

1. Git 설치하고 기본구성하기

1. <http://git-scm.com/downloads> 에서 OS 버전에 맞는 소프트웨어를 다운로드 하여 설치한다.
2. Git Bash 를 실행한다
3. \$git config --global user.name " your name"
4. \$git config --global user.email "your email"
5. \$git config --edit --global 를 통해서 변경하거나 추가할수도 있다.

2.작업디렉토리 생성하고 로컬 repo 생성하기

1. mkdir "skccdemo"
2. cd skccdemo
3. git init => /skccdemo/.git/ 생성
4. ls -al
5. git status => master 브랜치를 확인
6. vi readme.md -> visual studio 가 열리고 그냥 save 만 한다.
7. git status

```
yjkim@YJKIM MINGW64 ~/skccdemo (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        readme.md

nothing added to commit but untracked files present (use "git add" to track)
```

8. git add readme.md
9. git status

```
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   readme.md
```

10. 몇개의 파일을 추가로 더 생성해보자
11. vi new-file1.md
12. vi new-file2.md
13. vi new-file3.md
14. git add . => . 는 추가되지 되지 않은 모든 파일을 의미한다.

15. git status

```
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   new-file1.md
        new file:   new-file2.md
        new file:   new-file3.md
        new file:   readmo.md
```

16. git commit -am "new files added"

```
$ git commit -am "new files added"
[master (root-commit) 44c930b] new files added
4 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 new-file1.md
create mode 100644 new-file2.md
create mode 100644 new-file3.md
create mode 100644 readmo.md
```

17. git status

18. git log

3. GitHub 레포지토리 생성하기

1. <http://github.com>

계정이 없으신분은 계정을 생성하기

2. 레포지토리 생성하기

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *



youngjinkim817 ▾

Repository name *

/ skccdemo ✓

Great repository names are short and memorable. Need inspiration? How about **fluffy-system**?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

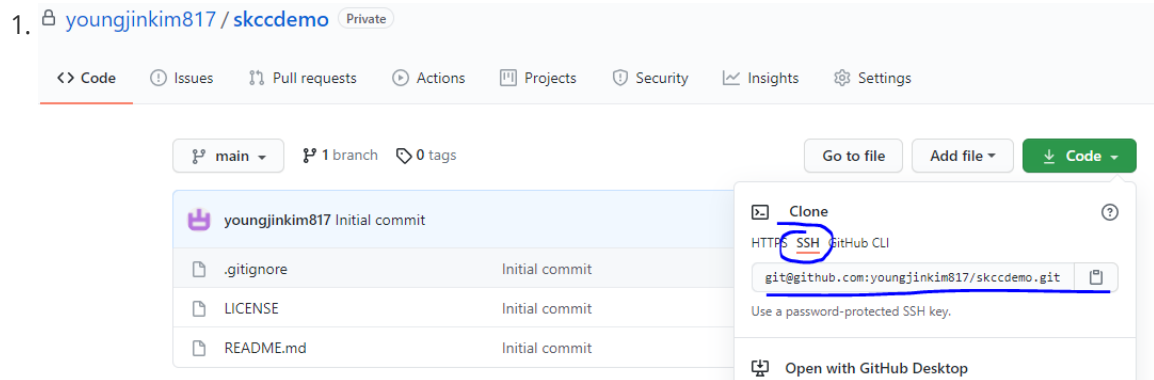
Create repository

3. Github 에 ssh 로 접속하기 위한 ssh key 생성하기

1. `ls -al ~/.ssh` : 이미 생성된 ssh key 가 있는지 확인
2. 있다면 `rm -rf ~/.ssh` 로 삭제
3. `ssh-keygen -t rsa -b 4096 -C "youngjinkim817@gmail.com"` => passphrase 는 입력하지 않아도 됩니다.
4. `eval $(ssh-agent -s)`
5. `ssh-add ~/.ssh/id_rsa`
6. `ls -al ~/.ssh` 로 확인해보기 (`id_rsa` : private key , `id_rsa.pub` : public key)
7. `cat ~/.ssh/id_rsa.pub` 에서 ssh 를 notepad 에 복사해둔다.
8. <http://github.com> 에 로그인해서 로그인계정 -> setting -> SSH and GPG key -> New ssh key
title : my desktop ssh
key : 복사해둔 키를 붙여넣기 한다. => add ssh key => confirm password
9. Git bash 에서
`ssh -T git@github.com` -> yes 입력

```
$ ssh -T git@github.com
The authenticity of host 'github.com (15.164.81.167)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWG17E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com,15.164.81.167' (RSA) to the list of known hosts.
Hi youngjinkim817! You've successfully authenticated, but GitHub does not provide shell access.
```

4. Github 와 함께 작업하기



2. git bash 에서

```
$pwd
```

```
$git clone git@github.com:youngjinkim817/skccdemo.git
```

```
$ls -al
```

```
$cd skccdemo
```

```
$ls -al
```

```
$git status
```

```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

origin : remote repo

main : branch 이름

3. 실습복사폴더에 있는 폴더와 파일을 skccdemo 레포아래에 복사해두기

4. \$git status

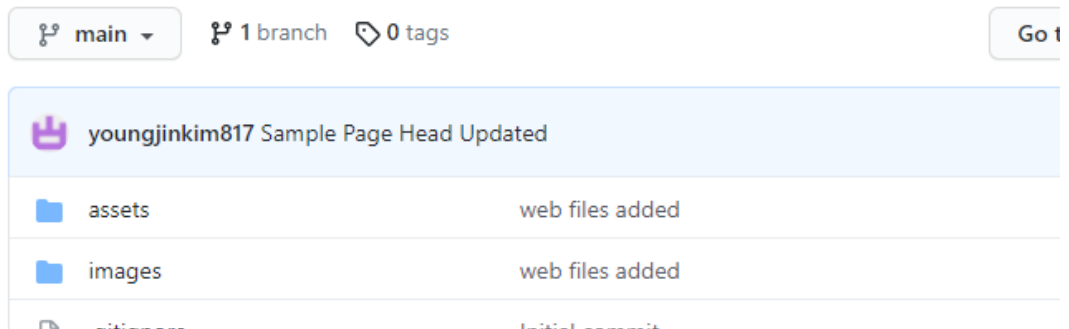
```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    assets/
    favicon.ico
    images/
    index.html

nothing added to commit but untracked files present (use "git add" to track)
```

5. git add .

6. git status
7. git commit -m "websile files added"
8. git status
9. git remote -v
10. git push origin main
11. github 페이지로 가서 skccdemo 레포지토리에 web 파일들이 추가 되었는지 확인해본다.
12. github 페이지에서 index.html 파일을 업데이트해본다.



13. git bash 로 와서

\$git status

\$git fetch : merging 하지 않는다

```
$ git fetch
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:youngjinkim817/skccdemo
   e1f70ae..703e4c3  main       -> origin/main

yjkim@YJKIM MINGW64 ~/skccdemo/skccdemo (main)
$ git status
On branch main
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)

nothing to commit, working tree clean

yjkim@YJKIM MINGW64 ~/skccdemo/skccdemo (main)
$ |
```

14. \$git pull

```
$ git pull
Updating e1f70ae..703e4c3
Fast-forward
 index.html | 4 ++--
 1 file changed, 2 insertions(+), 2 deletions(-)
```

15. cat index.html 를 통해 로컬 레포지토리가 update 된것을 확인
16. 로컬레포지토리에서 파일을 업데이트해본다.
17. vi index.html 를 통해 업데이트해본다.
18. git status 로 확인

```
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

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

19. \$git add .

\$git commit -am "local index.html updated"

\$git push origin main

```
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 296 bytes | 296.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:youngjinkim817/skccdemo.git
  703e4c3..1249322  main -> main
```

20. github 에 가서 레포지토리에 업데이트를 확인해본다.

main		1249322
youngjinkim817 local index.html updated		
assets	web files added	
images	web files added	
.gitignore	Initial commit	
LICENSE	Initial commit	
README.md	Initial commit	

21. 만약 로컬과 리모트 레포지토리에서 동시에 업데이트가 발생하면 어떻게 될까요?

21.1 로컬에서 update 해보기

\$vi index.html => title 을 sample site at the same time 으로 변경하고 저장한다.

\$git add .

\$git commit -am "change title at local "

21.2 리모트에서 update 해보기

<http://github.com> 에서 index.html 에서 title 을 sample title at remote 로 변경하고 commit 에 서 "title upate at remote " commit change 함

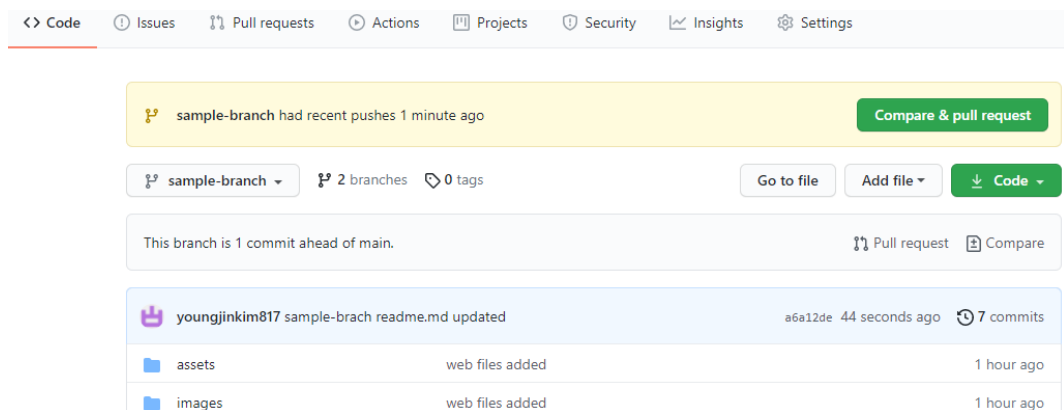
22. gitbash 에서 git push origin main

```
$ git push origin main
To github.com:youngjinkim817/skccdemo.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'git@github.com:youngjinkim817/skccdemo.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

23. 이럴땐 git pull를 먼저하고 나서 merge 를 해야 한다.

5. Branch 로 작업하기

1. github 에 default master 브랜치를 확인해본다 (ex. main)
2. sample-branch 를 생성해본다.
3. main 브랜치와 sample-branch 의 콘텐츠가 동일함을 확인한다.
4. sample-branch 에 있는 readme.md 를 업데이트해본다.
 1. sample-branch readme.md is updated 로 commit 해본다



5. main 과 sample-branch 의 readme.md 를 비교해본다.

6. gitbash 에서 브랜치생성해보기

```
$git checkout -b "local-sample-branch"
```

```
$git status
```

```
$vi local-sample-branch.txt 를 생성하고 local sample branch 라고 입력하고 저장한다.
```

```
$git status
```

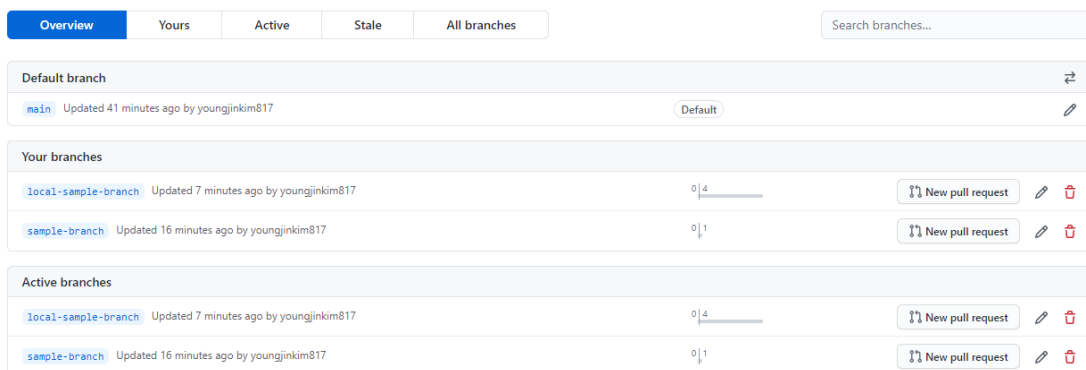
```
$git add .
```

```
$git commit -am "local branch file added"
```

```
$git push -u origin local-sample-branch
```

=> github에서 새로 생성된 local-sample-branch 를 확인해본다.

7. github 에서 브랜치를 클릭



8. New pull request 를 눌러 comparison 을 통해 변경사항을 확인하고 필요하다면 merge 할수 있다.

