

# Git & Github 사용법

# 들어가기 전 필수 용어설명

- 디렉토리(Directory) : 우리가 평소에 사용하는 폴더(Folder)입니다. 폴더를 디렉토리라고 부를뿐!!
- 커맨드(command) : git bash에서 사용하는 명령문입니다.
- 워킹 디렉토리(working directory) : 작업을 하는 project directory
- 스테이징 아레아(staging area) : git add를 한 파일들이 존재하는 영역
- 레포지토리(repository) : 워킹 디렉토리의 변경 이력이 저장되어 있는 영역
- 로컬 레포지토리(local repository) : 나의 컴퓨터에 만들어 놓은 git의 commi을 관리하는 숨겨진 디렉토리라고 생각하시면 됩니다.
- 리모트 레포지토리(remote repository) : Github에 만들어 놓은 원격 commit 저장소입니다.

# Git사용법



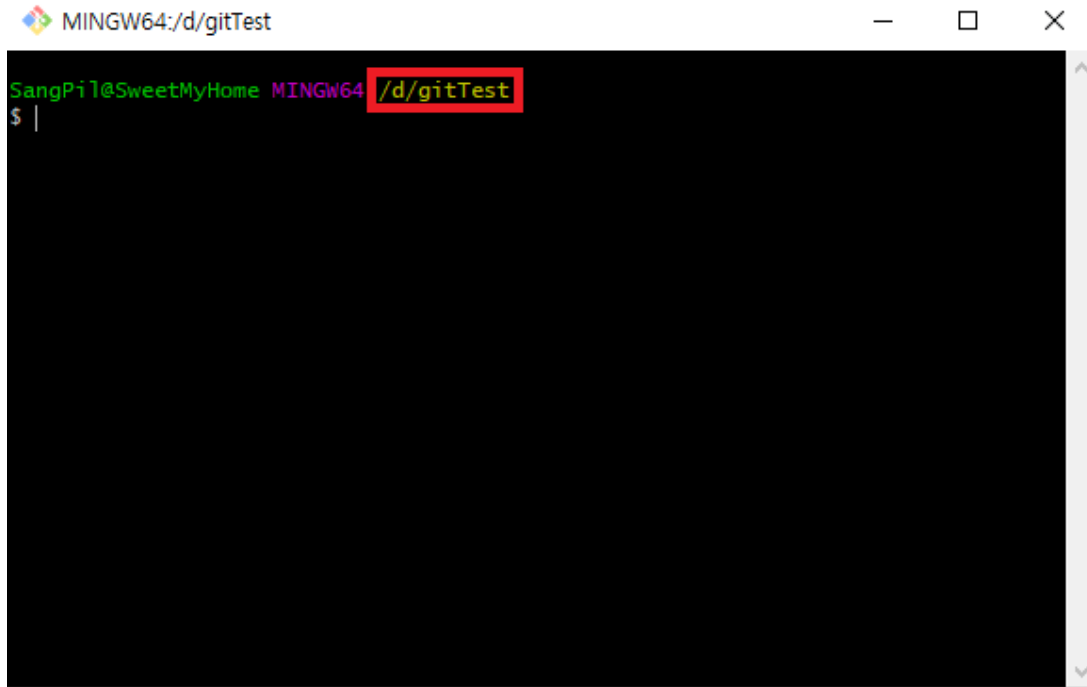
- git 다운로드하기(<https://git-scm.com/>)
- ‘Download 2.29.2 for Windows 버튼’을 누르면 잠시 후 자동으로 다운로드를 시작합니다.
- 다운로드 된 Git-2.29.2.2-64-bit.exe파일을 실행시키고 별다른 설정 건드리실 거 없이 체크되어 있는 그대로 끝까지 설치해주세요.

# Git사용법



- git 설치 완료 후 새로운 디렉토리에서 우클릭 해보면 좌측 화면처럼 git과 관련된 항목이 2개 생긴 걸 볼 수 있습니다.
- 저는 D드라이브 - gitTest라는 워킹 디렉토리를 만들고 그 안에서 우클릭을 한 상태입니다.(경로명 참고하세요.)
- gitTest라는 워킹 디렉토리에서 git기능을 사용하기 위해 작업을 해주어야 하는데 해당 작업은 Git Bash(Mac의 경우 터미널)에서 가능합니다.
- Git Bash Here을 클릭해줍니다.

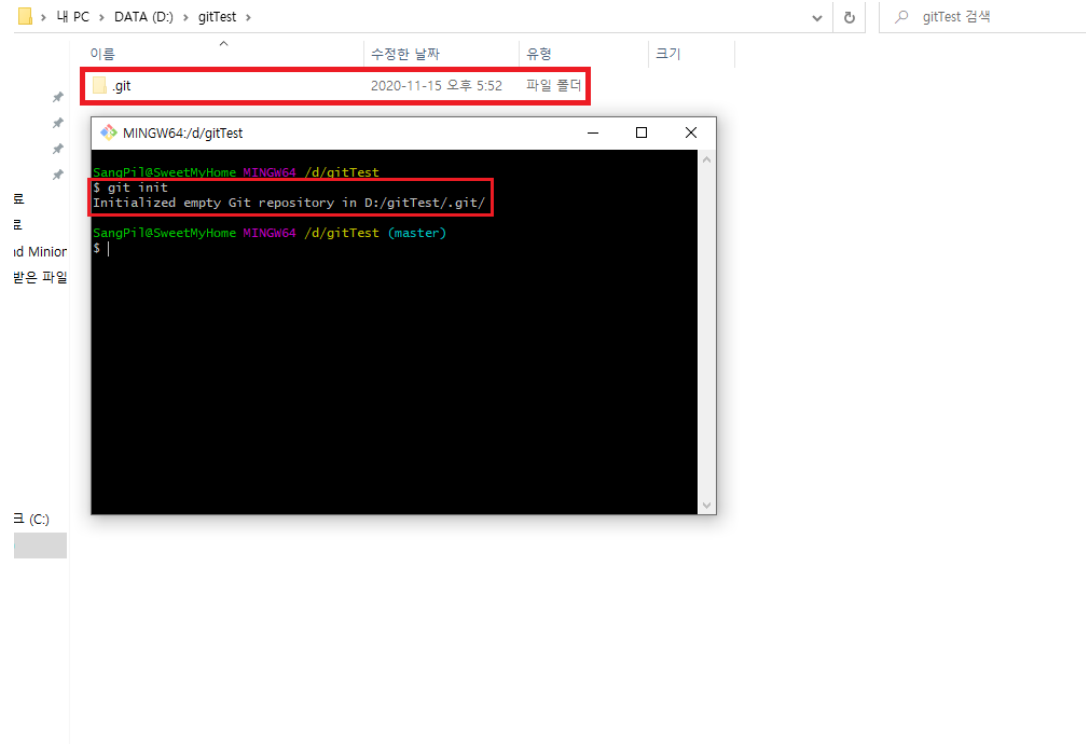
# Git사용법



```
MINGW64:/d/gitTest
SangPil@SweetMyHome MINGW64 /d/gitTest
$ |
```

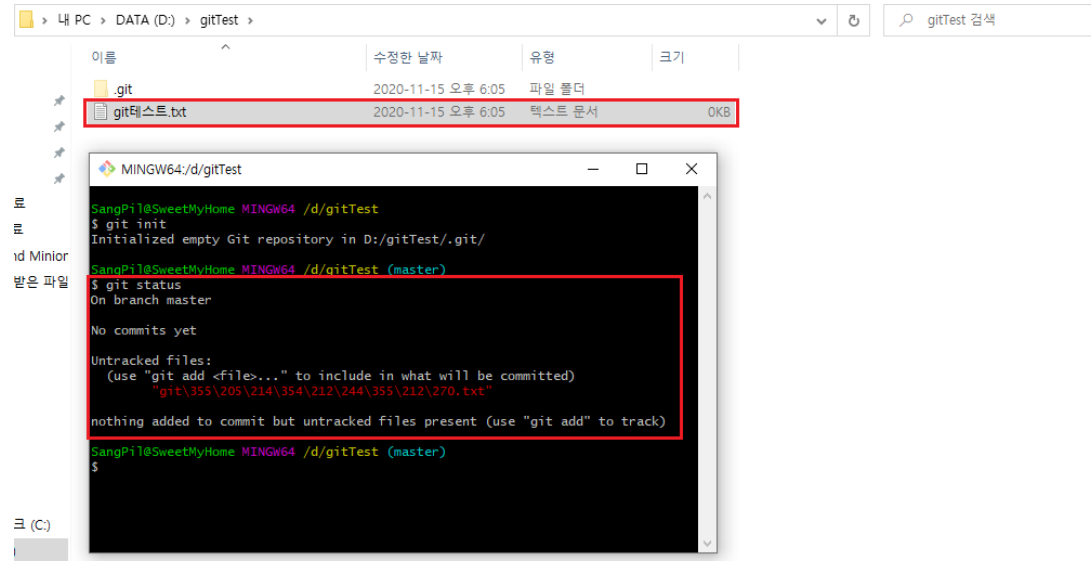
- git bash를 실행시키면 자동으로 /d/gitTest처럼 내가 git bash here한 디렉토리의 경로가 잡혀 있습니다 (Mac은 이 기능이 없어서 찾아서 들어 가셔야 합니다.)
- 경로가 다를 경우 cd 커맨드를 사용해서 디렉토리 이동을 하게 되는데 커맨드는 git bash 파일에 정리해 두었으니 참고하시기 바랍니다.
- 이 상태에서 git bash창에 차근차근 커맨드를 입력하여 해당 디렉토리에서 git을 사용해 보겠습니다.

# Git사용법



- git init 이라고 입력하시면 Initialized empty Git repository in D:/gitTest/.git/ 이라는 문구가 출력되고 워킹 디렉토리에 .git이라는 숨겨진 디렉토리가 추가되는 걸 확인할 수 있습니다. 지금 생성된 이 .git 디렉토리가 gitTest의 로컬 레포지토리입니다.
- 이렇게 git init을 가장 먼저 해주어야 해당 워킹 디렉토리에 git 기능을 사용 및 github와 연동할 수 있습니다.(git init을 해줘서 (master)가 생성된 거 확인!!)

# Git사용법

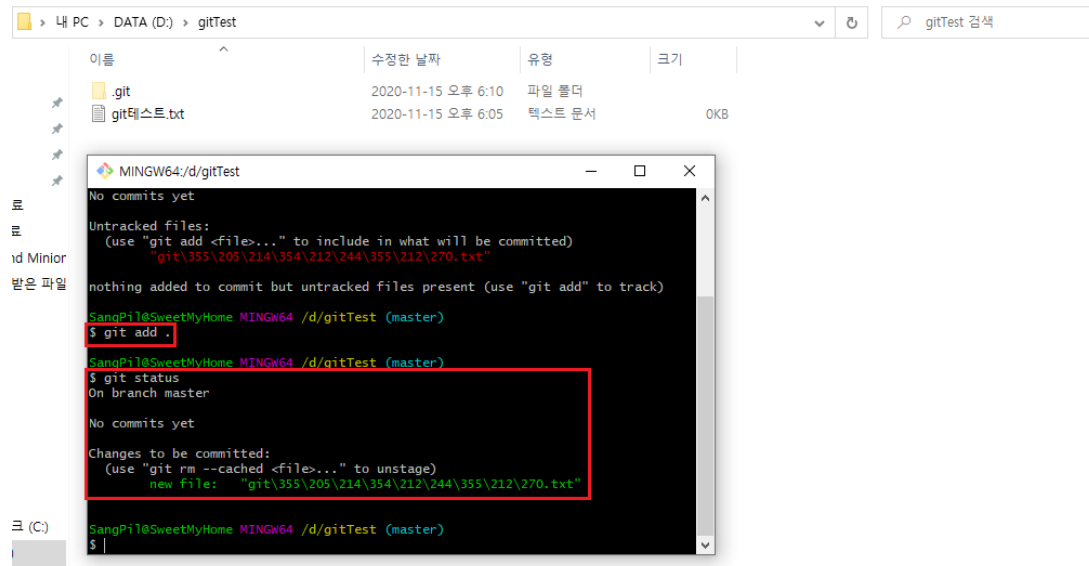


• 비어있던 gitTest 디렉토리에 git테스트.txt라는 빈 텍스트 파일을 추가하였습니다. 그리고 이 상태에서 git bash 창에 git status를 입력해보면 Untracked files가 있다고 나옵니다.

• git status는 git이 현재 해당 워킹 디렉토리를 어떻게 인식하고 있는지 알고 싶을 때 사용합니다.(수정사항이 있는지 확인할 때 보통 사용합니다.)

• Untracked files는 'Git에 의해 아직 추적되지 않고 있다 -> 버전 관리의 대상이 아니다.' 라는 의미이고 이 파일을 버전 관리하기 위해서는 staging area에 추가해줘야 합니다. 이 때 사용하는 커맨드가 git add 입니다.

# Git사용법



The image shows a Windows File Explorer window and a terminal window. The File Explorer window is titled '내 PC > DATA (D:) > gitTest' and shows a folder named '.git' and a file named 'git테스트.txt'. The terminal window is titled 'MINGW64:/d/gitTest' and shows the following commands and output:

```
No commits yet
Untracked files:
  (use "git add <file>..." to include in what will be committed)
  git\355\205\214\354\212\244\355\212\270.txt

nothing added to commit but untracked files present (use "git add" to track)
SangPil@SweetMyHome MINGW64 /d/gitTest (master)
$ git add .
SangPil@SweetMyHome MINGW64 /d/gitTest (master)
$ git status
On branch master

No commits yet

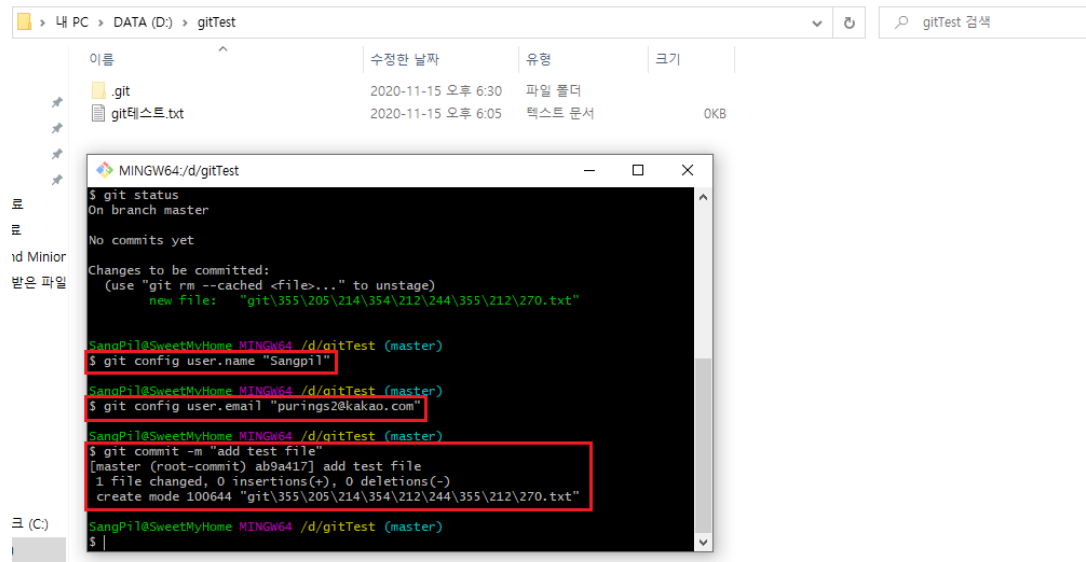
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   git\355\205\214\354\212\244\355\212\270.txt

SangPil@SweetMyHome MINGW64 /d/gitTest (master)
$
```

- git add <파일명> 으로 파일 하나씩 add 할 수 있고, 혹은 좌측에 보이는 것처럼 git add . 으로 모든 파일을 한 번에 add 할 수 있습니다.
- add 이후 git status를 해보시면 초록색으로 변경된 것이 보일텐데 이 상태가 되면 정상적으로 staging area에 변경사항이 add가 되었다는 겁니다.
- 다만 이 add가 되었다는 건 워킹 디렉토리 -> staging area로 add가 되었다는 것이고, 아직 흔히 말하는 commit이 된 건 아닙니다.



# Git사용법



```
$ git status
On branch master

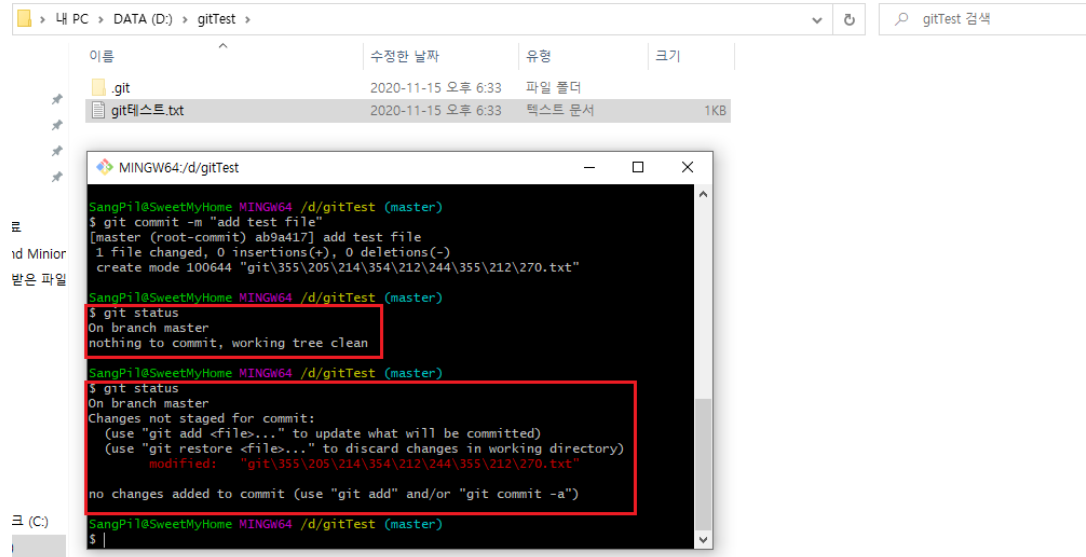
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   "git\355\205\214\354\212\244\355\212\270.txt"

SangPil@SweetMyHome MINGW64 /d/gitTest (master)
$ git config user.name "Sangpil"
SangPil@SweetMyHome MINGW64 /d/gitTest (master)
$ git config user.email "purings2@kakao.com"
SangPil@SweetMyHome MINGW64 /d/gitTest (master)
$ git commit -m "add test file"
[master (root-commit) ab9a417] add test file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "git\355\205\214\354\212\244\355\212\270.txt"
SangPil@SweetMyHome MINGW64 /d/gitTest (master)
$ |
```

- staging area에 올린 파일을 하나의 버전으로 남기고 싶다면 commit을 해주어야 합니다.(스냅샷!!)
- 다만 처음 commit을 할 때는 두가지 작업을 해주어야 하는데 user.name과 user.email을 저장하는 작업이 필요합니다.
- 먼저 git config 커맨드를 사용하여 name과 email을 저장한 상태에서 commit을 진행해주세요.  
(git config --global user.email “이메일” <요렇게!>)  
(옆에 사진에선 --global이 없어요 헛갈리지마세요!)
- git commit -m “커밋메시지” 형태로 commit을 하면 됩니다.

# Git사용법



The image shows a Windows File Explorer window and a terminal window. The File Explorer window displays the contents of the 'd:\gitTest' directory, which includes a '.git' folder and a 'git테스트.txt' file. The terminal window shows the output of several Git commands: 'git commit -m "add test file"', 'git status' (showing 'nothing to commit, working tree clean'), and 'git status' (showing 'Changes not staged for commit: (use "git add <file>..." to update what will be committed) (use "git restore <file>..." to discard changes in working directory) modified: "git\355\205\214\354\212\244\355\212\270.txt"').

```
SangPil@SweetMyHome MINGW64 /d/gitTest (master)
$ git commit -m "add test file"
[master (root-commit) ab9a417] add test file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "git\355\205\214\354\212\244\355\212\270.txt"

SangPil@SweetMyHome MINGW64 /d/gitTest (master)
$ git status
On branch master
nothing to commit, working tree clean

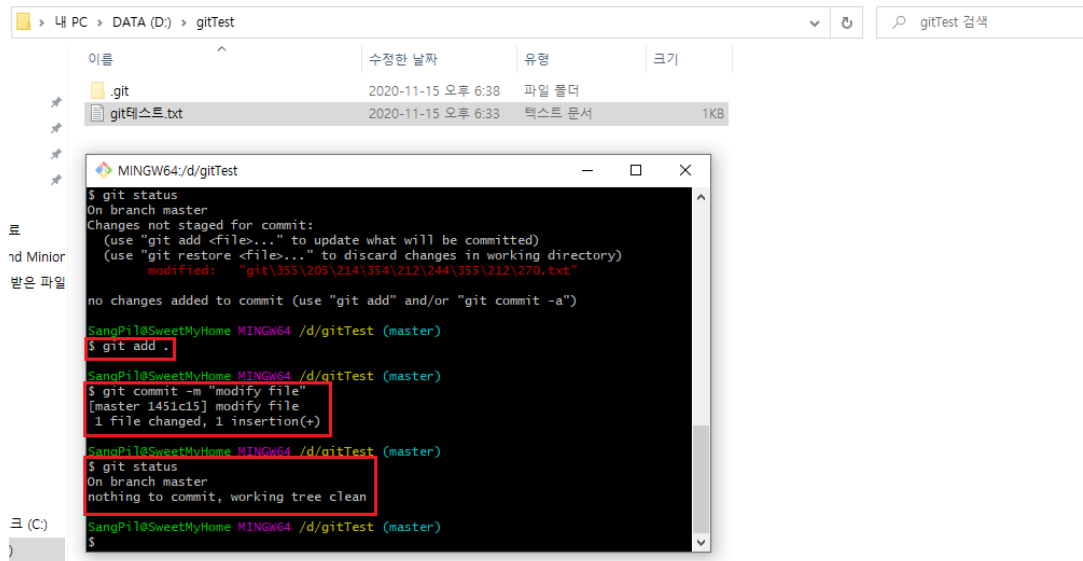
SangPil@SweetMyHome MINGW64 /d/gitTest (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   "git\355\205\214\354\212\244\355\212\270.txt"
no changes added to commit (use "git add" and/or "git commit -a")

SangPil@SweetMyHome MINGW64 /d/gitTest (master)
$
```

- git commit 직후 git status를 하면 nothing to commit, working tree clean 이라고 뜹니다. Commit 이후 어떠한 변경사항도 없기때문에 이런 메시지가 출력되는 것입니다.
- 하지만 이후에 git테스트.txt 파일에 일정 내용을 추가한 뒤 저장 -> git status를 다시해보면 Change not staged for commit 이 출력되는 걸 확인 할 수 있습니다.
- 처음엔 git에 add하지 않아 Untracked였지만 이제는 등록이 되었고, commit이후로 수정사항이 생겼기 때문에 Change not staged for commit 라고 표시되는 겁니다.
- 이 시점부터는 내가 어떤 파일을 수정하고 저장해두고 싶은 시점의 순간마다 git add, git commit을 활용하여 버전을 저장하면 됩니다.

# Git사용법

- 변경된 사항을 두번째 버전으로 남기기 위해  
git add . 이후 git commit을 한 모습입니다.
- 이렇듯 내가 남기고 싶은 순간마다
  1. git status로 상태 확인
  2. git add . 로 staging area에 추가
  3. git commit -m “커밋메시지” 로 commit  
을 해주시면 됩니다!
- 커밋메시지는 가급적 영어로 작성해주세요.



The image shows a Windows File Explorer window and a terminal window. The File Explorer window displays the contents of the 'gitTest' directory, which includes a '.git' folder and a 'git테스트.txt' file. The terminal window shows the execution of several Git commands and their output:

```
MINGW64/d/gitTest
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   "git\355\205\214\354\212\244\355\212\270.txt"

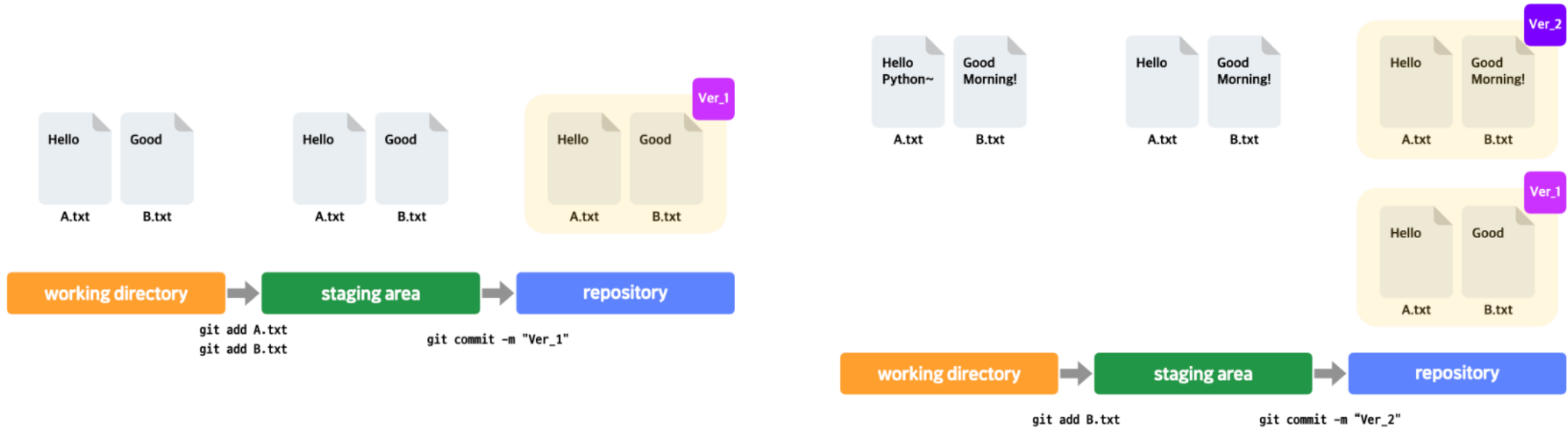
no changes added to commit (use "git add" and/or "git commit -a")

SangPil@SweetMyHome MINGW64 /d/gitTest (master)
$ git add .

SangPil@SweetMyHome MINGW64 /d/gitTest (master)
$ git commit -m "modify file"
[master 1451c15] modify file
1 file changed, 1 insertion(+)

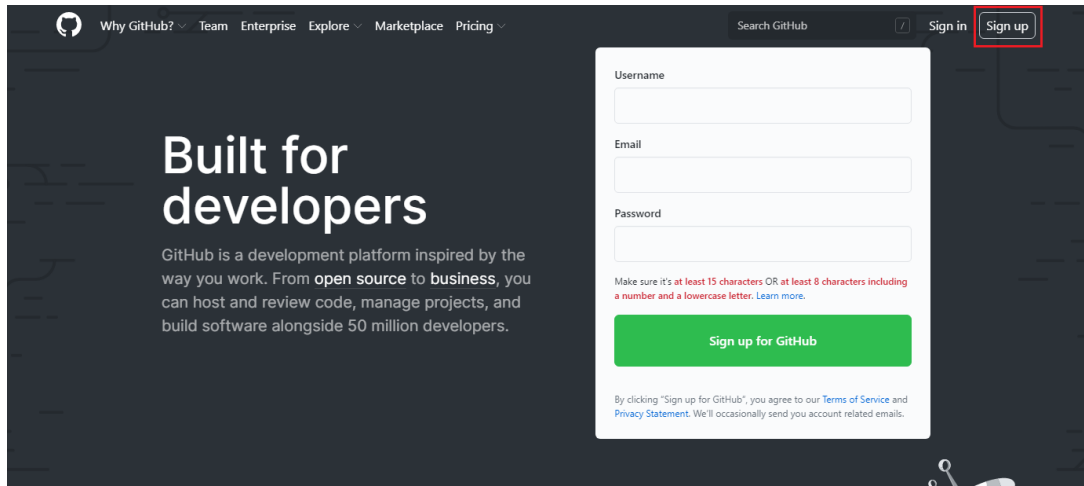
SangPil@SweetMyHome MINGW64 /d/gitTest (master)
$ git status
On branch master
nothing to commit, working tree clean
```

# Git의 3가지 영역 한눈에 보기



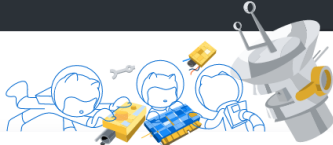
출처 : Codeit

# Git + Github 사용법




The screenshot shows the GitHub homepage with a dark background. On the left, the text "Built for developers" is prominent, followed by a description of GitHub as a development platform. On the right, there is a white sign-up form with fields for Username, Email, and Password. A green button labeled "Sign up for GitHub" is at the bottom of the form. Above the form, there are links for "Sign in" and "Sign up" (the latter is highlighted with a red box). The top navigation bar includes links for "Why GitHub?", "Team", "Enterprise", "Explore", "Marketplace", and "Pricing".


- Github (<https://github.com/>) 접속
- Sign up을 클릭하여 회원가입 진행해주세요.



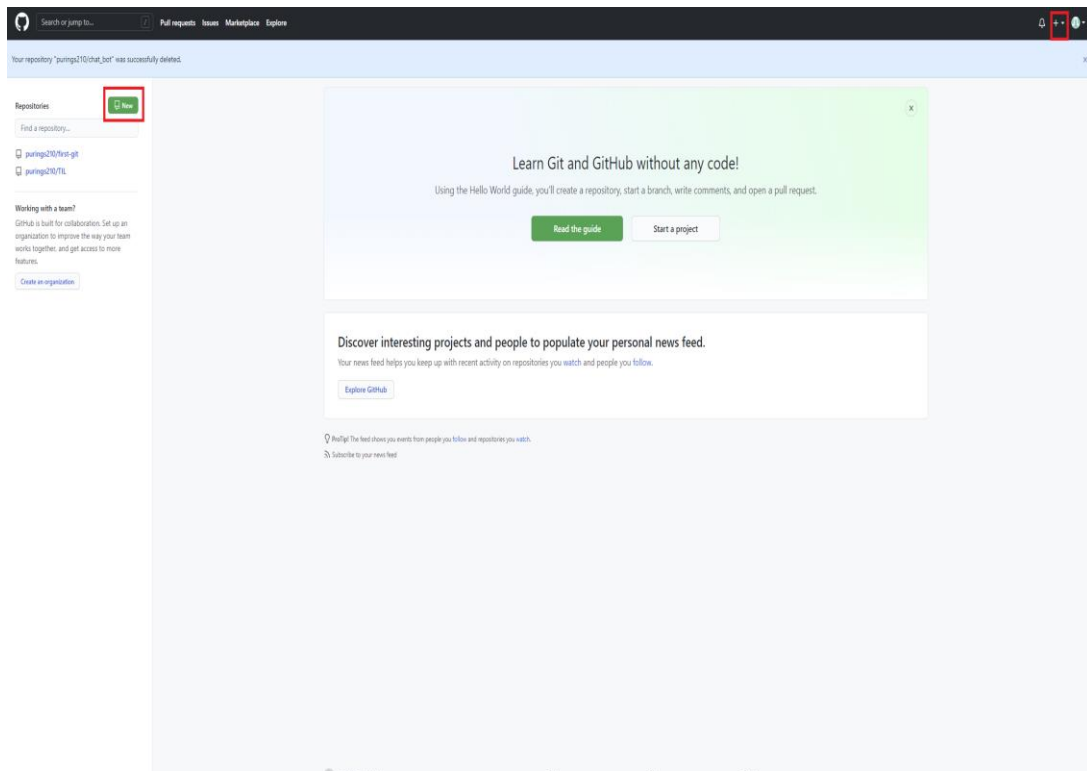
**Get started with GitHub Enterprise**

Take collaboration to the next level with security and administrative features built for teams.

**Enterprise**  
Deploy to your environment or the cloud.  
[Start a free trial](#)

**Talk to us**  
Need help?  
[Contact Sales →](#)

# Git + Github 사용법




- 로그인하시면 좌측과 같은 화면이 나올거예요! 좌측에 보이는 초록색 New를 클릭해주세요.
- 혹은 우측상단 + 버튼을 눌러서 'New repository'로 만드셔도 됩니다!
- 오늘 만들었던 챗봇프로그램 코드들을 한 번 github에 연동시켜 보겠습니다 잘 따라와주세요!

# Git + Github 사용법

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*  purings210

Repository name \* chat\_bot ✓

Great repository names are short and memorable. Need inspiration? How about fluffy-potato?

Description (optional)

처음 만들어본 챗봇 프로그램!

☒ Public  
Anyone on the internet can see this repository. You choose who can commit.

☐ Private  
You choose who can see and commit to this repository.


Initialize this repository with:  
Skip this step if you're importing an existing repository.

☐ Add a README file  
This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore  
Choose which files not to track from a list of templates. [Learn more.](#)

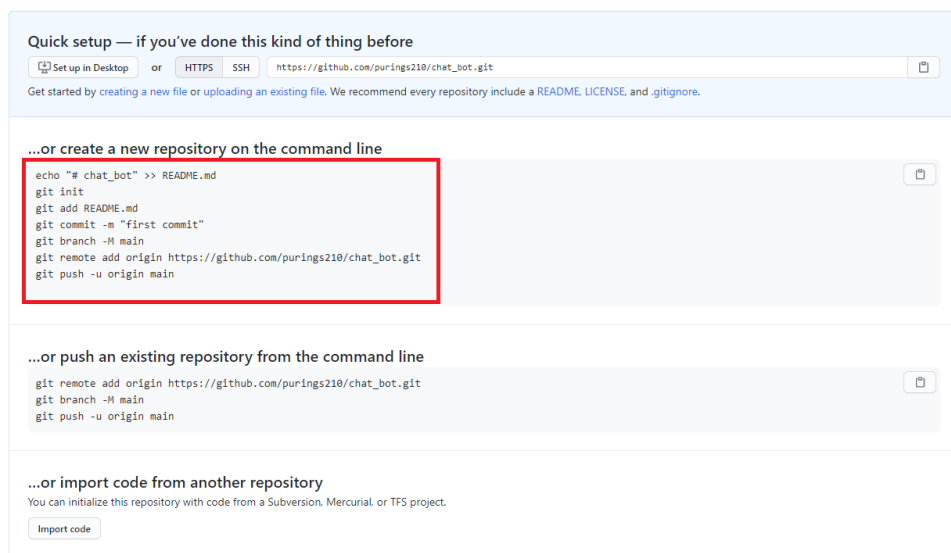
☐ Choose a license  
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

© 2021 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)  [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

- Repository name은 필수로 적으셔야 되며, 이 레포지토리의 이름을 적어주시면 됩니다! 내 컴퓨터에 있는 폴더명과 같지 않아도 아무런 상관은 없습니다!(다만 맞춰서 진행하시면 덜 헛갈리겠죠?)
- Description은 보시다싶이 optional이라 적지 않으셔도 무방하지만! 나중에 레포지토리가 많이 쌓이면 헛갈릴 수 있으니 가급적 적는 습관을 들여주세요.
- 위 2가지 내용을 적었으면 Create repository!

# Git + Github 사용법



• 생성이 완료되면 좌측과 같은 화면을 보실 수 있을텐데 아직은 내 컴퓨터에 만들어져있는 로컬 레포지토리와 지금 막 생성한 리모트 레포지토리가 연결되지 않은 상태입니다.

• 연결을 위해서 빨간 네모친 부분을 실행하게되는데 잘 보시면 우리가 배웠던 커맨드들이 많이 있습니다!(다음장에서 설명계속)



# Git + Github 사용법

```
MINGW64:/c:/Users/SangPil/chat_bot
SangPil@SweetMyHome MINGW64 ~
$ cd chat_bot

SangPil@SweetMyHome MINGW64 ~/chat_bot
$ git init
Initialized empty Git repository in C:/Users/SangPil/chat_bot/.git/

SangPil@SweetMyHome MINGW64 ~/chat_bot (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    __pycache__/
    app.py
    message.py
    ngrok.exe
    set_webhook.py

nothing added to commit but untracked files present (use "git add" to track)
SangPil@SweetMyHome MINGW64 ~/chat_bot (master)
```

- 지금 상황은 이미 내 컴퓨터에 chat\_bot이라는 폴더가 있고 그 안 오늘 진행했던 챗봇 프로그래밍 파일들이 있습니다!
- 다만 아직 이 폴더를 git init을 하지 않은 상태이기 때문에 git기능을 사용할 수 없습니다!
- chat\_bot 폴더에서 마우스우클릭 – git bash here 해주시고 git init명령어를 작성해주세요.
- ~/chat\_bot (master)와 같이 나온다면 성공!

# Git + Github 사용법

```
MINGW64:/c/Users/SangPil/chat_bot
SangPil@SweetMyHome MINGW64 ~
$ cd chat_bot

SangPil@SweetMyHome MINGW64 ~/chat_bot
$ git init
Initialized empty Git repository in C:/Users/SangPil/chat_bot/.git/

SangPil@SweetMyHome MINGW64 ~/chat_bot (master)
$ git status
On branch master

no commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    __pycache__/
    app.py
    message.py
    ngrok.exe
    set_webhook.py

nothing added to commit but untracked files present (use "git add" to track)

SangPil@SweetMyHome MINGW64 ~/chat_bot (master)
$ git add .

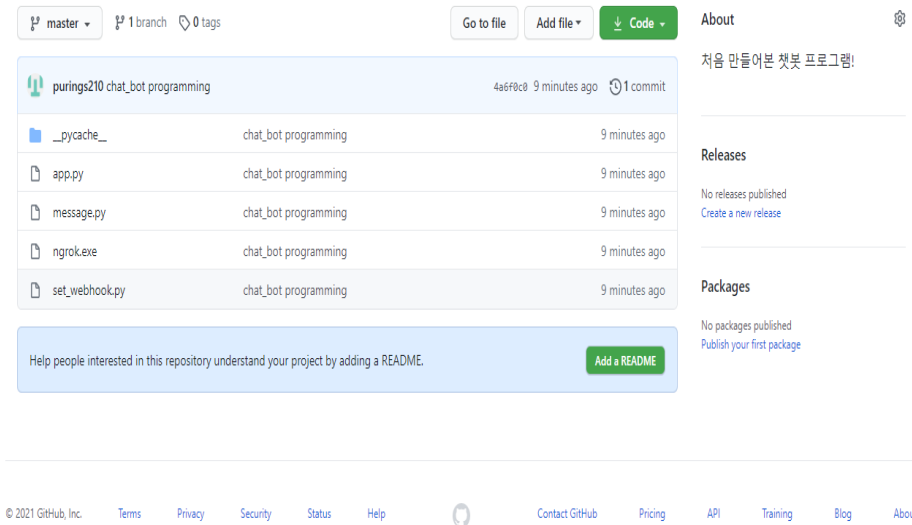
SangPil@SweetMyHome MINGW64 ~/chat_bot (master)
$ git commit -m "chat_bot programming"
[master (root-commit) 4a6f0c0] chat_bot programming
5 files changed, 89 insertions(+)
create mode 100644 __pycache__/app.cpython-38.pyc
create mode 100644 app.py
create mode 100644 message.py
create mode 100644 ngrok.exe
create mode 100644 set_webhook.py

SangPil@SweetMyHome MINGW64 ~/chat_bot (master)
$ git remote add origin https://github.com/purings210/chat_bot.git

SangPil@SweetMyHome MINGW64 ~/chat_bot (master)
$ git push origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 13.02 MiB | 2.17 MiB/s, done.
```

- 이후 git status를 해보시면 No commits yet과 함께 Untracked files들이 5개 있을거예요!
- 우선 이 파일들을 모두 add 해줘야겠죠? git add .을 활용하여 5개 파일모두 staging area로 add 해주세요
- add가 완료되면 commit을 진행해주세요.
- commit까지 완료되었다면 2슬라이드 전 빨간 네모 칸 아래서 두번째줄인 git remote~로 되어있는 부분을 복사 후 git bash에서 실행해주세요.
- 이후 push까지 입력해주고! github페이지를 새로고침하면!

# Git + Github 사용법



- 짜잔 github에 정상적으로 연결되었습니다!
- 이제부터 chat\_bot 프로그램을 업데이트 할 때마다 꼭 commit을 해주시면 좋을거 같아요!
- 아, 참고로 git commit까지만 하게 되면 github엔 반영이 안됩니다!
- git push origin master라는 한 단계를 더 입력해주셔야 내 컴퓨터에 쌓여있던 commit 내역들이 github에 연동되니 이 점 유의하세요!