

환경구성_github저장

1. 환경 구성하기

- 서버컴퓨터: 24시간 구동시켜주는 컴퓨터
 - a. 시각적으로 보기 좋은 UI, 기능 필요 X.
 - b. 코드를 정상적으로 구동시키는 역할만 하면 됨
(검은 화면에 흰색 글씨만 있는 그런 느낌)
- 개발서버: 프로젝트 개발을 진행 → local 개발
⇒ 가상환경; 각 프로젝트 맞는 라이브러리 사용
즉, 하나의 개발서버(컴퓨터)에서 가상으로 환경을 나눠 작업한다.
- 운영서버: 개발된 코드를 구동해서 운영 → cloud, server computer(pc) 배포+운영

1. git 설치(최신버전)

2. anaconda 설치

```
user@DESKTOP-HPS7MN3 MINGW64 ~  
$ conda --version  
conda 4.10.3  
  
user@DESKTOP-HPS7MN3 MINGW64 ~  
$ conda create --name envForDjango_ex python=3.8.8  
Collecting package metadata (current_repodata.json): ...working... done  
Solving environment: ...working... done  
  
## Package Plan ##
```

가상환경 만들고 확인

```
user@DESKTOP-HPS7MN3 MINGW64 ~  
$ conda info --envs  
# conda environments:  
#  
base * C:\Users\user\anaconda3  
Prophet C:\Users\user\anaconda3\envs\Prophet  
django C:\Users\user\anaconda3\envs\django  
envForDjango_ex C:\Users\user\anaconda3\envs\envForDjango_ex  
multi C:\Users\user\anaconda3\envs\multi  
multi2 C:\Users\user\anaconda3\envs\multi2
```

3. github 이용 순서

- a. cd install_check_project [원하는 파일로 이동]
- b. git init [사진기]
- c. git add . [. : all]
- d. git commit -m "블라블라"
- e. git push -u origin main

▼ 캡처

```
user@DESKTOP-HPS7MN3 MINGW64 ~/study/kd/django/project
$ cd install_check_project/

user@DESKTOP-HPS7MN3 MINGW64 ~/study/kd/django/project/install_check_project
$ git init
Initialized empty Git repository in C:/Users/user/Study/KD/Django/project/install_check_project/.git/

user@DESKTOP-HPS7MN3 MINGW64 ~/study/kd/django/project/install_check_project (master)
$ |
```

```
user@DESKTOP-HPS7MN3 MINGW64 ~/study/kd/django/project/install_check_project (master)
$ git add .

user@DESKTOP-HPS7MN3 MINGW64 ~/study/kd/django/project/install_check_project (master)
$ git commit -m "project init"
[master (root-commit) f058b96] project init
11 files changed, 200 insertions(+)
create mode 100644 db.sqlite3
create mode 100644 install_check_project/__init__.py
create mode 100644 install_check_project/__pycache__/__init__.cpython-38.pyc
create mode 100644 install_check_project/__pycache__/settings.cpython-38.pyc
create mode 100644 install_check_project/__pycache__/urls.cpython-38.pyc
create mode 100644 install_check_project/__pycache__/wsgi.cpython-38.pyc
create mode 100644 install_check_project/asgi.py
create mode 100644 install_check_project/settings.py
create mode 100644 install_check_project/urls.py
create mode 100644 install_check_project/wsgi.py
create mode 100644 manage.py
```

```

user@DESKTOP-HP57MN3 MINGW64 ~/study/kd/django/project/install_check_project (master)
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=85321887+lovegohome@users.noreply.github.com
user.name=lovegohome
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true

```

```

user@DESKTOP-HP57MN3 MINGW64 ~/study/kd/django/project/install_check_project (master)
$ git config --global --list
user.email=85321887+lovegohome@users.noreply.github.com
user.name=lovegohome
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true

```

```

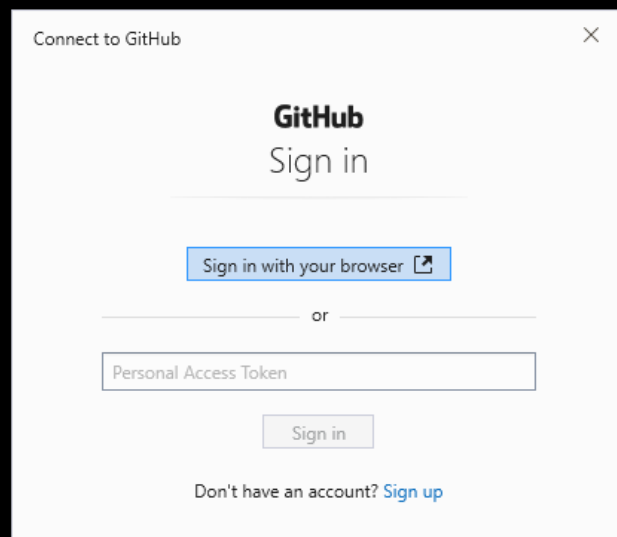
user@DESKTOP-HP57MN3 MINGW64 ~/study/kd/django/project/install_check_project (master)
$ git config --unset --global user.email

```

```

user@DESKTOP-HP57MN3 MINGW64 ~/study/kd/django/project/install_check_project (master)
$ git push -u origin master

```



▼ git push 부터 에러로 고생중.....

3개의 답변

2021. 3. 4.

```
remote: https://github.com/lovegohome/TIL/pull/new/master
```

```
user@DESKTOP-HP57MN3 MINGW64 ~/study/kd/django/project/install_check_project (master)
$ git branch -M main
```

4

```
git add .
git commit -m "clean push"
git push
기존 리포지토리 remote 제거
git remote remove origin
새 리포지토리 remote 추가
git remote add origin https://github.com/계정/리포지토리

git push -u origin main
```

```
user@DESKTOP-HPS7MN3 MINGW64 ~/study/kd/django/project/install_check_project (master)
$ git branch -M main

user@DESKTOP-HPS7MN3 MINGW64 ~/study/kd/django/project/install_check_project (main)
$ git remote remove origin

user@DESKTOP-HPS7MN3 MINGW64 ~/study/kd/django/project/install_check_project (main)
$ git remote add origin https://github.com/lovegohome/TIL.git

user@DESKTOP-HPS7MN3 MINGW64 ~/study/kd/django/project/install_check_project (main)
$ git push -u origin main
To https://github.com/lovegohome/TIL.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/lovegohome/TIL.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.
```

```
user@DESKTOP-HPS7MN3 MINGW64 ~/study/kd/django/project/install_check_project (main)
$ git pull
remote: Enumerating objects: 231, done.
remote: Counting objects: 100% (231/231), done.
remote: Compressing objects: 100% (217/217), done.
remote: Total 231 (delta 80), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (231/231), 34.27 MiB | 3.11 MiB/s, done.
Resolving deltas: 100% (80/80), done.
From https://github.com/lovegohome/TIL
 * [new branch]      main -> origin/main
 * [new branch]      master -> origin/master
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.

    git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=origin/<branch> main
```

▼ TIL 대신에 origin을 새로운 `lovegohome/remote-repo-example`으로 연결하
니깐 push 됨!!!

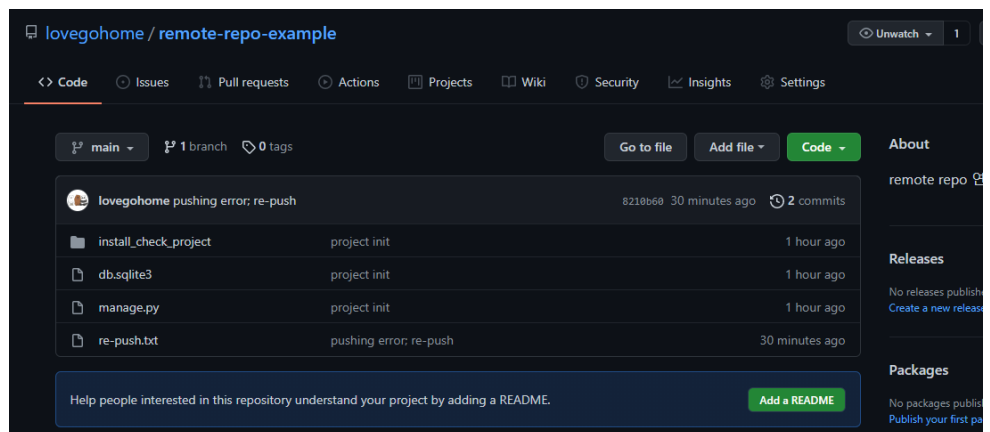
```

user@DESKTOP-HPS7MN3 MINGW64 ~/study/kd/django/project/install_check_project (main)
$ git remote remove origin

user@DESKTOP-HPS7MN3 MINGW64 ~/study/kd/django/project/install_check_project (main)
$ git remote add origin https://github.com/lovegohome/remote-repo-example.git

user@DESKTOP-HPS7MN3 MINGW64 ~/study/kd/django/project/install_check_project (main)
$ git push -u origin main
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 4 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (17/17), 4.93 KiB | 630.00 KiB/s, done.
Total 17 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), done.
To https://github.com/lovegohome/remote-repo-example.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.

```



git의 사본