



4강 - 원격 저장소

원격 저장소(Remote Repository)란??

Github 가입하기

예제 1. 깃허브에 원격 저장소 만들기

예제 2. 원격저장소와 로컬저장소 버전관리 정보 업로드 하기 - push

실습 1. 로컬저장소에 새로 추가한 브랜치와 커밋들 원격저장소에 업로드하기

예제 3. 원격저장소에서 프로젝트 내려받기 - clone

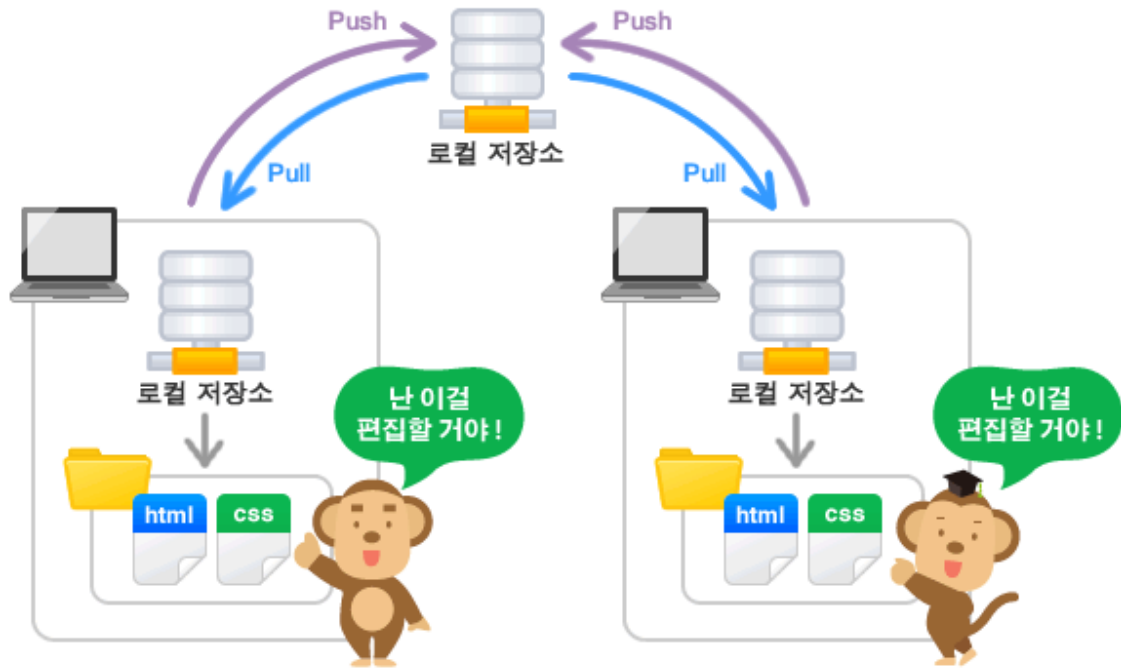
실습 2. 동일한 원격 저장소 다른 폴더에 클론하기

예제 4. 두 개 이상의 로컬저장소에서 하나의 원격저장소 관리하기

실습 3. academy 로컬저장소에서 버전관리 작업 후 my-home에 동기화하기

원격 저장소(Remote Repository)란??

- 원격저장소란 리모트 저장소라고도 불리며 로컬 저장소와는 달리 여러 사람 혹은 개인이 공유하면서 사용하기 위해 존재하는 원격 파일 서버 저장소입니다.
- 대표적인 원격 저장소 기능을 제공하는 서비스는 Github, GitLap, BitBucket 등이 있습니다.



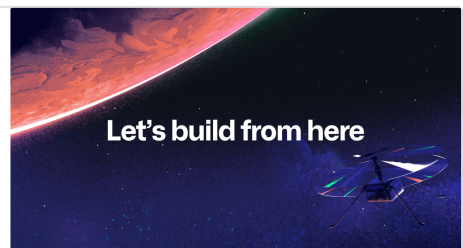
이미지 출처 : https://backlog.com/git-tutorial/kr/intro/intro1_2.html

Github 가입하기

GitHub: Let's build from here

GitHub is where over 100 million developers shape the future of software, together. Contribute to the open source community, manage your Git repositories, review code like a pro, track bugs

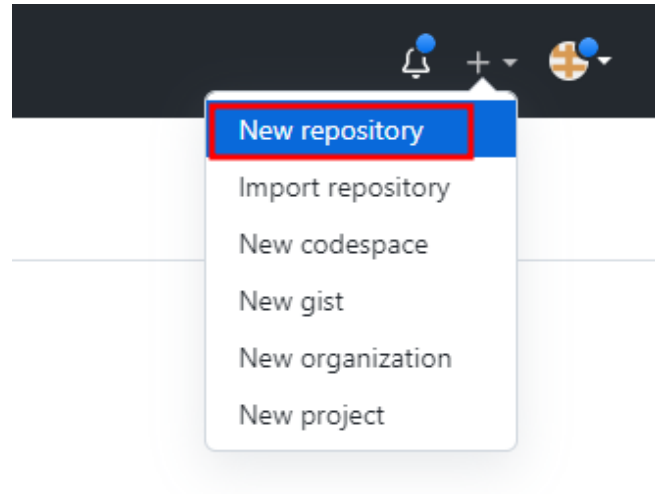
<https://github.com/>



- 원격 저장소 가입시 로컬 저장소에 등록한 이메일로 가입해주시는게 좋습니다

예제 1. 깃허브에 원격 저장소 만들기

- 깃허브에 로그인하면 우측 상단에 + 버튼을 누르면 'New repository' 메뉴가 보임



- 저장소 이름을 지어준 뒤 다른 옵션은 건드리지 말고 하단에 'Create repository' 클릭

Create a new repository

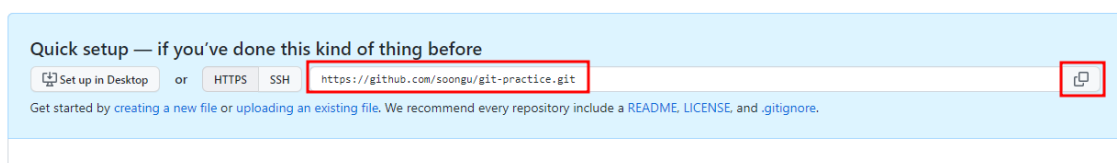
A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * soongu / Repository name * git-practice ✓

Great repository names git-practice is available. ple. Need inspiration? How about [fuzzy-computing-machine?](#)

Description (optional)

- 그러면 원격 저장소가 생성되며 해당 저장소의 주소가 보임.
- 해당 주소를 복사



예제 2. 원격저장소와 로컬저장소 버전관리 정보 업로드 하기 - push

- 우선은 로컬 저장소에 관리하던 프로젝트를 깃허브에 만든 원격 저장소와 연결합니다.
- 예제 1에서 복사해둔 원격저장소 주소를 아래 명령어와 함께 넣어주세요.

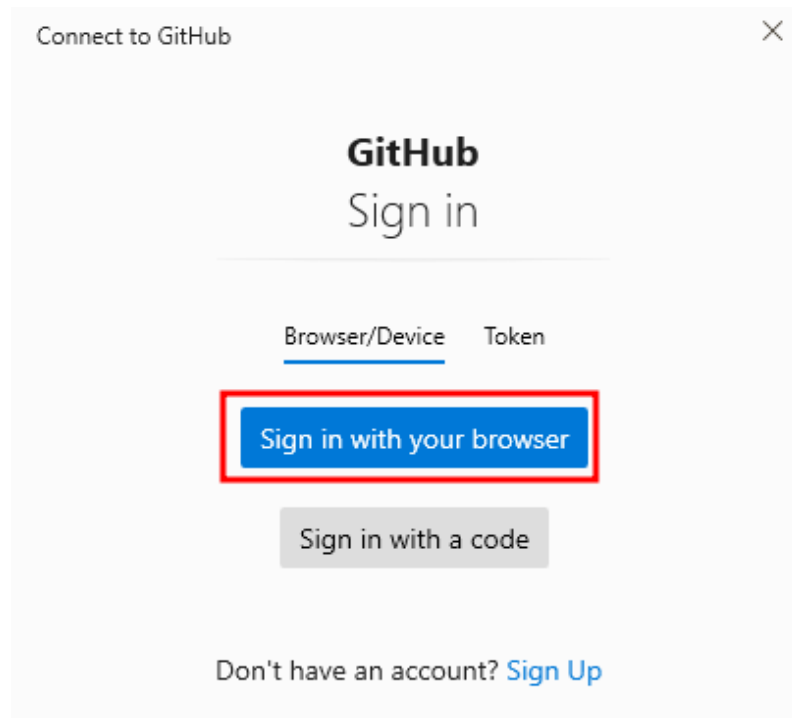
```
$ git remote add origin 원격저장소 주소 // 로컬저장소에 원격저장소 연결하기
$ git remote -v // 연결 정보 확인하기
```

- 이제 그동안 버전관리 했던 프로젝트를 깃허브에 업로드해봅시다.

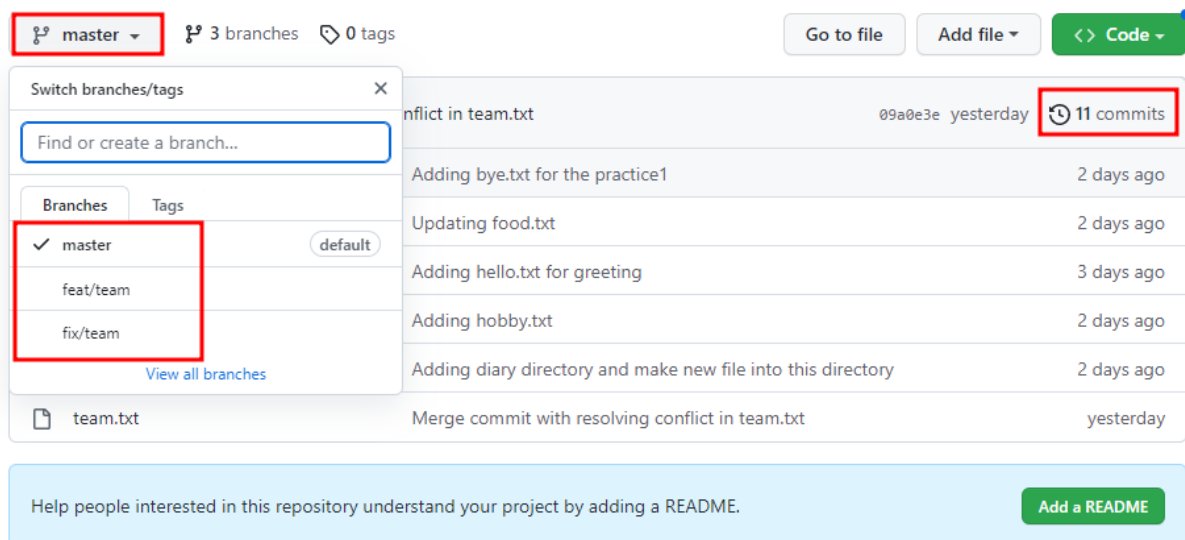
```
// 모든 브랜치와 태그 업로드
$ git push --all

// 참고 - 만약 특정 브랜치만 업로드하고 싶으면
$ git push origin 브랜치명
```

- 첫 push시에 깃허브 로그인창이 뜨면 로그인을 한번 해줍니다.



- 로컬 저장소와 모든 브랜치가 github에 업로드된 것을 볼 수 있습니다.



실습 1. 로컬저장소에 새로 추가한 브랜치와 커밋들 원격저장소에 업로드하기

1. 깃허브에 원격저장소 image-box를 생성하세요.
2. E드라이브에 새로운 폴더 my-favorite-image 폴더를 생성하세요.
3. my-favorite-image 폴더를 git 로컬저장소로 만들어 주세요.
4. 로컬저장소에서 브랜치 feat/image 를 master분기에서 생성하고 이동하세요.
5. 좋아하는 이미지 파일들을 다운로드 한 후 프로젝트 폴더에 포함하고 커밋하세요.
6. master에 병합하지 말고 feat/image 브랜치를 원격 저장소에 업로드하세요.

▼ 예시 답안

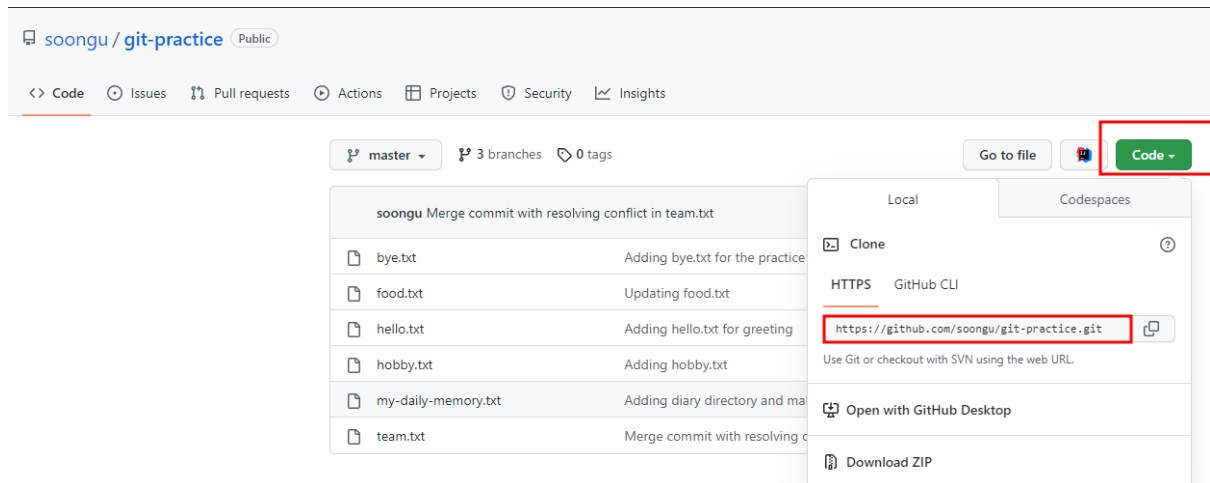
```
$ cd e:
$ mkdir my-favorite-image
$ cd my-favorite-image
$ git init
$ touch README.md
$ git add --all
$ git commit -m "initial commit" // 첫 커밋 이후 브랜치 생성 가능
$ git switch -c feat/image master
$ git add --all
$ git commit -m "message"
$ git remote add origin 원격저장소주소
$ git push origin feat/image
```

예제 3. 원격저장소에서 프로젝트 내려받기 - clone

- 우선 로컬에 저장된 프로젝트 폴더를 과감하게 지워주세요. (단, 원격저장소에 확실히 업로드 되었는지 꼭 확인!!)
- git bash에서 로컬에 내려받기 원하는 폴더로 이동해주세요. 예제에서는 D드라이브의 develop 폴더로 이동하겠습니다.

```
$ cd d:/develop
```

- 여기서 원격 저장소의 프로젝트를 내려받겠습니다. 깃허브에서 저장소로 간 후 code버튼을 눌러 URL 확인!



```
$ git clone 원격저장소URL
$ ls // 클론된 프로젝트 폴더 확인
$ cd 프로젝트폴더명
```

- 이전 예제에서 업로드한 폴더가 다시 다운로드 된 것을 확인할 수 있음.
- \$ git clone URL .** : URL 뒤에 도트(.) 를 붙이면 안에 있는 파일들만 복제함

실습 2. 동일한 원격 저장소 다른 폴더에 클론하기

- 이전 예제에서 클론한 폴더를 다시 삭제하세요.
- 원하는 위치에 my-home폴더와 academy폴더를 각각 생성하세요.
- 두 폴더에 각각 원격저장소의 프로젝트를 클론하세요.

▼ 예시 답안

```

$ cd d:/develop
$ rm -rf git-practice // 이전에 클론한 폴더 삭제
$ mkdir my-home academy // 폴더 2개 생성
$ cd my-home

// 마지막에 점을 붙이면 폴더를 통째로 내려받지않고 내부 파일들만 내려받음
$ git clone https://github.com/soongu/git-practice.git .

$ cd ../academy
$ git clone https://github.com/soongu/git-practice.git .

```

예제 4. 두 개 이상의 로컬저장소에서 하나의 원격저장소 관리하기

- 만약 여러분이 혼자 개발을 할 때 집에 있는 데스크탑과 노트북이 있다고 가정합니다.
- 집에서는 편하게 데스크탑으로 일정부분 개발을 진행했는데 갑자기 해외여행을 가게 되었습니다.
- 하지만 프로젝트 마감기한이 닥쳐서 해외여행중에도 개발을 진행해야 된다면 아마 노트북을 들고 가서 개발을 이어서 진행하겠죠?
- 그리고 해외여행을 마친 후 노트북에서 개발한 부분을 이어서 다시 집에 데스크탑에서 작업을 해야한다면 어떨까요?
- git과 github는 이 문제에 대한 해결방법을 제시합니다.

- 원격에서 사용하지 않는 브랜치인 feat/team과 fix/team 브랜치를 삭제하겠습니다.

```
$ git push origin --delete feat/team fix/team
```

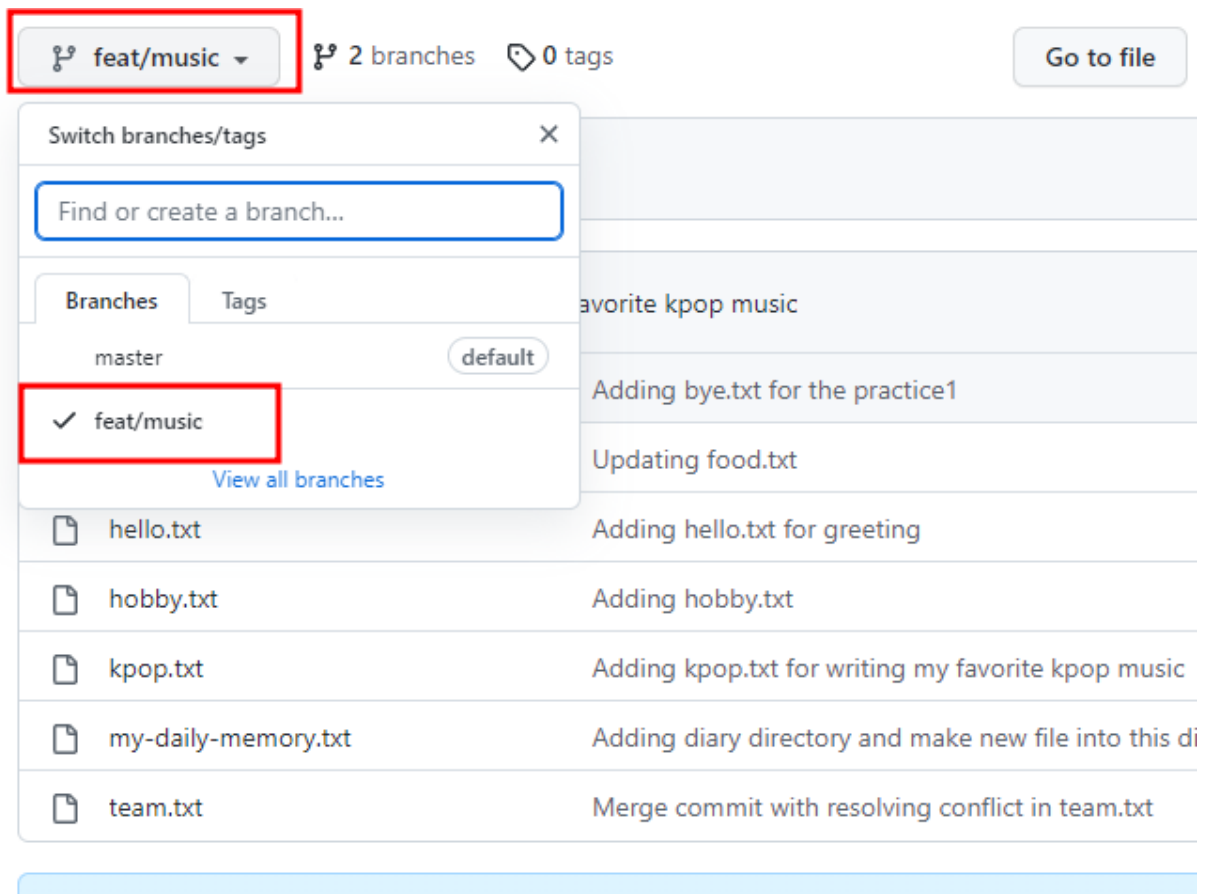
- my-home 디렉토리로 이동해서 브랜치를 생성하고 작업을 시작합니다.

```

$ cd d:/develop/my-home
$ git switch -c feat/music master
$ notepad kpop.txt
$ git add --all
$ git commit -m "message"
$ git push origin feat/music

```


- Github에 feat/music 브랜치가 업로드된 것을 확인할 수 있습니다.



- 자 그러면 이제 로컬 저장소 academy 디렉토리로 이동해서 git log를 보겠습니다.

```
$ cd ../academy
$ git log --all --oneline --graph
```

- 현재 academy 로컬저장소는 my-home 저장소와 원격을 공유하고 있지만 my-home에서 push한 이력이 아직 적용되지 않은 것을 볼 수 있습니다.
- 이 때 원격의 최신 변경내용을 받아오는 개념이 fetch와 pull입니다.
- fetch는 단순히 이력만 받아오는 것이고 pull은 실제 변경내용을 로컬에 적용하는 것입니다.
- 따라서 fetch로 이력을 먼저 받아 체크한 후 안전하게 pull을 하는 것이 좋습니다.

```

$ git fetch origin          // 원격 변경 이력 가져오기
$ git log --oneline --all --graph // 이력 변경 내용 확인
$ git switch -c feat/music master // 로컬에 땡겨올 브랜치를 생성
$ git pull origin feat/music // 원격에 업로드된 브랜치 다운로드

```

실습 3. academy 로컬저장소에서 버전관리 작업 후 my-home에 동기화하기

1. academy 디렉토리에서 fix/music 브랜치를 생성하고 체크아웃하세요.
2. kpop.txt를 수정한 후 커밋하고 원격에 fix/music이라는 브랜치에 푸시하세요.
3. my-home 디렉토리에서 변경사항을 동기화하세요. 단, my-home 디렉토리에서도 fix/music 브랜치가 생성되어야 합니다.
4. master 브랜치에 feat/music 브랜치를 병합하고 커밋, 푸시하세요.
5. master 브랜치에 fix/music 브랜치를 병합하고 커밋, 푸시하세요.
6. feat/music, fix/music브랜치를 로컬에서도 삭제하고 원격에서도 삭제하세요.
(git push origin --delete 브랜치이름)
7. academy 디렉토리에서 원격 origin/master에 병합된 내용을 동기화하세요.

▼ 예시 답안

```

// 1번 답안
$ git switch -c fix/music

// 2번 답안
$ notepad kpop.txt
$ git add --all
$ git commit -m "message"
$ git push origin fix/music

// 3번 답안
$ cd ../my-home
$ git fetch origin
$ git switch -c fix/music origin/fix/music // 원격 브랜치를 pull함과 동시에 로컬브랜치 생성효과
$ git log --oneline --all

// 4번 답안
$ git switch master
$ git merge feat/music
$ git push origin master

```

```
// 5번 답안
$ git merge fix/music
$ git push origin master

// 6번 답안
$ git branch -d feat/music fix/music          // 로컬에서 브랜치 삭제
$ git push origin --delete feat/music fix/music // 원격에서 브랜치 삭제

// 7번 답안
$ cd ../academy
$ git fetch origin
$ git pull origin master
```