

# Python Django (1.8)

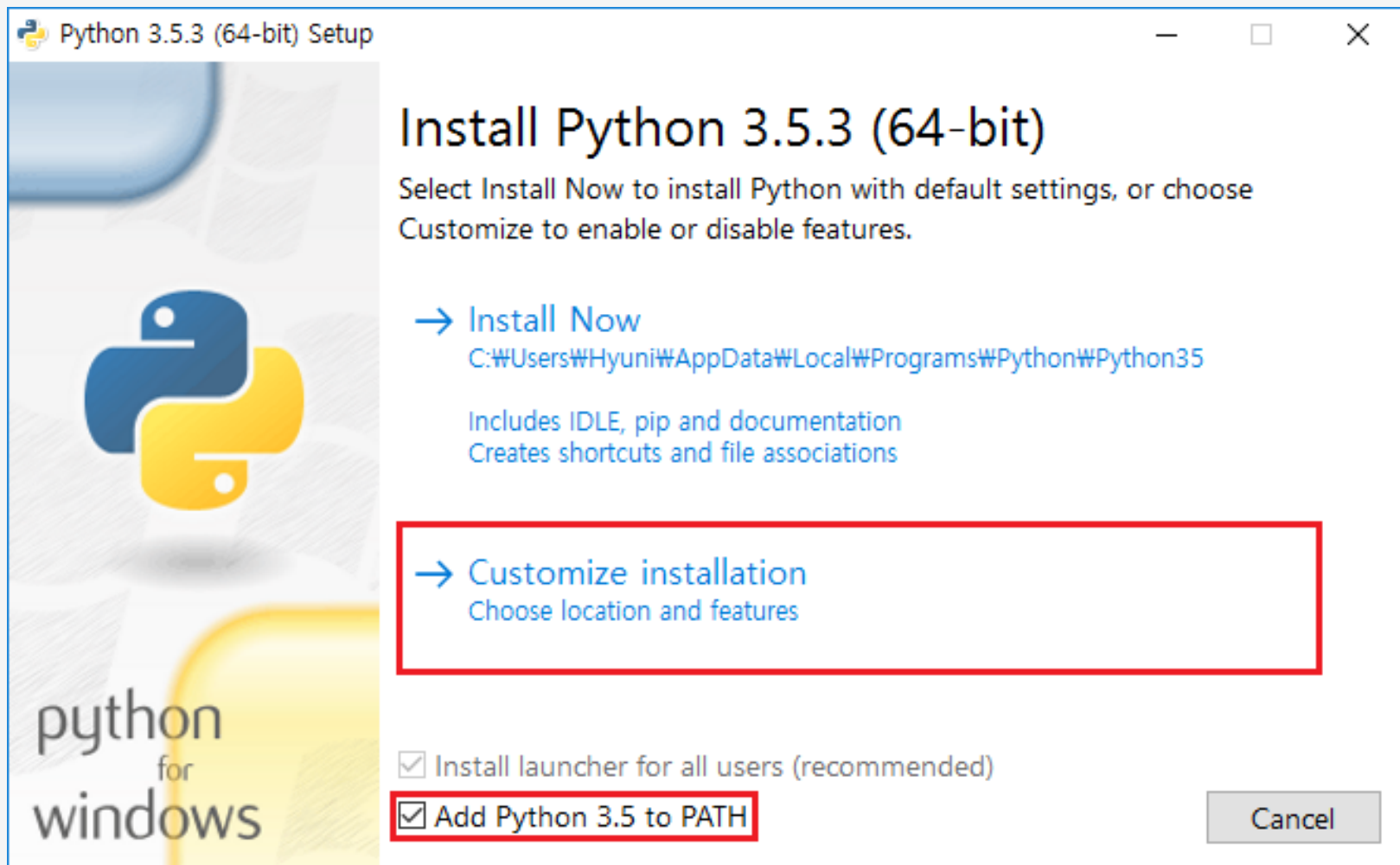
Web Programming

# Django 설치

- Python 설치
- pip 사용
- 가상환경 생성 및 동작
- 가상환경에서 Django 설치
- Django Project 생성 및 서버 동작
- Project 폴더 구조

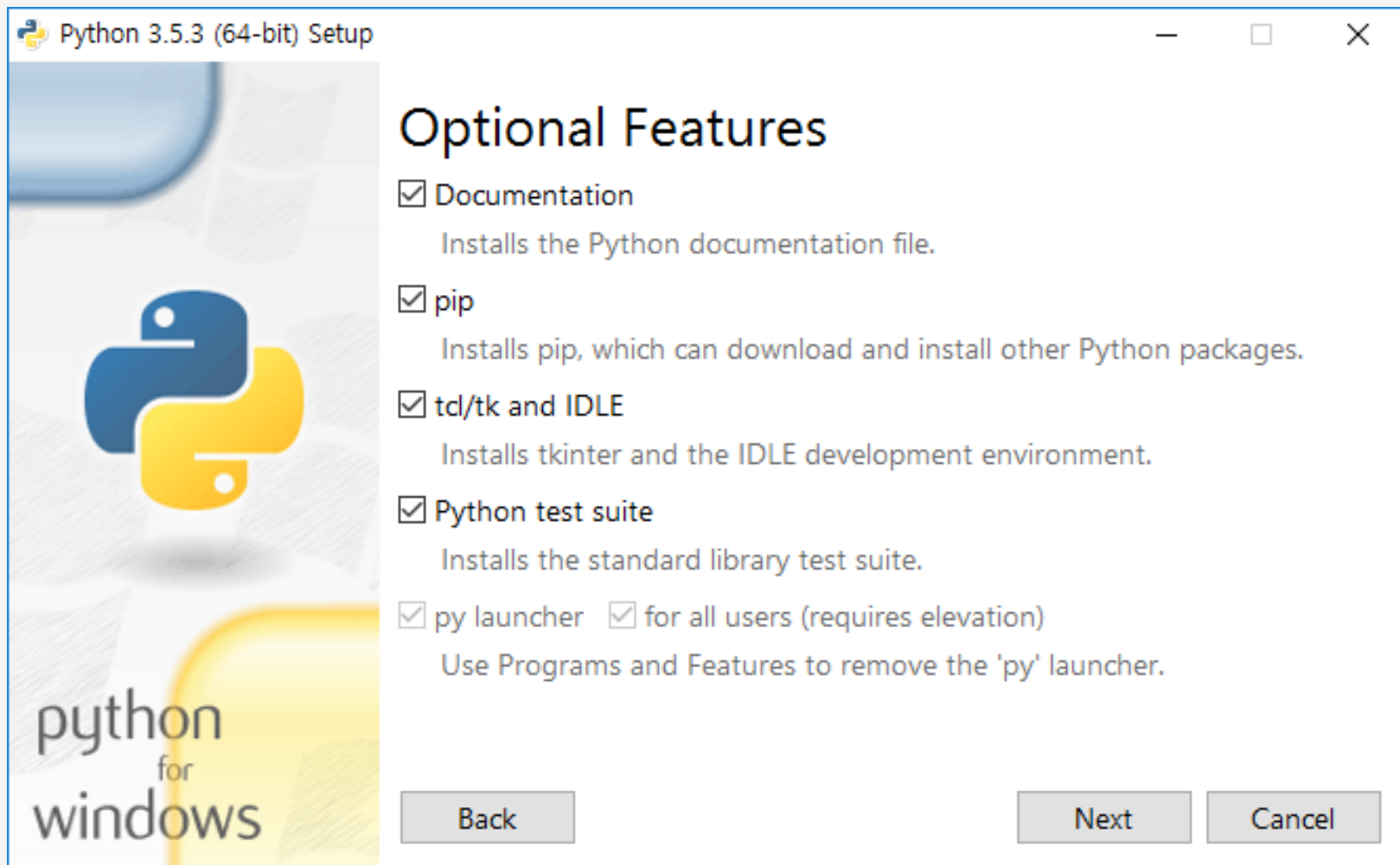
# Django 설치

## 1. Python 설치 (1/5)



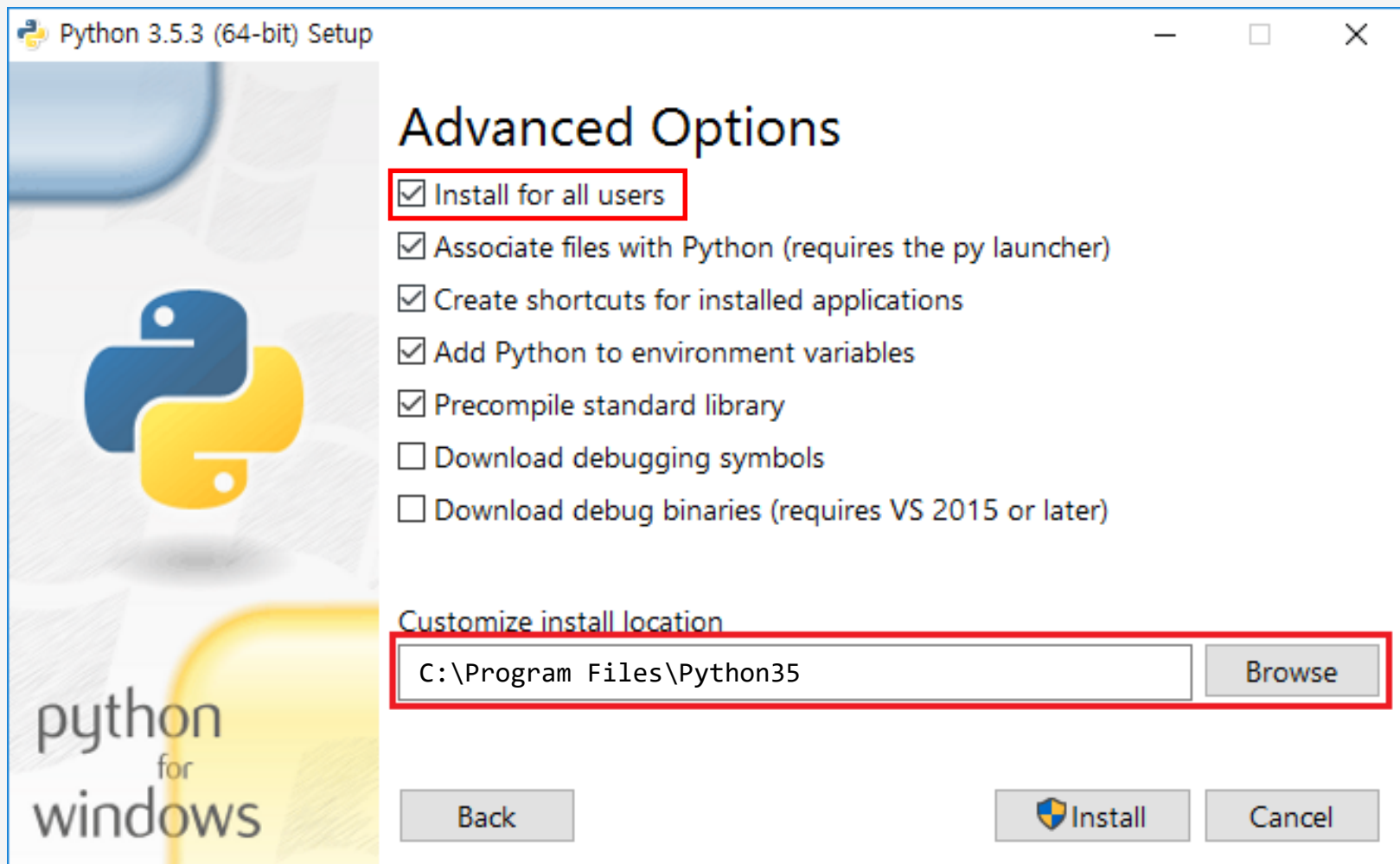
# Django 설치

## 1. Python 설치 (2/5)



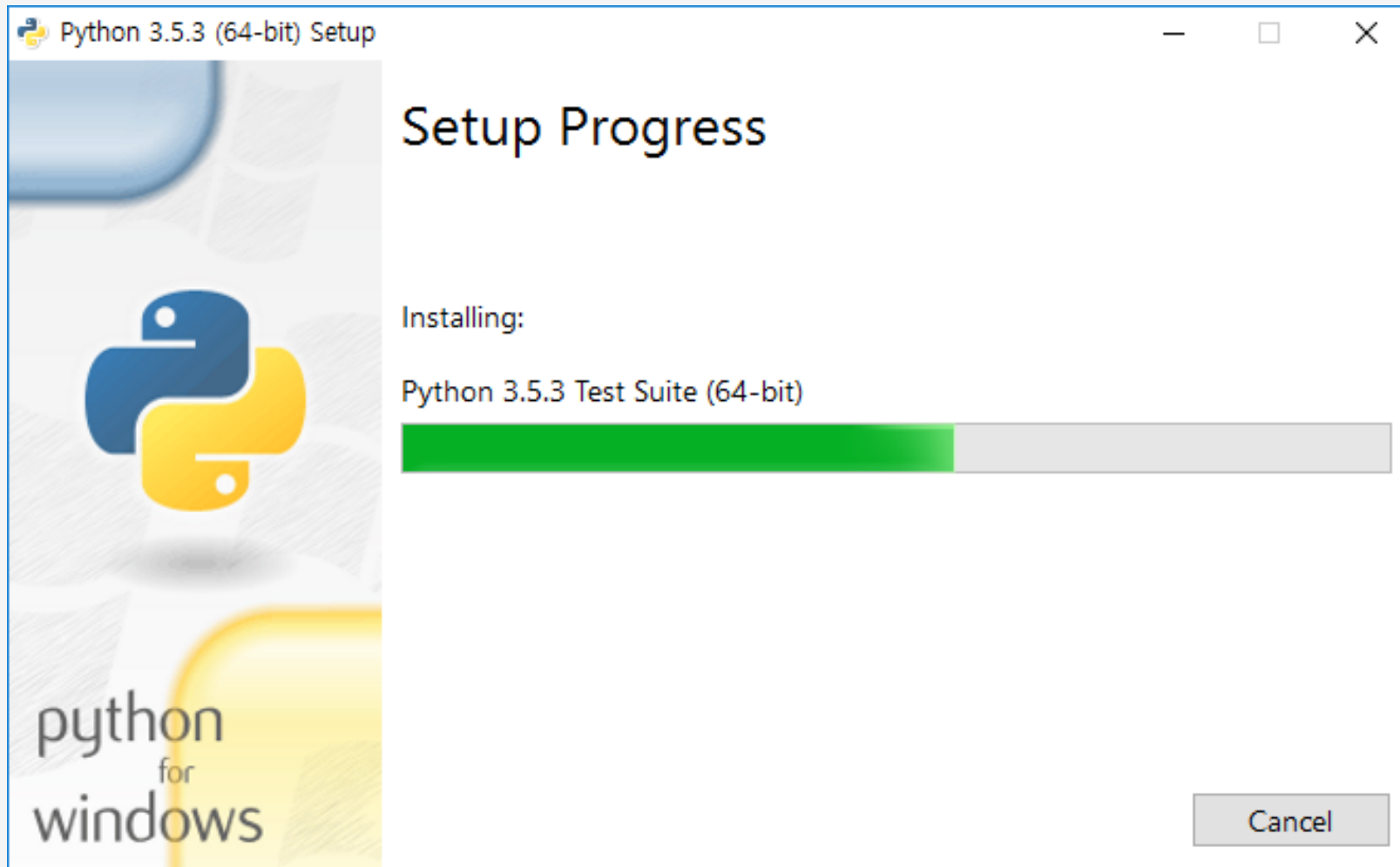
# Django 설치

## 1. Python 설치 (3/5)



# Django 설치

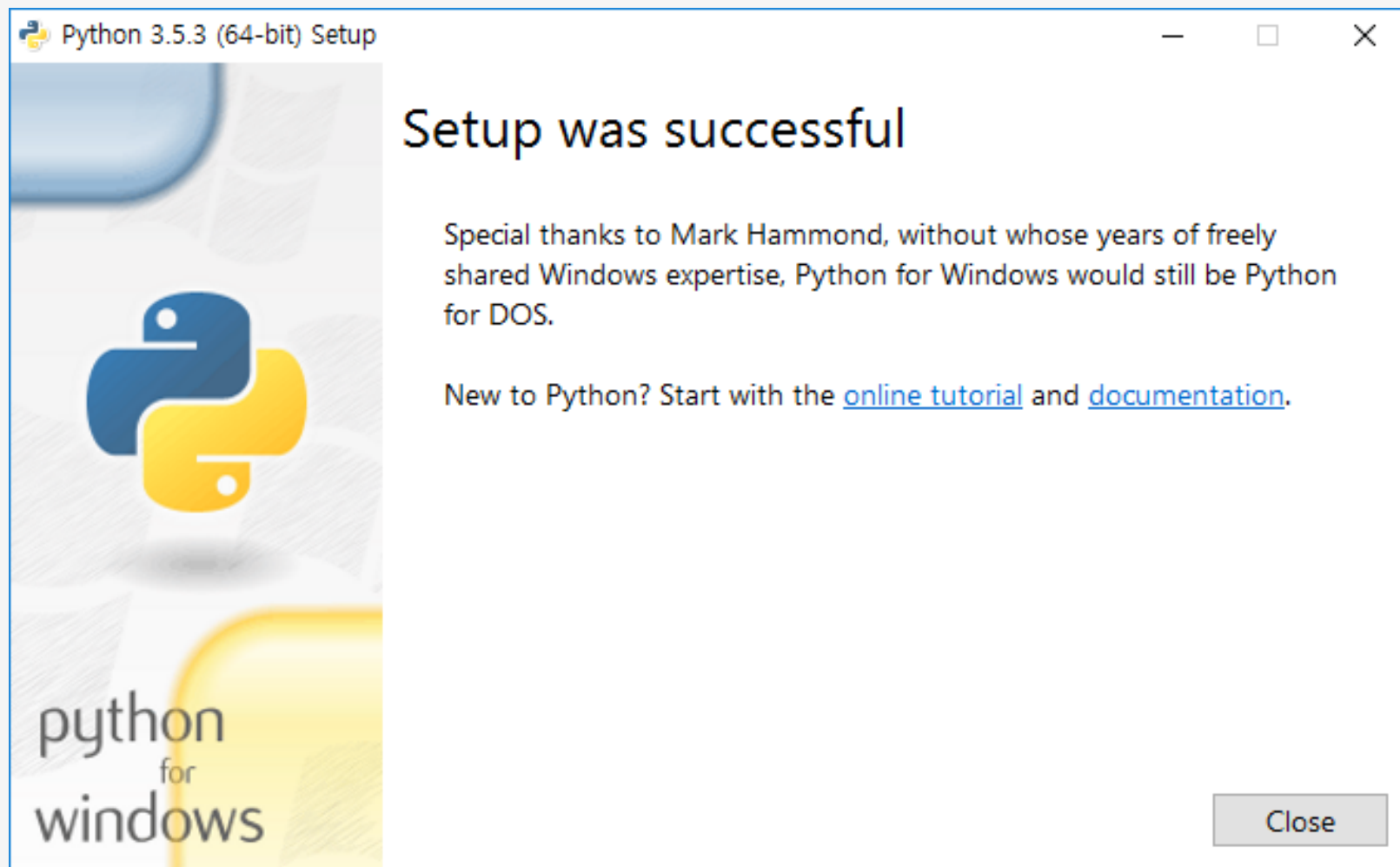
## 1. Python 설치 (4/5)



# Django 설치

---

## 1. Python 설치 (5/5)



# Django 설치

---

## 2. pip 사용

### PIP Command

**pip list** : 현재 설치되어 있는 Python Package 목록을 출력

**pip search <package name>** : 외부 Package 검색

**pip download <package name>** : Python Package 파일 다운로드

**pip install <package name>** : 외부 Package를 다운로드 받아 설치

**pip install <whl 파일>** : 다운로드 받은 .whl 파일로 설치

**pip uninstall <package name>** : 설치되어 있는 Package 삭제



# Django 설치

---

## 3. 가상환경 생성 및 동작 (1/2)

### 가상환경 생성

```
python -m venv <가상환경 폴더명>
```

### 가상환경 동작

```
<가상환경 폴더명>\Scripts\activate.bat
```

### 가상환경 종료

```
<가상환경 폴더명>\Scripts\deactivate.bat
```

# Django 설치

---

## 3. 가상환경 생성 및 동작 (2/2)

```
D:\workspace> python -m venv VirPy
```

```
D:\workspace> VirPy\Scripts\activate.bat
```

```
(VirPy) D:\workspace> VirPy\Scripts\deactivate.bat
```

```
D:\workspace>
```

# Django 설치

---

## 4. 가상환경에서 Django 설치 (1/2)

1. 가상환경을 활성화 시킨다.  
`<가상환경 폴더명>\Scripts\activate.bat`
2. pip를 사용하여 Django 1.8 Version을 설치 한다.  
`pip install django==1.8`
3. 설치 프로그램을 확인 한다.  
`pip list`

# Django 설치

---

## 4. 가상환경에서 Django 설치 (2/2)

```
D:\workspace> VirPy\Scripts\activate.bat
```

```
(VirPy) D:\workspace> pip install django==1.8
```

```
Collecting django==1.8
```

```
  Downloading Django-1.8-py2.py3-none-any.whl (6.2MB)
```

```
    100% |#####| 6.2MB 142kB/s
```

```
Installing collected packages: django
```

```
Successfully installed django-1.8
```

```
(VirPy) D:\workspace> pip list
```

```
.....
```

```
Django (1.8)
```

```
pip (9.0.1)
```

```
setuptools (28.8.0)
```

# Django 설치

---

## 5. Django Project 생성 및 서버 동작 (1/3)

### Django Project 생성

```
django-admin startproject <project_name>
```

### Django Project로 이동

```
cd <project_name>
```

### Django Server 동작

```
manage.py runserver
```

# Django 설치

---

## 5. Django Project 생성 및 서버 동작 (2/3)

```
(VirPy) D:\workspace>django-admin startproject myWeb
```

```
(VirPy) D:\workspace>cd myWeb
```

```
(VirPy) D:\workspace\myWeb>manage.py runserver  
Performing system checks...
```

```
System check identified no issues (0 silenced).
```

```
You have unapplied migrations; your app may not work properly until they are applied.  
Run 'python manage.py migrate' to apply them.
```

```
September 20, 2017 - 21:46:03
```

```
Django version 1.8, using settings 'myWeb.settings'
```

```
Starting development server at http://127.0.0.1:8000/
```

```
Quit the server with CTRL-BREAK.
```

# Django 설치

---

## 6. Project 폴더 구조

```
myWeb
├── myWeb
│   ├── settings.py # 기본 설정을 위한 파일
│   ├── urls.py     # URL 패턴을 위한 파일
│   └── wsgi.py      # 웹 서버와 통신을 위한 파일
└── manage.py       # Django 관리를 위한 파일
```

# Django Basic Setting

- App 생성 및 등록
- 기본 Database 등록 및 관리자 계성 생성
- admin-site 접속
- 기본 언어 설정 및 시간 설정
- App 폴더 구조



# Basic Setting

---

## 1. App 생성 및 등록

### App 생성

```
manage.py startapp <app_name>
```

### App 등록 - settings.py

```
INSTALLED_APPS = (  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    '<app_name>',  
)
```

# Basic Setting

---

## 2. 기본 DataBase 등록 및 관리자 계정 생성 (1/3)

### 기본 DataBasae 등록

```
manage.py migrate
```

### 관리자 계정 생성

```
manage.py createsuperuser
```

# Basic Setting

---

## 2. 기본 DataBase 등록 및 관리자 계정 생성 (2/3)

(VirPy) G:\workspace\myWeb> `manage.py migrate`

Operations to perform:

Synchronize unmigrated apps: staticfiles, messages

... 생략 ...

Synchronizing apps without migrations:

Creating tables...

... 생략 ...

Running migrations:

Rendering model states... DONE

... 생략 ...

Applying sessions.0001\_initial... OK

... 생략 ...

# Basic Setting

---

## 2. 기본 DataBase 등록 및 관리자 계정 생성 (3/3)

```
(VirPy) G:\workspace\myWeb> manage.py createsuperuser
Username (leave blank to use 'user'): admin
Email address:      # Enter로 생략 가능
Password: admin1234  # 입력되는 것은 보이지 않는다.
Password (again): admin1234  # 입력되는 것은 보이지 않는다.
Superuser created successfully.
```

# Basic Setting

---

## 3. admin-site 접속 (1/3)

```
(VirPy) G:\workspace\myWeb> manage.py runserver  
Performing system checks...
```

```
System check identified no issues (0 silenced).
```

```
You have unapplied migrations; your app may not work properly until they are applied.  
Run 'python manage.py migrate' to apply them.
```

```
September 20, 2017 - 21:46:03
```

```
Django version 1.8, using settings 'myWeb.settings'
```

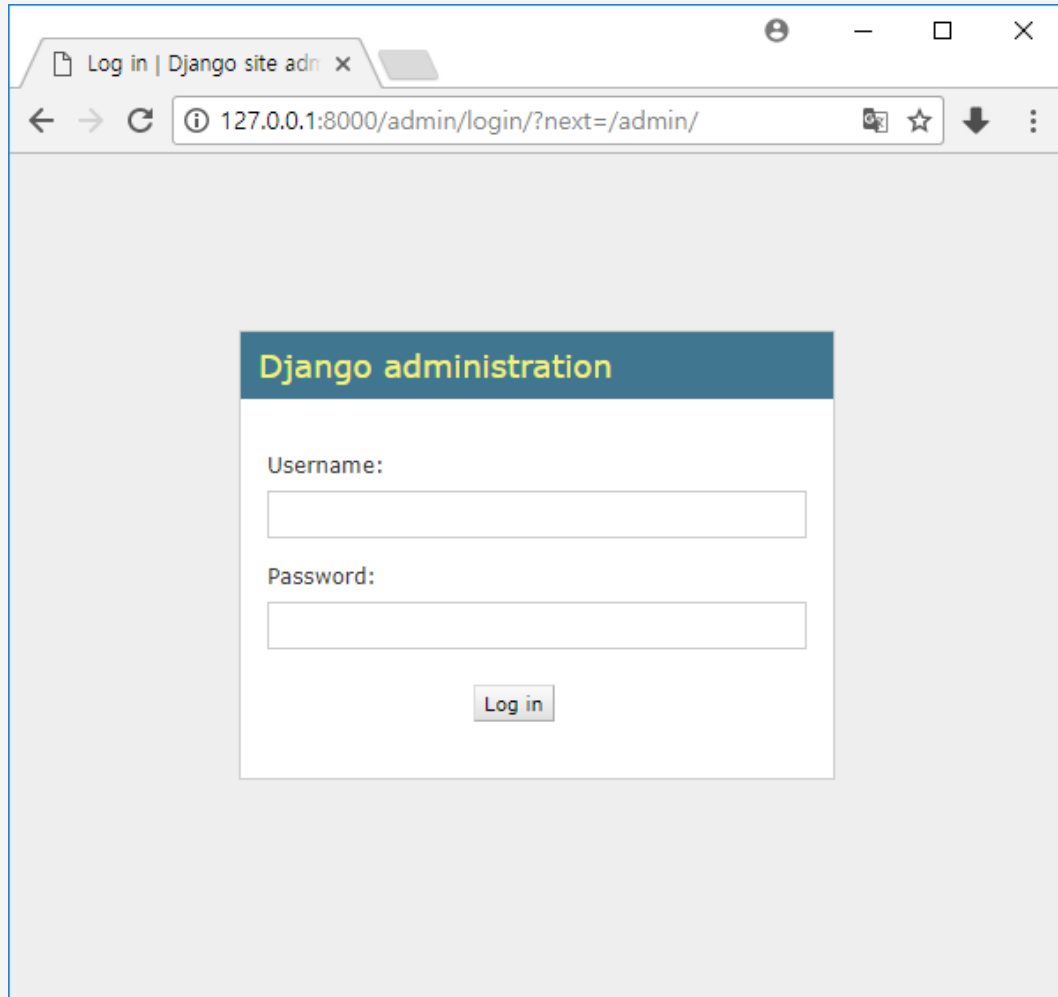
```
Starting development server at http://127.0.0.1:8000/
```

```
Quit the server with CTRL-BREAK.
```

# Basic Setting

---

## 3. admin-site 접속 (2/3)



The screenshot shows a web browser window with a single tab titled "Log in | Django site admin". The address bar displays the URL "127.0.0.1:8000/admin/login/?next=/admin/". The main content area features a login form with a dark blue header bar containing the text "Django administration" in yellow. Below the header, the form has two input fields: "Username:" and "Password:". A "Log in" button is positioned at the bottom of the form.

Log in | Django site admin x

127.0.0.1:8000/admin/login/?next=/admin/

Django administration

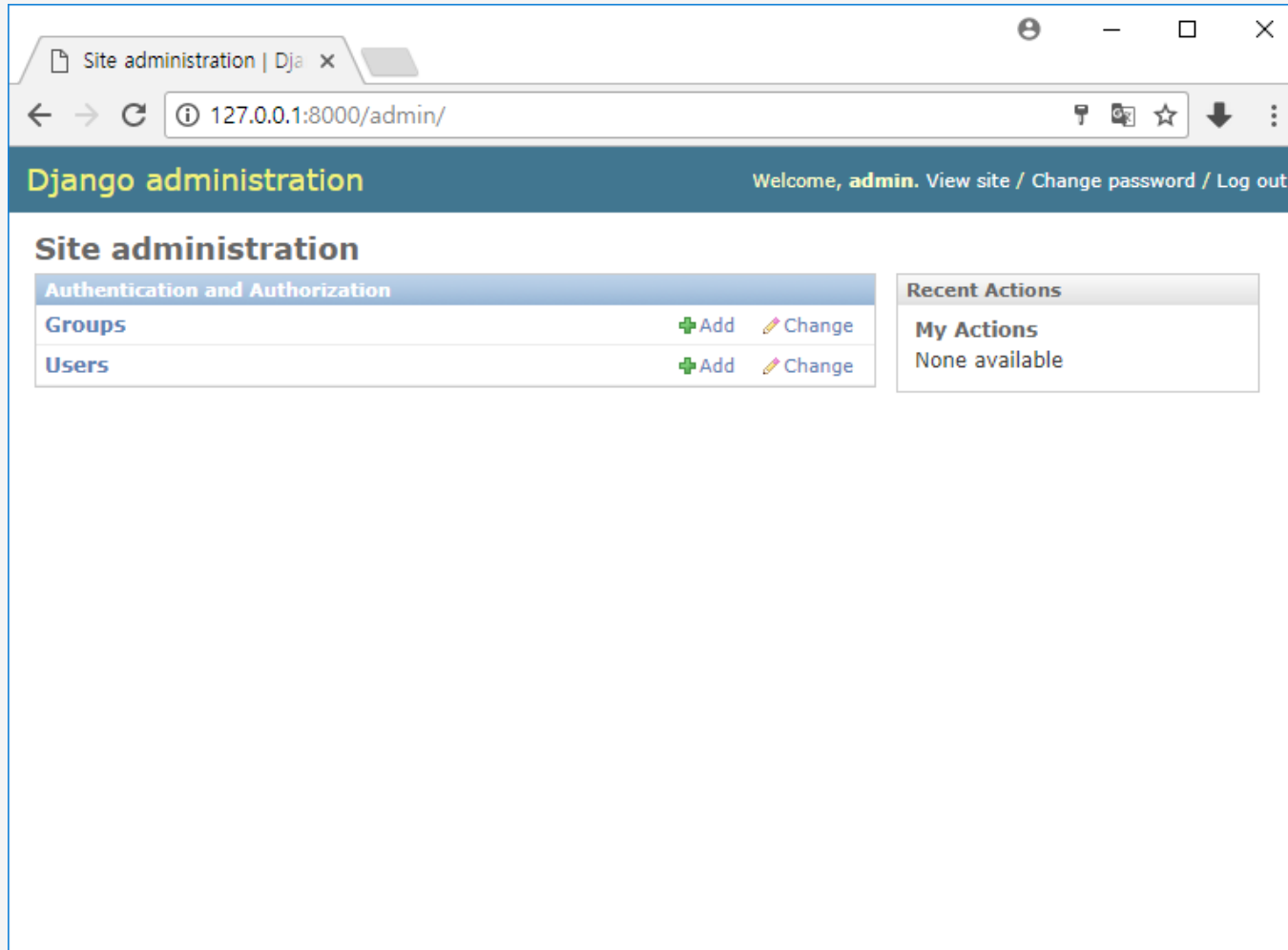
Username:

Password:

Log in

# Basic Setting

## 3. admin-site 접속 (3/3)



# Basic Setting

---

## 4. 기본 언어 설정 및 시간 설정 (1/2)

myWeb > settings.py

... 생략 ...

```
LANGUAGE_CODE = 'ko-kr'
```

```
TIME_ZONE = 'Asia/Seoul'
```

... 생략 ...



# Basic Setting

## 4. 기본 언어 설정 및 시간 설정 (2/2)



# Basic Setting

---

## 5. App 폴더 구조

```
myWeb
├── myWeb
│   ├── settings.py
│   └── .....
├── sub_app
│   ├── migrations      # DataBase Table Change Log
│   ├── admin.py        # Admin-Site Register
│   ├── models.py       # DataBase Model
│   └── views.py         # Control
├── db.sqlite3
└── manage.py
```

# Django MTV Model

- 웹 서버 기초
- Model 1 / Model 2
- Django MTV Model
- Django Folder/File 구조로 보는 MTV Model

# MTV Model

---

## 1. 웹 서버 기초

### 웹 클라이언트(Web Client)

웹 서비스를 요청하는 장치(Client Computer)로 사용자가 요청하는 Data를 웹 서버와 웹 클라이언트가 해석 할 수 있는 형태의 Data로(HTTP) 변환하여 정보를 교환하며, 최종적으로 교환된 정보를 사용자가 읽을 수 있는 형태의 문서로 변환하는 처리 까지 담당한다.

대표적인 웹 클라이언트 프로그램으로 **IE, Chrome, FireFox, Edge...** 등이 있다.

### 웹 서버(Web Server)

웹 서비스를 하기 위한 Data가 존재하는 장치(Server Computer)로 사용자가 요청하는 Data를 찾아 응답해 준다.

대표적인 웹 서버 프로그램으로 **IIS, Apache, Nginx...** 등이 있다.

# MTV Model

## 1. 웹 서버 기초



# MTV Model

---

## 1. 웹 서버 기초

### 클라이언트 사이드(Client Side) 언어

웹 클라이언트 프로그램에서 동작하는 언어로 웹 서버로부터 받은 Data를 사용자가 읽을 수 있는 형태의 문서로 변환을 하거나 문서 안에서의 여러 동작을 제어하기 위한 코드로 구성되어 있다. 대표적인 클라이언트 사이드 언어로는 **HTML, CSS, JavaScript...** 등이 있다.

### 서버 사이드(Server Side) 언어

웹 서버에서 동작하는 프로그래밍 언어로 사용자가 요청하는 Data를 처리하기 위한 Logic으로 구성되어 있다. 대표적인 서버 사이드 언어로는 **PHP, ASP, JSP, Python...** 등이 있다.

### 데이터베이스(DataBase)

웹 서버에서 사용자들에게 서비스하기 위한 각종 Data가 모여져 있는 저장소. 대표적인 데이터베이스 엔진으로는 **MS-SQL, ORACLE, MYSQL...** 등이 있다.

# MTV Model

---

## 1. 웹 서버 기초

### **프레임워크(Framework)**

프레임워크는 특정 프로그램 또는 특정 시스템, 서비스 등에서 범용적으로 재사용 할 수 있도록 구성되어 있는 라이브러리이다.

대표적으로 웹 프레임워크로는 Spring과 Django가 있으며, 이는 웹 서비스를 위한 여러 기능을 구현한 라이브러리 코드가 된다.

# MTV Model

---

## 2. Model1 / Model2 (1/2)

### **Model 1**

**DataBase Connection or Logic Code  
(PHP, JSP, Python)**

**Logic Code / HTML, CSS Control (JavaScript)**

**Design Code (CSS)**

**Document Structure Code (HTML)**



# MTV Model

---

## 2. Model1 / Model2 (2/2)

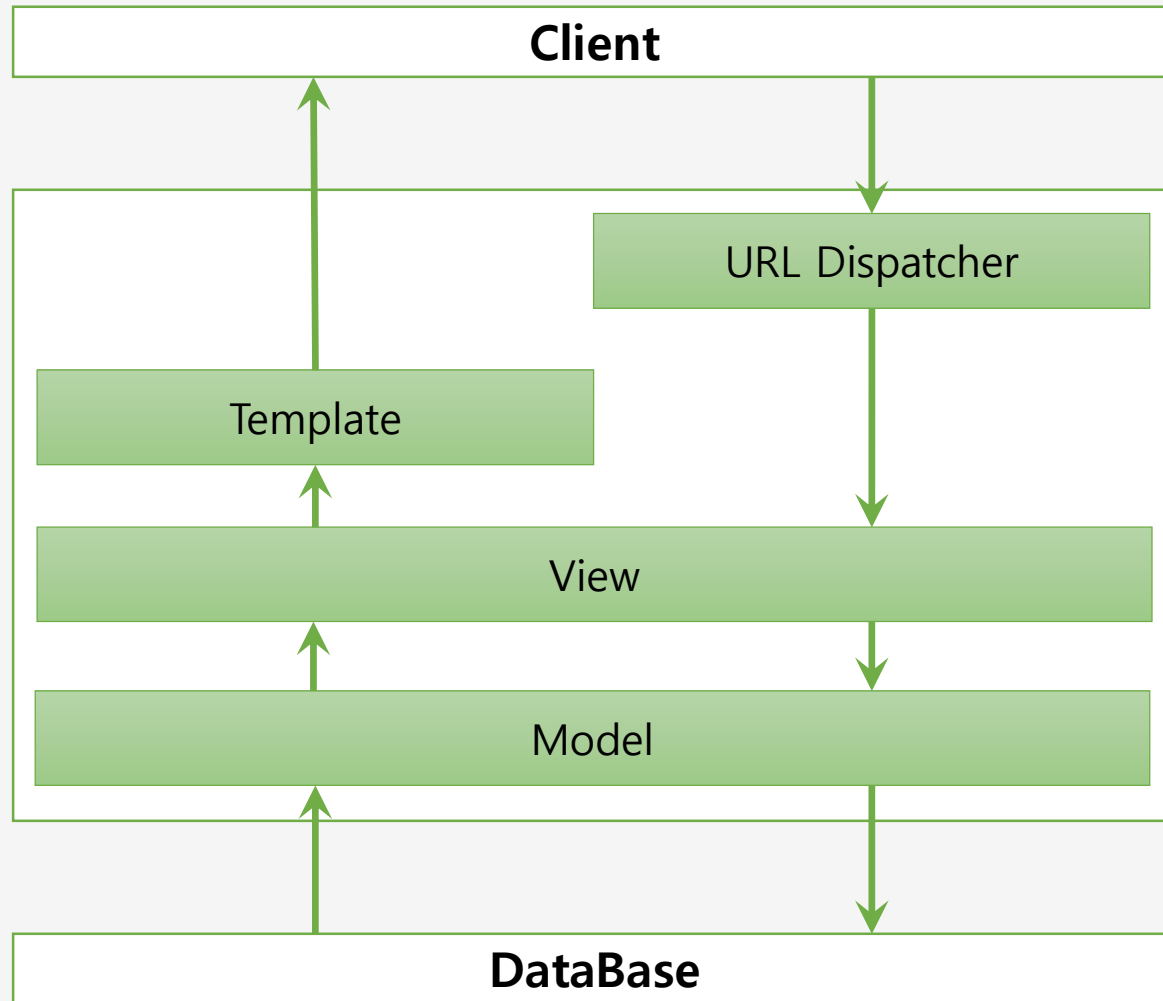
**Database (Model)**

**Document Design (Template)**

**Control (View)**

# MTV Model

## 3. Django MTV Model



# MTV Model

---

## 4. Django Folder/File 구조로 보는 MTV Model

