**Git 환경 구축 매뉴얼**

**Git이란**?

git은 소스코드 버전 관리시스템으로 로컬에서 변경 사항을 추적하고 원격 리소스에서 변경사항을 푸시하거나 가져올 수 있으며, 하나의 프로젝트를 여러 사람들과 함께 효과적으로 관리하여 협업할 수 있다.

**GitHub란?**

GitHub는 Git 원격 저장소 서비스입니다. Git에 대해서 잘 알지 못하는 사람들은 Git과 GitHub가 같다고 생각할 수 있다. 그러나 Git과 GitHub는 별개이며 Git을 통해 프로젝트를 진행하면서 저장소를 여러 사람이 공유하고 협업할 수 있는데, 이때 협업을 위해 프로젝트를 공유할 수 있는 네트워크상의 저장공간이 필요하고 이러한 네트워크상의(인터넷) 저장공간을 제공해 주는 서비스 중 하나가 GitHub인 것이다. 원격 저장소는 인터넷이 아니더라도 사내 사설 네트워크 안에 Git 원격 저장소 서버를 두고 관리할 수도 있으며, GitHub 이외에도 다양한 인터넷 Git 원격 저장소 서비스들이 존재한다.

**[Git Downloads]**

1. https://git-scm.com/downloads 접속하여 Downloads에서 windows 선택



2. 자신의 컴퓨터 bit에 맞춰 setup

텍스트이(가) 표시된 사진

자동 생성된 설명

3. 다운로드 완료 후 setup 프로그램 실행

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

4. “Git Bash Here”에 체크 표시를 해준 뒤 Next → 또 Next (시작메뉴에 추가할 것인지 물어보는 것)

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

5. 그 다음은 git에 관련된 세부설정 해주는 단계로, 모두 다 Next를 해주면서 진행하면 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

※ 왼쪽은 기본 설정이며, 오른쪽은 VS Code(Visual Studio Code)용 설정 (이 설정을 진행하려면 VS Code가 먼저 설치되어야 한다.)

텍스트이(가) 표시된 사진

자동 생성된 설명

<새 레파지토리 생성할 때 initial branch name 어떻게 지정할건지 묻는 것으로 그냥 Let Git Decide로 지정해주면 됨>

텍스트이(가) 표시된 사진

자동 생성된 설명

<환경설정 PATH 지정해주는 화면>

■ Use Git from Git Bash Only

Git Bash에서만 Git을 사용할 수 있게 하는 것

■ Git from the command line and also from 3rd-party software

커맨드라인과 다른 소프트웨어에서도 git 사용할 수 있는 것

■ Use Git and optional Unix tools from the Command Prompt

git뿐만 아니라 유닉스 도구들도 일부 사용할 수 있도록 PATH에 추가된다는건데 경고에 윈도우 tool과 충돌날 수 있다고 되어 있음

텍스트이(가) 표시된 사진

자동 생성된 설명

<https 커넥션 방식을 선택하는 화면>

■ Use the OpenSSL library

SSL 인증을 통과하는 것

■ Use the Native Windows Secure Channel library

윈도우 인증서 저장소를 사용하는 것 CA 방식 인증

텍스트이(가) 표시된 사진

자동 생성된 설명

<체크아웃과 커밋 방식을 묻는 화면>

체크아웃은 윈도우에서 작업하기 편하도록 하고, 커밋은 개행이 무조건 Unix-Style로 올라가게끔 해줌

텍스트이(가) 표시된 사진

자동 생성된 설명

<터미널 에뮬레이터를 어떤 것으로 쓸지에 대해 묻는 항목>

기본인 MinTTY를 선택

텍스트이(가) 표시된 사진

자동 생성된 설명

<Git pull 작업 방식을 물어보는 것으로, 기본 Default로 지정>

텍스트이(가) 표시된 사진

자동 생성된 설명

<DEPRECATED는 곧 사라질 예정으로 권장하지 않음. 비번 같은 신용 관련된 관리방법을 물어보는 화면으로 첫 번째 방식을 권장.>

텍스트이(가) 표시된 사진

자동 생성된 설명

<추가 옵션으로 파일시스템 캐싱과 심볼릭 링크가 가능하게 할지에 대한 선택항목으로 기본으로 지정된 것으로 해도 되지만, 속도면에서 좀 더 나아질 수 있다고 하여 둘 다 Check해도 된다.>

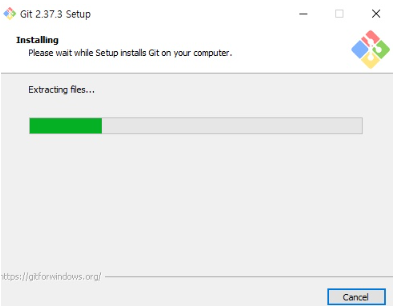
텍스트이(가) 표시된 사진

자동 생성된 설명

<개발중인 옵션 선택항목>

굳이 선택하지 않아도 되어 생략한 다음 Install 누르고 설치 시작

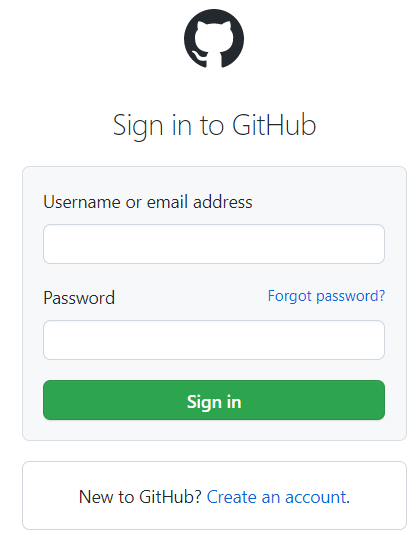
6. Git 설치



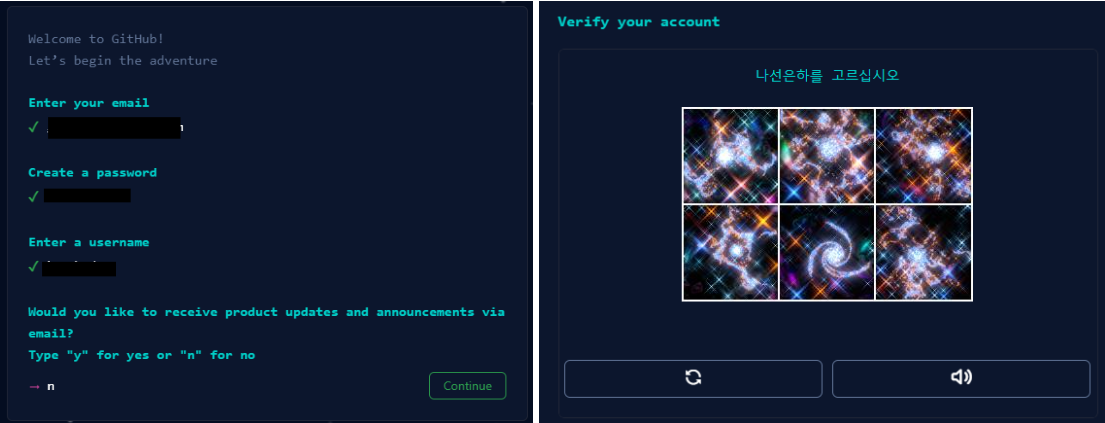
**[Github]**

**• 회원가입**

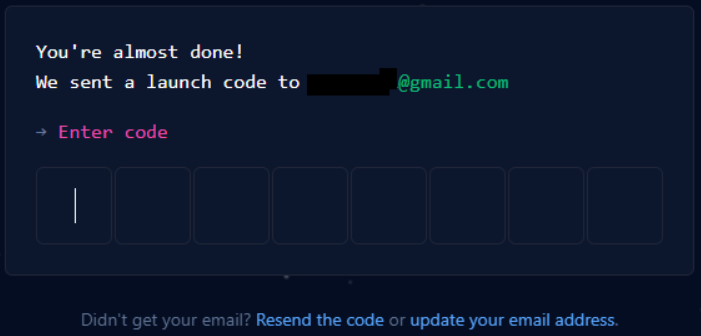
1. https://github.com/login 회원가입



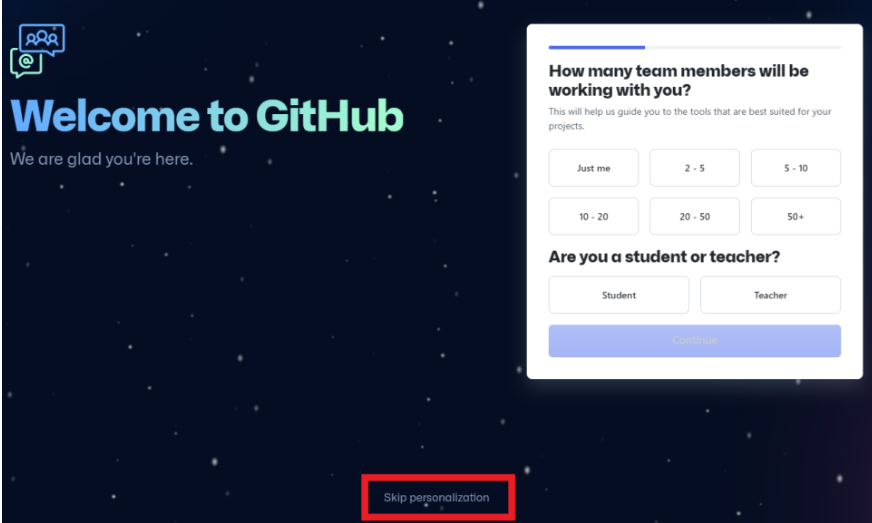
2. email, password, username 입력



3. 작성한 이메일로 입증 코드 발송



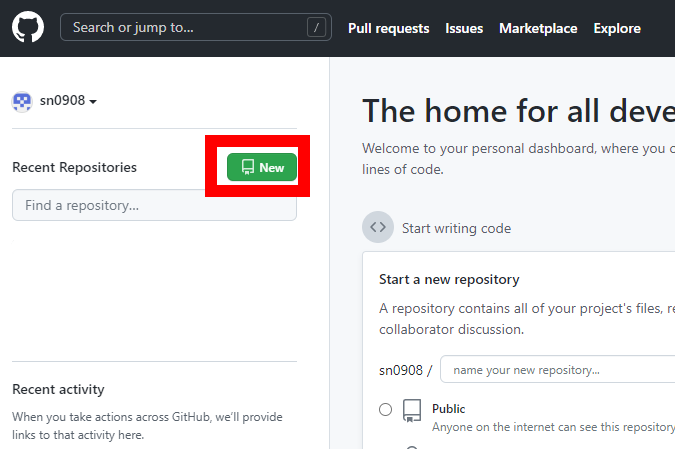
4. 추가 설정은 Skip



**[Github 사용 방법]**

**• Repository 생성**

1. log in 후 new 클릭 (새로운 Repository 생성)



2. Repository name 생성 → Public / Private 선택 (Private는 유료) → Add .gitignore / Add a license 선택 (None 선택) → 설정 완료 후 “Create repository”

gitignore이나 license에 대해서는 나중에 추가하거나 변경할 수 있으므로 None을 선택.

텍스트이(가) 표시된 사진

자동 생성된 설명

0

0

**• 파일 올리는 방법**

1. 작업을 저장할 폴더(텍스트이(가) 표시된 사진

자동 생성된 설명) 를 생성해 준다.

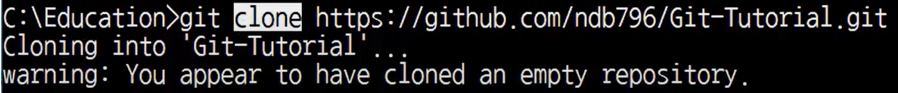
2. github에서 HTTPS에서 주소복사

텍스트이(가) 표시된 사진

자동 생성된 설명

3. 명령 프롬프트 창에서 아래와 같이 명령어와 함께 복사한 주소를 넣어준다.

작업 저장할 폴더로 이동 → git clone 복사한 주소 입력



\* clone은 다운로드라고 생각하면 된다.

4. clone이 완료되면 Github에서 생성했던 저장소의 이름이 폴더로 들어간 것을 확인할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

5. Git-Tutorial 폴더로 들어가면 아래와 같이 “.git” 이라는 폴더가 보인다.

텍스트이(가) 표시된 사진

자동 생성된 설명

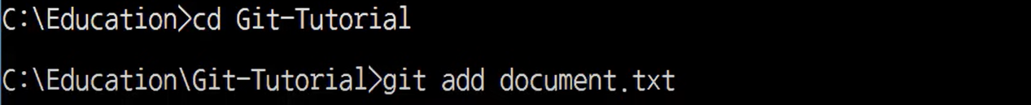
6. document.txt 파일을 만들어준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

이제 txt 파일을 github에 올리는 방법에 대하여 알아본다.

먼저 올릴 파일이 있는 폴더로 이동하고 아래 명령어를 순서대로 입력해준다.



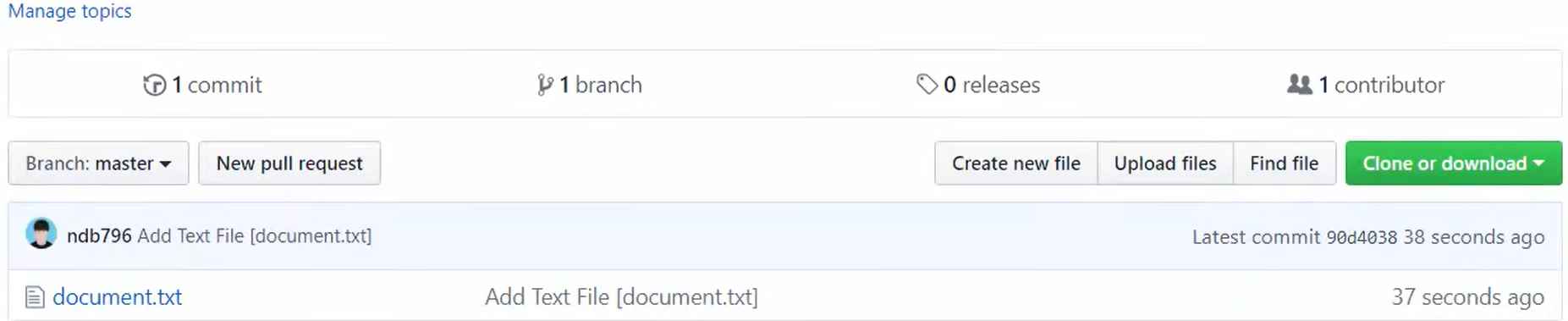
텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

이후 github 사이트에서 새로고침을 하게 되면 아래와 같이 파일이 올라온 것을 확인할 수 있다.



github에 파일을 올리는 방법에 대하여 정리하면 다음과 같다.

그러나 위의 방법처럼 git add 파일명(또는 git add .) / git commit -m / git push로도 파일을 업로드 할 수가 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

git init

그리고 올릴 파일을 add 해준다.

git add .

git add.을 하면 해당 폴더의 모든 파일들이 선택된다.

따라서 원하는 폴더가 있다면 add 뒤에 파일명 입력하시면 된다. ex) git add mysound.py

​

git commit -m "first commit"

커밋해줍니다.

​

git remote add origin https://github.com/본인git이름/repository이름.git

이걸로 로컬저장소와 원격저장소를 연결시켜 준다.

​혹시 이러한 에러가 뜨게 되면

fatal: remote origin already exists.

이 명령어로 제거를 해주면 된다.

git remote rm origin

​

그리고 push 명령어로 github에 올려준다.

git push origin master

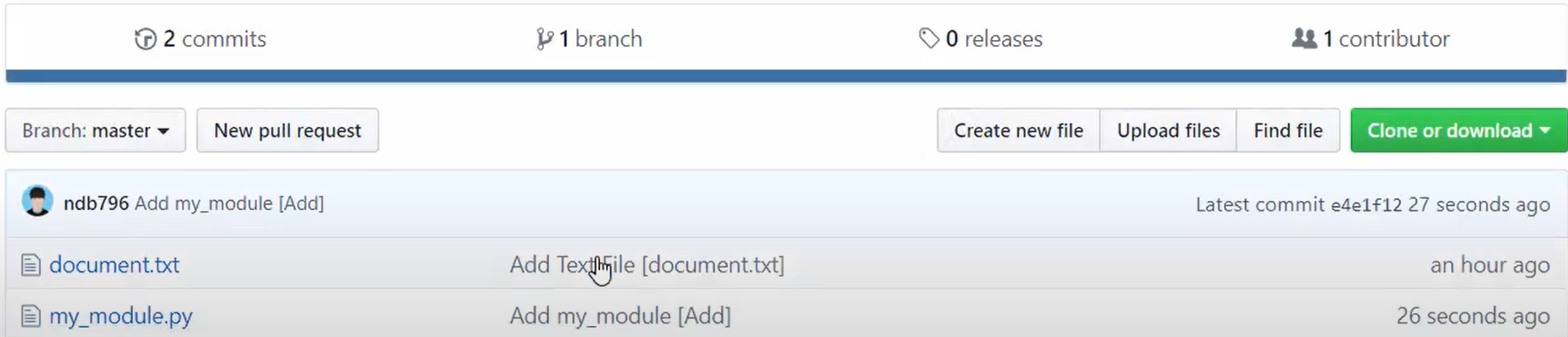
사용자이름, 비밀번호(토큰) 입력해준다.

그리고 난 후 github에서 확인해보면 올라가 있다.

**• 파일 수정하는 방법**

텍스트이(가) 표시된 사진

자동 생성된 설명



my\_module.py 파일을 수정하고자 한다. 수정 방법 순서는 다음과 같이 하면 된다.

1. my\_module.py을 수정한다.

2. git add. 또는 git add 파일명 입력

3. git commit -m “작성하고자 하는 메시지” 입력

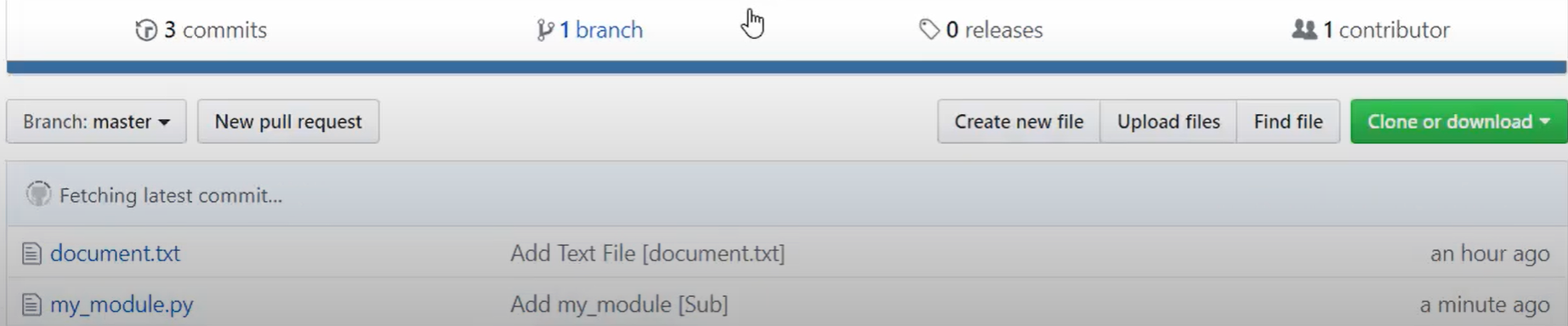
- 수정한 파일은 commit에 들어가는 메시지를 다르게 해준다.

예) 원본 commit : git commit -m “Add my\_module [Add]”

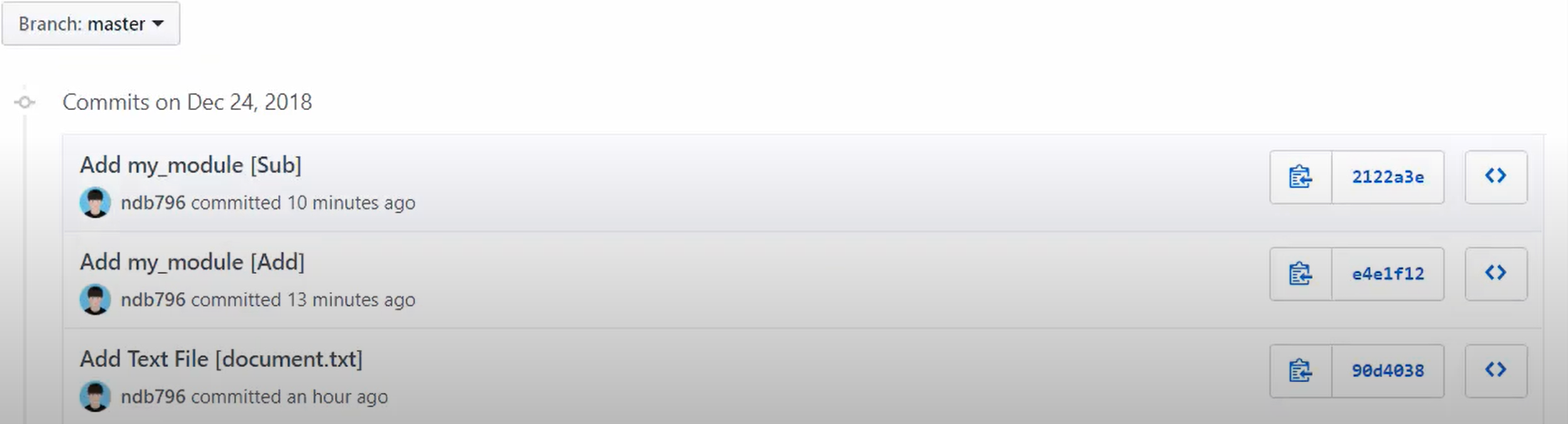
수정 commit : git commit -m “Add my\_module [Sub]”

4. git push 입력

5. github 사이트에서 새로고침을 하게 되면 수정한 내용이 반영된 것을 볼 수 있다.



6. 또한 commit에 들어가면 커밋한 내역을 확인할 수 있다.



**• 파일 내려 받기**

<처음 내려 받기>

1. commit 까지 프로젝트 완전 복사 : git clone "깃 주소"

\* clone을 한 후 push를 하면 복사해온 주소로 수정된 값이 올라간다.

​

<이미 있는 폴더에 내려 받기>

1. 수정된 내용 다운로드 : git pull

<파일 하나만 내려 받기>

1. Github 저장소 선택

2. 받고자 하는 파일 선택

텍스트이(가) 표시된 사진

자동 생성된 설명

3. Github 화면에서 [Raw] 버튼을 마우스 오른쪽 버튼으로 클릭

4. ‘다른 이름으로 링크 저장’ 선택 또는 cmd 명령창에서 curl 명령 사용

**[다른 이름으로 링크 저장]**

텍스트이(가) 표시된 사진

자동 생성된 설명

**[‘링크 주소 복사’ 선택 후 cmd 명령창에서 curl 명령 사용]**

텍스트이(가) 표시된 사진

자동 생성된 설명

링크 주소 복사한 파일을 다운로드하는 curl 명령어

-L 옵션 : 리디렉션 응답을 처리 (서버가 요청 페이지가 이동했다고 알리면 새로운 장소로 다시 요청함)

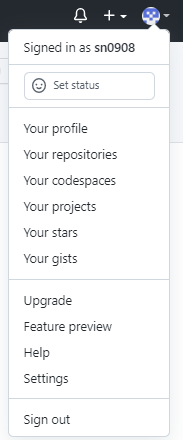
-O 옵션 : 저장할 파일명으로 리소스의 파일명을 그대로 사용

텍스트이(가) 표시된 사진

자동 생성된 설명

**• 단체(Organization) 계정 생성하기**

1. Settings 선택



2. Organization → New rganization

텍스트이(가) 표시된 사진

자동 생성된 설명

3. 단체에서 적용할 Plan을 선택

**Free**

• public/private 저장소 제한 없음

• 저장소 Collaborators 제한 없음

• 2,000 Action min per month

• 500MB of Github Packages

**Team**

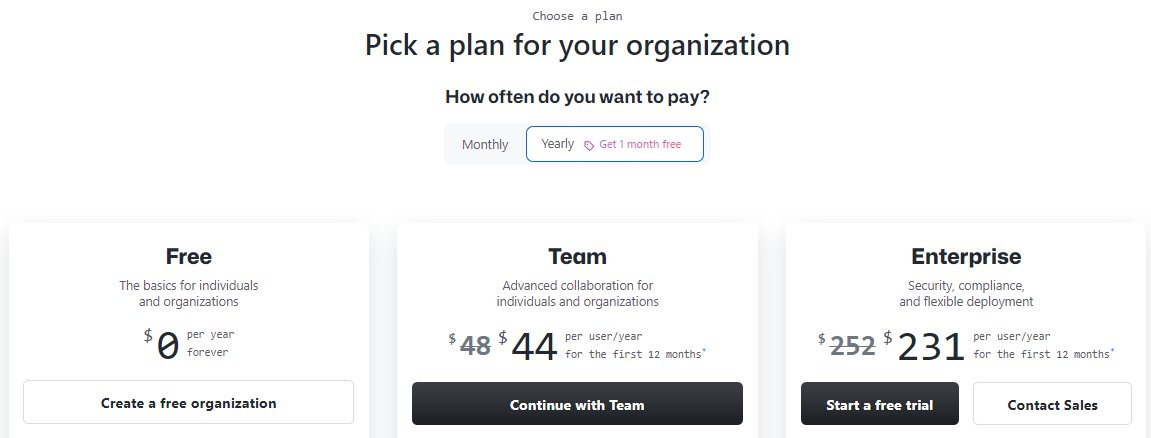
• Free에서 지원해주는 모든 것을 지원

• 이메일을 통한 Github 지원

• 3,000 Action min per month

• 2GB of Github Packages

• 이 외 다양한 기능

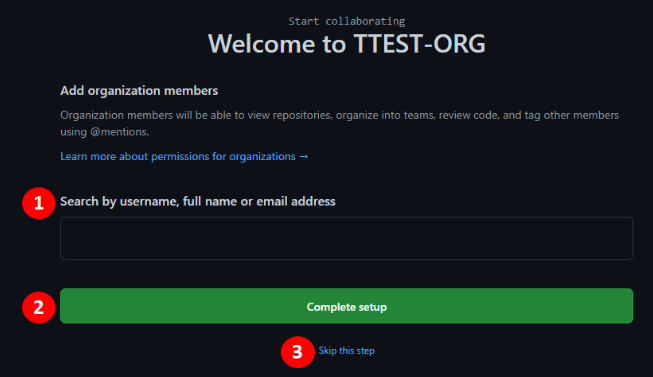


4. ① 단체명, ② 단체와 연결되는 Email, ③ 단체의 소속, ④ 퍼즐, ⑤ Next

텍스트이(가) 표시된 사진

자동 생성된 설명

5. ① 단체에 소속시킬 Github User, ② 완료, ③ 스킵



6. 설문과 관련된 내용이므로, Submit를 눌러 그냥 넘어가도 된다.

텍스트, 스크린샷, 모니터, 화면이(가) 표시된 사진

자동 생성된 설명

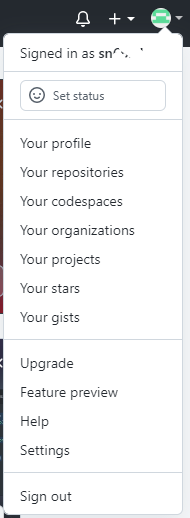
7. 저장소 생성 시 Owner 탭을 눌러서 개인 계정에 저장소를 소속시킬지 아니면 단체 계정에 저장소를 소속시킬지를 지정할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**• 권한 설정하기**

1. organization 선택



2. Obesis-Team 클릭



3. Invite someone : 초대하기 / Create new repository : 새롭게 저장소 생성하기

텍스트이(가) 표시된 사진

자동 생성된 설명

4. 초대할 username 또는 email address 입력

텍스트이(가) 표시된 사진

자동 생성된 설명

5. Member 또는 Owner 선택

텍스트이(가) 표시된 사진

자동 생성된 설명

• Member : Member는 다른 모든 구성원을 볼 수 있으며 리포지토리에 대한 액세스 권한을 부여할 수 있고 새로운 팀과 리포지토리를 만들 수 있음

• Owner : Owner는 조직에 대한 모든 관리 권한을 가지며 모든 리포지토리 및 팀에 대한 완전한 액세스 권한을 가짐

10. 권한 설정

텍스트이(가) 표시된 사진

자동 생성된 설명

**• No permission** : 권한 없음. 구성원은 공용 리포지토리를 복제하고 꺼낼 수만 있음

구성원에게 추가 액세스 권한을 부여하려면 해당 사용자를 팀에 추가하거나 개별 리포지토리의 공동작업자로 만들어야 함

**• Read** : 구성원은 모든 리포지토리를 복제하고 pull할 수 있음

**• Write** : 구성원은 모든 리포지토리를 복제, pull 및 push할 수 있음

**• Admin** : 구성원은 모든 리포지토리에 새 공동작업자를 복제, pull, push 및 추가할 수 있음