**만든 리액트 사이트 build & Github Pages로 배포해보기**

이번시간은 간단한 내용이기 때문에 글로 빠르게 진행합니다.

여러분이 만든 사이트를 배포하려면 그냥 작업하던 App.js 파일 그대로 올리는게 아니라

build용 파일을 생성하신 후 그걸 올려야합니다.

왜냐고요? 웹브라우저는 HTML CSS JS 이 세개의 언어만 해석할 수 있습니다. 리액트의 이상한 state, JSX 이런거 전혀 못알아듣습니다.

그래서 리액트 프로젝트를 build 라는걸 하시면 브라우저 친화적인 HTML CSS JS 파일로 바꿔줍니다.

그리고 그걸 어딘가 서버에 올려야 사용자들이 여러분의 사이트를 구경할 수 있는 것임

우리도 build를 해본 후 무료로 HTML 파일을 호스팅해주는 Github Pages를 이용해 배포까지 해보도록 합시다.

**빨리 github.com 들어가서 계정을 먼저 하나 생성해오십시오.**

**Q. 웹서버 가지고 있는데 여기에 배포는 어떻게 합니까?**

서버를 만들줄 아는 똑똑이라면

리액트는 이쁜 HTML파일 만들어주는 툴이기 때문에

리액트로 열심히 프로젝트 만들고 npm run build 입력하면 build/index.html 파일이 생성됩니다.

그리고 **"어떤 놈이 메인페이지로 접속하면 /build/index.html 파일 전송해라"**

라고 서버 API를 작성하면 간단한 배포가 끝납니다.

[collapse]

**0. 배포하기 전 체크할 사항**

**(1) 일단 미리보기 띄워볼 때 콘솔창, 터미널에 에러만 안나면 됩니다.**

warning 메세지는 사이트 구동에 별 영향이 없기 때문에 테스트해보실 땐 개무시해도 됩니다.

**(2) 혹시 사이트를 배포할 때**

http://codingapple.com/ 여기에 배포하는 경우엔 따로 설정이 필요없이 대충 해도 되지만

http://codingapple.com/blog/ 이런 하위 경로에 배포하고 싶으면 프로젝트에 설정이 따로 필요합니다.

여러분의 프로젝트 파일 중 package.json 이라는 파일을 오픈해서

"homepage": "http://codingapple.com/blog",

homepage라는 항목을 추가한 후

여러분이 배포할 사이트 경로를 뒤에 추가하면 됩니다. (혹은 /blog 이렇게 경로 쓰셔도 됩니다)

그리고 리액트 라우터가 설치되어있다면 라우터가 제공하는 basename="" 속성을 추가하는게 라우팅 잘될겁니다.

자세한 내용은 <https://create-react-app.dev/docs/deployment/#building-for-relative-paths>

**1. 별 문제가 없다면 이제 터미널에 build 명령어를 입력하십시오.**

여러분이 작성한 state, JSX, <컴포넌트>, props 이런 문법들은 브라우저가 해석할 수 없으니 그대로 배포할 수 없습니다.

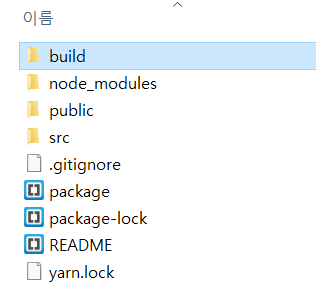
그래서 이런 문법들을 역사와 전통의 CSS, JS, HTML 문법으로 바꿔주는 작업이 필요합니다.

이것을 컴파일 또는 build라고 합니다.

build 하려면 여러분의 리액트프로젝트에서 터미널을 켠 후

npm run build

입력하면 됩니다.



▲ 그럼 여러분 작업 프로젝트 폴더 내에 build 라는 폴더가 하나 생성되는데

여러분이 짰던 코드를 전부 .html .css .js 파일로 변환해서 거기 담아줌

이제 build 폴더 안에 안에 있는 내용을 모두 서버에 올리면 됩니다.

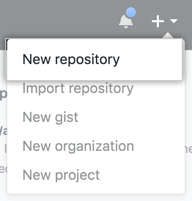
index.html이 메인페이지입니다.

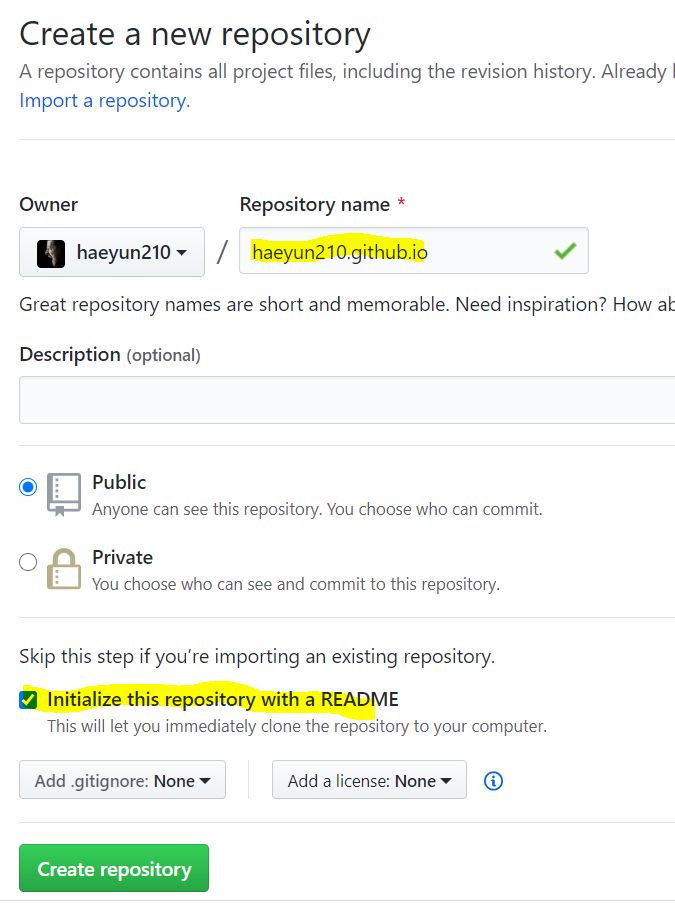
끝

**2. 근데 우린 무료 호스팅해주는 github pages에 올릴겁니다**

간단하게 HTML/CSS/JS 파일을 무료로 호스팅해주는 고마운 사이트입니다.

일단 github.com에 들어가셔서 로그인까지 하십시오.

▼ 그 다음엔 우측 상단 + 버튼을 누르셔서 New Repository 버튼을 누르십시오. 

   ▼ 그 다음엔 노란 곳에 다음과 같이 입력합니다. 

▲ Repository name 은 꼭 **왼쪽에 뜨는 여러분아이디.github.io** 라고 입력하셔야합니다.

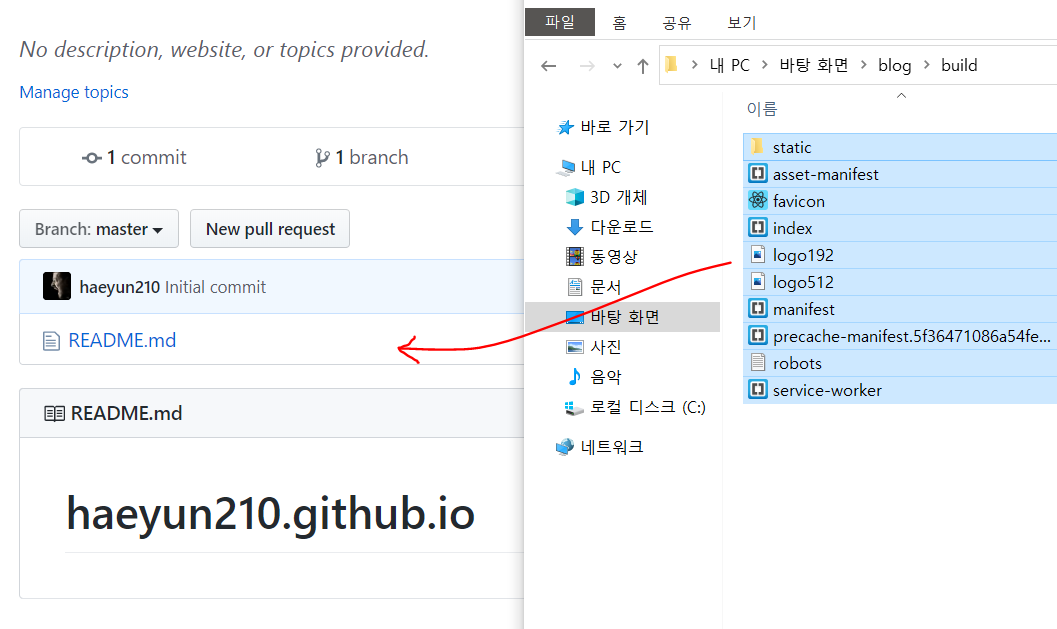
여러분아이디.github.io 말고 임의로 설정하시면 여러분 코딩인생 끝납니다.

그리고 README 파일 생성도 체크한 뒤에 생성해주시면 됩니다.

**3. Repository에 html 파일 올리기**

Repository 생성이 끝나면 repository로 자동으로 들어가질겁니다.

▼ 그럼 거기에 build 폴더 내의 파일을 전부 드래그 앤 드롭하면 됩니다.



(주의) build 폴더를 드래그 앤 드롭하는게 아닙니다. build 폴더 안의 내용물이요.

드래그 앤 드롭하시고 초록버튼까지 눌러주면 배포 끝입니다.

실수했다면 repository 과감하게 삭제하고 다시 만들면 됩니다.

이제 10분 정도 후에 **https://여러분아이디.github.io** 라고 주소창에 입력하면 여러분의 사이트가 보입니다.

이제 집가서 부모님께 자랑합시다.

**(흔한 github pages 에러) 왜 사이트 주소로 접속했는데 404 페이지가 뜨죠?**

- 10분 더 기다려보십시오.

- ctrl + shift + r 을 이용해 새로고침 해보십시오.

- 혹은 repository 생성하실 때 **님들아이디.github.io**가 아니라 뭔가 잘못적은겁니다.

정확히 안적었으면 그냥 다시 하나 새로 만듭시다.

**(추가) github이 좋아졌습니다.**

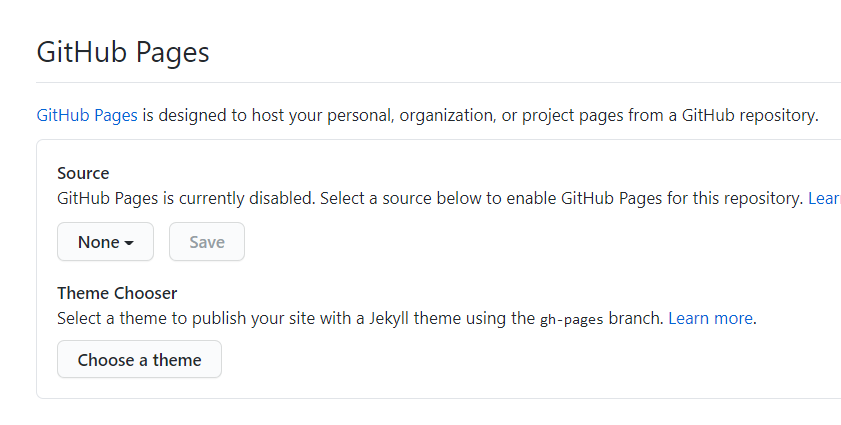
이제 여러 repository를 동시에 호스팅해준다고합니다. 다른 HTML 페이지도 호스팅받고 싶으면

0. 위에서 만든 내이름.github.io 라는 repository 잘 있죠? 그거 지우면 안됩니다.

1. 남에게 자랑하고픈 새로운 프로젝트를 올릴 repository를 새로 만들어줍니다. 이름은 아무렇게나 하시면 됩니다.

2. 그 repository에 아까처럼 드래그앤드롭으로 원하는 HTML CSS JS 파일을 업로드하고 확인까지 누릅니다.

3. repository setting 메뉴에 들어가서 Github pages 부분을 찾습니다.



 ▲4. 저기 source 부분을 None이 아니라 main 이런걸로 바꿔주고 저장하셈

5. 그럼 끝입니다. 이제 **님아이디.github.io/repository이름/** 으로 들어가시면 아까 업로드했던 HTML파일을 볼 수 있습니다.

안보이면

**님아이디.github.io/repository이름/html파일명.html**

이렇게 직접 들어가봅시다. 그리고 첫 업로드 후엔 보통 10~20분넘게 기다려야 반영됩니다.

**Q1. 첫 페이지 로딩 속도를 빠르게 하려면**

원래 리액트, 뷰로 만든 웹앱들은 첫 방문시 필요한 파일을 전부 로드합니다.

전송되는 파일 사이즈를 조금이라도 줄여서 빠르게 만들고 싶으면 컴포넌트들을 lazy하게 로딩하는 방법을 사용할 수도 있습니다.

공식 튜토리얼에 있는 lazy 함수 : <https://reactjs.org/docs/code-splitting.html#route-based-code-splitting>

**Q2. 업데이트 사항이 생기면 배포 또 어떻게하죠?**

build 또 하시고 그 파일 그대로 다시 업로드하면 됩니다.

build 할 때 마다 CSS, JS 파일 명이 무작위로 다시 생성됩니다.

그래서 새로 배포할 때마다 사이트 방문자들은 새로운 CSS,JS 파일을 이용할 수 있습니다.

**Q3. build 할 때 압축 시키지말고 남기고 싶은 파일은?**

여러분이 ./ 부터 시작하는 경로로 첨부한 이미지, js 파일들은 전부 짜부되고 이름이 변합니다.

이름이 변하지 않게 하고 싶으면 public 폴더안에 넣고 build 해보십시오.

그럼 build 하고 나서도 그대로 루트경로에 파일이 남아있습니다.

(개발시 그런 파일들을 이용하고 싶으면 public 폴더에 보관하고 ./ 이게 아닌 / 경로로 import 해오면 됩니다)

**Q4. 서버에 올렸는데 왜 접속하면 이상한 페이지가 나오거나 일부 img, css파일이 로드가 안되는 것이죠?**

대부분 경로 문제입니다.

- build 할 때 에러 안났겠죠

- 혹시 하위폴더에 배포한거 아닙니까

- 배포한 페이지가 안나오면 크롬개발자도구 열어서 index.html이 쓰고있는 css, js, img 파일들의 경로가 제대로 되어있는지 체크해보도록 합시다.

**Q5. 메인페이지 말고 왜 특정 페이지로 접속하면 404 에러가 뜨나요?**

**어쩌구.github.io/detail/1** 이렇게 세부 페이지 URL을 주소창에 직접 입력하시면

찾는 페이지가 없어요~ 이렇게 404 에러가 날 수 있습니다.

이건 서버에서**"누군가 어쩌구.github.io/어쩌구 로 접속하면 메인페이지로 안내하세요~"**라고 API 개발을 해놔야합니다.

근데 github은 우리가 서버를 만지고 어찌할 수 있는게 아니고 그냥 HTML 파일 올린것만 샤락 보여주는 곳이기 때문에

사이트 메뉴에다가 페이지 이동버튼을 잘 만들어두시면 되겠습니다.

아니면 URL에 #기호가 붙는 hashRouter를 리액트 라우터 코드짤 때 쓰든가요.