**GIT1 – 생활코딩 220402**

<https://opentutorials.org/module/3733>

**3대 목적**

1 version

2 backup

3 collaborate

**git 의 목적 1 -- version관리**

나의 folder 를 version 관리하겠다

– create a new repository : local path 지정

version 명 작성 + “commit to master” click

같은 Path 에 있는 파일은 version 명에 따라 관리

**git 의 목적 2 – 백업**

github.com 사용하여 가능

github.com > public repository 에 Name 작성

push to github.com / pull from github.com

**git 의 목적 3 – 협업**

pull from github.com 🡪 “fetch origrin”

작업 후 push to github.com

**git 의 종류**

TortoiseGit – windows 에서 사용

GithubDesktop –

**Sourcetree – 추천(생활코딩)**

**git original**

**git1 🡨 git2(versoin) 🡨 git3(backup) 🡨 git4(collaborate)**

**GIT2 -- version관리 – 생활코딩 220402**

**CLI – Command Line Interface 이용**

https://opentutorials.org/module/3762

**버전관리의 시작 / 버전의 생성 / 버전간의 차이점 비교**

gitbash 에서 실행

git .init // initialize repository : git 시작

git add hello1.txt // add to Staging ara : Staging area 에 올리기

git commit -m "Message 1" // create version : Repository 에 올리기

**git status // working tree status : git 상태**

**git log –stat // show version –stat**

**git log –p //**

**git diff // show changes**

**git reset –hard // working area 에서 전 상태로 돌리기**

git config --global user.email “[ggkimsh@gmail.com](mailto:ggkimsh@gmail.com)” // 나의 영역 지정하기

**checkout 과 시간 여행**

git checkout 622855db4ef38bb49c4f1d4fd70f04f0734646c7 // Message 2 상태로 돌아가기

git checkout master // 현재 상태로 돌아가기

**보충수업**

git add **.**  // add to Staging area : 현재 디렉토리 전체를 add to Staging area

git add **src**  // add to Staging area : src 디렉토리 전체를 add to Staging area

git commit **-am “4”** // add to Staging area & commit

git config --global core.editor "nano" // git 에서 사용하는 default editor 변경하기

touch 파일이름 // 파일 생성

rm 파일이름 // 파일 삭제

**버전 삭제**

git reset //

git reset --hard 622855db4ef38bb49c4f1d4fd70f04f0734646c7

// 버전까지 지우기. 지정한 상태를 최종으로 만들기

**버전 되둘리기**

git revert //

git revert d5332e082f5651e61895f469ec2771cd914a192b

반드시 역순으로 revert 진행해야 한다

**수업을 마치며**

**.gitignore folder 만들어서 버전관리하지 않을 파일을 둘 수 있다**

**branch 저장소에서 일부 내용만 수정 가능**

**tag 버전의 id를 줄 수 있다**

**backup**

**GIT3 -- CLI백업 – 생활코딩 220403**

**https://opentutorials.org/module/3801**

**Dropbox**

**Google Drive**

**One Drive**

**이용하라**

hosting : 인터넷에 연결되어 사용할 수 있는 서버 제공

git hosting : backup server 원격 저장 가능

**수업의 목표와 용어의 정리**

local repository –--(push) 🡪 remote repository

복제하기 : clone (remote repository 에서 local repository 로 가져오기)

local repository 🡨 (pull) --- remote repository

remote repository 마련하기

**Git hosting 선택**

GitHub

GitLab.com

**저장소 생성**

**공부의 방향**

http ssh 중에서 http 를 사용할 것임

**원격저장소와 연결**

git remote add origin <https://github.com/ggkimsh/my-repo.git> // github 연결, 별명:origin

git remote –v // 연결내용 보기

**git push**

git push --set-upstream origin master // 처음에만 한 번 사용

git push

**git clone**

원격저장소를 복제하여 지역저장소를 만드는 방법

git clone <https://github.com/ggkimsh/my-repo.git> // 원격저장소를 똑같이 복제

**git pull**

원격저장소의 버전을 지역저작소로 땡겨오는 방법

git pull //

**git 과 오픈소스**

오픈소스를 git를 통해 내 컴퓨터로 가져오는 방법

git site 에 들어가서 source code 를 선택하여 가져올 수 있다 //

**수업을 마치며**

**공부할 만한 주제**

SSH 저장소간의 통신 방법

issues 발생한 이슈에 대한 사항 follow up

협업 준비 완료됨

**GIT3 – CLI Branch & 충돌 – 생활코딩 220403**

**https://opentutorials.org/module/3927**

예시> 제품 사용설명서 git 로 관리

- 제품 출시 후 고객사 마다 차이가 있게 됨

**브랜치의 사용법**

git branch

git branch apple

git branch google

git branch ms

**git log --all --graph --oneline**

$ git log --all --graph --oneline

\* 8463960 (HEAD -> master) master work 4

\* 660f8af (ms, google, apple) work 3

\* 2135daf work 2

\* d5035f5 work 1

git checkout apple

$ git log --all --graph --oneline

\* 8463960 (master) master work 4

\* 660f8af (HEAD -> apple, ms, google) work 3

\* 2135daf work 2

\* d5035f5 work 1

**브랜치 병합**

서로 다른 브랜치의 버전들을 병합하는 방법을 살펴봅시다

**4-1 merge**

**base – merge 하려는 branch의 공통 조상**

**merge commit**

git init manual-merge

Initialized empty Git repository in D:/MyITSkill/Documents/git/git3-cli/manual-merge/.git/

git commit –amend // version 이름 변경

git checkout master

git merge o2

**3 way merge**

**외부 도구를 이용해서 병합하는 방법**

git mergetool p4Merge

**수업을 마치며**

git-flow

cherry-pick

rebase

**부록 : reset vs checkout**

**GIT4 – CLI협업 – 생활코딩 220403**

**https://opentutorials.org/module/3967**

hosting : 인터넷에 연결되어 사용할 수 있는 서버 제공

git hosting : backup server 원격 저장 가능

**git로 혼자 작업하기**

mkdir git4-collaboration

cd git4-collaboration

git init a

cd a

nano work.txt

git add work.txt

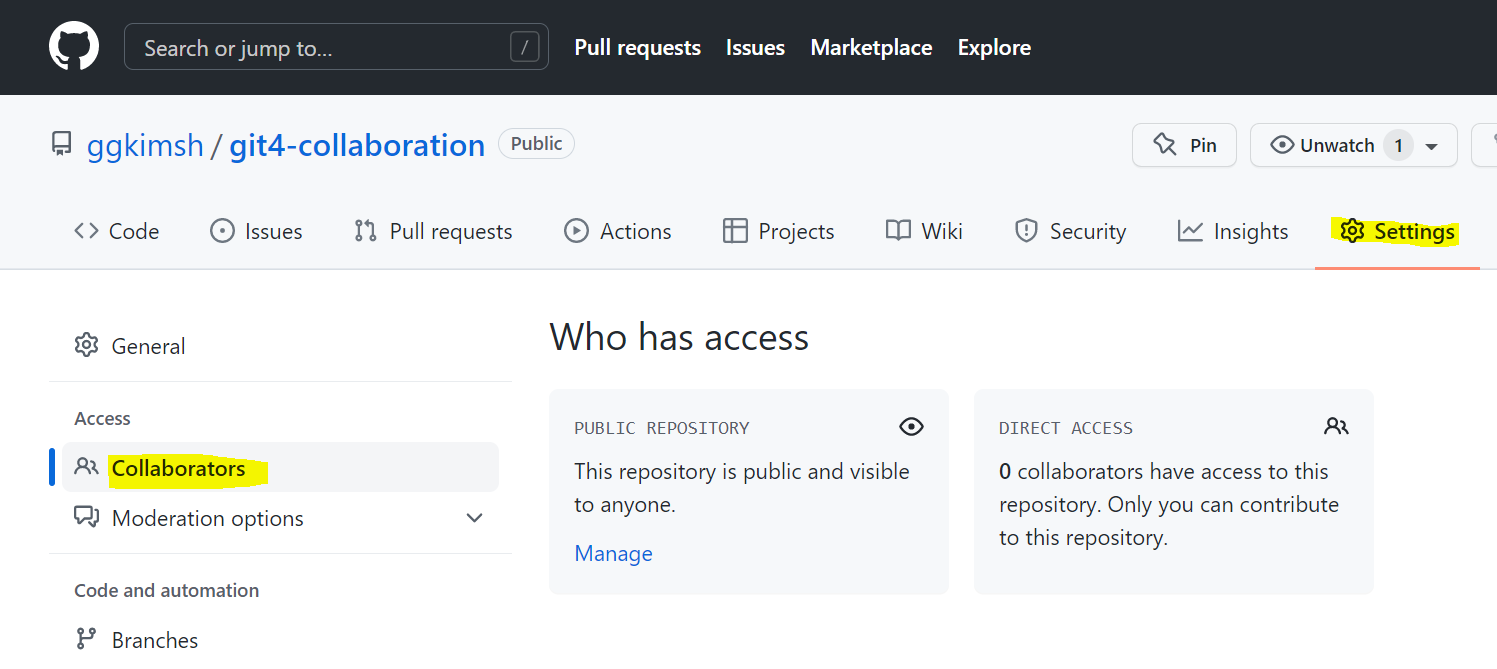
git commit -m "work 1"

(git 에 new repository 만들기, git4-collaboration.git)

git remote add origin https://github.com/ggkimsh/git4-collaboration.git

git push -u origin master

**git로 같이 작업하기**



git clone https://github.com/ggkimsh/git4-collaboration.git b

**git push & pull**

cd ../a

nano work.txt

git commit -am "work 2a"

git push

$ git log --oneline

ac09978 (HEAD -> master, origin/master) work 2a

a162e44 work 1

cd ../b

git pull

nano work.txt

git add work.txt

git status

git push

**git pull vs fetch 그리고 원격 브랜치**

git pull = git fetch + git merge FETCH\_HEAD

git fetch : remote branch 만 가져오는 방법

**수업을 마치며**

gerrit

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**vi 사용**

**https://blockdmask.tistory.com/25**

