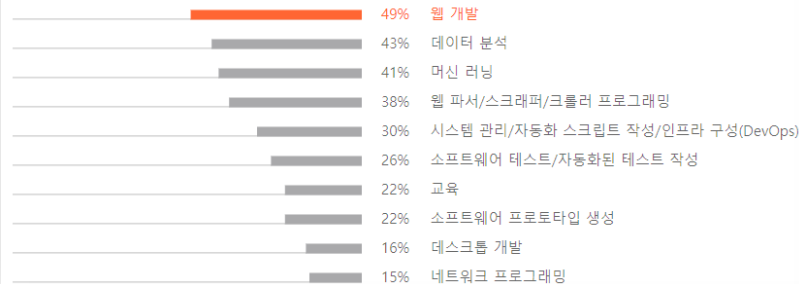


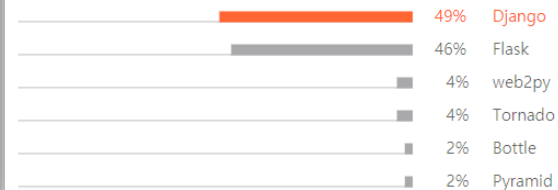
장고 (Django)

- 2003년에 Lawrence Journal-World newspaper에서 사용하기 위해 개발
- 파이썬으로 만들어진 무료 오픈소스 웹 애플리케이션 프레임워크

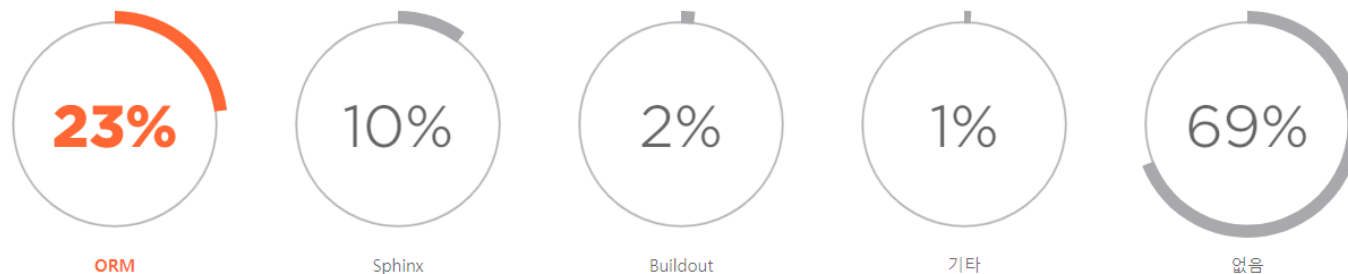
Python를 어떤 용도로 사용하십니까?



Python과 함께 어떤 웹 프레임워크 또는 라이브러리를 사용하십니까(복수 선택 가능)?



Python과 함께 어떤 다른 기술을 사용하십니까(복수 선택 가능)?



2020년 파이썬 라이브러리 순위

- <https://www.jetbrains.com/ko-kr/lp/devecosystem-2020/python/>

■ 장고 (Django)

● MTV 패턴 (Model, Template, View)

- Model : 데이터 처리 (DB의 테이블)
- Template : 클라이언트에게 보여줄 UI (HTML)
- View : Controller 역할, 요청을 받은 후 [모델을 실행한 후] 응답

● Model은 app/models.py 파일 내의 클래스 형식으로 작성

- 반드시 **app** 필요

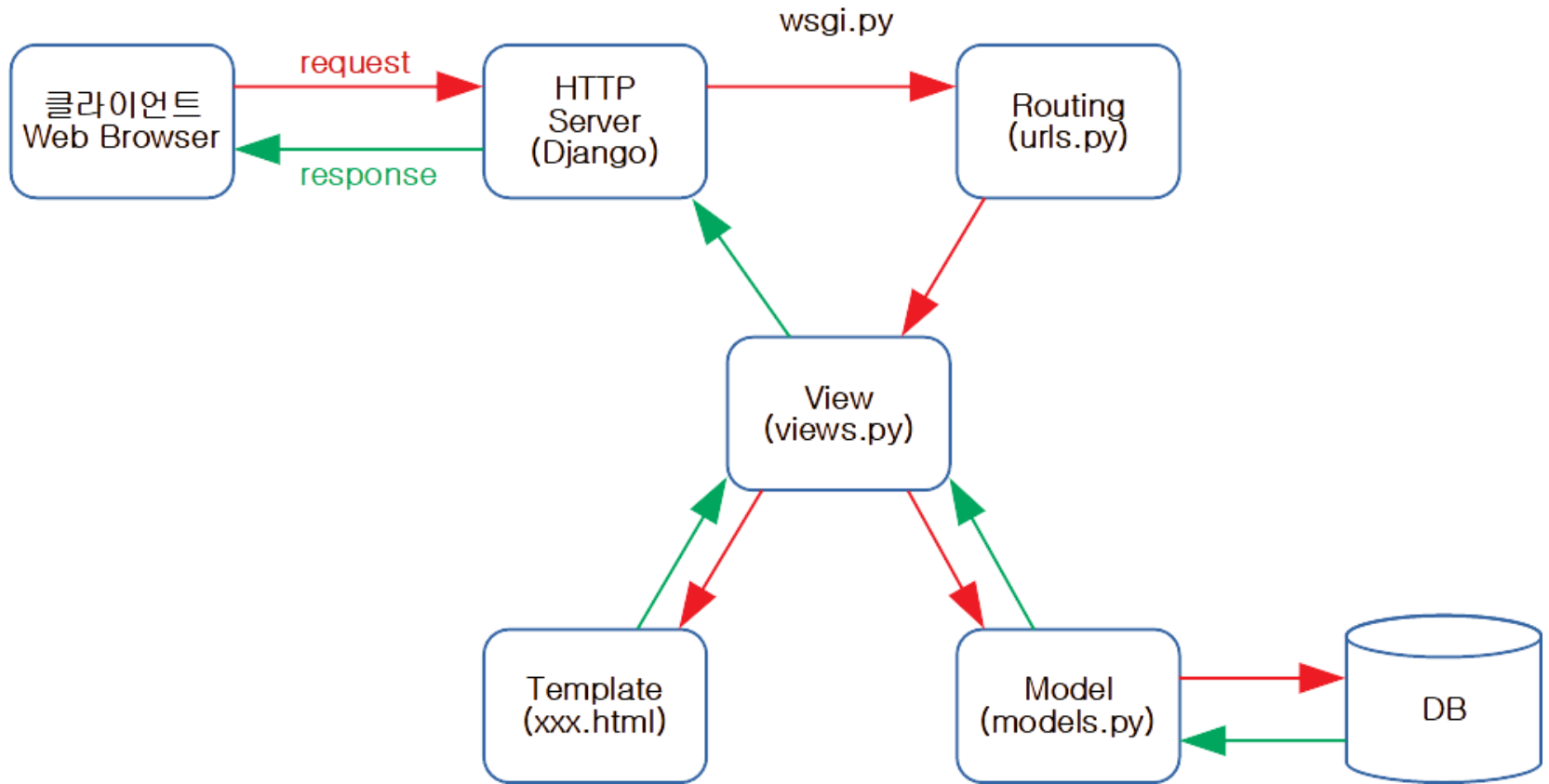
● Template은 templates 라는 디렉토리를 프로젝트 루트 또는 App 하위에 작성

- **settings.py** 설정 필요
- **CSS, JS 등 static 자원 사용 시 static 폴더 필요**

● View는 Project 또는 App에 views.py 파일의 함수 또는 클래스 형식으로 작성

- **app을 생성하지 않아도 사용 가능**

장고 (Django)



■ 장고 (Django)

● 디렉토리 및 파일 구조

sample/	최상위 경로, 프로젝트 내용을 담고 있는 단순 컨테이너 / 이름 변경 무관
manage.py	프로젝트 관리, 여러가지 기능을 동작시키는 커맨드 라인 유틸리티
config/	프로젝트의 패키지명, 파이썬 코드에서 import 시 사용
__init__.py	패키지 구성 시 기본 생성
settings.py	프로젝트의 환경설정
urls.py	접속 가능한 url 정보 파일, 장고 프로젝트의 콘텐츠 목록
wsgi.py	Web Server Gateway Interface
app/	일반적으로 프로젝트 내의 실제 기능을 담당하는 module

● 프로젝트 생성 명령어

django-admin startproject config .

(앱 생성 : django-admin startapp app)

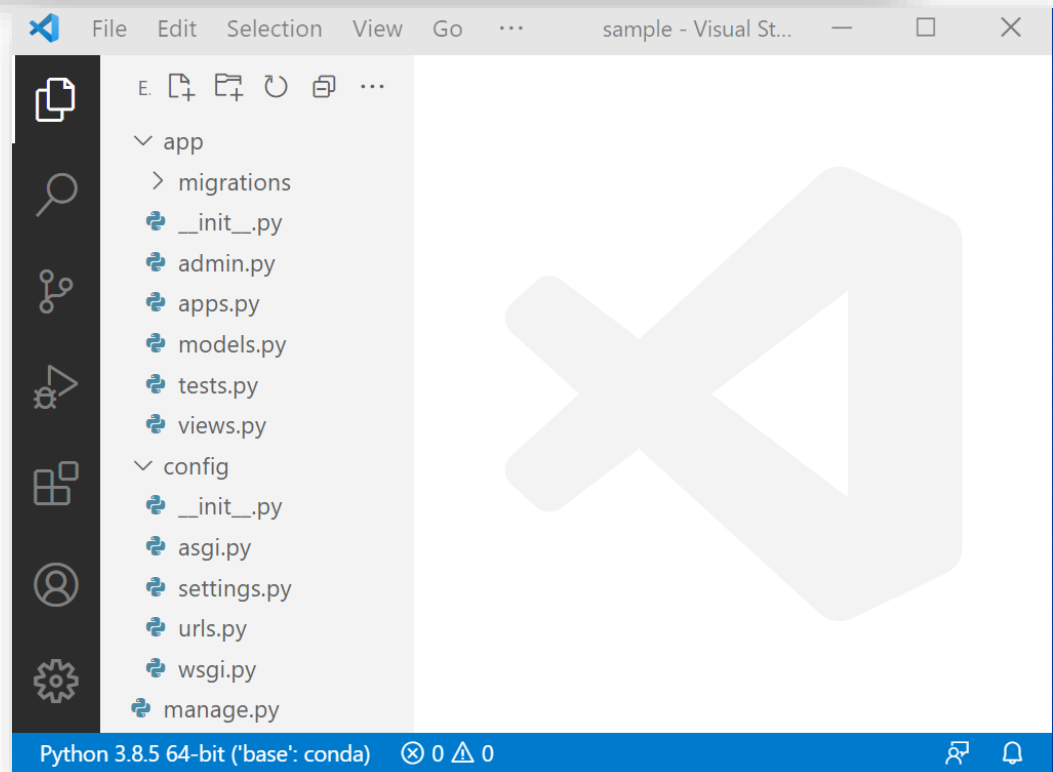
■ 프로젝트 생성

```
C:\dev\django>mkdir sample
```

```
C:\dev\django>cd sample
```

```
C:\dev\django\sample>django-admin startproject config .
```

```
C:\dev\django\sample>django-admin startapp app
```



■ 프로젝트 구성 파일

● settings.py : 프로젝트 환경설정 파일

- DEBUG : 개발 시 필요한 메시지 (오류 등) 확인
- ALLOWED_HOSTS : 서버로 접속 가능한 URL 지정
 - default : 'localhost' 또는 '127.0.0.1'
 - deploy : '0.0.0.0' 또는 서비스 도메인

```
from pathlib import Path

BASE_DIR = Path(__file__).resolve().parent.parent

SECRET_KEY = 'django-insecure-d-zp_h9(nu_)oadr3$du+7jt6n'

DEBUG = True

ALLOWED_HOSTS = []
```

■ 프로젝트 구성 파일

- settings.py : 프로젝트 환경설정 파일
 - DATABASES : 데이터베이스 엔진 설정
 - default : SQLite

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': BASE_DIR / 'db.sqlite3',  
    }  
}
```

※ MySQL(MariaDB) 설정 예시

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'mydatabase',  
        'USER': 'mydatabaseuser',  
        'PASSWORD': 'mypassword',  
        'HOST': '127.0.0.1',  
        'PORT': '3306',  
    }  
}
```

■ 프로젝트 구성 파일

● settings.py : 프로젝트 환경설정 파일

- TEMPLATES : html 등의 템플릿 관련 설정
 - default : 각 App 내의 templates 폴더

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [],  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.debug',  
                'django.template.context_processors.request',  
                'django.contrib.auth.context_processors.auth',  
                'django.contrib.messages.context_processors.messages',  
            ],  
        },  
    ],  
]
```

※ 프로젝트 공통 Template 사용 예시

```
'DIRS': [ BASE_DIR / 'templates' ],
```


■ 프로젝트 구성 파일

● settings.py : 프로젝트 환경설정 파일

- STATIC_URL : Web에서 사용되는 자원 (css, js, image 등) 경로 설정
 - default : 각 App 내의 static 폴더

```
STATIC_URL = '/static/'
```

※ 프로젝트 공통 Template 사용 예시

```
STATICFILES_DIRS = [  
    BASE_DIR / "static"  
]
```

■ 프로젝트 구성 파일

● manage.py : 프로젝트 관리 파일

- runserver : 서버 실행
- makemigrations app : app 의 모델 변경사항 확인
- migrate : 변경사항을 데이터베이스로 반영
- createsuperuser : 관리자 생성
- shell : 셸을 통해 데이터 확인 & 간단한 기능 테스트
- collectstatic : 여러개의 App 에서 사용하는 static 파일을 한 곳으로 모음

● 프로젝트 구동 명령어

python manage.py runserver

python manage.py runserver 포트번호

python manage.py runserver 접속주소:포트번호