

day32-react

☰ 태그	
📅 날짜	@2022년 11월 16일

※집에서 해보니 npm 명령어가 먹히지 않았다.

찾아보니 nodejs가 설치 되어 있지 않으면 먹히지 않는다고 한다.

starter-code 열기 → 터미널 npm install (dependencies에 있는것들을 다 설치해줌) →

```
ow() and performance.timeOrigin.  
npm WARN deprecated svgo@1.3.2: This SVGO version is no longer supported. Upgrade to v2.x.x.  
  
added 1441 packages, and audited 1442 packages in 1m  
  
229 packages are looking for funding  
  run `npm fund` for details  
  
6 high severity vulnerabilities  
  
To address all issues (including breaking changes), run:  
  npm audit fix --force  
  
Run `npm audit` for details.  
npm notice  
npm notice New major version of npm available! 8.19.2 -> 9.1.2  
npm notice Changelog: https://github.com/npm/cli/releases/tag/v9.1.2  
npm notice Run npm install -g npm@9.1.2 to update!  
npm notice  
D:\Develop\react\Starter-Code>
```

(npm install로 다른것들을 설치하면 package.json에 추가된다.)



Edit `src/App.js` and save to reload.

[Learn React](#)

리액트 화면이 뜨는것 까지 확인했다.

app.css와 index.css는 삭제했다. 또한 안쓰는 코드도 주석처리 한다.

```
index.js
import {BrowserRouter} from 'react-router-dom';
링크를 통해서 데이터를 주고 받고 하는걸 리액트에서도 가능하게 해준다.
```

Styles폴더를 생성하고 GlobalStyles.js파일 생성 한다.

fontsource 라는 사이트에서 폰트를 가져온다. 구글 폰트도 있다.

오늘은 구글 폰트에서 가져온다.

sora라는 폰트를 받는다.

yarn add @fontsource/sora ← 이 명령어로 했으나 yarn이 먹히지 않아서

npm add @fontsource/sora로 다운로드를 진행 하였다.

yarn add @fontsource/akaya-telivigala 도 설치를 진행한다. 위와 같이

npm add @fontsource/akaya-telivigala 으로 설치를 진행하였다.

폰트 관련된 설치를 마쳤다.

```
yarn add @fontsource/akaya-telivigala
```

```
import "@fontsource/akaya-telivigala"
```

```
body { font-family: "Akaya Telivigala", cursive; }
```

import로 적용하는 방법이 설명되어있다.

```

src > styles > JS GlobalStyles.js > ...
1  import { createGlobalStyle } from "styled-components";
2  import "@fontsource/akaya-telivigala"
3  import "@fontsource/sora"
4
5  const createGlobalStyle = createGlobalStyle`
6
7  *,*::before,*::after{
8      margin:0;
9      padding:0;
10 }
11 body{
12     font-family: ;
13 }
14
15
16
17 `

```

import로 적용시켜준다.

styles폴더에 Themes.js라는 파일을 만든다. 테마를 잡아주는 역할을 할것이다.

```

return (
  <>
    <GlobalStyles />
    <ThemeProvider theme={light}>
      App File
    </ThemeProvider>
  </>
);
}

export default App;

```

light 나 dark냐에 따라 테마가 달라진다.

src 밑에 components 폴더를 생성하고 그 밑에 Navigation.js라는 파일을 생성한다.

```
1  import React from "react";
2
3  const Navigation = () => {
4
5      return (
6          <div>
7
8
9          </div>
10
11      );
12
13  }
14
15
16  export default Navigation;
```

함수형으로 기본 구조를 만들어준다.

Tagged Template Literal 문법 (``)

을 통해 해당 컴포넌트의 props 를 읽어올 수도 있다.