

# Git

## [이론]

### What is Git?

Git은 프로그램 등의 **소스 코드 관리를 위한 분산 버전 관리 시스템**. 기하학적 불변 이론을 바탕으로 설계됐고, **빠른 수행 속도**에 중점을 두고 있는 것이 특징. 최초에 리누스 토발즈가 리눅스 커널 개발에 이용하려고 개발하였으며, 현재는 다른 곳에서도 널리 사용되고 있음.

Git의 **작업 폴더는 모두, 전체 기록과 각 기록을 추적할 수 있는 정보를 포함하고 있으며, 완전한 형태의 저장소임. 네트워크에 접근하거나 중앙 서버에 의존하지 않음.**

- wikipedia -

## 버전 관리 시스템

- 버전 관리 시스템 : 프로젝트에 포함된 파일의 변경사항을 추적할 수 있도록 돕는 방법론이나 도구
  - 사용자가 변경한 모든 내용을 추적 관리 가능한 저장소(Repository)를 통해 버전 관리.
  - 사용자는 변경된 내용을 저장소에 저장(commit)
  - 하나의 중앙 저장소가 있고 모든 사용자는 이 저장소에 변경사항을 전송
  - 중앙 저장소에 접근가능한 모든 사람들이 변경 사항을 조회 가능하고 최신 버전을 복사해서 작업할 수 있음

## 분산버전관리시스템

- 분산 버전 관리 시스템 : 각 사용자가 프로젝트의 전체 이력을 관리할 수 있는 자신만의 저장소를 가지고 있음
  - 기존 버전 관리 시스템과 같이 중앙 저장소를 통해 협업 가능
  - 사용자는 변경된 내용을 개인 저장소에 저장 (commit)
  - 개인 저장소에 있는 변경된 내용을 중앙 저장소에 저장(Pushing)
  - 중앙 저장소에 있는 최신 내용을 개인 저장소로 가져와 최신 버전을 유지 가능(Pulling)
  - 네트워크 연결이 안된 상태에서도 저장소에 저장 가능하고, 네트워크 연결시 중앙 저장소와 동기화 실시
  - 개인 저장소와 중앙 저장소 모두 프로젝트 전체 이력을 가지고 있어 네트워크 연결이 안된 상태에서도 변경 이력 조회 가능

## Git의 특징

1. 빠른 속도와 성능
  - git의 모든 동작은 local에서 실행되기 때문에 매번 Network 연결을 할 필요가 없어 타 버전관리 시스템에 비해 속도가 빠르다
2. 분산 작업
  - 중앙 저장소를 통해 한 프로젝트를 여러명의 개발자가 동시에 작업을 분담하여 진행할 수 있다.
3. 데이터의 보장성
  - git의 모든 파일은 암호화 시켜주며, commit시 checksum이란 검사를 통해 data가 유효한지 검사, 또, commit을 할 때마다 user name, 시간, 내용등을 기록하여 history 관리가 용이하다.
4. Staging area
  - 작업한 내용을 바로 commit 시키지 않고 index영역에 올려둔 내용만 commit할 수 있다.
5. branch model

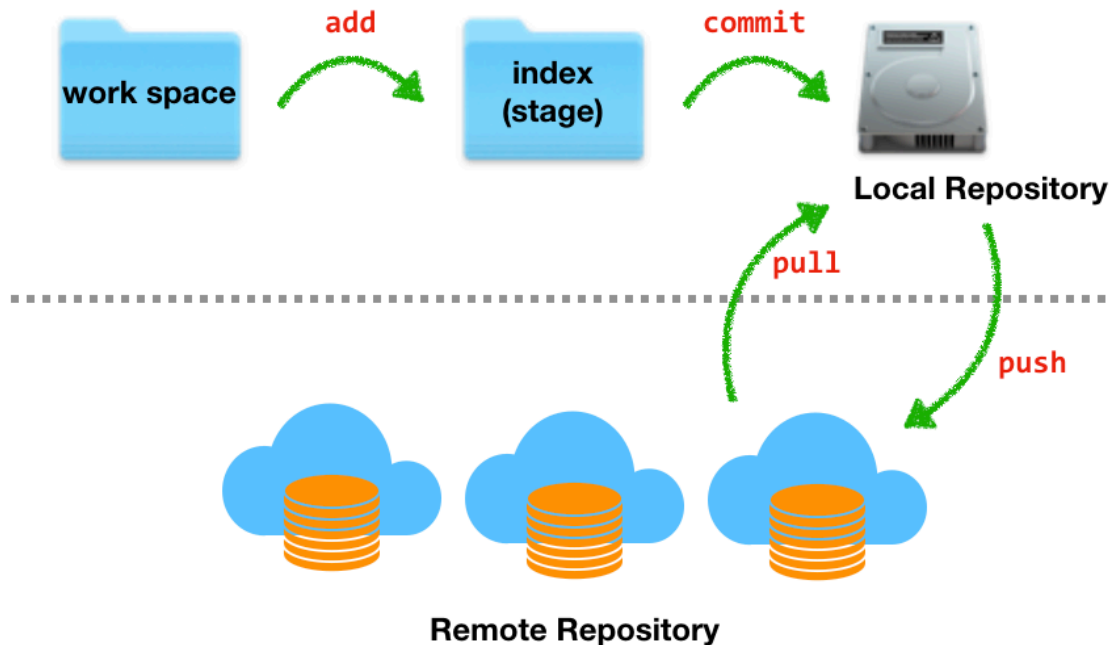
# Git의 구조

1. Work Space
  - 현재 프로젝트의 작업이 이뤄지는 장소
  - work tree/work directory라고도 함
2. index
  - work space에 있는 내용을 내부저장소에 저장하기전 올려두는 중간 단계 공간
  - index공간을 통해서 수정된 파일 중 원하는 파일만 저장소에 저장 시킬 수 있음
3. 내부 저장소
  - git이 설치된 내부 컴퓨터
4. 외부 저장소
  - 외부에 저장하는 저장소, github란 서비스도 외부 저장소에 속함.

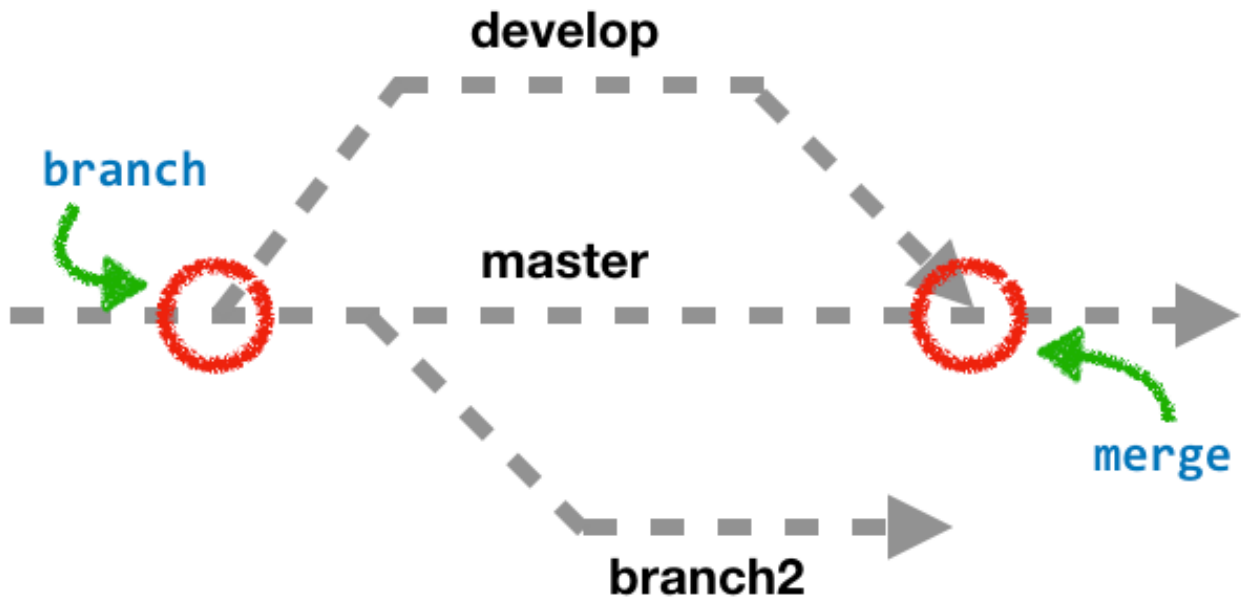
## git 명령어

- init: git 초기화
- clone : 외부 저장소에서 프로젝트 내려 받기
- add : 변경된 내용을 index 영역에 올려두는 명령어
- commit : index 영역에 있는 내용 내부저장소에 저장
- status : 현재 git 상태 확인
- branch : 가지치기
- checkout : 작업 환경 이동
- push : 외부 저장소에 데이터 저장
- pull : 외부 저장소에서 최신 변경사항 받아오기

## Git 요약



# Branch Model



## Workflow - Commit

- commit을 하는 순간 project의 상태를 index(stage)에 저장.
- commit하는 순간의 상태를 중요시 여기며 그 상태를 snapshot으로 찍는다고 함



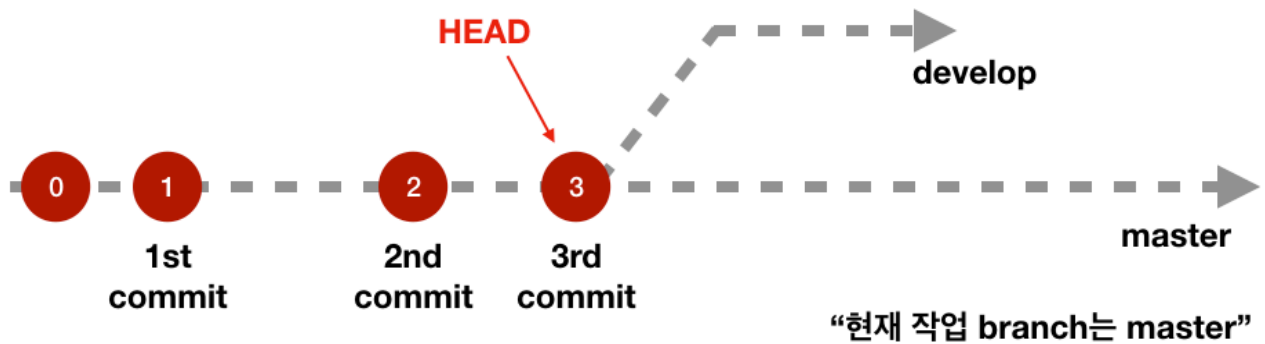
## Workflow - Commit

- head는 현재 사용자가 바라보고 있는 곳을 가르키며 commit시 자동으로 이동 된다.



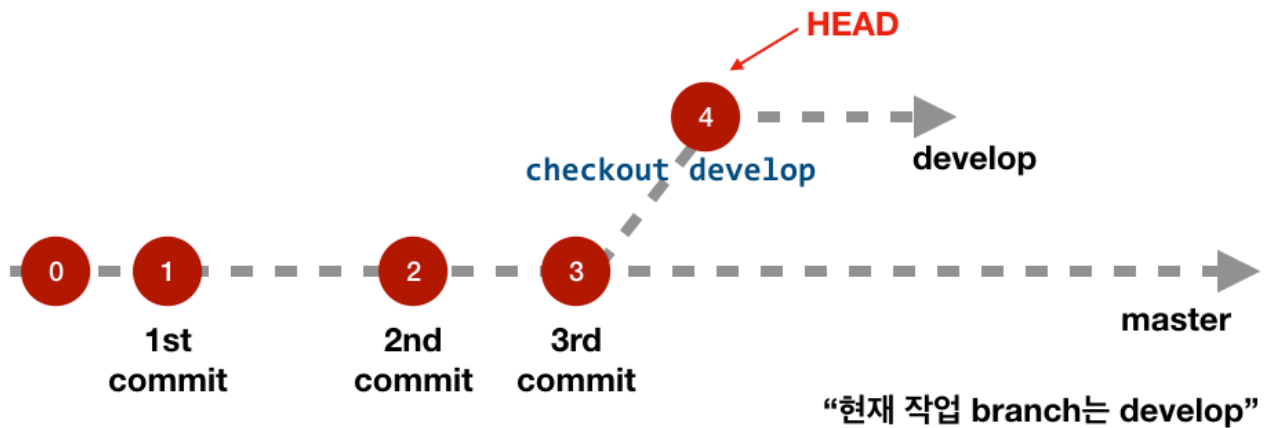
## Workflow - Branch

- 새로운 branch 생성(develop)



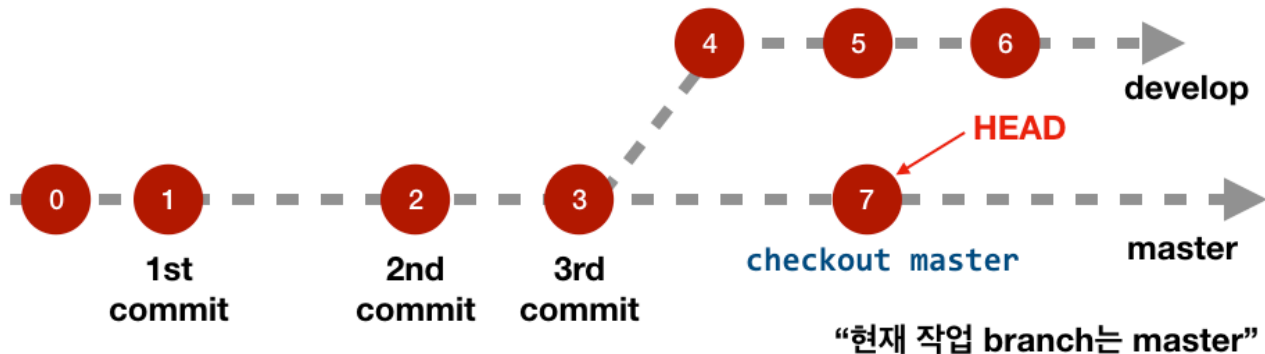
## Workflow - checkout

- 새로운 branch로 작업 환경 변경 : checkout



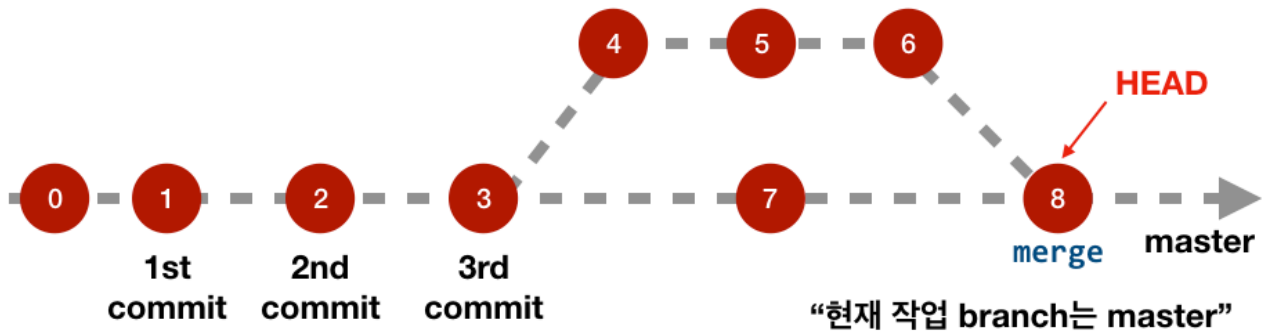
## Workflow - checkout

- develop branch에서 master branch로 작업 환경 변경: checkout master



# Workflow - Merge

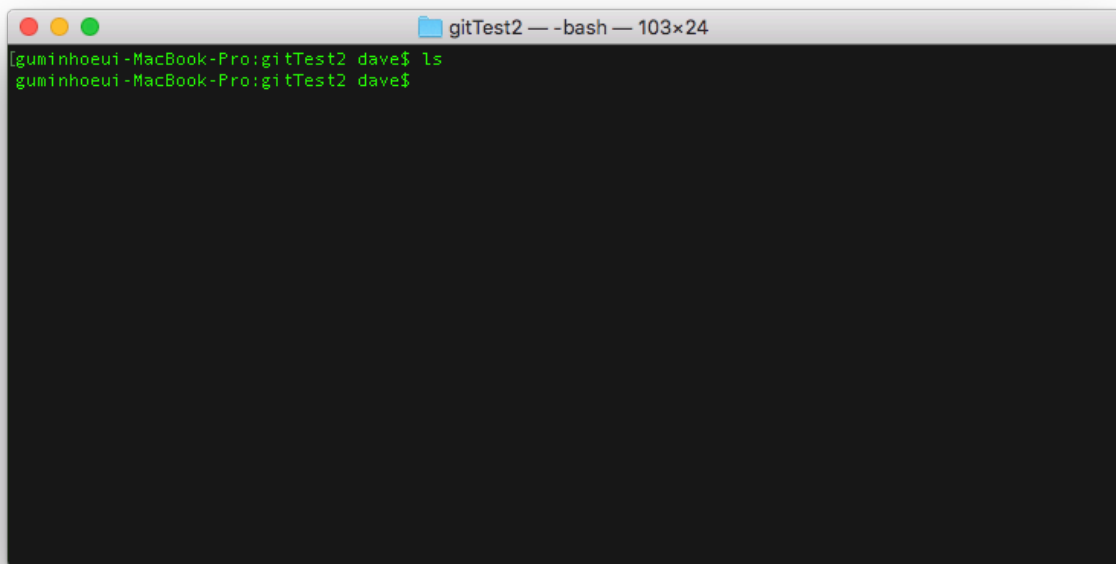
- develop branch에서 변경된 내용을 master branch와 합치기: merge develop



## [실습]

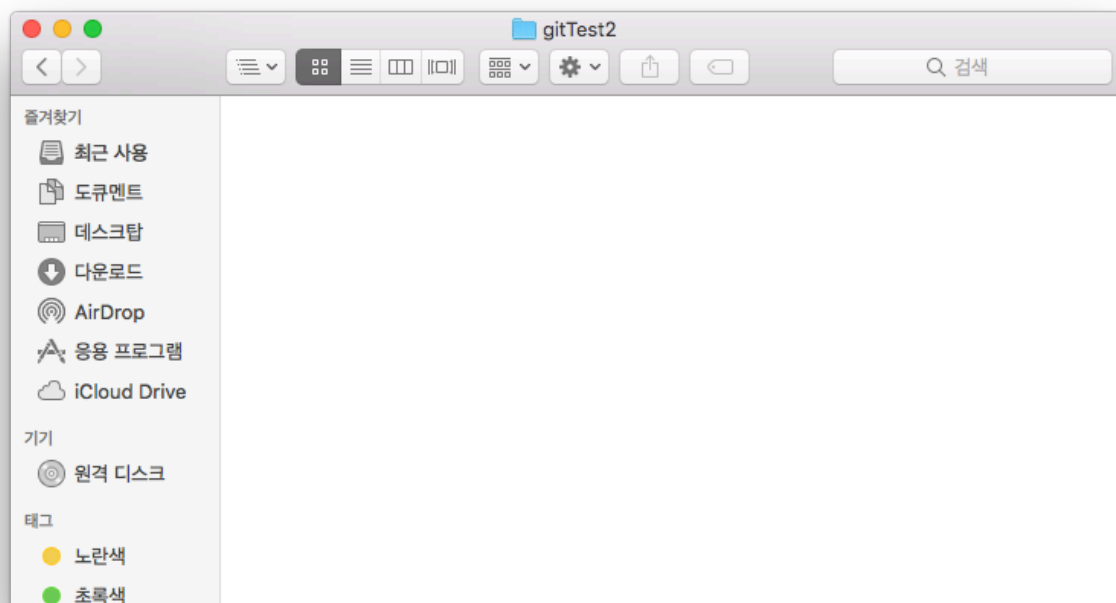
### git 저장소 설정 및 내부 파일 설치

command window에서 “gitTest2”라는 directory를 생성하고 “ls”란 명령어를 입력하여 directory 내부에 파일이 있는지 확인한다.

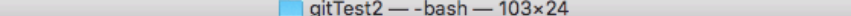


```
gitTest2 — -bash — 103x24
[guminhoeui-MacBook-Pro:gitTest2 dave$ ls
guminhoeui-MacBook-Pro:gitTest2 dave$
```

Finder에서 본 폴더(디렉토리)에도 아무 파일이 보이지 않는다.



vi editor : “i”를 keyboard에서 누르고, 원하는 내용을 기입하고, esc키를 누르고 “:wq”를 누르면 저장됨



A terminal window titled "gitTest2 — -bash — 103x24". The prompt is "guminhoeui-MacBook-Pro:gitTest2 dave\$". The first command is "ls", showing "newFile.txt". The second command is "vi newFile.txt", which opens a vim editor with a blank file.

A screenshot of a terminal window titled "gitTest2 — vi newFile.txt — 103x24". The window has standard macOS window controls (red, yellow, green buttons) at the top left. The main area is dark gray. On the far left, there are several light blue tilde (~) characters indicating line numbers or cursor positions. At the bottom left, the text "newFile.txt [New File]" is displayed in red.

gitTest2 — vi newFile.txt — 103x24

New File 생성 |

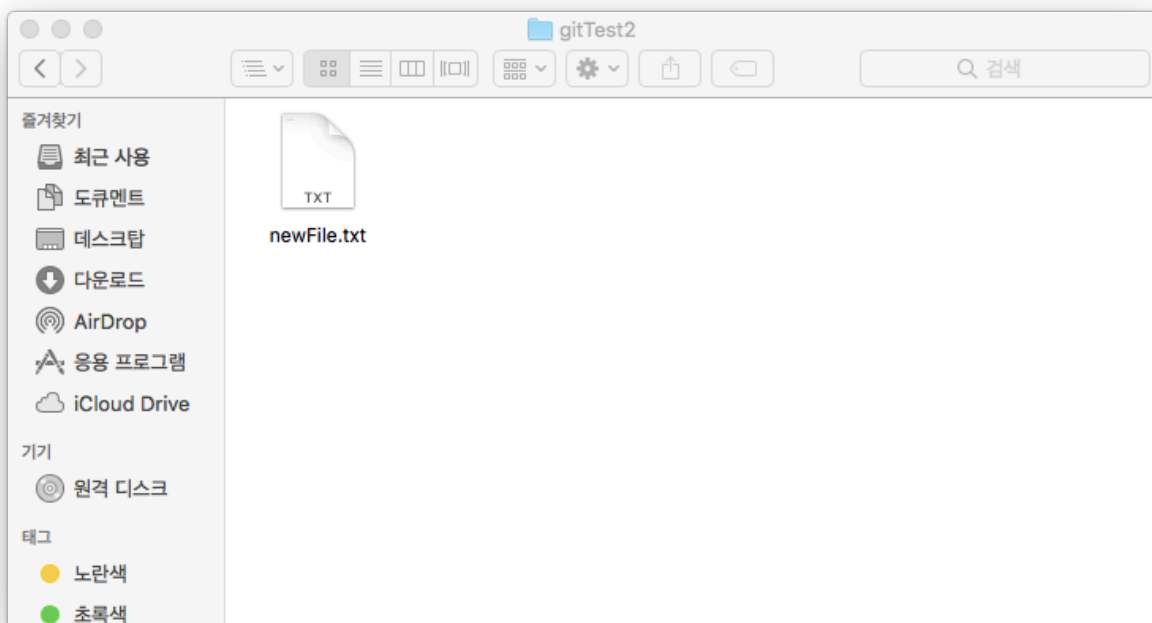
-- INSERT --

새로운 파일이 생성되었음을 확인

(참고로 “git version”이란 키워드를 입력하면 git설치유무와 버전을 확인할 수 있다.)

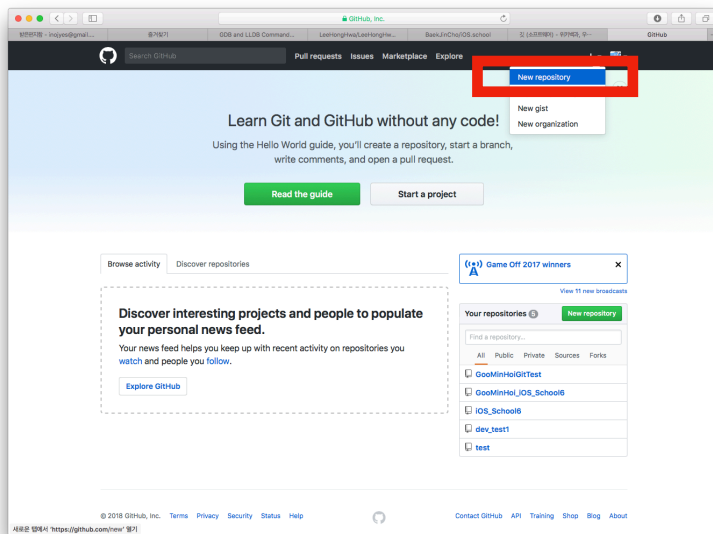
```
gitTest2 — -bash — 103x24
[guminhoeui-MacBook-Pro:gitTest2 dave$ ls
[guminhoeui-MacBook-Pro:gitTest2 dave$ vi newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ ls
newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git version
git version 2.10.1
guminhoeui-MacBook-Pro:gitTest2 dave$
```

Folder에서도 새로운 파일이 생성되었음을 확인 할 수 있다.

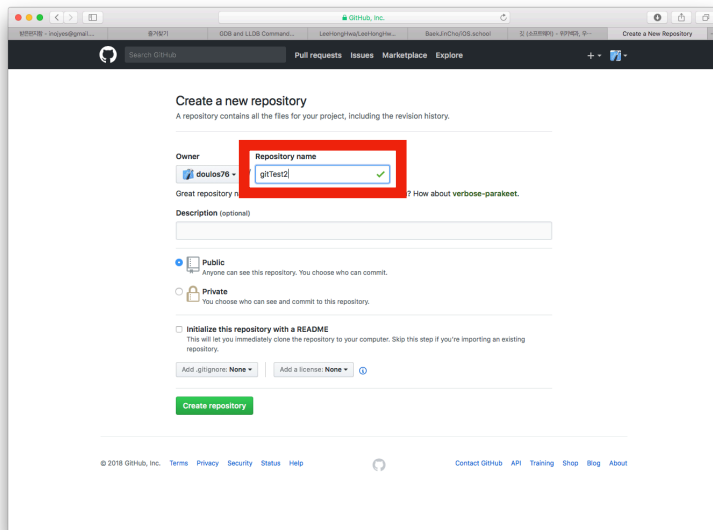




## gitHub에서 원격 저장소 설정 방법

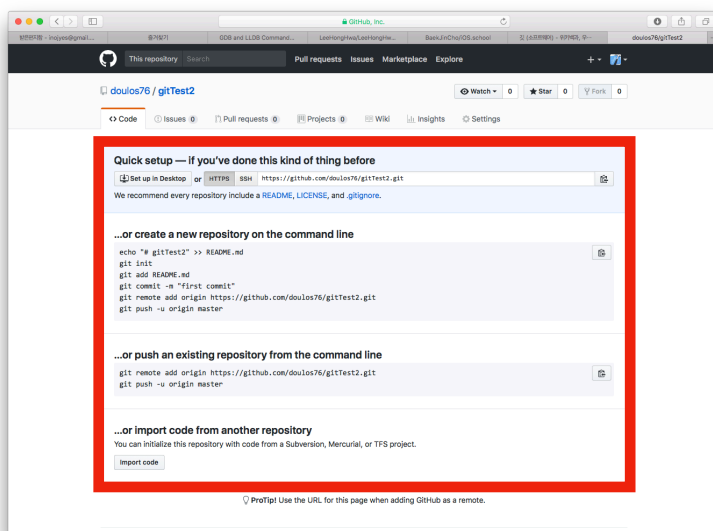


New repository 선택



Repository name 을 directory name 과 동일하게 설정

Creat repository를 클릭한다.



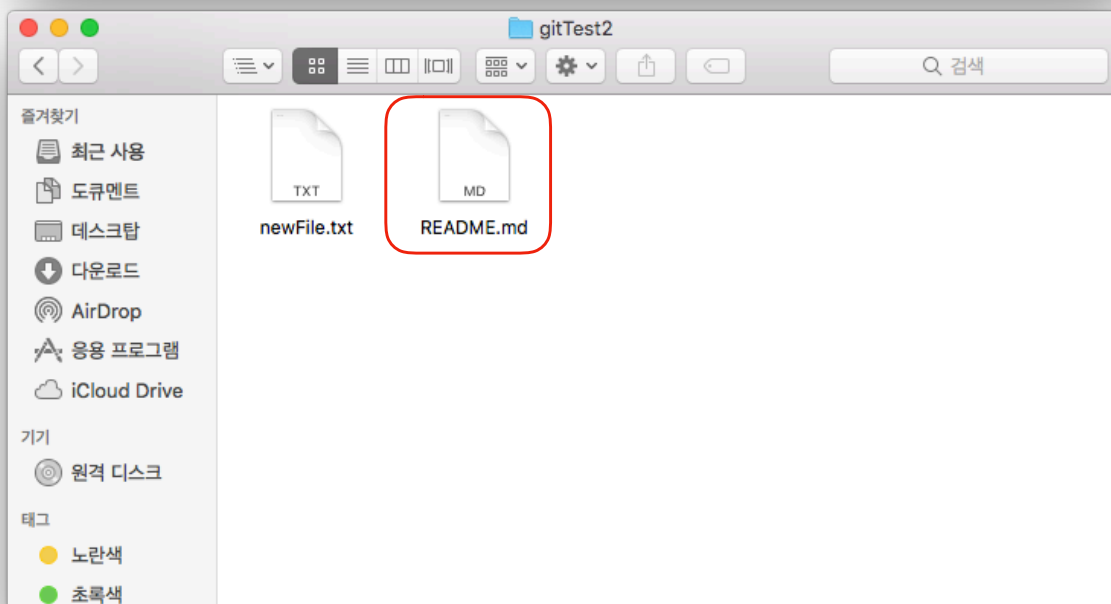
옆의 그림 처럼 원격 저장소가 설정 되었음을 알 수 있다.

```
gitTest2 — -bash — 103x24
[guminhoey-MacBook-Pro:gitTest2 dave$ ls
[guminhoey-MacBook-Pro:gitTest2 dave$ vi newFile.txt
[guminhoey-MacBook-Pro:gitTest2 dave$ ls
newFile.txt
[guminhoey-MacBook-Pro:gitTest2 dave$ git version
git version 2.10.1
[guminhoey-MacBook-Pro:gitTest2 dave$ echo "# gitTest2" >> README.md
```

README.md 라일 생성

```
gitTest2 — -bash — 103x24
[guminhoey-MacBook-Pro:gitTest2 dave$ ls
[guminhoey-MacBook-Pro:gitTest2 dave$ vi newFile.txt
[guminhoey-MacBook-Pro:gitTest2 dave$ ls
newFile.txt
[guminhoey-MacBook-Pro:gitTest2 dave$ git version
git version 2.10.1
[guminhoey-MacBook-Pro:gitTest2 dave$ echo "# gitTest2" >> README.md
[guminhoey-MacBook-Pro:gitTest2 dave$
```

“README.md”가 이상 없이 생길 경우 별도 메시지가 표시되지 않는다.



Finder에서 “README.md”생겼음을 확인 할 수 있다.

## git init으로 git 내부 저장소 생성

```
gitTest2 — -bash — 103x24
[guminhoeui-MacBook-Pro:gitTest2 dave$ ls
[guminhoeui-MacBook-Pro:gitTest2 dave$ vi newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ ls
newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git version
git version 2.10.1
[guminhoeui-MacBook-Pro:gitTest2 dave$ echo "# gitTest2" >> README.md
[guminhoeui-MacBook-Pro:gitTest2 dave$ git init
```

```
gitTest2 — -bash — 103x24
[guminhoeui-MacBook-Pro:gitTest2 dave$ ls
[guminhoeui-MacBook-Pro:gitTest2 dave$ vi newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ ls
newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git version
git version 2.10.1
[guminhoeui-MacBook-Pro:gitTest2 dave$ echo "# gitTest2" >> README.md
[guminhoeui-MacBook-Pro:gitTest2 dave$ git init
Initialized empty Git repository in /Users/dave/gitTest2/.git/
[guminhoeui-MacBook-Pro:gitTest2 dave$ █
```

## “git add . “ 으로 index(stage)에 저장

```
gitTest2 — -bash — 103x24
[guminhoeui-MacBook-Pro:gitTest2 dave$ ls
[guminhoeui-MacBook-Pro:gitTest2 dave$ vi newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ ls
newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git version
git version 2.10.1
[guminhoeui-MacBook-Pro:gitTest2 dave$ echo "# gitTest2" >> README.md
[guminhoeui-MacBook-Pro:gitTest2 dave$ git init
Initialized empty Git repository in /Users/dave/gitTest2/.git/
[guminhoeui-MacBook-Pro:gitTest2 dave$ git add . █
```

```
gitTest2 — -bash — 103x24
[guminhoeui-MacBook-Pro:gitTest2 dave$ ls
[guminhoeui-MacBook-Pro:gitTest2 dave$ vi newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ ls
newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git version
git version 2.10.1
[guminhoeui-MacBook-Pro:gitTest2 dave$ echo "# gitTest2" >> README.md
[guminhoeui-MacBook-Pro:gitTest2 dave$ git init
Initialized empty Git repository in /Users/dave/gitTest2/.git/
[guminhoeui-MacBook-Pro:gitTest2 dave$ git add .
[guminhoeui-MacBook-Pro:gitTest2 dave$
```

“git status “로 현재 상태 확인

```
gitTest2 — -bash — 103x24
[guminhoeui-MacBook-Pro:gitTest2 dave$ ls
[guminhoeui-MacBook-Pro:gitTest2 dave$ vi newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ ls
newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git version
git version 2.10.1
[guminhoeui-MacBook-Pro:gitTest2 dave$ echo "# gitTest2" >> README.md
[guminhoeui-MacBook-Pro:gitTest2 dave$ git init
Initialized empty Git repository in /Users/dave/gitTest2/.git/
[guminhoeui-MacBook-Pro:gitTest2 dave$ git add .
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   README.md
        new file:   newFile.txt

guminhoeui-MacBook-Pro:gitTest2 dave$
```

“git commit -m “first commit” 명령어로 commit과 commit message 기록

```
gitTest2 — -bash — 103x24
[guminhoeui-MacBook-Pro:gitTest2 dave$ ls
[guminhoeui-MacBook-Pro:gitTest2 dave$ vi newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ ls
newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git version
git version 2.10.1
[guminhoeui-MacBook-Pro:gitTest2 dave$ echo "# gitTest2" >> README.md
[guminhoeui-MacBook-Pro:gitTest2 dave$ git init
Initialized empty Git repository in /Users/dave/gitTest2/.git/
[guminhoeui-MacBook-Pro:gitTest2 dave$ git add .
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   README.md
        new file:   newFile.txt

guminhoeui-MacBook-Pro:gitTest2 dave$ git commit -m "first commit"
```

```
gitTest2 — -bash — 103x24
newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git version
git version 2.10.1
[guminhoeui-MacBook-Pro:gitTest2 dave$ echo "# gitTest2" >> README.md
[guminhoeui-MacBook-Pro:gitTest2 dave$ git init
Initialized empty Git repository in /Users/dave/gitTest2/.git/
[guminhoeui-MacBook-Pro:gitTest2 dave$ git add .
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   README.md
        new file:   newFile.txt

[guminhoeui-MacBook-Pro:gitTest2 dave$ git commit -m "first commit"
[master (root-commit) e880b4d] first commit
 2 files changed, 2 insertions(+)
 create mode 100644 README.md
 create mode 100644 newFile.txt
guminhoeui-MacBook-Pro:gitTest2 dave$
```

```
gitTest2 — -bash — 103x24
[guminhoeui-MacBook-Pro:gitTest2 dave$ echo "# gitTest2" >> README.md
[guminhoeui-MacBook-Pro:gitTest2 dave$ git init
Initialized empty Git repository in /Users/dave/gitTest2/.git/
[guminhoeui-MacBook-Pro:gitTest2 dave$ git add .
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   README.md
        new file:   newFile.txt

[guminhoeui-MacBook-Pro:gitTest2 dave$ git commit -m "first commit"
[master (root-commit) e880b4d] first commit
 2 files changed, 2 insertions(+)
 create mode 100644 README.md
 create mode 100644 newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
nothing to commit, working tree clean
guminhoeui-MacBook-Pro:gitTest2 dave$
```

파일 수정 - commit 하기 위한 가정 상태 만들

```
gitTest2 — -bash — 103x24
[guminhoeui-MacBook-Pro:gitTest2 dave$ echo "# gitTest2" >> README.md
[guminhoeui-MacBook-Pro:gitTest2 dave$ git init
Initialized empty Git repository in /Users/dave/gitTest2/.git/
[guminhoeui-MacBook-Pro:gitTest2 dave$ git add .
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   README.md
        new file:   newFile.txt

[guminhoeui-MacBook-Pro:gitTest2 dave$ git commit -m "first commit"
[master (root-commit) e880b4d] first commit
 2 files changed, 2 insertions(+)
 create mode 100644 README.md
 create mode 100644 newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
nothing to commit, working tree clean
guminhoeui-MacBook-Pro:gitTest2 dave$ vi newFile.txt
```

```
gitTest2 — vi newFile.txt — 103x24
New File 생성

add commit
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

```
gitTest2 — vi newFile.txt — 103x24
New File 생성

add commit
~
~
~
~
~
~
~
~
~
~
~
~
:wd
```

파일이 수정되었음이 표시됨

```
gitTest2 — -bash — 103x24
(use "git rm --cached <file>..." to unstage)
    new file:   README.md
    new file:   newFile.txt

[guminhoeui-MacBook-Pro:gitTest2 dave$ git commit -m "first commit"
[master (root-commit) e800b4d] first commit
 2 files changed, 2 insertions(+)
 create mode 100644 README.md
 create mode 100644 newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
nothing to commit, working tree clean
[guminhoeui-MacBook-Pro:gitTest2 dave$ vi newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   newFile.txt

no changes added to commit (use "git add" and/or "git commit -a")
[guminhoeui-MacBook-Pro:gitTest2 dave$
```

## index에 저장

```
gitTest2 — -bash — 103x24
(use "git rm --cached <file>..." to unstage)

new file:   README.md
new file:   newFile.txt

[guminhoeui-MacBook-Pro:gitTest2 dave$ git commit -m "first commit"
[master (root-commit) e880b4d] first commit
 2 files changed, 2 insertions(+)
 create mode 100644 README.md
 create mode 100644 newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
nothing to commit, working tree clean
[guminhoeui-MacBook-Pro:gitTest2 dave$ vi newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   newFile.txt

no changes added to commit (use "git add" and/or "git commit -a")
guminhoeui-MacBook-Pro:gitTest2 dave$ git add .
```

```
gitTest2 — -bash — 103x24
create mode 100644 README.md
create mode 100644 newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
nothing to commit, working tree clean
[guminhoeui-MacBook-Pro:gitTest2 dave$ vi newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   newFile.txt

no changes added to commit (use "git add" and/or "git commit -a")
guminhoeui-MacBook-Pro:gitTest2 dave$ git add .
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   newFile.txt

guminhoeui-MacBook-Pro:gitTest2 dave$ █
```

## index에 있는 내용을 내부 저장소에 스냅샷으로 저장 (commit )

```
gitTest2 — -bash — 103x24
create mode 100644 README.md
create mode 100644 newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
nothing to commit, working tree clean
[guminhoeui-MacBook-Pro:gitTest2 dave$ vi newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   newFile.txt

no changes added to commit (use "git add" and/or "git commit -a")
guminhoeui-MacBook-Pro:gitTest2 dave$ git add .
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   newFile.txt

guminhoeui-MacBook-Pro:gitTest2 dave$ git commit -m "add commit"
```

```
gitTest2 — -bash — 103x24
On branch master
nothing to commit, working tree clean
[guminhoeui-MacBook-Pro:gitTest2 dave$ vi newFile.txt
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   newFile.txt

no changes added to commit (use "git add" and/or "git commit -a")
[guminhoeui-MacBook-Pro:gitTest2 dave$ git add .
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   newFile.txt

[guminhoeui-MacBook-Pro:gitTest2 dave$ git commit -m "add commit"
[master afbd715] add commit
1 file changed, 4 insertions(+)
guminhoeui-MacBook-Pro:gitTest2 dave$
```

```
gitTest2 — -bash — 103x24
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   newFile.txt

no changes added to commit (use "git add" and/or "git commit -a")
[guminhoeui-MacBook-Pro:gitTest2 dave$ git add .
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   newFile.txt

[guminhoeui-MacBook-Pro:gitTest2 dave$ git commit -m "add commit"
[master afbd715] add commit
1 file changed, 4 insertions(+)
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
nothing to commit, working tree clean
guminhoeui-MacBook-Pro:gitTest2 dave$
```

```
gitTest2 — -bash — 103x24
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   newFile.txt

no changes added to commit (use "git add" and/or "git commit -a")
[guminhoeui-MacBook-Pro:gitTest2 dave$ git add .
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   newFile.txt

[guminhoeui-MacBook-Pro:gitTest2 dave$ git commit -m "add commit"
[master afbd715] add commit
1 file changed, 4 insertions(+)
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
nothing to commit, working tree clean
guminhoeui-MacBook-Pro:gitTest2 dave$ remote add origin https://github.com/doulos76/gitTest2.git
```



## 원격 저장소에 저장

```
gitTest2 — -bash — 103x24

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   newFile.txt

no changes added to commit (use "git add" and/or "git commit -a")
[guminhoeui-MacBook-Pro:gitTest2 dave$ git add .
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   newFile.txt

[guminhoeui-MacBook-Pro:gitTest2 dave$ git commit -m "add commit"
[master afbd715] add commit
 1 file changed, 4 insertions(+)
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
nothing to commit, working tree clean
[guminhoeui-MacBook-Pro:gitTest2 dave$ remote add origin https://github.com/doulos76/gitTest2.git
-bash: remote: command not found
guminhoeui-MacBook-Pro:gitTest2 dave$ git remote add origin https://github.com/doulos76/gitTest2.git
```

```
gitTest2 — -bash — 103x24

  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   newFile.txt

no changes added to commit (use "git add" and/or "git commit -a")
[guminhoeui-MacBook-Pro:gitTest2 dave$ git add .
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   newFile.txt

[guminhoeui-MacBook-Pro:gitTest2 dave$ git commit -m "add commit"
[master afbd715] add commit
 1 file changed, 4 insertions(+)
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
nothing to commit, working tree clean
[guminhoeui-MacBook-Pro:gitTest2 dave$ remote add origin https://github.com/doulos76/gitTest2.git
-bash: remote: command not found
[guminhoeui-MacBook-Pro:gitTest2 dave$ git remote add origin https://github.com/doulos76/gitTest2.git
guminhoeui-MacBook-Pro:gitTest2 dave$
```

```
gitTest2 — -bash — 103x24

        modified:   newFile.txt

no changes added to commit (use "git add" and/or "git commit -a")
[guminhoeui-MacBook-Pro:gitTest2 dave$ git add .
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

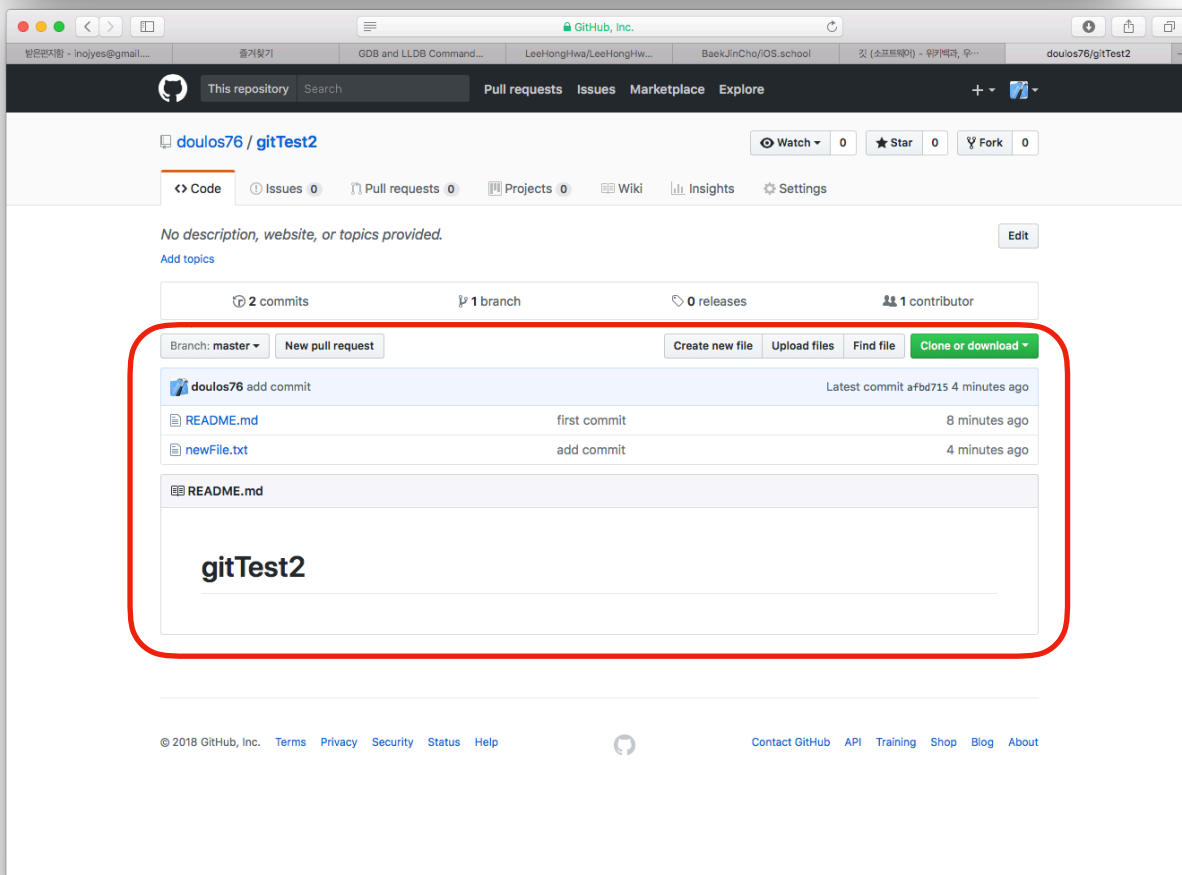
        modified:   newFile.txt

[guminhoeui-MacBook-Pro:gitTest2 dave$ git commit -m "add commit"
[master afbd715] add commit
 1 file changed, 4 insertions(+)
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
nothing to commit, working tree clean
[guminhoeui-MacBook-Pro:gitTest2 dave$ remote add origin https://github.com/doulos76/gitTest2.git
-bash: remote: command not found
[guminhoeui-MacBook-Pro:gitTest2 dave$ git remote add origin https://github.com/doulos76/gitTest2.git
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
nothing to commit, working tree clean
guminhoeui-MacBook-Pro:gitTest2 dave$ git push -u origin master
```

```
gitTest2 — -bash — 103x24
modified:   newFile.txt

[guminhoeui-MacBook-Pro:gitTest2 dave$ git commit -m "add commit"
[master afbd715] add commit
1 file changed, 4 insertions(+)
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
nothing to commit, working tree clean
[guminhoeui-MacBook-Pro:gitTest2 dave$ remote add origin https://github.com/doulos76/gitTest2.git
-bash: remote: command not found
[guminhoeui-MacBook-Pro:gitTest2 dave$ git remote add origin https://github.com/doulos76/gitTest2.git
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
nothing to commit, working tree clean
[guminhoeui-MacBook-Pro:gitTest2 dave$ git push -u origin master
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (7/7), 552 bytes | 0 bytes/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To https://github.com/doulos76/gitTest2.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
guminhoeui-MacBook-Pro:gitTest2 dave$
```

```
gitTest2 — -bash — 103x24
1 file changed, 4 insertions(+)
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
nothing to commit, working tree clean
[guminhoeui-MacBook-Pro:gitTest2 dave$ remote add origin https://github.com/doulos76/gitTest2.git
-bash: remote: command not found
[guminhoeui-MacBook-Pro:gitTest2 dave$ git remote add origin https://github.com/doulos76/gitTest2.git
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
nothing to commit, working tree clean
[guminhoeui-MacBook-Pro:gitTest2 dave$ git push -u origin master
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (7/7), 552 bytes | 0 bytes/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To https://github.com/doulos76/gitTest2.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
[guminhoeui-MacBook-Pro:gitTest2 dave$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working tree clean
guminhoeui-MacBook-Pro:gitTest2 dave$
```



## 정리

1. git은 분산 버전 관리 시스템으로 프로젝트 변경 이력을 관리하는데 도움을 줌
2. git의 작업이 이뤄지는 저장소의 시점을 work tree라고 하며 branch생성을 통해서 다양한 시점의 작업이 가능하.
3. branch를 현재 작업 tree로 변경하기 위해서는 “checkout”명령을 통해서 변경 가능함(\$git checkout branch\_name)
4. index영역이 존재해서 work tree에서 작업한 내용을 곧바로 내부 저장소에 저장하지 않고 선별적으로 저장할 내용을 add할 수 있다.
5. commit 명령을 통해 index에 있는 변경 내용을 내부저장소에 저장할 수 있다.
6. 외부 저장소를 이용 중앙 집중 저장소로 사용할 수 있다.
7. 대표적인 외부 저장소로는 github이란 서비스가 있고, 설정을 통해 외부 서버를 만들 수도 있음
8. push & pulld을 통해 외부 저장소에 데이터를 저장하고 불러올 수 있다.
9. clone 명령어를 통해서 외부 저장소를 내부 저장소로 복제 할 수 있다.

## git Data Transport Commands

