

Django Hands-on {Beginner}

Created by Sangkon Han ([@sigmadream](#))

[PyBusan](#) / [DjangoBusanUserGroup](#)

핵심 내용

1. Python, VirtualEnv 그리고 Django 설치
2. HelloWorld
3. Todo(POST/GET)

Python, VirtualEnv 그리고 Django 설치

1) Python 설치

- Python(≥ 3.6 이상)을 설치하세요.
 - 될 수 있으면 Python(≥ 3.5 , 64bit) 이상을 사용하세요.
- Python을 여러개 설치하지 마세요.
 - 개인적으로 Anaconda를 추천하진 않지만, Anaconda를 사용하실꺼면 Anaconda만 사용하세요.

2) 설치확인

- 콘솔/컨맨드(cmd)에서 아래 명령어를 사용해서 파이썬이 올바르게 실행되는지 확인하세요.

```
$ python --version  
Python 3.7.2
```

3) VirtualEnv 설정

- VirtualEnv 만들기

```
$ python -m venv venv
```

- VirtualEnv 활성화

```
$ venv\Scripts\activate  
or  
$ source venv/bin/activate
```

- venv가 활성화가 된다면 아래와 같은 화면을 볼 수 있음

```
(venv) $
```

4) Django 설치

- pip를 사용한 django 설치

```
(venv) $ pip install django
```

5) PyCharm 설치

- 개인적으로 VSCode를 주로 사용하지만 교육의 편의성을 위해서 PyCharm을 사용.
- PyCharm 다운로드 -> 설치
 - Project Interpreter 설정
 - django 설치

2. HelloWorld

0) Django의 기본 구성 요소

- Model
 - DB 사용을 위한 인터페이스
- URLConf
 - URL을 정의
- View
 - HTTP 요청을 처리
- Template
 - HTML 생성
- Admin

1) Django 프로젝트 생성 및 실행

```
(venv) $ django-admin startproject helloworld  
(venv) $ cd helloworld  
(venv) $ python manage.py runserver  
// 실행 취소는 Ctrl + C
```

2) CoffeeShop 앱 추가

```
(venv) $ python manage.py startapp coffeeshop
```

3) 앱 추가

- `helloworld/settings.py` 수정

```
INSTALLED_APPS = [  
    ...  
    'coffeeshop',  
]
```

- 실행 확인

```
(venv) $ python manage.py startapp runserver
```

4) Model 수정(coffeeshop/models.py)

```
from django.db import models

class Menu(models.Model):
    name = models.CharField(max_length=100)
    price = models.PositiveIntegerField()
    create_at = models.DateTimeField(auto_now_add=True)
    update_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.name
```

5) Model 적용

```
(venv) $ python manage.py makemigrations  
(venv) $ python manage.py migrate
```

6) 데이터 입력을 위해서 Admin 설정

- 관리자 추가

```
(venv) $ python manage.py createsuperuser
```

- Admin 추가

```
from django.contrib import admin  
from .models import Menu  
  
admin.site.register(Menu)
```


7) View 수정(coffeeshop/views.py)

```
from django.shortcuts import render
from .models import Menu

def menu_list(request):
    menus = Menu.objects.all()
    return render(request, 'shop/menu_list.html', {
        'menus': menus
    })
```

8) URL 수정(helloworld/urls.py)

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('shop/', views.menu_list)
]
```

9) Template 작성 (coffeeshop/templates/shop/menu_list.html)

```
<table>
  <tbody>
    {% for menu in menus %}
      <tr>
        <td>{{ menu.id }}</td>
        <td>{{ menu.name }}</td>
        <td>{{ menu.price }}</td>
      </tr>
    {% endfor %}
  </tbody>
</table>
```

3. FBV for Todo

1) 프로젝트 설정

```
(venv) $ django-admin startproject tasker  
(venv) $ cd tasker  
(venv) $ python manage.py runserver
```

2) Todo 앱 추가

```
(venv) $ python manage.py startapp todo
```

3) Todo 앱 설정

```
INSTALLED_APPS = [  
    ...  
    'todo',  
]
```

4) Todo Model 작성(tood/model.py)

```
from django.db import models

class Todo(models.Model):
    title = models.CharField(max_length=50)
    text = models.TextField(blank=True, null=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    def __str__(self):
        return self.title
```


5) Model 적용

```
(venv) $ python manage.py makemigrations  
(venv) $ python manage.py migrate
```

6) 데이터 입력을 위해서 Admin 설정

- 관리자 추가

```
(venv) $ python manage.py createsuperuser
```

- Admin 추가

```
from django.contrib import admin  
from .models import Todo  
  
admin.site.register(Todo)
```

7) View 수정(todo/views.py)

```
from django.shortcuts import render
from .models import Todo

def todo_list(request):
    todos = Todo.objects.all()
    return render(request, 'todo/todo_list.html', {
        'todos': todos
    })
```

8) URL 추가

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('todo/', views.todo_list),
]
```

9) Template 작성 (todo/templates/todo/todo_list.html)

```
<table>
  <tbody>
    {% for todo in todos %}
      <tr>
        <td>{{ todo.id }}</td>
        <td>{{ todo.title }}</td>
        <td>{{ todo.text }}</td>
      </tr>
    {% endfor %}
  </tbody>
</table>
```

10) POST를 사용한 데이터 추가를 위한 Template 추가

```
<h2>New Todo</h2>
<form action="" method="post">
  {% csrf_token %}
  <table class="table table-bordered">
    <tbody>
      <tr>
        <th>title</th>
        <td>
          <input type="text" name="title" value="{{ title }}" />
        </td>
      </tr>
      <tr>
        <th>text</th>
        <td>
          <textarea name="text">{{ text }}</textarea>
        </td>
      </tr>
    </tbody>
  </table>
</form>
```

11) POST 처리를 위한 View 추가

```
def todo_new(request):  
    if request.method == 'POST':  
        title = request.POST.get('title')  
        text = request.POST.get('text')  
  
        // 데이터 검증  
        if title and text:  
            todo = Todo()  
            todo.title = title  
            todo.text = text  
            todo.save()  
            return render(request, 'todo/todo_new.html')  
    else:  
        return render(request, 'todo/todo_new.html')
```

12) URL 추가

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('todo/', views.todo_list),
    path('todo/new/', views.todo_new),
]
```


