Terminal, Git command

터미널이란?(Terminal)

- 터미널이란 컴퓨터와 사용자간의 서로 소통시켜주는 인터페이스(번역기와 같은느낌)
- 터미널은 shell을 기본적으로 사용하고 있다.
  - shell은 문자기반의 명령어들을 컴퓨터 언어로 변환하여 컴퓨터와 소통을 가능하게 해준다.
  - shell은 기본적으로 bash,tcsh등이 사용되며 window에서는 cmd를 기반으로 사용되고있다.

GIT, GIT 명령어

Distributed Version Control System 그래서 (VCS라고 줄여서 씀.)

분류	명령어	설명
새 GIT 생성	git init	.git 하위 디렉토리 생성
	git add .	init 된 디렉토리의 모든 변경사항 저장. add는 stage에 올리는 작업.
추가 및 확정 (add/commit)	git add -A	commit에 파일의 변경사항을 한번에 모두 포함.
	git commit -m 'message to recognize the modification'	commit 생성. 변경사항을 확정한다. Stage에 올라온 사진을 찍어서 hash로 박제.
	git status	파일 상태 확인. 현재 어떤 상황인지 찍먹.
	git commit -a -m ‘message’	Staging과 commit을 한번에 처리. -a는 싹 add해서 commit 한다는 뜻

분류	명령어	설명
가지치기(branch)	git branch	브랜치 목록
	git granch <브랜치 이름>	새로운 branch 생성 (local로 만듬)
	git checkout -b <브랜치 이름>	
	git checkout -d <브랜치 이름>	
	git checkout master	
	git push origin <브랜치 이름>	만든 브랜치를 github에 전송.
갱신 및 병합 (merge)	git pull	
	git merge <다른 브랜치 이름>	
	git add <파일명>	
	git diff <브랜치 이름> <다른 브랜치 이름>	
태그작업	git log	현재 위치한 브랜치 commit내역 확인, hash값으로 식별자(id) 부여
	git log —oneline	각 commit 내역을 1줄씩으로 간단하게.
	git fetch origin	github에 저장된 git 프로젝트들의 현 상태를 다운로드
복제(clone)	git clone <https://……>	해당 github에 올라와 있는 repository 복제해 오기.

▼ 아 난 모르겠다 일단 당장 시작이나 하고 싶어!

1. 파일 “수정”
2. 파일 “저장”
3. git “애드”
4. git “커밋”

5. github에 “푸쉬”

- Commit이란?
  - 이것들을 (다 넣을지 일부만 넣을지는 뒤에 명세) local VCS에 기록해줘!

- push란?
  - Local VCS에서 만든 변화들을 remote VCS로 넣어줘! (remote가 원격이라는 뜻이니까)

- pull
  - remote VCS로부터 변화들을 받아줘!
  - 대부분 update가 수반된다. → 이 경우, Local repository를 update해서 변경사항 follow up
- git push origin master?? 이게 뭐야?
  - git push는 알잖아.
  - origin?
    - remote VCS에 올려져있는 project의 기본 닉네임이다.
  - master?
    - push 하고 있는 branch를 뜻함. 기본이 되는 main branch
    -
  - 그러니까 내가 master branch”를” origin이라는 닉네임의 remote VCS에 push한다는 뜻임.
    - 다른 예시로 mypojects라는 remote VCS에 push하고 싶으면 다음과 같은 명령어를 쓰면 됨.
    - git push myprojects master

▼ Clone 따오는 방법 의식의 흐름순서로 다시. origin이 뭐고 어떻게 바꾸는지.

1. Github 들어가서 remote repository만듬. (부동산 계약. 방을 팠음.)

2. URL주소 복사
3. TERMINAL 들어가서(가져오길 원하는 Directory에서 GIT BASH켜거나 경로 찾아서 들어감.), git clone https://~~~~
4. 이러면 자동으로 링크는 이어진거임.
5. 잘 되었는지 확인해보자. git remote -v(-v명령어는 자세하게 보여달라는 뜻이다.)
6. origin id@emil.com : ~~~~~~~~~~~~~~~~~~(fetch)
   origin id@emil.com : ~~~~~~~~~~~~~~~~~~(push) 이렇게 뜸
7. 왜 origin이라는 이름을 제 맘대로 쓸까?
   a. 내가 CESSNAJ_TIL이라고 이름 지어서 둔거는 Github server에 들어가있는 repository 이름이다.
   b. origin은 loacl환경에서 부르는 Default short name이다.
   c. 다 origin이라고 하면 헷갈리니까 다른 사람/프로젝트를 위해 다른 github의 repository를 받아올때는 이 명령어를 사용해서 origin 대신 다른 이름을 써주자!
      remote add project_of_friend <URL주소>
   d. 이렇게 되면 origin 대신 내가 설정한 이름으로 local에 clone될것이다.
      remote -v를 했을때
      project_of_friend git@github.com:CessnaJ/TIL.git (fetch)
      project_of_friend git@github.com:CessnaJ/TIL.git (push)

▼ Head, branch, Tag가 뭔지?

- Head
  - stage는 commit을 준비하는 공간. (add를 통해 올릴 수 있다.)
  - master는 가장 기본이 되는 branch
  - head는 현재 branch. 즉, 현재 commit이 무엇인지 알려주는 포인터.
  - Head는 master를 가리키는 레퍼런스가 되는 포인터
  - master는 해당 커밋의 해쉬값을 가리킴.

  - 그러니까 파일 수정해서 새로 commit을 하면, head는 같은 master를 가리키니까 안변하는데, master값은 변함.

달라진 commit의 hash값을 가리키니까.

- tag
  - Head와는 또 다른 기능을 하는 pointer. tag.
  - git tag 명령 이용해서 만들 수 있음.
  - 버전 배포를 위해서 특정 지점 commit을 tag로 고정해둔것.
  - 다른 commit과 구분되는 책갈피를 끼워둔다고 생각.
- branch
  - 특정 단계의 commit을 기준으로 새로운 분기점을 만들 수 있다.
  - 동일한 commit hash를 가리키는 branch를 만든다.
  - checkout 명령어를 통해서 head가 가리키는 branch를 자유롭게 왔다갔다 할 수 있다.
    처음 만들었을때 가장 메인이 되는 기본 branch의 이름이 master임.

▼ 참조..

https://velog.io/@delilah/GitHub-Git-명령어-모음

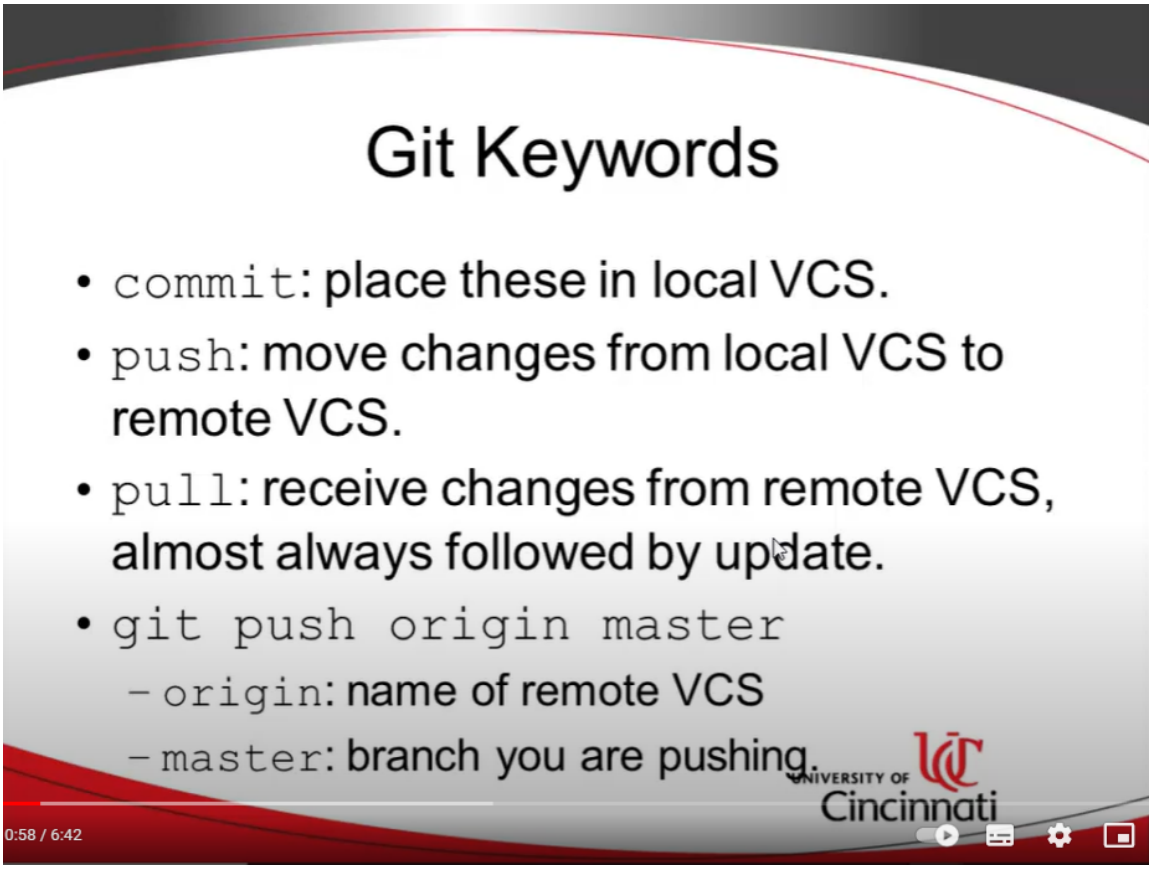
https://blog.naver.com/newyoung124/222620605205

https://www.youtube.com/watch?v=UGkT8w91qXQ (commit, push, pull, origin master)

https://www.youtube.com/watch?v=LIHIRBz5ZXk (what is origin?, how can i change origin)

https://www.youtube.com/watch?v=oQ4kT8KhRY8(Head, Branch, Tag)

https://blog.naver.com/gksthf4140/222487337759 git 관련 좋은 블로그


Git Keywords
• commit: place these in local VCS.
• push: move changes from local VCS to remote VCS.
• pull: receive changes from remote VCS, almost always followed by update.
• git push origin master
– origin: name of remote VCS
– master: branch you are pushing.
UNIVERSITY OF Cincinnati
0:58 / 6:42