



Django 설치 세팅과 기본 사용법

장고 설치하기

▼ 가장 최신 버전 다운로드하기

```
pip install django
```

▼ 특정 버전 설치하기

```
pip install django==2.1.15
```

▼ 장고 버전 확인하기

- 깔린 것들 중에서 장고 버전 확인 `pip list` (깔린 것들 중에서 장고 확인)
- 장고만 찾아서 버전 확인 `python -m django --version`

▼ vscode 에서 extension 설치 후 세팅 업데이트

- extension 에서 Django 검색
- 설치 후 Usage에서 코드 복사

```
"files.associations": {  
  "**/*.html": "html",  
  "**/templates/**/*.html": "django-html",  
  "**/templates/**/*.txt": "django-txt",  
  "**/requirements{/**,*.txt,in}": "pip-requirements"  
},  
  
"emmet.includeLanguages": {"django-html": "html"},
```

- `ctrl + shift + p` 누른 후 Open Settings(Json) 파일에 추가하기

```
{  
  "liveServer.settings.donotShowInfoMsg": true,  
  "window.zoomLevel": 1,
```

```

"terminal.integrated.shell.windows": "C:\\Program Files\\Git\\bin\\bash.exe",
"explorer.confirmDelete": false,
"python.dataScience.sendSelectionToInteractiveWindow": false,
"explorer.confirmDragAndDrop": false,

"files.associations": {
  "**/*.html": "html",
  "**/templates/**/*.html": "django-html",
  "**/templates/**/*.txt": "django-txt",
  "**/requirements{/**,*}.txt,in": "pip-requirements"
},

"emmet.includeLanguages": {"django-html": "html"}
}

```

1. 프로젝트 만들기

`django-admin startproject` 프로젝트이름 입력

e.g. `django-admin startproject first_project`

2. 서버 실행하기

`manage.py` 있는 디렉토리로 `cd` 한 후 `python manage.py runserver` 입력

잘 실행되었다면 로켓 페이지가 뜬다.

3. 앱 생성하기

`python manage.py startapp` 앱이름

e.g. `python manage.py startapp article`

참고) 앱 이름에 - 사용할 수 없고, 파이썬이나 장고에서 이미 쓰이는 이름을 쓰면 안된다.

앱을 생성하면 앱이름을 가진 폴더(여기서는 `article`)가 생기고 그 안에 부수적으로

`views.py` , `models.py` 등의 파일들이 자동으로 생성된다.

4. 앱을 프로젝트에 등록하기

```
INSTALLED_APPS = [
    #내가 만든 apps
    'articles',

    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

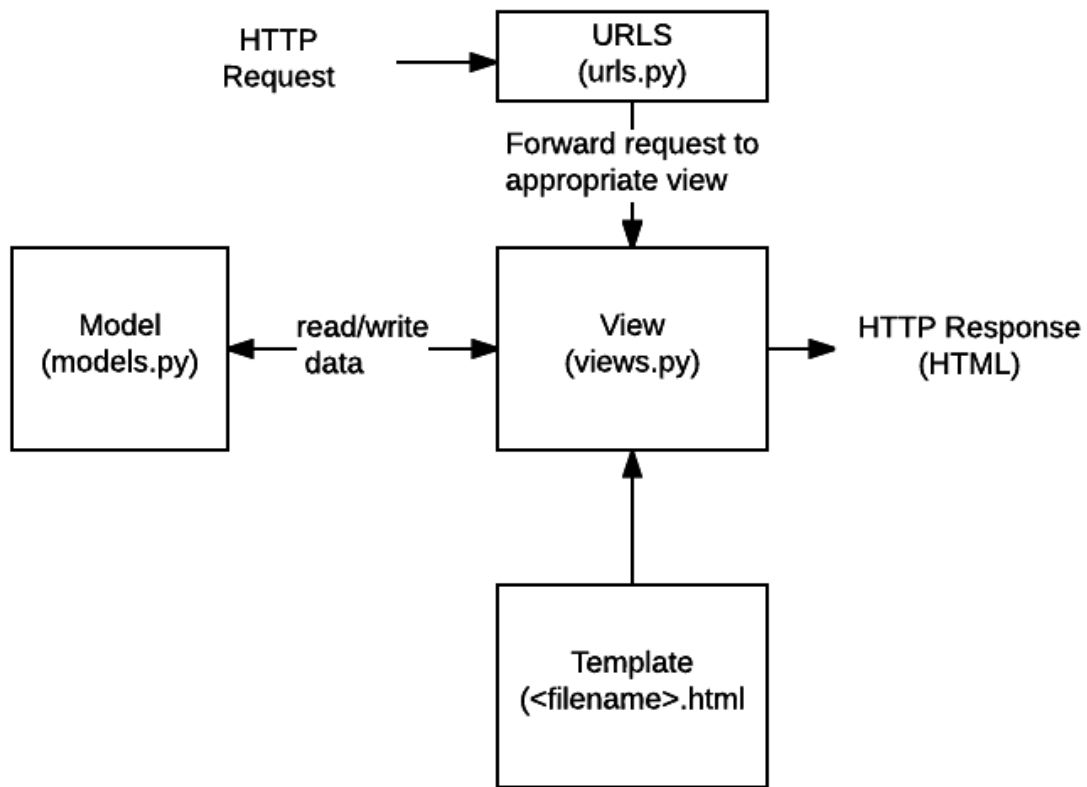
- `settings.py` 파일에서 `Installed_apps` 에 내가 방금 만든 app 이름을 상단에 추가한다.
- `settings.py` 파일 최하단에 `LANGUAGE_CODE = 'ko-kr'` 로 바꾸면 한글 설정이 된다.
- `TIME_ZONE = 'Asia/Seoul'` 시간 설정도 바꿀 수 있다.

참고사항)

- startapp으로 앱 생성 후 `settings.py` 에 앱 등록을 한다.
반대 순서로 진행하지 않게 주의하자.
- django에는 trailing comma이다. 코드뒤에 아무 것도 없어도 `,` 를 붙인다.
- 터미널 clear 단축키 `ctrl + l`
- Django imports style guide
 1. standard library
 2. erd party
 3. django
 4. local django

장고를 사용하기 위한 기본 세팅이 끝났고, 이제 코드 작업을 하면 된다.

코드 작업 순서는 (M - U - V - T) 이다.



MTV 패턴

<MTV 패턴> 을 참고하여 개발하자.

1. `urls.py` 파일에 주소를 추가한다.

```
from django.contrib import admin
from django.urls import path
from articles import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('index/', views.index),
    path('dinner/', views.dinner),
    path('hello/<str:name>/', views.hello),
]
```

- `urlpatterns` 안에 `path` 로 주소를 추가한다.
- `path()` 의 첫 번째 인자로써 유저가 리퀘스트를 보낼 주소 를 적는다.
- `path()` 의 두 번째 인자로써 리턴해 줄 함수 이름을 적는다.
- 이 때 아까 만든 앱을 `import` 하지 않으면 `views.py` 안에 정의한 함수를 불러올 수 없으니 `from 앱이름 import views` 를 해야 한다.
- 참고) url 뒤에 `/` 을 까먹지 말고 붙여주자.

2. `views.py` 에서 함수를 만든다.

```
def index(request):
    return render(request, 'index.html')
```

- 위와 같이 사용자가 특정 url 에 접속했을 때 불러질 함수를 만든다. 이 때 함수의 첫 번째 인자는 무조건 `request` 이다.
- `render()` 를 사용하여 `html` 페이지를 리턴해주는데, 이 때도 `render` 의 첫 번째인자는 무조건 `request` 이다.
- 두 번째 인자로써 `return` 할 `html` 페이지 명을 적는다.
- 이 때, 장고에서는 `앱/templates/` 위치에서 `html` 파일을 찾기로 정해져있기 때문에, 추가적으로 경로를 적어줄 필요 없이 파일명만 적어주면 된다.
- 따라서 앱 안에 `templates` 라는 이름을 가진 폴더를 생성하고, 그 안에서 `html` 파일을 작업해야한다.

3. `template` 반환

- `앱/templates/` 위치에서 반환해줄 `html` 페이지 템플릿을 생성한다.
- 파일명을 `render()` 함수 두 번째 인자와 같게 설정하면 된다.

