

Git

코드를 관리하는 도구(버전을 통해)

- 코드 관리도구
- 코드 협업도구
- 코드 배포도구

Git 명령어

git [명령어]

(1)git init

Git 으로 코드 관리를 시작

- 코드 관리 단위(기준): 폴더
- (master) : 현재 브랜치명
- .git 폴더 생성 : Git이 관리와 관련된 파일

(2)git status

현재 상태를 출력(Git에게 현재 상태를 물어봄)

- git init 직후,

```
On branch master
-> master 라는 브랜치 위에 있어요.

No commits yet
-> 아직 commit이 없어요.

nothing to commit (create/copy files and use "git add" to track)
-> commit 할 것이 없어요.
```

- test.txt 파일 생성 후,

```
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    test.txt
-> 추적되지 않은 파일이 있어요.
    (파일명)

nothing added to commit but untracked files present (use "git add" to track)
-> commit 하기 위해 add된 것이 없어요. 그러나 추적되지 않은 파일은 있어요.
```

- `git add test.txt` 직후,

```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   test.txt
-> commit 할 변경들이 있어요.
    (파일명)
```

- `git commit` 이후

```
nothing to commit, working tree clean
-> commit 할 게 없어요. 작업 폴더는 깔끔해요.
```

- 파일 수정 후,

```
On branch master
Changes not staged for commit:
-> commit 하기 위해 stage 되지 않은 변경 사항이 있어요.
    (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
        modified:   test.txt

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

(3) `git add` [파일명]

commit을 위한 stage

- `git add .` : 현재 폴더의 모든 변경 사항 stage
- `git add *.txt` : 현재 폴더의 .txt 확장자 스테이징
- `git add src/` : 디렉토리의 모든 파일 스테이징
- `for문 응용`

```
myList=("a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k")

for item in "${myList[@]}; do
    echo "Add $item.txt" > "$item.txt"
done
```

(4) `git commit -m "커밋 메시지"`

commit == 버전을 생성 == 현재 상태의 스냅샷 촬영

- 처음으로 commit을 할 경우,

Author identity unknown

-> 작자미상

*** please tell me who you are.

-> 당신이 누군지 알려주세요.

Run

-> 아래의 명령어를 실행해주세요.

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

- for문 응용

```
for item in "${myList[@]}; do
    git add "$item.txt"
    git commit -m "Add $item.txt"
done
```

- git config 설정 후(vim 에디터 창), git config --global user.name "Your Name"

Please enter the commit message for your changes.

-> 변경사항에 대한 commit message를 입력해주세요.

Lines starting with '#' will be ignored, and an empty message aborts the commit.

-> #로 시작하는 라인은 무시됩니다. 아무것도 없는 메시지는 종료됩니다.

On branch master

#

Initial commit

#

Changes to be committed:

new file: test.txt

(5) git config

Git에 관한 설정

- git config --global user.email "이메일" : global(전역으로) user의 email을 설정
- git config --global user.name : 설정값 확인

(6) git log

현재까지의 commit을 출력

- `git log` 출력화면

```
commit 159f15d7738a925555ed9d21ae7625acb848fcb1 (HEAD -> master)
Author: unknown <foodgoon@naver.com>
-> 작성자
Date: Thu Jul 13 17:07:37 2023 +0900
-> 날짜

    first commit
-> 커밋 메시지
```

- `git log --oneline` : 한줄로 출력

```
159f15d (HEAD -> master) A1
41a742e Add test.txt file
```

(7) git remote

- `git remote add` [저장소 이름] [저장소 주소] : `git remote add origin https://github.com/hkeryfonttlxisdlw/basic_git`
 - git에게 원격저장소(remote) 추가(add)를 명령
- 저장소이름 : `origin`
- 저장소주소 : https://github.com/hkeryfonttlxisdlw/basic_git
- `git remote -v` 란?

```
origin https://github.com/seonjaechoi0307/intro.git (fetch)
-> 로컬 저장소에서 해당 파일을 갖고와서
origin https://github.com/seonjaechoi0307/intro.git (push)
-> 깃 허브로 해당 파일 푸쉬
```

(8) git push

- `git push origin master`

`git push` 원격저장소명 브랜치명

- 업로드 시간은 commit 을 찍은 시간이 달려온다.

(9) git clone

- `git clone url` 폴더명

```
git clone https://github.com/choibest/pratice collab
-> 해당 url 주소의 코드를 collab 디렉토리를 만들어서 저장해라
```

(10) git pull

- git pull origin master

```
git pull 원격저장소명 브랜치명
```

(11) git 초대 및 수락

- 초대 방법
 - 1. Repositories 들어가기
 - 2. 공유할 Repositories 들어가기
 - 3. 상단에 Settings 들어가기
 - 4. 좌측에 Collaborators 들어가기
 - 5. Manage access 에 협업할 계정 입력
 - 1. ex) id = choibest
- 수락 방법
 - 우측 상단의 편지함 들어가기
 - Accpet 눌러서 동의하기

(12) .git 삭제

- `rm -rf .git`
 - .git 삭제