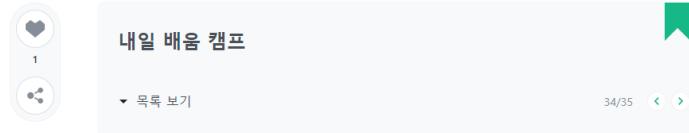


# Visual Studio Code에서 Django 프로젝트 기본 설정 하는 법(파일 생성부터 Git 설정까지)

rjsekaehdhkw · 2022년 6월 22일

django git python vscode 내일 배움 캠프



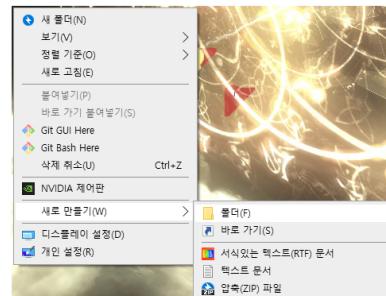
1. 프로젝트 폴더 만들기
2. Vscode로 폴더 열기
3. Django, Python 확장 설치
4. 가상환경 실행
5. 가상환경 실행
6. 패키지 설치
7. gitignore 설정
8. Github repository 생성
9. 설치 패키지 관리
10. VScode 터미널로 git 연동



이번에 Django 프로젝트들을 하면서 느낀 점들은 처음에 설정하는 부분에서 헷갈리는게 많아, 시간을 너무 잡아 먹었던 것 같습니다. 처음으로 Visual Studio Code로 이번에 넘어오게 되어서, Django 프로젝트를 할 때 필요한 기본 설정에 대해 알아보겠습니다. (원도우 기준으로 작성되었습니다.)

## 1. 프로젝트 폴더 만들기

우선 프로젝트를 진행할 폴더를 만들어 줍니다.



바탕화면에서 바탕화면을 우클릭하고 새로 만들기 -> 폴더를 클릭해서 새 폴더를 만들어 줍니다.



새 폴더를 만드셨다면, 이미지처럼 앞으로 진행할 프로젝트의 이름을 적어줍니다.

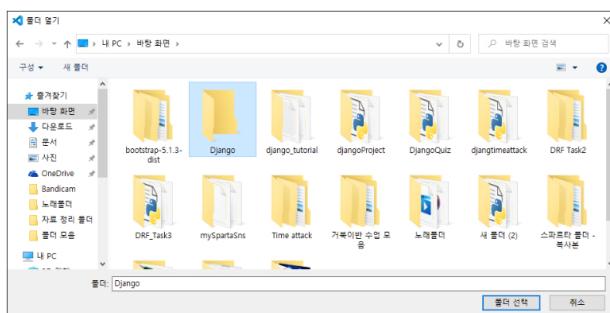
그리면 이제 Vscode를 실행시켜 주시고, Vscode로 만든 폴더를 열어보겠습니다.



Vscode를 실행 시키시면 이미지와 같은 화면이 됩니다.  
여기서 왼쪽 상단에 있는 파일을 눌러 줍니다.



파일을 누르시면 이렇게 설정들이 나옵니다, 여기서 폴더 열기를 눌러줍니다.



그리고 아까전에 만든 프로젝트 파일을 선택해서 열어 주시면 됩니다.



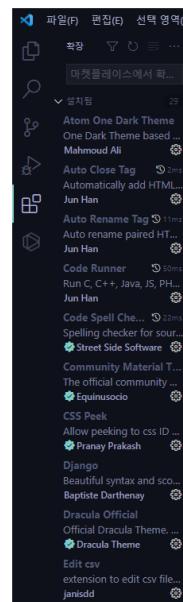


제대로 열어주셨다면 왼쪽 상단 탐색기라고 써진 부분 아래에 만들었던 프로젝트의 이름이 보이시는 걸 확인 할 수 있습니다. 그러면 제대로 성공하신 겁니다.

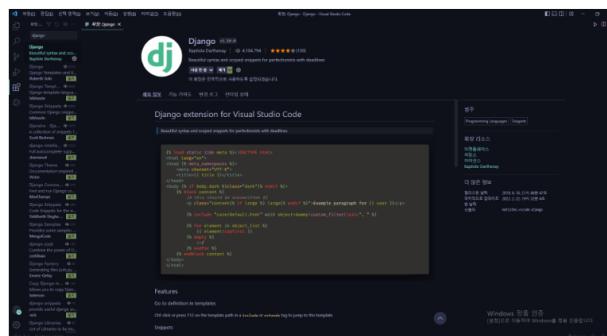
### 3. Django, Python 확장 설치



Vscode화면에서 오른쪽 아이콘 부분에서 "확장"을 눌러줍니다.



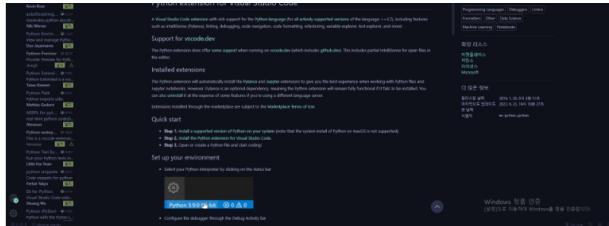
확장을 누르시면 이렇게 화면이 뜨고, 먼저 검색창에 Django를 입력해줍니다.



입력한 후 맨 위에있는 Django를 눌러서 화면과 같은 버전을 설치 해줍니다.  
(저는 이미 설치를 해둔 상태입니다.)

Django를 설치 하셨다면 검색창에 Python을 검색해 줍니다.





Python도 Django 설치할 때와 같이 똑같이 해주시면 끝입니다!

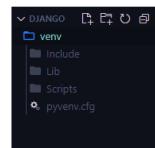
#### 4. 가상환경 설치



우선 현재 상태에서 **Ctrl + `** 을 눌러서 터미널 창을 열어줍니다.  
터미널 창을 열었다면 위의 이미지처럼 나옵니다.

```
python -m venv venv
```

위에 코드를 터미널에 입력하면 python 자체적으로 가지고 있는 가상환경을 설치해줍니다.



제대로 설치가 되었다면 해당 이미지처럼 나오게 됩니다.

#### 5. 가상환경 실행

이제 가상환경을 실행 시켜보도록 하겠습니다. 터미널에 아래의 코드를 입력합니다.

[Windows 기준]

```
venv\Scripts\activate
```

[Mac 기준]

```
venv/bin/activate
```

```
PS C:\Users\user\Desktop\django> venv\Scripts\activate  
(venv) PS C:\Users\user\Desktop\django> [ ]
```

이미지처럼 터미널 옆에 (venv) 이렇게 생기게 된다면 가상환경이 성공적으로 실행 된 것입니다.

제대로 실행이 안되신다면 코드를 앞에 source를 적어주시면 됩니다!  
예시) source venv/Scripts/activate

가상환경을 나가고 싶을때는 아래 코드를 적어주시면 됩니다.

```
deactivate
```

```
PS C:\Users\user\Desktop\ Django> venv\Scripts\activate  
(venv) PS C:\Users\user\Desktop\ Django> deactivate  
PS C:\Users\user\Desktop\ Django> █
```

이미지를 보시면 위의 코드를 입력한 후 (venv)가 사라진 모습입니다.

## 6. 패키지 설치

패키지를 설치할때는 꼭 가상환경을 실행시킨 후에 진행해주세요!

터미널 창을 열어주시고 아래의 코드를 입력해줍니다.

`pip install django`

```
[venv] C:\Users\user\Desktop\django> pip install django
Collecting django
  Using cached Django-3.2.11-py3-none-any.whl (7.9 MB)
    Preparing metadata (setup.py) ... done
  Using cached wheel-0.36.2-py3-none-any.whl (42 kB)
  Using cached pytz-2022.1-py2.py3-none-any.whl (583 kB)
  Using cached astroid-2.6.5-py3-none-any.whl (22 kB)
  Using cached cffi-1.15.1-py3-none-any.whl (22 kB)
  Using cached expiringdict-3.5.2-py3-none-any.whl (22 kB)
Collecting typing_extensions; python_version < "3.8"
  Using cached typing_extensions-4.2.0-py3-none-any.whl (24 kB)
  Installing collected packages: astroid, cffi, expiringdict, pytz, typing-extensions, django
    Successfully installed astroid-2.6.5 cffi-1.15.1 expiringdict-3.5.2 pytz-2022.1 django-3.2.11 typing_extensions-4.2.0
WARNING: You are using pip version 20.1.1; however, version 22.1.2 is available.
You should consider upgrading via the C:\Users\user\Desktop\django\venv\scripts\python.exe --upgrade pip command.
```

가상환경에 Django 패키지를 설치한 것 입니다.

이제 DRF를 설치 해보겠습니다. 아래 코드를 터미널 창에 입력해주세요

```
pip install djangorestframework
```

아래 이미지 처럼 나오면 설치 완료입니다!

## 7. gitignore 설정

`gitignore` 설정을 해보겠습니다. `gitignore` 설정을 하게되면 git에 올릴때 자동으로 올라가지 않도록 설정 할 수 있습니다.

[gitignore.io 주소]

<https://www.toptal.com/developers/gitignore/>

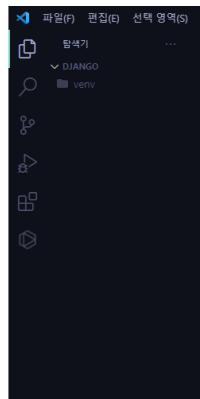
[gitignore.io](https://gitignore.io)

자신의 프로젝트에 꼭 맞는 .gitignore 파일을 만드세요.

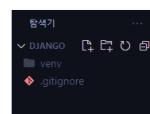
Windows × VisualStudioCode × Django ×

주소에 접속하시면 이미지의 화면이 보이는데 검색창에 자신이 사용하는 은연체/개발 화면/프로그램 및 어려들을 적어주시고 샌선을 눌러 줍니다.

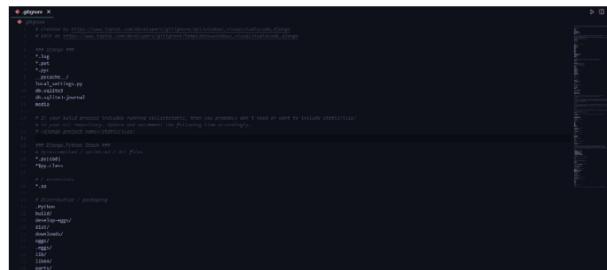
생성 버튼을 누르시면 이렇게 화면이 나오는데  
Ctrl + A를 해서 전체선택을 하고  
Ctrl + C를 해서 복사 해줍니다.



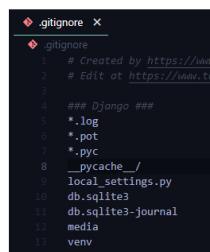
그 후 다시 VScode로 돌아가서 venv 아래쪽에 있는 빈칸을 우클릭 해서 새 파일을 만들어 줍니다. 이름은 .gitignore로 만들어 줍니다.



이미지와 같이 만들어지면 성공입니다. 그리고 .gitignore에 들어가서 아까 복사해온 부분들을 Ctrl + V를 해서 붙여넣기를 해줍니다.



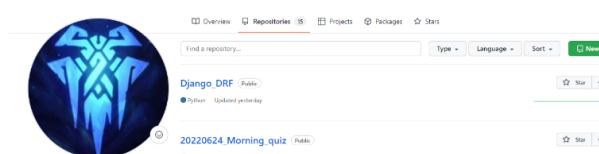
그리고 12번째줄에 있는 `media` 아래에다가 `venv`를 입력해주고 저장해줍니다.  
`Ctrl + C`를 해서 저장을 할 수 있습니다.



`venv`를 적고 저장까지 하고나면 이제 git에 업로드 할때 `venv` 파일은 업로드 되지 않습니다.  
저 부분에다가 파일 이름을 적어주면 그 파일은 업로드 되지 않습니다.

## 8. Github repository 생성

이제 Git에 업로드 하기 위해서 repository를 만들어 보겠습니다.



자신의 Github의 계정으로 로그인 한 후 이미지 화면의 New 버튼을 눌러줍니다.

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

Owner \* Repository name \*

rjskaehdhkw /  Django

Great repository names are short and memorable. Need inspiration? How about shiny-broccoli?

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.

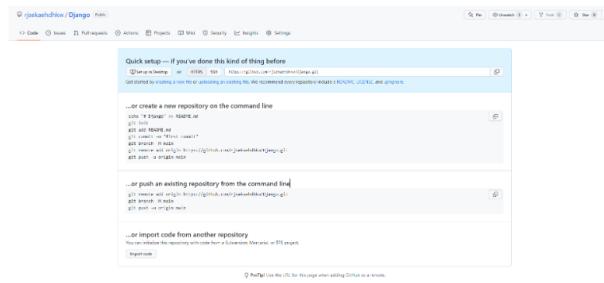
Add a README file This is where you can write a long description for your project. [Learn more.](#)

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.](#)

ⓘ You are creating a public repository in your personal account.

이름을 정하고 아래에 있는 Create repository를 눌러줘서 repository를 생성해 줍니다.



이미지의 화면이 나오면 성공적으로 만드신 겁니다!

## 9. 설치 패키지 관리

gitignore를 사용할때는 꼭 해줘야하는 부분이 있습니다.

`pip freeze > requirements.txt`

```
requirements.txt
1 asgiref==3.5.2
2 Django==3.2.13
3 pytz==2022.1
4 sqlparse==0.4.2
5 typing-extensions==4.2.0
```

코드를 적어주면 이미지처럼 txt파일이 생기고 안에는 설치한 패키지들이 있습니다.  
새로운 패키지를 추가할때마다 해당 코드를 입력해줘야 합니다.

이 파일을 업로드 하면 설치한 패키지를 편하게 관리 할 수 있고, 다른 사람이 이 파일을  
클론을 받아서 사용할때

`pip install -r requirements.txt`

위의 코드를 입력해주면 제가 설치했던 패키지를 한번에 다운받을 수 있습니다.

## 10. VScode 터미널로 git 연동

VScode에 있는 터미널로 git과 연동 해보겠습니다.

```
git init
```

해당 코드를 적으면 git을 시작합니다.

```
git remote add origin [Github repository 주소]
```

repository의 주소를 넣어줘서 앞으로 push 할 repository와 연결합니다.

```
git add .
```

현재 있는 파일들을 생성 해줍니다.

```
git commit -m '설명'
```

업로드할 파일의 설명을 적어줍니다.

```
git branch -M main
```

메인 브랜치의 이름을 설정합니다.

```
git push origin main
```

repository에 모든 파일들이 올라갑니다.



이미지처럼 뜬다면 성공적으로 파일 업로드에 성공한 것입니다!

다음 내용으로는 Django 기본 셋팅을 알아보겠습니다! 긴 글 읽어주셔서 감사합니다.



이전 포스트 장고 DRF 특강 1일차 과제

다음 포스트 403 Forbidden 오류에 대해 알...

월 비용 없는 풀서비스 사무실  
[40% 할인] 풀옵션 풀서비스로 청소부터 소모품까지 무료로 제공하는 중대형 프리미엄 사무실

fastfive.co.kr [견적 받기 >](#)

0개의 댓글

댓글을 작성하세요

댓글 작성

관심 있을 만한 포스트