Git_01_commit

Git 온라인 리소스

https://git-scm.com/docs/git-commit

https://docs.github.com/en/get-started/using-git

https://www.atlassian.com/git/tutorials

https://git-scm.com/docs

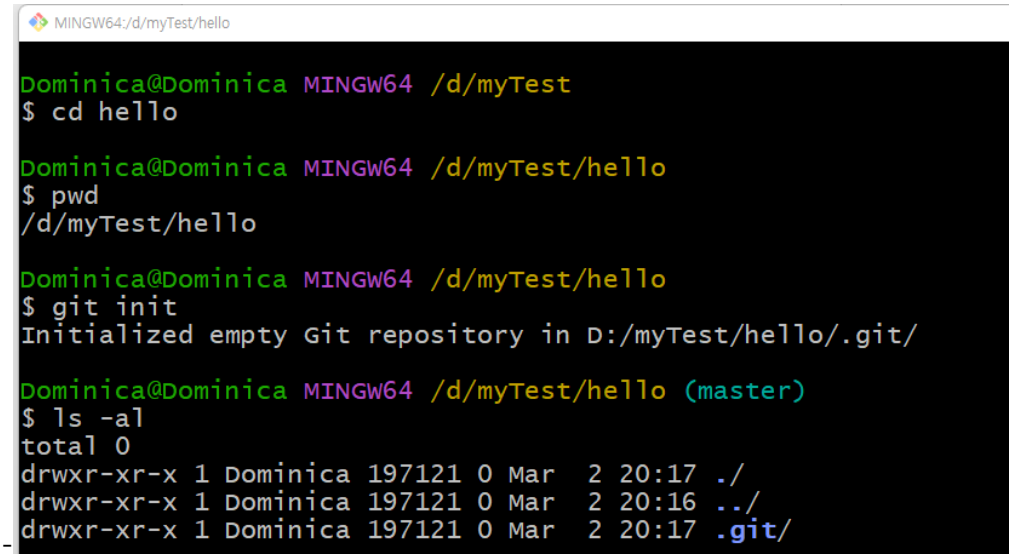
[Step 01] 해당운영체제 맞게 다운로드 한다.

[Step 02] 작업 폴더를 생성한다.

Ex) D:₩myTest₩hello

[Step 03] 생성된 작업 폴더에서 Git Bash 실행한다.

1. **현재 작업 디렉토리 확인 pwd**
2. **Git 저장소 초기화 (git init) 실행**
3. **숨김 파일 포함하여 디렉토리 내 파일 목록 확인 ls -al**



[Step 04] Git을 사용하여 파일 추가

---

1. **Hello.txt 파일 생성**
2. **cat 명령어를 사용하여 hello.txt 파일의 내용을 확인**
3. **Git 에 파일 추가 (git add)**
4. **hello.txt 를 Git 스테이징 영역에 추가한다**

*경고 메시지(LF will be replaced by CRLF)가 나타날 수 있으나, 이는 Windows 환경에서 Git이 자동 변환하는 것으로 문제없음."

[Step 05] 파일 확인, 추적 해제

1. 현재 상태 확인 (git status) : 파일이 스테이징 영역에 있는지 확인한다

```
Dominica@Dominica MINGW64 /d/myTest/hello (master)
$ git status
On branch master

No commits yet

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

파일을 Git 에서 추적 해제 (git rm —cached) :파일을 Git 의 스테이징 영역에서 제거하고, 로컬에는 남겨둔다.

```
Dominica@Dominica MINGW64 /d/myTest/hello (master)
$ git rm --cached hello.txt
rm 'hello.txt'

Dominica@Dominica MINGW64 /d/myTest/hello (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        hello.txt

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

2. 다시 Git 에 파일 추가 (git add) : hello.txt 를 다시 스테이징 영역에 추가한다.
3. 첫 번째 커밋 (git commit) : hello.txt 를 Git 에 커밋한다.

```
Dominica@Dominica MINGW64 /d/myTest/hello (master)
$ git add hello.txt
warning: in the working copy of 'hello.txt', LF will be replaced by CRLF the n
ext time Git touches it

Dominica@Dominica MINGW64 /d/myTest/hello (master)
$ git commit -m "[initialize] hello.txt"
[master (root-commit) 4a7a645] [initialize] hello.txt
 1 file changed, 1 insertion(+)
 create mode 100644 hello.txt
```

4. Git 상태 확인 (git status)

```
Dominica@Dominica MINGW64 /d/myTest/hello (master)
$ git status
On branch master
nothing to commit, working tree clean

Dominica@Dominica MINGW64 /d/myTest/hello (master)
$ pwd
/d/myTest/hello
```

- 현재 master 브랜치에 있음.
- "nothing to commit, working tree clean" →

  → Git 에 추가하거나 변경할 파일이 없음.

  → 즉, 현재 작업 디렉토리와 Git 저장소가 완전히 동기화된 상태.

5. 현재 커밋이 완료되었는지 확인한다.
6. 현재 디렉토리를 확인한다.

```
HELLO 폴더는 GIT 저장소로 초기화됨 (GIT INIT 실행됨).

모든 파일이 커밋되었고, 현재 GIT 저장소와 작업 디렉토리가 동일한 상태.

추가할 파일(GIT ADD)이나 커밋할 내용(GIT COMMIT)이 없음.

즉, 현재 GIT 저장소는 깔끔(CLEAN)한 상태로 유지되고 있음
```

[Step 06] 파일 수정 → 변경 사항 확인 → 커밋

1. **Vi hello.txt를 열어 "임의의 내용 한 줄의 텍스트를 추가해 본다 "**
2. **상태 확인을 한다. => 커밋 => 상태확인**

```
Dominica@Dominica MINGW64 /d/myTest/hello (master)
$ vi hello.txt

Dominica@Dominica MINGW64 /d/myTest/hello (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:   hello.txt

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

Dominica@Dominica MINGW64 /d/myTest/hello (master)
$ git commit -a  -m "[edit] hello.txt"
warning: in the working copy of 'hello.txt', LF will be replaced by CRLF the n
ext time Git touches it
[master 963c0f8] [edit] hello.txt
 1 file changed, 2 insertions(+)

Dominica@Dominica MINGW64 /d/myTest/hello (master)
$ git status
On branch master
nothing to commit, working tree clean
```

git status로 변경사항 확인 시

- 현재 MASTER 브랜치에 있음.
- HELLO.TXT 파일이 수정되었으나(GIT 이 추적하고 있지만), 아직 스테이징 영역(GIT INDEX)에 추가되지 않음.
- 다음 중 하나를 수행해야 함:
  - GIT ADD HELLO.TXT → 변경 사항을 스테이징 영역에 추가.
  - GIT COMMIT -A → 자동으로 수정된 파일을 스테이징하고 바로 커밋.

이후 git commit -a -m "[edit] hello.txt" → 커밋 수행 하게 되면 -a 옵션을 사용했기 때문에 변경된 파일(hello.txt)이 자동으로 스테이징 영역에 추가된 후 커밋됨.

커밋 후 git status를 실행하면 "working tree clean" 상태가 되어 변경할 내용이 없음을 확인

[Step 07 Log 확인]

```
Dominica@Dominica MINGW64 /d/myTest/hello (master)
$ git log
commit 963c0f8d8d9e1ba8ee865420fa0d1da406e53030 (HEAD -> master)
Author:
Date:   Sun Mar 2 20:23:53 2025 +0900

    [edit] hello.txt

commit 4a7a6451297cdfad9afc042d7c47b46fe1f746b9
Author:
Date:   Sun Mar 2 20:21:23 2025 +0900

    [initialize] hello.txt

Dominica@Dominica MINGW64 /d/myTest/hello (master)
$ git  log --oneline
963c0f8 (HEAD -> master) [edit] hello.txt
4a7a645 [initialize] hello.txt
```

Exam_첫 번째 커밋부터 커밋 로그까지 단계별 실습

1 단계: 작업 폴더 만들기 및 Git 저장소 초기화

Q1. D 드라이브에 git_test 폴더를 생성하고 이동하는 명령어를 작성하시오.
Q2. 현재 디렉토리에 Git 저장소를 생성하는 명령어를 작성하시오.

---

2 단계: 첫 번째 파일 생성 및 첫 번째 커밋

Q3. hello.txt 파일을 생성하고 "Hello Git!"이라는 내용을 추가하는 명령어를 작성하시오.
Q4. 현재 디렉토리에 있는 파일 목록을 확인하는 명령어를 작성하시오.
Q5. Git 저장소에 변경 사항이 있는지 확인하는 명령어를 작성하시오.
Q6. hello.txt 를 Git 에 추가하는 명령어를 작성하시오.
Q7. Git 에 추가된 파일이 스테이징 영역에 있는지 확인하는 명령어를 작성하시오.
Q8. "첫 번째 커밋"이라는 메시지와 함께 커밋하는 명령어를 작성하시오.

---

3 단계: 파일 수정 후 커밋

Q9. hello.txt 파일을 수정하고 "Git 을 배우는 중!"이라는 문장을 추가하는 명령어를 작성하시오.
Q10. 파일이 수정되었는지 확인하는 명령어를 작성하시오.
Q11. 수정된 파일을 Git 에 추가하지 않고 바로 커밋하는 명령어를 작성하시오.
Q12. 커밋이 완료되었는지 확인하는 명령어를 작성하시오.

---

4 단계: 커밋 로그 확인

Q13. Git 커밋 이력을 확인하는 명령어를 작성하시오.
Q14. 한 줄 요약된 커밋 로그를 확인하는 명령어를 작성하시오.