

2022년 3월 23일 수요일

Git 총 정리

1. git 설치 for mac

1. Brew install git 으로 설치 (홈브루 설치했다고 가정)

2. github Repository 연결

git 작업 공간으로 설정할 폴더의 정확한 경로를 확인

ex: /users/in/documents/gitTest

terminal 에서 cd /users/in/documents/gitTest 입력해 로컬 저장소에 접근

github 닉네임 및 이메일을 이 컴퓨터와 연동시켜야 함 ->

1. git config --list 로 user.name과 user.email 확인

2. git config --global user.name "확인 값"

3. git config --global user.email "확인 값"

git 저장소 선언

1. cd 명령어로 접근한 로컬 저장소를 git init 명령어 실행하여 깃 repository로 설정

git add README.md

```
git commit -m "first commit"
```

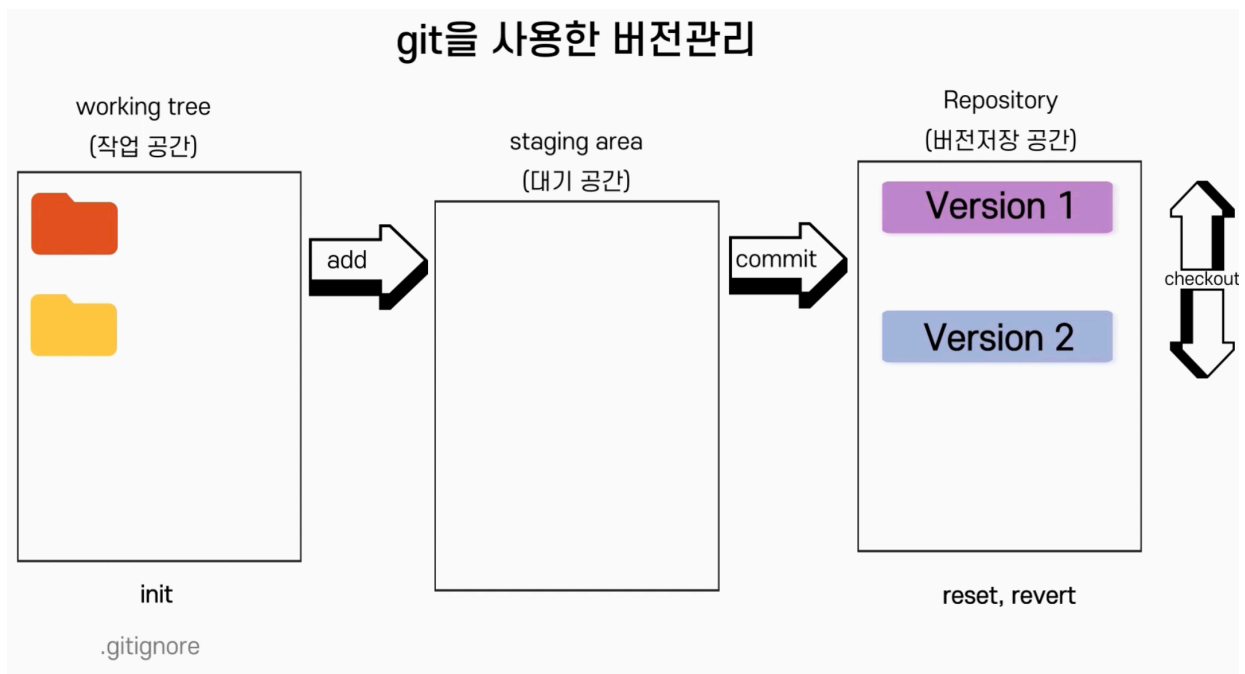
```
git branch -M main
```

```
git remote add origin https://github.com/사용자이름/레파지토리
```

주소. git

```
git push -u origin main
```

2. git 구조



git을 통해 우리가 작업하는 local directory와 리모트 저장소(github)를 연결시킬 수 있다.

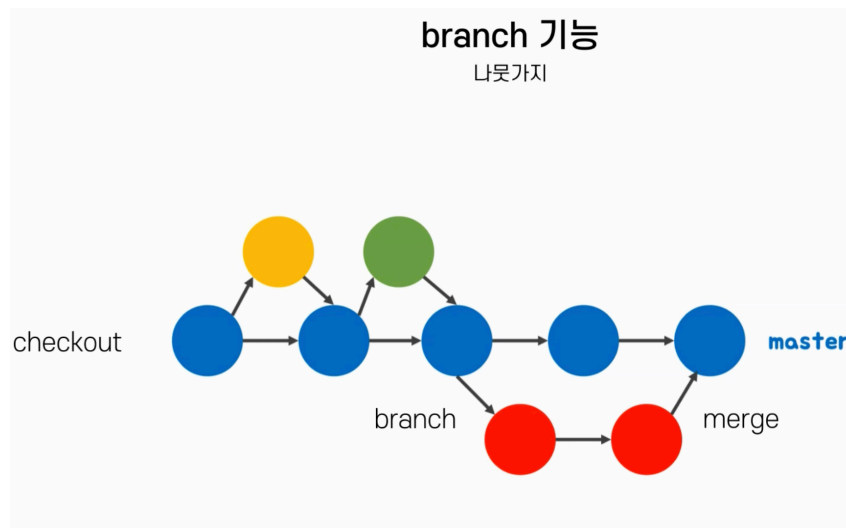
리모트 저장소란 쉽게 말하면 클라우드 서비스와 같다.

local directoy(=working tree)에서 작업한 내용을 리모트 저장소(클라우드)에 올리기 위해서는

add -> commit -> push 의 과정을 거친다(또는 merge). **git remote -v**를 통해 자신의 local directory와 연결된 리모트 저장소들의 목록을 확인할 수 있다.

사용자는 리모트 저장소를 여러개를 가질 수 있고, 리모트 저장소의 속성이 public이라면, 타인의 리모트 저장소를 내 local directory에 추가로 연결할 수 있다, 방법은 다음과 같다.

\$ git remote add 임의의명칭 <<https://github.com/사용자이름/레파지토리주소>>git >



3. git 명령어

git add '파일이름' :

로컬 저장소(= working directory, working tree)에 저장된 파일을 staging area(= cashe, index)로 전달 (인자로 . 를 전달하면 로컬 저장소 안의 모든 파일을 area로 전달)

git commit -옵션 "#버전이름 " : staging area의 수정된 파일들을 repo로 전달, 옵션을 통해 버전 관리 가능

commit 옵션들 :

-m : 별도의 메세지를 작성할 필요없이 인라인 형식으로 바로 커밋 메세지(=버전 이름) 작성.

-a : 별도의 add명령어를 사용하지 않고 수정된 파일에 대해 add, commit을 한번에 수행함. (단, 한번도 add되지 않은 파일은 add를 따로 작업 해주어야함.)

-am : a, m의 옵션을 합친 형태.

git log : commit 내역을 확인 가능하다

git remote add <리모트 저장소 이름> <<https://github.com/사용자이름/레파지토리주소>> : 리모트 저장소를 **working tree**에 연결한다 (데이터를 가져오는 것이 아님)

git checkout <Version> : 커밋 버전에 따른 고유 아이디를 입력하여 해당 버전에 접근할 수 있다.

git branch <브랜치이름> : 기동에 가지를 치는 것 처럼 세부 버전을 생성해 나갈 수 있는 branch를 생성한다, checkout을 통해 접근도 가능하다

git branch -d <브랜치이름> : 해당 브랜치 삭제

git fetch <리모트 저장소 이름> : 자신의 working tree와 연결된 리모트 저장소의 이름을 통해 해당 리모트 저장소에서 데이터를 모두 가져올 수 있다. 그러나 자동으로 merge 하지 않기에 로컬 작업을 마무리한 후 수동으로 Merge 해주어야 한다.

***merge 하고 싶은 상황일 때 :**

main(master) branch에서 합치고 싶은 branch에 대해 다음 명령어를 실행한다.

git checkout master (or main)

git merge <브랜치 이름>

git push <리모트 저장소 이름> <브랜치 이름> : 리모트 저장소에 변경사항(데이터) 전달

###

git clone 명령은 자동으로 로컬의 master 브랜치가 리모트 저장소의 master 브랜치를 추적하도록 한다 (물론 리모트 저장소에 master 브랜치가 있다는 가정에서). 그리고 **git pull** 명령은 Clone 한 서버에서 데이터를 가져오고 그 데이터를 자동으로 현재 작업하는 코드와 Merge 시킨다.(확실히 모름)

####