

CLI (Command Line Interface)

CLI

- interface : inter(사이) + face(접면)
- command line : 줄로 입력하는 명령어
- CLI : 줄로 입력하는 명령어 인터페이스

Git

코드를 관리하는 도구(버전을 통해)

- 코드 관리도구
- 코드 협업도구
- 코드 배포도구

SCM & VCS

- SCM(Sourcr Code Managment) : (소스) 코드 관리 도구
- VCS(Version Cotrol System) : 버전 관리 도구
- DVCS(Distributed Version Control System) : 분산 버전 관리 도구

Git 명령어

프로젝트(폴더) 기반 코드 관리 도구

특정한 프로젝트를 깃으로 관리할 수 있게 함.

`git` [명령어]

(1) `git init`

Git으로 특정 폴더(프로젝트) 관리 시작

```
& git init
Initialized empty Git repository in C:/Users/bean/intro/.git
```

개인 사용자 안에 intro 폴더 안의 git이 숨겨진 파일로 생성됨.

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

git status

Git의 현재 상태 출력

- `git init` 직후

```
$ git status
On branch master
-> master 라는 branch에 있음

No commits yet
-> 아직 commit 없음

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

- `touch a.txt` 파일 추가 후, `git status`

```
$ git status
On branch master -> master 라는 branch에 있음

No commits yet -> 아직 commit 없음

_____ 동일

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  -> commit 하려면 add 해야함, 파일 추적이 안됨

nothing added to commit but untracked files present (use "git add" to track)
-> commit 하기 위해 add 된 것은 없으나 추적되지 않은 파일은 존재함
```

- `git add a.txt` 직후, `git status`

```
$ git status
On branch master

No commits yet

Changes to be committed: -> commit 될 변경사항이 있음
  (use "git rm --cached <file>..." to unstage) -> 초기버전의 a.txt로 돌아가는 코드
    new file:   a.txt
```

- `git commit` 이후

```
nothing to commit, working tree clean
-> commit 할 게 없이 작업 폴더가 깔끔함
```

- 파일 수정 후

On branch master

Changes not staged for commit:

-> commit하기 위해 stage 되지 않은 변경 사항이 있음

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: a.txt

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

(3) git add [파일/폴더]

스냅샷을 찍기 위해 stage (사진대)에 올려둔 상태

- `git add .` : 현재 폴더의 모든 변경 사항 stage

(4) git rm --cached [파일/폴더]

스냅샷을 찍기 위한 모든 준비가 완료됨을 확인함

(5) git commit -m "커밋 메세지"

Git 버전 현재 상태에 대한 스냅샷 실행 `-m` 부분은 사진을 찍는다는 것을 알려려는 메세지

```
$ git commit -m "First commit"
```

- 최초 commit 실행시

Author identity unknown

-> 작자 미상

*** Please tell me who you are.

-> 실행자가 누군지 모르는 상태

Run

-> 아래 명령어를 실행해라

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

```
git config --global user.name "Your Name"
```

- `git config` 설정 후 (`vim` 에디터 창)

```
# Please enter the commit message for your changes.
-> 변경사항에 대한 commit message를 입력해라
# Lines starting with '#' will be ignored, and an empty message aborts the
commit.
-> #로 시작하는 라인은 무시하며, 아무것도 없는 메시지는 종료됨
# On branch master
#
# Initial commit
#
# Changes to be committed:
# new file: a.txt
```

(6) git config

Git에 관한 설정

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

(7) git log

현재까지의 commit을 출력함

- `git log` 출력화면

```
commit 1a7d9384d2f9a064e2ddb4719306defeb51ac3cf (HEAD -> master)
Author: John Kang <hphk.john@gmail.com>
-> 작성자
Date: Tue Mar 16 15:55:10 2021 +0900
-> 날짜

    first commit
-> 커밋 메시지
```

- `git log --oneline` : 한줄로 간결하게 출력

```
7c7abf0 (HEAD -> master) Add title
1a7d938 first commit
```

(8) git remote

- `git remote add [저장소 이름] [저장소 주소]`
: `git remote add origin https://github.com/hkeryfonttlxisdlw/basic_git`
- 저장소 이름 : `origin`

- 저장소 주소 : https://github.com/hkeryfonttlxisdlw/basic_git

(9) `rm -r .git`

master 상태에서 일반 계정으로 상태 변경