

실습_07주차_연습문제

1. 다음과 같은 디렉토리 구조일 때, 질문에 답하시오.

- 현재 my_folder 디렉토리에 있다.

```
my_folder
├── subfolder1
│   ├── file1.txt
│   ├── file2.txt
│   └── file3.txt
└── subfolder2
    ├── file4.txt
    ├── file5.txt
    └── subfolder3
        └── file6.txt
```

▼ 문제

1. subfolder1 에 file4.txt를 생성하세요.
2. subfolder1에 file2.txt를 subfolder2의 subfolder3으로 이동하세요
3. subfolder2의 모든 파일과 하위 폴더를 삭제하세요.

▼ 정답

- 디렉토리 구조 생성

```
mkdir -p my_folder/subfolder1 my_folder/subfolder2/subfolder3
```

```
touch my_folder/subfolder1/file1.txt my_folder/subfolder1/file2.txt
my_folder/subfolder1/file3.txt my_folder/subfolder2/file4.txt
my_folder/subfolder2/file5.txt my_folder/subfolder2/subfolder3/file6.txt
```

1. `touch subfolder1/file4.txt`
2. `mv subfolder1/file2.txt subfolder2/subfolder3`
3. `rm -rf subfolder2/*`

2. Git branch 관련 문제

▼ 문제

1. 새로운 로컬 브랜치 "feature"를 만드세요.
2. "feature" 브랜치에서 새로운 파일 "feature.txt"를 만드세요.
3. "feature" 브랜치에서 commit을 생성하고, 변경 사항을 원격 저장소로 푸시하세요.
4. 로컬 브랜치 "main"으로 이동하고, "feature" 브랜치를 "main" 브랜치에 병합하세요.
5. "feature" 브랜치를 삭제하세요.

▼ 정답

1. `git branch feature`
2. `git checkout feature && touch feature.txt`
3. `git add feature.txt`
`git commit -m 'Add feature.txt'`
`git push -u origin feature`
→ "u" 옵션은 로컬 저장소의 feature 브랜치와 원격 저장소의 feature 브랜치를 연결
→ 이후에는 "git push" 명령만 입력해도 feature 브랜치로 푸시 가능
4. `git checkout main`
`git merge feature`
5. `git branch -d feature`

3. Git rebase

▼ 문제

- 다음은 'main' 브랜치와 'develop' 브랜치의 커밋 히스토리입니다.

```
      H---I---J  main
      /
A---B---C---D---E---F---G  develop
```

‘main’ 브랜치를 ‘develop’ 브랜치의 최신 커밋으로 이동시키고, ‘main’ 브랜치의 커밋을 ‘develop’ 브랜치의 최신 커밋을 기준으로 재배치하려면 어떤 명령을 사용해야 할까요?

1. `git merge develop`
2. `git rebase develop`
3. `git checkout develop && git merge main`
4. `git checkout main && git rebase develop`

▼ 정답

4. 해설: `git checkout main && git rebase develop` main 브랜치로 이동해 develop 브랜치를 base 삼아 rebase 하겠다는 의미입니다.

명령을 실행하면 main 브랜치를 develop 브랜치의 최신 커밋으로 이동시키고, main 브랜치의 커밋을 develop 브랜치의 최신 커밋을 기준으로 재배치합니다. 따라서, 정답은 4번입니다.

4. Git 충돌

실습 과제 참고하기

5. Git 협업

실습_05주차_Git_(최종) 파일의 ‘Github으로 협업하기’ 참고