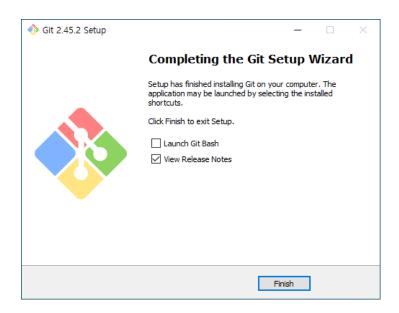
1. Git&Github 개요

1) Git 설치

₩실습1-1. Git 설치하기



- ✔ Install 클릭 이후 총 15단계 Next 진행
- ✔ 각 단계별 기본값은 수정하지 않음



- ✔ 설치완료 후 Finish 클릭
- ✓ 바탕화면에서 마우스 오른버튼 Open Git Bash here 클릭

2) Git bash 기본 명령어

명령어	사용 예	설명
	ls	- 현재 디렉터리 조회
	ls -1	- 현재 디렉터리 상세 조회
ls	ls -al	- 현재 디렉터리 숨김 포함 조회
	cd /	
	cd ./	
cd	cd/	디렉터리 이동
	rm <file></file>	
rm	rm -rf <directory></directory>	파일 또는 디렉터리 삭제
mkdir	mkdir <directory></directory>	디렉터리 생성
touch	touch <file></file>	파일 생성
cat	cat <file></file>	파일 내용 출력
clear	clear	화면 지우기
vi	vi <file></file>	편집기 실행

₩실습2-1. Git bash 기본 명령어 실습하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop

$ cd ./Workspace

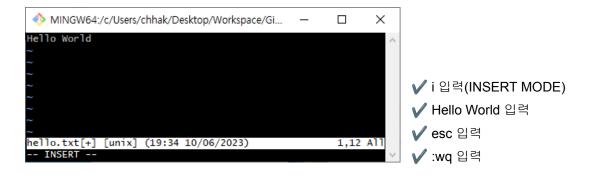
$ mkdir Git

$ cd ./Git

$ mkdir ch01

$ cd ch01

$ vi hello.txt
```



2. 버전관리

1) 버전관리 기본

주요 명령어	사용 예	설명
git init	git init	새로운 Git 저장소 초기화
git config	git configglobal user.name git configglobal user.email	Git 환경 설정
git status	git status	Git 저장소 상태 확인
git add	git add <file></file>	- 특정 파일 스테이징 - 모든 파일 및 디렉터리 스테이징
git commit	<pre>git commit -m <message> git commit -am <message></message></message></pre>	- 스테이지 파일 커밋 - 동시에 스테이지와 파일 커밋
git log	git log git logstat git logoneline git loggraph	- 커밋 기록 확인- 커밋 기록 통계 정보 확인- 커밋 기록 요약 정보 확인- 커밋 기록 그래프 정보 확인

₩실습1-1. Git 저장소 생성 및 환경 설정

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop
$ cd ./Workspace/Git

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git
$ git config --global user.email "chhak@503@gmail.com"
$ git config --global user.name "chhak@503"
$ git config --list

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git
$ git init

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git (master)
$ git status
On branch master
No commits yet
nothing to commit (create/copy files and use "git add" to track)
```

₩실습1-2. 문서 생성 및 내용 입력

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git (master)

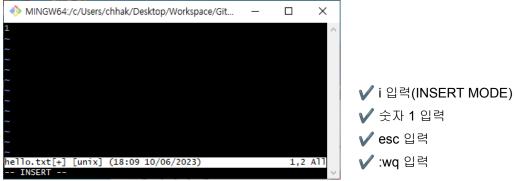
$ mkdir ch02

$ cd ch02

$ vi hello.txt

MINGW64:/c/Users/chhak/Desktop/Workspace/Git... — 

X
```



```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git (master)
$ ls -l
total 1
-rw-r--r-- 1 chhak 197121 2 Jun 10 16:16 hello.txt
```

₩실습1-3. 문서 현재 상태 확인

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)

$ git status
...
Untracked files:
    hello.txt
```



알실습1-4. 문서 Staging 하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git add hello.txt

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git status
...
Changes not staged for commit:
    new file: hello.txt
```



⇔실습**1-5**. 문서 **Commit** 하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git commit -m "add 1"
...
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git status
nothing to commit, working tree clean
```



⇔실습**1-6**. 문서 버전 이력 확인하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git log
commit cb8e0f15247a5691fdb7e768d6d6ec9360724046 (HEAD -> master)
...
    add 1
```

₩실습1-7. 문서 버전 수정하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ vi hello.txt
```

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)

$ git status
...
Changes not staged for commit:
    modified: hello.txt
```

일실습1-8. 문서 버전 Staging & Commit 후 이력 확인

₩실습1-9. 새 문서 버전 추가하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ vi welcome.txt
```

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git status
...
Untracked files:
    welcome.txt

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git add welcome.txt
$ git status
...
Changes to be committed:
    new file: welcome.txt

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git commit -m "add a, b, c, d"
...
create mode 100644 ch02/welcome.txt

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git status
...
nothing to commit, working tree clean
```

₩실습1-10. 문서 버전 최종 수정하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ vi welcome.txt
```

⇔실습**1-11**. 문서 버전 이력 상세 확인하기

2) 버전관리 고급

주요 명령어	사용 예	설명	
git checkout	git checkout <file></file>	- 특정 커밋 파일 내용 복원	
git reset	git reset git reset . git reset HEAD git reset HEAD^ git reset HEAD^1 git reset HEAD~2 git resetsoft HEAD^ git resetmixed HEAD^ git resethard HEAD^ git reset <-COMMIT_ID>	- 스테이지 내용 내리기 - git reset 동일 - git reset 동일 - 한 단계 이전 커밋으로 되돌아가기 - 두 단계 이전 커밋으로 되돌아가기 - HEAD^ 동일 - HEAD^ 동일 - 커밋 이동, 스테이지 유지, 작업트리 유지 - 커밋 이동, 스테이지 수정, 작업트리 유지 - 커밋 이동, 스테이지 수정, 작업트리 수정 - 특정 커밋으로 기록 없이 되돌아가기	
git revert	git revert HEAD git revert <commit_id></commit_id>	- 이전 커밋으로 기록 남기고 되돌아가기(취소하기) - 특정 커밋으로 기록 남기고 되돌아가기(취소하기)	

╩실습2-1. checkout 실습하기

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
\$ vi hello.txt



```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)

$ git status
Changes not staged for commit:
    modified: hello.txt
```

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ cat hello.txt
1
2
3
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git checkout hello.txt
Updated 1 path from the index
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git status
nothing to commit, working tree clean
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ cat hello.txt
1
2
```

😀실습2-2. reset 실습하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ vi hello.txt
```



```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git status
Changes not staged for commit:
       modified: hello.txt
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git commit -am "add 3"
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git log --oneline
0ac637b (HEAD -> master) add 3
b3beefd add 2
cb8e0f1 add 1
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git reset HEAD^
Unstaged changes after reset:
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git log --oneline
b3beefd (HEAD -> master) add 2
cb8e0f1 add 1
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git status
Changes not staged for commit:
       modified: hello.txt
```

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ cat hello.txt
2
3
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git checkout hello.txt
Updated 1 path from the index
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ cat hello.txt
2
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git status
nothing to commit, working tree clean
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git reset --hard HEAD^
HEAD is now at cb8e0f1 add 1
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git log --oneline
cb8e0f1 (HEAD -> master) add 1
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git status
nothing to commit, working tree clean
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ cat hello.txt
1
```

⇔실습2-3. revert 실습하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ vi hello.txt 🁈 숫자 2 입력 후 저장/종료

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git commit -am "add 2"
...

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ vi hello.txt 🁈 숫자 3 입력 후 저장/종료

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git commit -am "add 3"
...

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git log --oneline
ae2cf3f (HEAD -> master) add 3
de2bf7a add 2
cb8e0f1 add 1

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git revert HEAD
```

```
● MINGW64:/c/Users/java/Desktop/Workspace/git/ch02 ー □
Revert "add 3"

This reverts commit 92d6f79367ab054534bd9f395868c96d80a42738.

# Please enter the commit message for your changes. Lines starting # with '#' will be ignored, and an empty message aborts the commit.

# # On branch master # Changes to be committed: # modified: hello.txt # cgit/COMMIT_EDITMSG [unix] (15:27 17/06/2024) 1,1 Wq
```

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)

$ git log --oneline
6d6355d (HEAD -> master) Revert "add 3"
...

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)

$ cat hello.txt

1
2
```

₩실습2-4. revert 충돌 실습하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git reset --hard HEAD~1
HEAD is now at ae2cf3f add 3
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git log --oneline
ae2cf3f (HEAD -> master) add 3
de2bf7a add 2
cb8e0f1 add 1
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git revert <add 2 COMMIT_ID>
Auto-merging hello.txt
CONFLICT (content): Merge conflict in hello.txt
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master|REVERTING)
$ git status
Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
       both modified: hello.txt
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master|REVERTING)
$ vi hello.txt
```

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master|REVERTING)
$ git add hello.txt
```

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master|REVERTING)
$ git status
...
Changes to be committed:
   (use "git restore --staged <file>..." to unstage)
        modified: hello.txt

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master|REVERTING)
$ git revert --continue
```

```
MINGW64:/c/Users/java/Desktop/Workspace/git/ch02 — □

Revert "add 2"

This reverts commit de2bf7a653e99f958073ca094e7f21deaa27e785.

# Conflicts:
    hello.txt

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.

# On branch master
# You are currently reverting commit de2bf7a.

# Changes to be committed:
# modified: hello.txt

# Changes to be committed:
# Work This Total Area (기본값 사용)

* Revert 메시지 작성(기본값 사용)

* .git/COMMIT_EDITMSG [unix] (15:50 17/06/2024)
```

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git status
nothing to commit, working tree clean

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ git log --oneline
f234dab (HEAD -> master) Revert "add 2"
ae2cf3f add 3
de2bf7a add 2
cb8e0f1 add 1

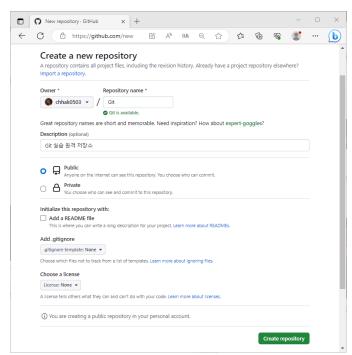
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch02 (master)
$ cat hello.txt
1
```

3. 원격 저장소

1) 원격 저장소 생성과 연결

주요 명령어	사용 예	설명
git remote	git remote add origin <remote-url></remote-url>	- 원격 저장소 연결 - 원격 저장소 연결 확인

₩실습 1-1. Github 원격 저장소 생성하기



- ✔ 저장소 이름(Git 입력)
- ✔ 저장소 공개 여부(Public 선택)
- ✔ Create repository 버튼 클릭

○ 실습 1-2. 원격 저장소 연결하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop
$ cd ~/Desktop/Workspace/Git

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git (master)
$ git remote add origin https://github.com/계정명/저장소명.git

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git (master)
$ git remote -v

origin https://github.com/계정명/저장소명.git (fetch)

origin https://github.com/계정명/저장소명.git (push)
```

2) 원격 저장소 push와 pull

명령어	사용 예	설명
git push	git push origin master git push -u origin master	- 원격 저장소 커밋 등록 - 추적 브랜치 설정 이후 커밋 등록
git pull	git pull origin master	원격 저장소 커밋 병합

⇔실습 2-1. 문서 생성 및 내용 입력

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git (master)
$ mkdir ch03
$ cd ch03
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch03 (master)
$ vi test1.txt 👈 알파벳 a 입력 후 저장/종료
$ vi test2.txt 👈 알파벳 a 입력 후 저장/종료
$ vi test3.txt → 알파벳 a 입력 후 저장/종료
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch03 (master)
$ cd ..
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git (master)
$ git add ch03
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git (master)
$ git commit -m 'add ch03 git study'
[master 59ca7be] add ch03 git study
3 files changed, 3 insertions(+)
create mode 100644 ch03/test1.txt
create mode 100644 ch03/test2.txt
create mode 100644 ch03/test3.txt
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git (master)
$ git log
commit 59ca7be2f049f03453de48214a8489555f7d1e38 (HEAD -> master)
Author: chhak0503 <chhak0503@gmail.com>
   add ch03 git study
```

일실습 2-2. 원격 저장소 push 하기

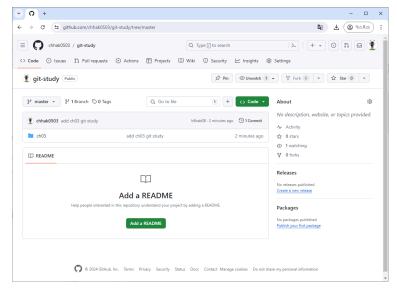
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git (master)

\$ git push origin master
...

Total 13 (delta 0), reused 0 (delta 0), pack-reused 0

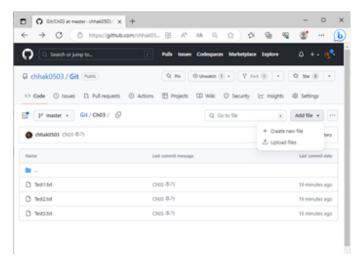
To https://github.com/chhak/Git.git

* [new branch] master -> master

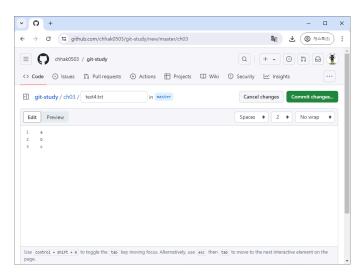


- ✔ 원격 저장소 확인
- ✔ 커밋 기록 확인

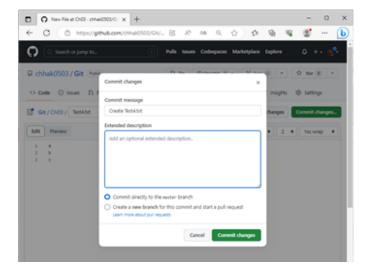
₩실습 2-3. 원격 저장소 문서 생성하기



- ✓ Add file 클릭
- ✔ Create new file 클릭



- ✔ 파일명 test4.txt 입력
- ✔ a, b, c 입력
- ✓ Commit changes... 클릭



- ✓ Commit message 확인
- ✓ Commit changes 클릭
- ✔ test4.txt 파일 생성 확인
- ✓ Commit 이력 확인

₩실습 2-4. 원격 저장소 pull 하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git (master)
$ git pull origin master
Fast-forward
ch03/test4.txt | 3 +++
1 file changed, 3 insertions(+)
create mode 100644 ch03/test4.txt
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git (master)
$ cd ch03
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch03 (master)
$ 1s -1
total 4
-rw-r--r-- 1 chhak 197121 2 Jun 11 11:58 test1.txt
-rw-r--r-- 1 chhak 197121 2 Jun 11 11:58 test2.txt
-rw-r--r-- 1 chhak 197121 2 Jun 11 11:58 test3.txt
-rw-r--r-- 1 chhak 197121 9 Jun 11 14:54 test4.txt
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch03 (master)
$ cat test4.txt
b
c
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch03 (master)
$ git log
commit fa482d13f64903ddfc7e503ecca42ec2e0d447bf (HEAD -> master, origin/master)
Author: chhak0503 <64509878+chhak0503@users.noreply.github.com>
    Create test4.txt
. . .
```

3) 원격 저장소 복제

명령어	사용 예	설명
git clone	git clone <remote-url></remote-url>	원격 저장소 복제

₩실습 3-1. 원격 저장소 복제하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop
$ cd ~/Desktop
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop
$ git clone https://github.com/계정명/저장소명.git Git_home
Cloning into 'Git_home'...
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop
$ git clone https://github.com/계정명/저장소명.git Git_office
Cloning into 'Git_office'...
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop
$ cd Git_home/
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Git_home (master)
$ git log
commit fa482d13f64903ddfc7e503ecca42ec2e0d447bf (HEAD -> master, origin/master)
Author: chhak0503 <64509878+chhak0503@users.noreply.github.com>
   Create test4.txt
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Git_home (master)
$ cd ..
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop
$ cd Git office/
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Git_office (master)
$ git log
commit fa482d13f64903ddfc7e503ecca42ec2e0d447bf (HEAD -> master, origin/master)
Author: chhak0503 <64509878+chhak0503@users.noreply.github.com>
   Create test4.txt
. . .
```

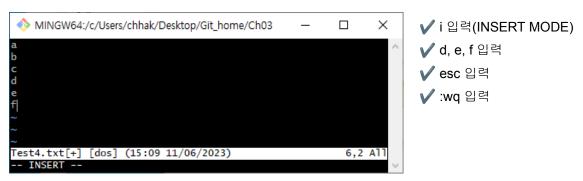
₩실습 3-2. 원격 저장소 push 하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop

$ cd ~/Desktop/Git_home/ch03

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Git_home/ch03 (master)

$ vi test4.txt
```



```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Git_home/ch03 (master)

$ git add test4.txt

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Git_home/ch03 (master)

$ git commit -m 'add d, e, f'

[master 7ccd5ba] add d, e, f

1 file changed, 3 insertions(+)

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Git_home/ch03 (master)

$ git push origin master
...

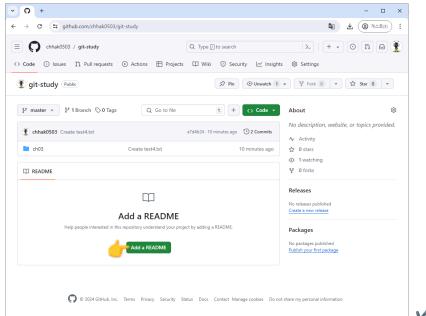
To https://github.com/chhak/Git.git
fa482d1..7ccd5ba master -> master
```

₩실습 3-3. 원격 저장소 pull 하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop
$ cd ~/Desktop/Git_office
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Git_office (master)
$ git pull origin master
Fast-forward
ch03/test4.txt | 3 +++
1 file changed, 3 insertions(+)
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Git_office (master)
$ cd ch03
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Git_office/ch03 (master)
$ 1s -1
total 4
-rw-r--r-- 1 chhak 197121 3 Jun 11 15:09 test1.txt
-rw-r--r-- 1 chhak 197121 3 Jun 11 15:09 test2.txt
-rw-r--r-- 1 chhak 197121 3 Jun 11 15:09 test3.txt
-rw-r--r-- 1 chhak 197121 18 Jun 11 15:29 test4.txt
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Git_office/ch03 (master)
$ cat test4.txt
b
С
d
f
```

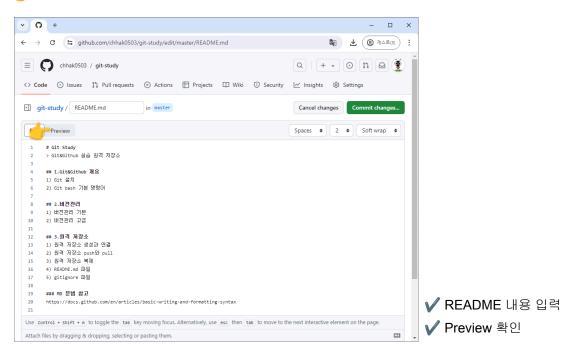
4) README.md

₩실습 4-1. README 파일 생성하기

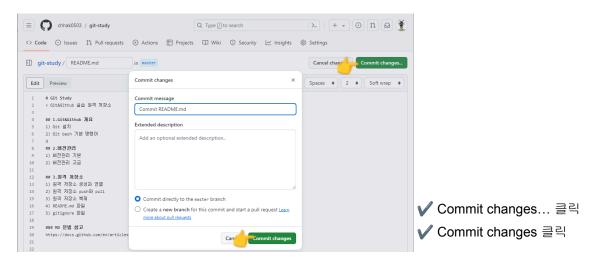


✔ Add a README 클릭

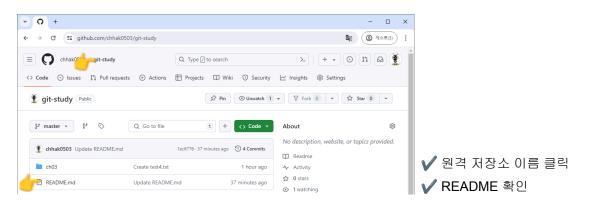
알실습 **4-2**. **README** 작성하기



알실습 **4-3**. **README** 커밋하기



알실습 **4-4**. **README** 확인하기



일실습 **4-5**. 원격 저장소 pull 하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop
$ cd ~/Desktop/Workspace/Git

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git (master)
$ git pull origin master
...

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git (master)
$ ls -l
total 5
-rw-r--r-- 1 java 197121 457 Jun 18 12:29 README.md
...
```

5) .gitignore

😀실습 5-1. .gitignore 파일 생성하기

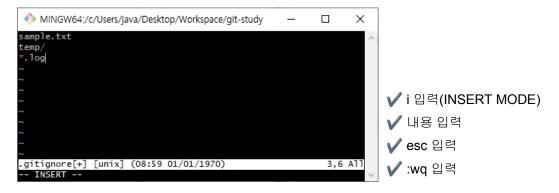
```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop
$ cd ~/Desktop/Workspace/Git/ch03

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch03 (master)
$ touch sample.txt sample.log test5.txt

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch03 (master)
$ mkdir sub temp

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch03 (master)
$ touch sub/test.txt temp/test.txt

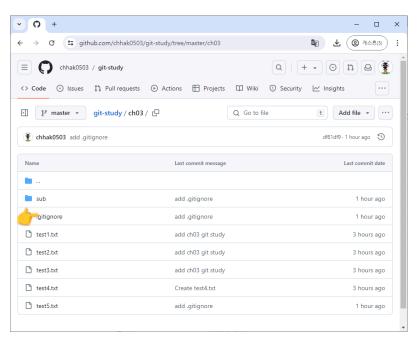
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch03 (master)
$ vi .gitignore
```



```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch03 (master)
$ ls -al
total 13
...
-rw-r--r-- 1 java 197121 23 Jun 18 11:57 .gitignore
-rw-r--r-- 1 java 197121 0 Jun 18 12:12 sample.log
-rw-r--r-- 1 java 197121 0 Jun 18 12:12 sample.txt
drwxr-xr-x 1 java 197121 0 Jun 18 12:14 sub/
drwxr-xr-x 1 java 197121 0 Jun 18 12:14 temp/
...
-rw-r--r-- 1 java 197121 0 Jun 18 12:12 test5.txt
```

일실습 5-2. .gitignore commit & push하기

₩실습 5-3. 원격 저장소 확인하기



- ✔ .gitignore 파일 확인
- ✔ 업로드 파일 확인

4. 브랜치

1) 브랜치 기본

주요 명령어	사용 예	설명
git branch	<pre>git branch git branch <branch_name> git branch -d <branch_name></branch_name></branch_name></pre>	- 현재 브랜치 나열 - 새 브랜치 생성 - 브랜치 삭제
git checkout	<pre>git checkout <branch_name> git checkout -b <branch_name></branch_name></branch_name></pre>	- 브랜치 전환 - 새 브랜치 생성 후 브랜치 전환

₩실습 1-1. 실습 디렉터리 및 파일 생성하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop
$ cd ~/Desktop/Workspace/Git

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git (master)
$ mkdir ch04

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git (master)
$ cd ch04

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ vi work1.txt
```



😀실습 1-2. 파일 커밋하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git add work1.txt
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git commit -m 'content 1'
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git log
commit ea2d890285646763f7d20cf98bbaf2811eda0576 (HEAD -> master)
Author: chhak0503 <chhak0503@gmail.com>
        content 1
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ vi work1.txt → 'content 2' 추가 입력 후 저장/종료
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git commit -am 'content 2'
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ vi work1.txt 👈 'content 3' 추가 입력 후 저장/종료
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git commit -am 'content 3'
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git log
commit 1cc55821e798613e54ced48889d1687282... (HEAD -> master)
Author: chhak0503 <chhak0503@gmail.com>
       content 3
. . .
```

₩실습 1-3. 브랜치 확인 및 생성하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git branch
* master
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git branch apple
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git branch
apple
* master
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git log
commit 1cc55821e798613e54ced48889d1687282... (HEAD -> master, apple)
Author: chhak0503 <chhak0503@gmail.com>
        content 3
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git branch banana
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git branch cherry
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git branch
apple
banana
cherry
* master
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git log
commit 1cc55821e798613e54ced48889d1687282... (HEAD -> master, cherry, banana, apple)
Author: chhak0503 <chhak0503@gmail.com>
        content 3
```

₩실습 1-4. 파일 수정 후 커밋하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ vi work1.txt  'content 4' 추가 입력 후 저장/종료

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git commit -am 'content 4 by master'
...
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git log --oneline

45fb5f7 (HEAD -> master) content 4 by master

1cc5582 (cherry, banana, apple) content 3

b6cac4c content 2

ea2d890 content 1
...
```

⇔실습 1-5. 브랜치 전환 후 파일 내용 확인하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git checkout apple
Switched to branch 'apple'

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (apple)
$ git log --oneline
1cc5582 (HEAD -> apple, cherry, banana) content 3
b6cac4c content 2
ea2d890 content 1
...
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (apple)
$ cat work1.txt
content 1
content 2
content 3
```

😀실습 1-6. 파일 수정 후 커밋하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (apple)
$ vi work1.txt i 'content 4 by apple' 추가 입력 후 저장/종료

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (apple)
$ vi apple.txt i 'content 4 by apple' 입력 후 저장/종료

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (apple)
$ git add work1.txt apple.txt

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (apple)
$ git commit -m 'content 4 by apple'

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (apple)
$ git log --oneline --branches --graph

* 83bd291 (HEAD -> apple) content 4 by apple

[ * 45fb5f7 (master) content 4 by master

]/

* 1cc5582 (cherry, banana) content 3

* b6cac4c content 2

* ea2d890 content 1
```

⇔실습 **1-7**. 브랜치 전환 후 삭제하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (apple)

$ git checkout master

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)

$ git branch -d apple
error: the branch 'apple' is not fully merged
...

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)

$ git branch -d banana cherry

Deleted branch banana (was 5c0e4ba).

Deleted branch cherry (was 5c0e4ba).

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)

$ git log --oneline --branches --graph
...
```

2) 브랜치 병합

주요 명령어	사용 예	설명
git merge	<pre>git merge <branch_name> git mergeabort</branch_name></pre>	- 브랜치 병합 - 브랜치 병합 취소

₩실습 2-1. 파일 생성 후 커밋하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop
$ cd ~/Desktop/Workspace/Git/ch04

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ vi work2.txt → 첫번째 줄에 'content 1 by master' 입력 후 저장/종료

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git add work2.txt

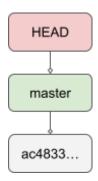
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git commit -m 'commit 1 by master'

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git log

commit ac4833d5fd2cbd81cf619fbe3508496f25d69403 (HEAD -> master)

...

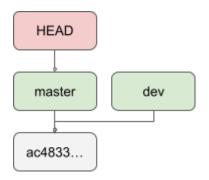
commit 1 by master
```



⇔실습 2-2. 브랜치 생성하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git branch dev

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git log --oneline
ac4833d (HEAD -> master, dev) commit 1 by master
```



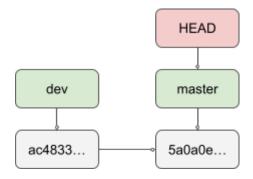
₩실습 2-3. 추가 파일 생성 후 커밋하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ vi work3.txt 🁈 첫번째 줄에 'content 1 by master' 입력 후 저장/종료

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git add work3.txt

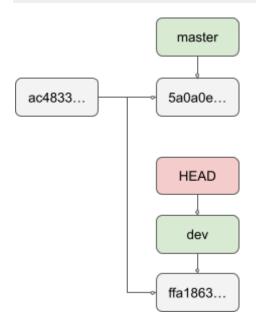
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git commit -m 'commit 2 by master'

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git log --oneline
5a0a0eb (HEAD -> master) commit 2 by master
ac4833d (dev) commit 1 by master
```

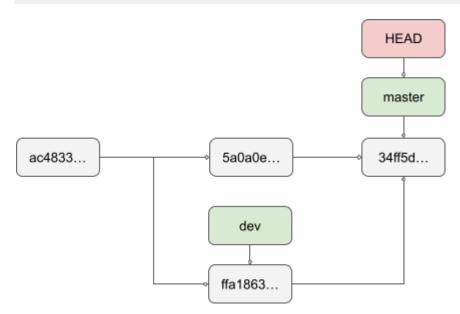


😃실습 2-4. 브랜치 전환 후 파일 생성 및 커밋하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git checkout dev
Switched to branch 'dev'
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (dev)
$ vi work4.txt → 첫번째 줄에 'content 1 by dev' 입력 후 저장/종료
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (dev)
$ git add work4.txt
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (dev)
$ git commit -m 'commit 3 by dev'
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (dev)
$ git log --oneline
ffa1863 (HEAD -> dev) commit 3 by dev
ac4833d commit 1 by master
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (dev)
$ git log --oneline --branches --graph
* ffa1863 (HEAD -> dev) commit 3 by dev
* 5a0a0eb (master) commit 2 by master
|/
* ac4833d commit 1 by master
```



₩실습 2-5. master 브랜치 전환 후 병합하기



✓ esc 입력

✓ :wq 입력

3) 브랜치 충돌

₩실습 3-1. 파일 생성 후 커밋하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop
$ cd ~/Desktop/Workspace/Git/ch04

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ vi work5.txt

MINGW64:/c/Users/java/Desktop/Workspace... - □ X

1 #section1
2
3
4 #section2
~ 1,4번 라인 내용 입력
```

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git commit -am 'work5 commit-1'
```

4,10 All

₩실습 3-2. 새 브랜치 생성 및 전환 후 커밋하기

work5.txt[+] [dos] (10:38 20/06/2024)

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git checkout -b feature
Switched to branch 'feature'

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (feature)
$ vi work5.txt
```

```
      I #section1
      ✓ :set nu

      3
      ✓ i 입력(INSERT MODE)

      5 content2 by feature
      ✓ 5번 라인 내용 입력

      Work5.txt[+] [dos] (10:38 20/06/2024)
      5,20 A11

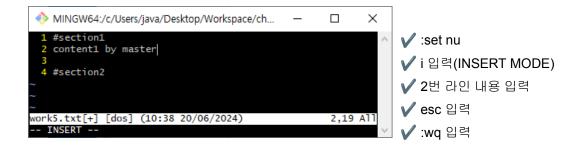
      -- INSERT --
      ✓ :wq 입력
```

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (feature)
$ git commit -am 'work5 commit-2'
```

₩실습 3-3. master 브랜치 전환 후 병합하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (feature)
$ git checkout master
Switched to branch 'master'

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ vi work5.txt
```



```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)

$ git commit -am 'work5 commit-3'

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)

$ git merge feature
...

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)

$ cat work5.txt
...

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)

$ git log --oneline --branches --graph
...
```

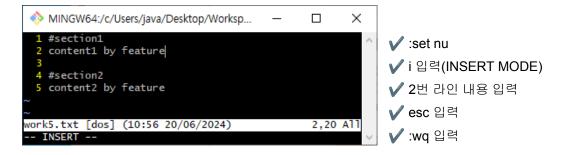
₩실습 3-4. 브랜치 전환 후 파일 수정하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)

$ git checkout feature
Switched to branch 'feature'

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (feature)

$ vi work5.txt
```



```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (feature)
$ git commit -am 'work5 commit-4'

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (feature)
$ git log --oneline --branches --graph
...
```

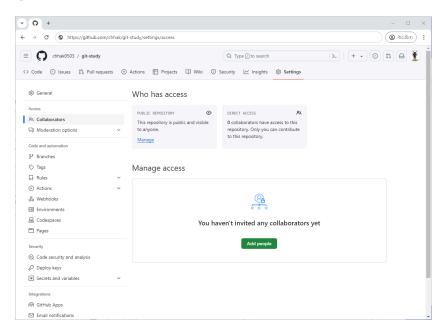
₩ 실습 3-5. master 브랜치 전환 후 병합하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (feature)
$ git checkout master
...
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git merge feature
...
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master|MERGING)
$ vi work5.txt
```



4) 풀 리퀘스트

일실습 **4-1**. Github Collaborators 추가하기



- ✓ Github Repository > Settings > Collaborators ০া হ
- ✓ Add people 클릭 후 협업자 email 등록(최소 4명 팀 구성)
- ₩실습 4-2. 파일 생성 및 커밋 후 푸시하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop
$ cd ~/Desktop/Workspace/Git/ch04

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ vi work6.txt 한 첫번째 줄에 'content by master' 입력
...

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git commit -am 'work6 commit by master'

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/Git/ch04 (master)
$ git push -u origin master
```

😀실습 4-3. 원격 저장소 복제 후 git_user1 브랜치 생성 및 푸시하기

```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop

$ git clone <github-repository-url> git_user1

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop

$ cd ./git_user1/ch04

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/git_user1/ch04 (master)

$ git checkout -b feature/git_user1
...

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/git_user1/ch04 (feature/git_user1)

$ vi work6.txt 	 두번째 줄에 'content by git_user1' 입력

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/git_user1/ch04 (feature/git_user1)

$ git commit -am 'work6 commit by git_user1'

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/git_user1/ch04 (feature/git_user1)

$ git push origin feature/git_user1
...
```

😀실습 4-4. 원격 저장소 복제 후 git_user2 브랜치 생성 및 푸시하기

```
$ git clone <github-repository-url> git_user2

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop
$ cd ./git_user2/ch04

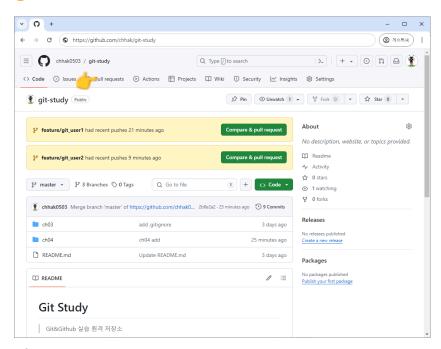
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/git_user2/ch04 (master)
$ git checkout -b feature/git_user2
...

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/git_user2/ch04 (feature/git_user2)
$ vi work6.txt → 두번째 줄에 'content by git_user2' 입력

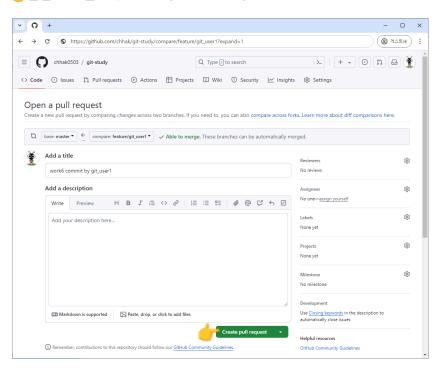
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/git_user2/ch04 (feature/git_user2)
$ git commit -am 'work6 commit by git_user2'

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/git_user2/ch04 (feature/git_user2)
$ git push origin feature/git_user2
...
```

알실습 **4-5**. **Github** 확인하기

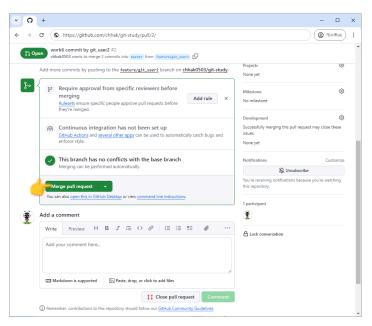


- ✔ 각 브랜치 사용자 별 Compare & pull request 클릭 또는 상단 Pull requests 클릭
- ₩실습 4-6. 풀리퀘스트 요청 메시지 작성하기

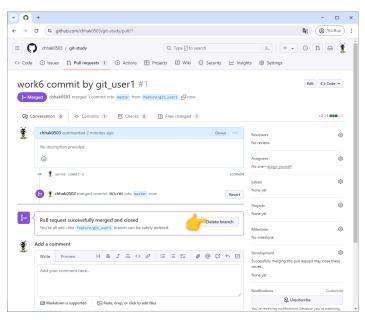


✓ title, description 입력 후 Create pull request 클릭

₩실습 **4-7**. 브랜치 충돌 여부 및 병합하기



- ✔ Merge pull request는 Github 담당자 또는 팀장이 수행하는 것이 일반적
- ✔ 브랜치 충돌이 발생하면 충돌 해결 후 Merge 수행
- ✓ 최종 Confirm merge 수행
- ₩실습 4-8. 깃허브 확인하기



- ✔ merge 수행 완료 후 해당 브랜치 삭제(권장)
- ✔ 병합된 파일 내용 확인하기

5. Github Actions

1) 스프링 프로젝트 생성 및 작업

₩실습 1-1. 스프링 프로젝트 생성하기

구분	항목	설명
프로젝트 정보	Name	spring-github-actions-app
	Location	~\Desktop\Workspace
	Туре	Gradle-Groovy
	Group	kr.chhak
의존성	Developer Tools	- Spring Boot DevTools - Lombok
	Web	Spring Web

₩실습 1-2. 스프링 프로젝트 작업하기

```
src > main > resources > application.properties
```

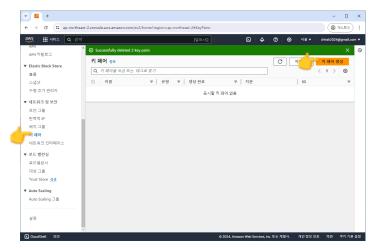
```
{\tt spring.application.name=spring-github-action-app} \\ {\tt app.version=0.0.1}
```

src > main > java > package > SpringGithubActionsAppApplication.java

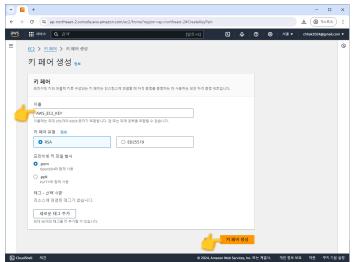
✔ http://localhost:8080 확인

2) AWS 비밀키 생성 및 설정

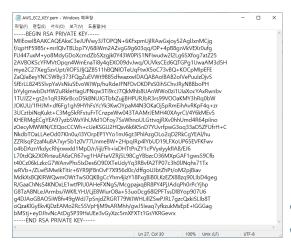
😀실습 2-1. 비밀키 생성하기



- ✔ 네트워크 및 보안 > 키 페어
- ✔ 키 페어 생성 클릭



- ✔ AWS_EC2_KEY 입력
- ✓ RSA 선택
- ✔ pem 선택
- ✔ pem 파일 다운로드



- ✔ pem 파일 메모장 열기
- ✔ pem 파일 내용 복사

₩실습 2-2. 공개키 생성하기

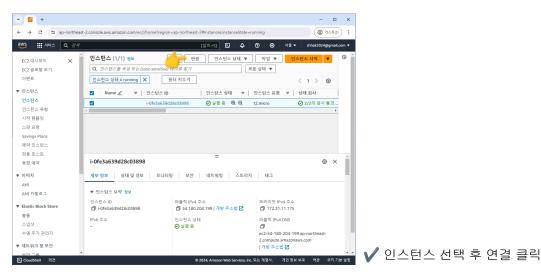
chhak@DESKTOP-J3DSA4P MINGW64 ~/Downloads \$ ssh-keygen -f AWS_EC2_KEY.pem -y

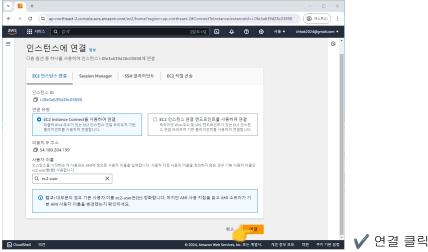
ssh-rsa

AAAAB3NzaC1yc2EAAAADAQABAAABAQCTELd41R9XDLc1M49A37ooXGmZSOVEDAbGOjLnYCCUnGcwK Ab2DrTeqr84/7ikHyCdWRURevS5+AVTjju7Ez7Kjwx3IYOhdyZ1ltJfOCOTRjjdbQ+JLU0UjC53Da XYuDpJd+Rc7T128nCr85RWSwIR3pBMY+DVTBoAzd3kebJ7YLbtd6DnO9Sm3+Ig4VImWNBkHnXWEFA 2U5N5SoW3BfmgLveRVa30Hzl1/BrPH

✔ 공개키 내용 복사하기

쓸실습 2-3. AWS EC2 공개키 설정하기





😀실습 2-4. 공개키 입력하기

[ec2-user@ip-172-31-37-192 \sim]\$ vi .ssh/authorized_keys

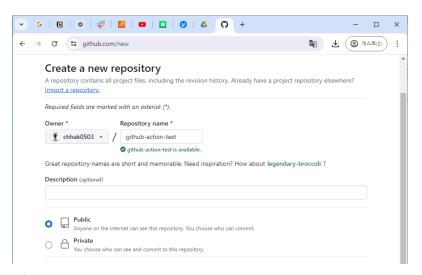
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCTELd4lR9XDLc1M49A37ooXGmZSOVEDAbGOjLnYCCUnGc wKAb2DrTeqr84/7ikHyCdWRURevS5+AVTjju7Ez7Kjwx3IYOhdyZ1ltJf0C0TRjjdbQ+JLU0UjC53DaXYuD pJd+Rc7T128nCr85RWSwIR3pBMY+DVTBoAzd3kebJ7YLbtd6Dn09Sm3+Ig4VImWNBkHnXWEFA2U5N5SoW3B fmgLveRVa30Hz11/BrPH

✔ 실습 2-2에서 복사한 공개키 내용 그대로 입력하기(붙여넣기)

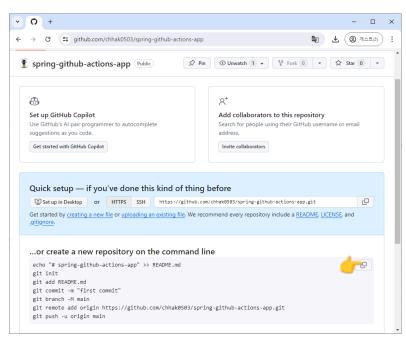
✔ ssh-rsa로 시작, 줄바꿈, 띄어쓰기 금지

3) 원격 저장소 생성 및 설정

₩실습 3-1. 원격 저장소 생성하기

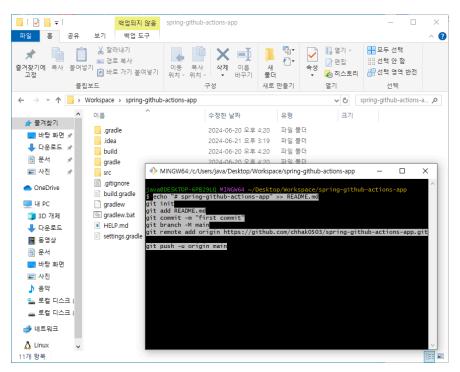


- ✔ 저장소 이름 : spring-github-actions-app 입력
- ✔ 저장소 공개 : public 선택
- ✔ 나머지 설정은 손대지 말것
- ✔ Create repository 버튼 클릭
- ₩실습 3-2. 원격 저장소 초기화 코드 복사하기

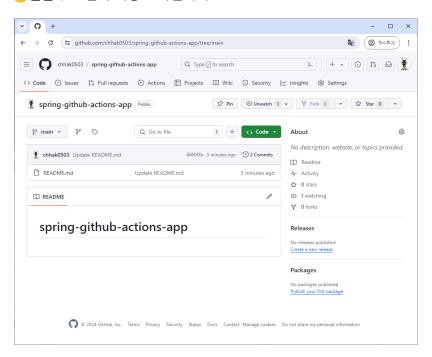


✓ ...or create a new repository on the command line 코드 복사하기

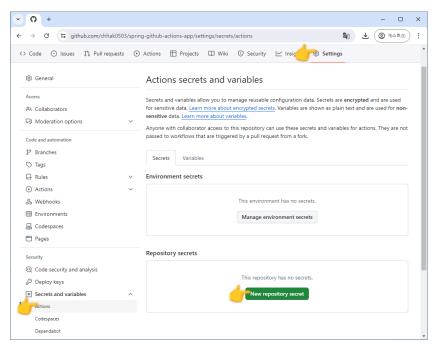
₩실습 **3-3**. 원격 저장소 프로젝트 설정하기



- ✓ 프로젝트 폴더 > 마우스 오른버튼 > Open Git Bash here 클릭
- ✔ 원격 저장소 초기화 코드 붙여넣기 후 Enter
- ✔ .git 확인
- ₩실습 3-4. 원격 저장소 확인하기



⇔실습 3-5. 저장소 설정하기



- ✔ 저장소 > Settings 클릭
- ✔ 사이드 메뉴 > Secrets and variables > Actions 클릭
- ✓ New repository secret 버튼 클릭

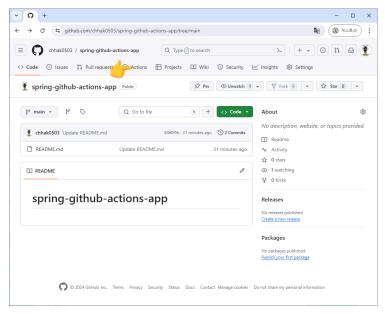
╝실습 3-6. Github Secrets 환경변수 입력하기

Name	Secret	설명
AWS_EC2_HOST	xxx.xxx.xxx	EC2 서버 아이피 주소 입력
AWS_EC2_USER	ec2-user	EC2 서버 기본 사용자 입력
AWS_EC2_KEY	EC2 secret 내용	실습 2-1 비밀키 내용 입력

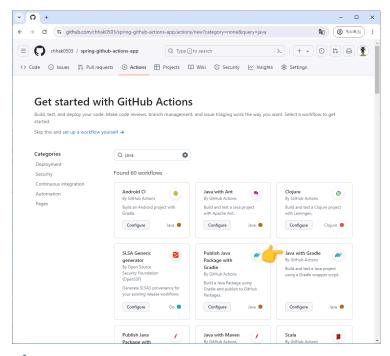


4) Github Actions 실습

≌실습 **4-1**. 워크 플로우 생성하기

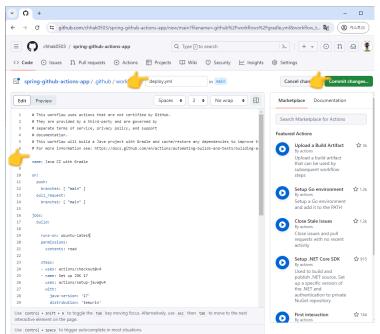


✓ Actions 메뉴 클릭



✓ 'java' 검색 후 'Java with Gradle' Configure 클릭

일실습 **4-2**. Workflow 작성하기



⊙ First interaction ☆ 744 ✔ 파일명 cicd.yml 입력

cicd.yml

https://github.com/chhak0503/git-study/blob/master/ch05/cicd.yml

- ✔ 기존 내용 삭제 후 위 주소의 스크립트 내용 복사&붙여넣기
- ✓ 내용 입력 후 Commit changes... 클릭
- 알실습 4-3. 프로젝트 pull & push 하기

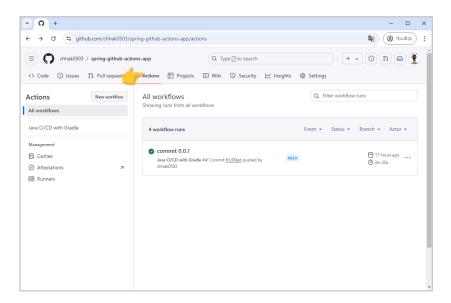
```
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/spring-github-actions-app (main)
$ git pull

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/spring-github-actions-app (main)
$ git add .

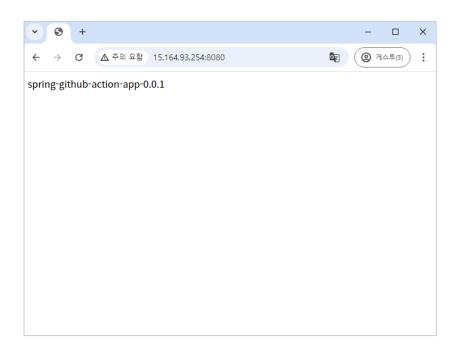
chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/spring-github-actions-app (main)
$ git commit -m 'commit 0.0.1'

chhak@DESKTOP-J3DSA4P MINGW64 ~/Desktop/Workspace/spring-github-actions-app (main)
$ git push
```

╩실습 **4-4**. Github Actions 확인하기



✔ Github > Actions 클릭, 진행 상태 확인



✔ Github Actions 진행 완료 후 브라우저 확인