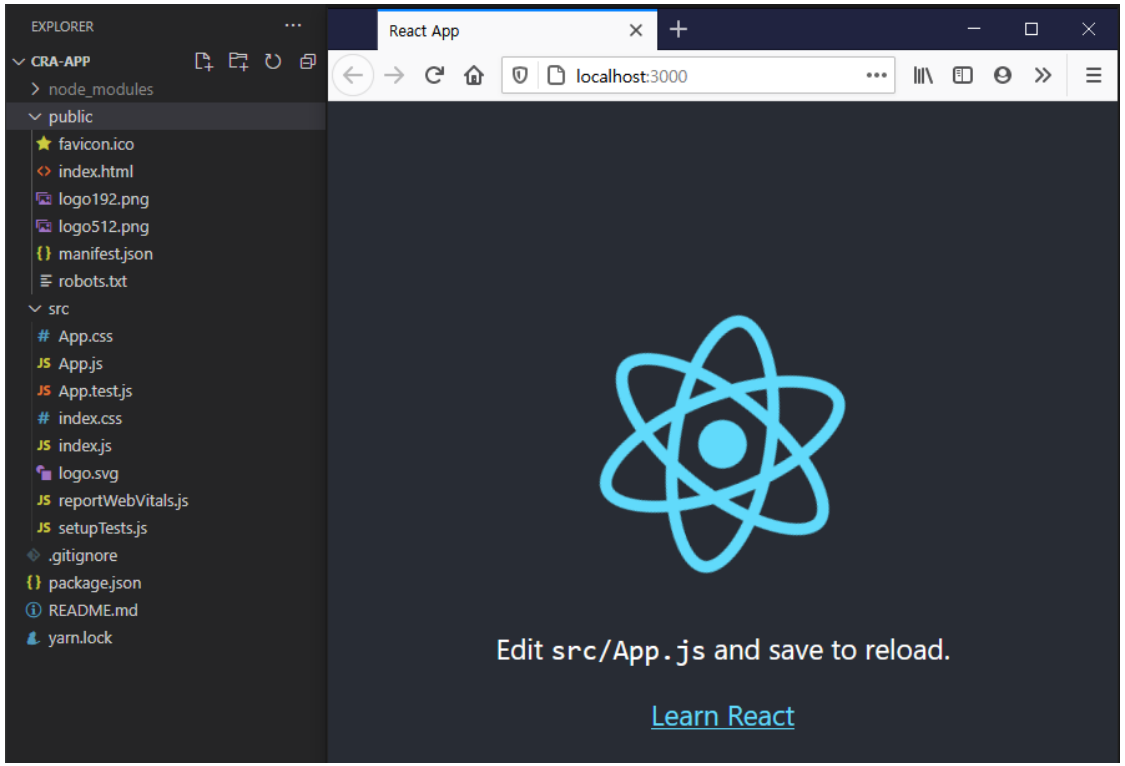
```
src > index.js
1  import React from "react";
2  import ReactDOM from "react-dom";
3
4  import App from "./App";
5
6  ReactDOM.render(<App />, document.getElementById("root"));
```

App.js

```
src > App.js > ...
1  import React from "react";
2  import { hot } from "react-hot-loader";
3
4  import "./App.css";
5
6  const App = () => {
7      return (
8          <div className="App">
9              <h1>Hello, React from Scratch!</h1>
10         </div>
11     );
12 }
13
14 export default hot(module)(App);
```

예제


EXPLORER
REACT-FROM-SCRATCH
node_modules
public
index.html
src
App.css
App.js
index.js
.babelrc
package-lock.json
package.json
U
webpack.config.js
Document
localhost:3000
Hello, React from Scratch!


EXPLORER
CRA-APP
node_modules
public
favicon.ico
index.html
logo192.png
logo512.png
manifest.json
robots.txt
src
App.css
App.js
App.test.js
index.css
index.js
logo.svg
reportWebVitals.js
setupTests.js
.gitignore
package.json
README.md
yarn.lock
React App
localhost:3000
Edit src/App.js and save to reload.
Learn React

Todo

- Redux 및 Flux 패턴
- CATLAS의 프론트엔드 도구 적용 현황