

HOW to use git (git bash)

<https://ittrue.tistory.com/92>

Clone

복제한 리포지토리의 주소를 로컬 저장소에 복제해오는 작업

`http://계정/리포지토리이름.git`

```
$ git clone 리포지토리 주소
```

Init

내 pc에서 생성한 디렉토리를 Git에 등록하여 관리할 수 있도록 하는 명령어
기존 디렉토리를 로컬 저장소로 사용할 수 있도록 한다.

(깃허브의 원격 저장소와 연결시키거나 새로운 저장소를 초기화 하는데 사용된다.)

```
$ git init
```

Remote

`git remote add origin` 명령어는 `init` 명령어를 통해 로컬 저장소로 변환한 디렉토리 (폴더)를 Github에서 원격으로 관리할 수 있도록 연결하는 명령어이다.

Github에 디렉토리(폴더) 이름과 같은 원격 저장소를 생성하고, `git remote add` 명령어를 입력함으로 원격 저장소와 로컬 저장소가 연결이 완료된다.

```
$ git remote add origin <http://gildong/git-bash.git>
```

`git remote -v`를 통해 연결된 원격 저장소 확인가능

```
git remote -v
```

Status

git status는 로컬 리포지토리의 변경된 사항이 있는지 상태를 보여주는 명령어이다

```
$ git status
```

원격 저장소의 상태와 변경된 로컬 저장소의 상태를 비교하여 상태를 나타내는데, Staging area와 Untracked files 목록을 확인할 수 있다. untracked files은 git의 트래킹 되고 있지 않은 파일들을 뜻한다. 즉, 로컬 리포지토리 내에 존재는 하지만, git의 관리를 받지 못하고 있는 파일이다. modified에는 변경된 파일들을 나타내 준다.

Add

git 에 등록하여 commit을 할 수 잇는 상태로 만든다.

git add. 명령어를 입력하면, 모든 파일을 등록한다.

```
$ git add .  
$ git add filename
```

Restore

변경된 사항을 폐기

commit , staged 되지 않은 변경 사항을 폐기한다.

(git add를 통해 등록되지 않은 파일 폐기)

```
$ git restore filename
```

Commit

커밋이 가능한 파일들의 변경 사항을 저장하는 명령어.

즉 staging area 에 있는 변경된 파일들을 의미.

-m 옵션으로 커밋할 내용의 코멘트 작성가능

```
$ git commit -m '커밋 메시지'
```

Reset

git reset는 commit 한 기록을 취소하여 수정하고 싶을 때 사용하는 명령어이다.

즉, git commit 명령어를 사용한 이후 에러가 발생하였다면 commit 한 기록을 reset 명령어를 통해 취소할 수 있다.

아직 Remote Repository에 올라가지 않은 commit일 경우에만 reset 명령어로 취소할 수 있다.

```
$ git reset HEAD^  
$ git reset HEAD^^  
$ git reset HEAD3  
$ git reset HEAD~1
```

Push

local에서 변경, commit 된 사항을 원격 저장소에 업로드하는 명령어다.

git commit -m '커밋 메시지'명령어를 통해 커밋을 했다면 이를 등록 요청해야한다.

```
$ git push  
$ git push origin main(or master)  
$ git push origin branch
```

여러개의 브랜치를 둘 수 있으며 메인브랜치는 가장 중심.

Pull

원격 레포지토리의 작업 내용을 가져오면서 병합 작업을 실행

git pull 명령어는 git fetch와 git merge를 실행한 결과를 나타낼 수 있다.

즉 이는 Local 에 Remote내용을 덮어 씌운다고 할 수 있다.

```
$ git pull
$ git pull origin main(or master)
$ git pull origin branch
```

협업

만약, 협업하는 동료의 이름이 gildong이라 가정하고, 동료의 작업 내용을 자신의 로컬 레포지토리에 가져오면서 병합을 하고자 한다면 다음과 같이 사용할 수 있다. 만약, 동료의 main 브랜치가 아닌 다른 브랜치에서 가져오고자 한다면, main 대신

```
$ git pull gildong main(or master or branch)
```

git pull 명령어의 주의할 점은 git fetch와 git merge 동작을 한 번에 수행하기 때문에, 여러 사람들과 협업을 진행할 시 충돌이 일어날 수 있다.

Fetch

git fetch명령어는 원격 저장소의 변경 사항을 가져오기만 한다.

<병합을 수행하지 않는다,>

Merge

git merge 명령어는 git fetch 명령어를 통해 원격 저장소의 변경 사항을 가져온 후 사용하여 로컬 저장소와 병합을 하기 위해 사용한다.

<일반적으로는 push, pull을 사용하지만 일부 병합 문제로 인해 push가 안되거나 안전하게 내용을 가져오려 할 때 fetch, merge를 사용한다.>

Log

git log 는 현재까지 commit 된 내역들을 터미널 창에 출력해주는 명령어다.

벗어나려면 q를 입력하면 된다.