

Week 6

Backend-Basic

[Django Framework]

01. Basic Concept

02. Django Framework

03. API Design

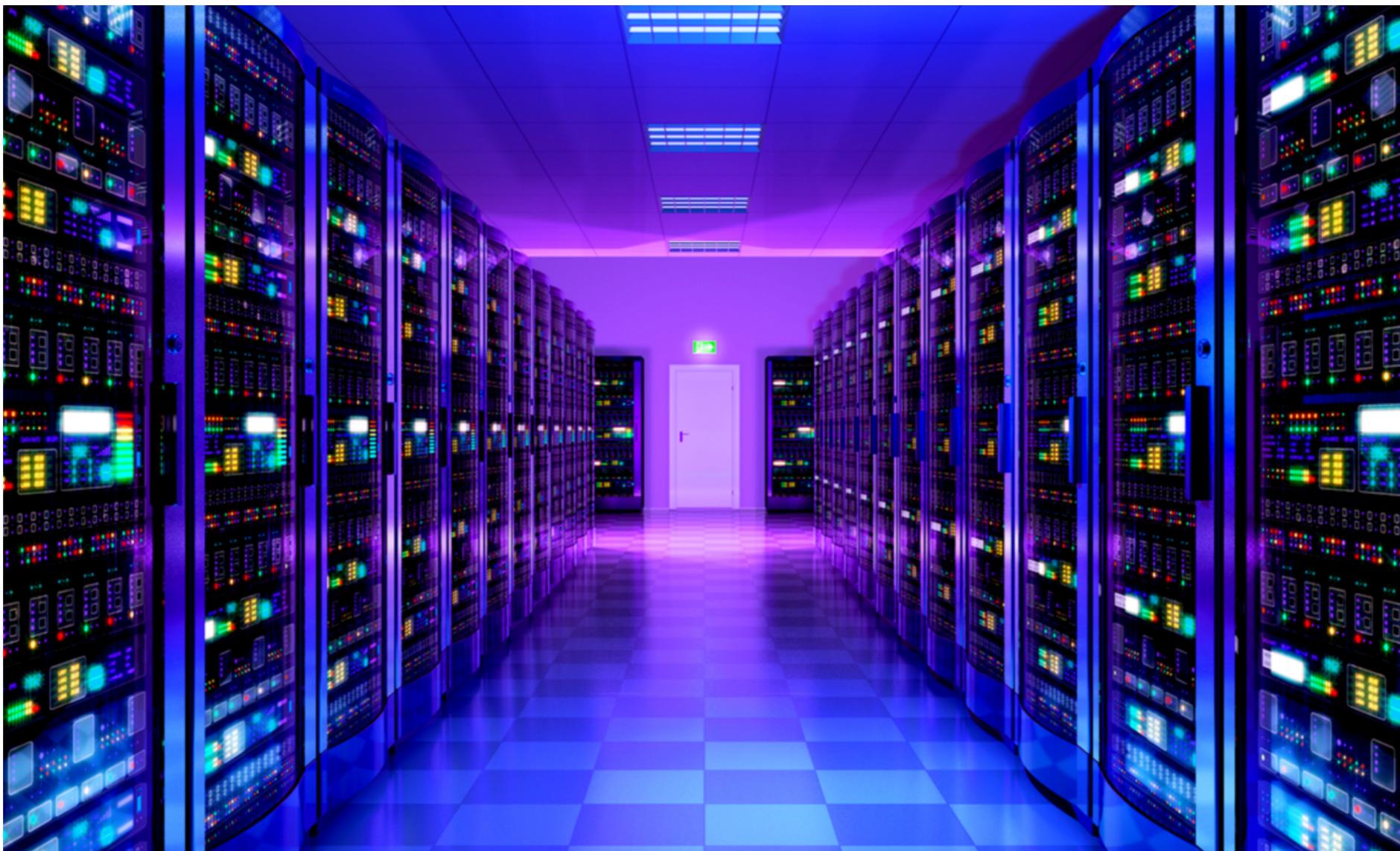
04. Terminal 실습

05. Django Server 생성

06. Django Server 구축

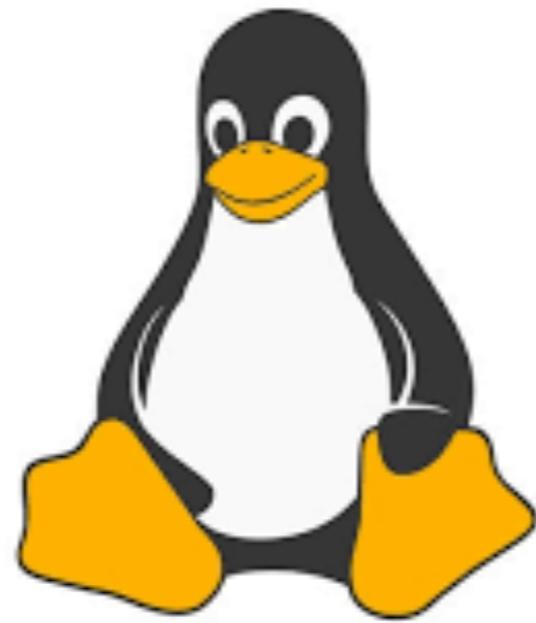
01

Basic Concept



Server

클라이언트에게 정보나 서비스를 제공하는 컴퓨터 프로그램 또는 장치



Linux

Windows, Mac과 같은 오픈소스 컴퓨터 운영체제

Shell

02

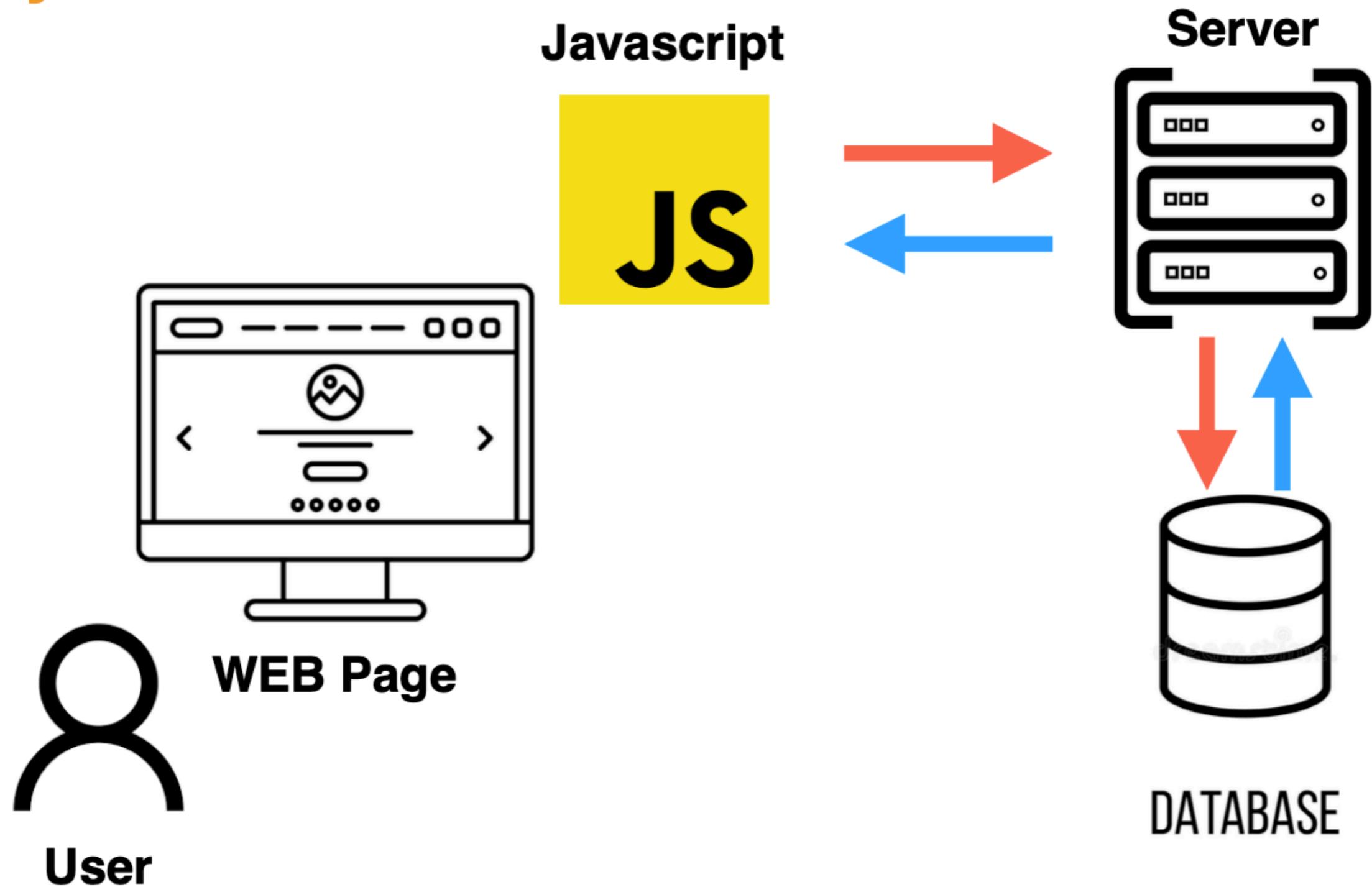
Django Framework



Django Framework

파이썬 언어 기반의 웹 프레임워크
데이터베이스 기반 웹사이트를 작성에 있어서 빠른 개발이 가능

Dynamic Web



Django, Web Framework

장고는 빠르게 Dynamic Web Page를 구현하는데 최적화된
Web Framework 입니다.

내부적으로 Template engine, Database를 지원하며, 원한
다면 오직 Django만을 사용하여 온전한 웹서비스를 개발할 수
있습니다!

API Server

하지만!! 저희는 보다 일반적인 서버에 대해 학습하기 위해, JSON 응답을
클라이언트에게 제공하는 API Server 개발을 실습할 겁니다 😊

03

API Design

API

Application **P**rogramming **I**nterface의 약자로..

그래서 그게 뭔데?

API Design (API Document)

HTTP METHOD	End Point	Description
GET	/member	사용자 정보 조회
GET	/footprint	footprint 조회
POST	/footprint/send	footprint 전송

API Design (Footprint 조회 응답)

```
{  
  "status": 200,  
  "message": "Footprint 조회 성공",  
  "data": {  
    "messages": [  
      "Javascript 4차 세미나 핫팅!",  
      "안녕하세요~~"  
    ]  
  }  
}
```

API Design

(Footprint 전송 요청 및 응답)

Request Body

```
{  
  "content": "안녕 영권아~",  
  "receiverName": "김영권",  
}
```

Success Response

```
{  
  "status": 200,  
  "message": "메시지 전송 성공"  
}
```

쉬는시간

04

Terminal 실습

꼭 알아야하는 Unix 명령어

cd: Change Directory

ls: List (optional)

mkdir: Make Directory (optional)

pwd: Print Working Directory

. : Current Directory

..: Outer Directory

~: Home Directory

/ : Root Directory

05

Django Server 생성

6차세션 사전 과제

- ★ 4주차 Javascript Data-Fetching 세션에서 사용했던 소스코드 복습 및 잘 동작하는지 확인해서 오기!
- ★ Python 추가 세션에서 다른 파이썬 기본 문법내용 등, 파이썬 복습해오기
- ★ 터미널 명령어 공부해서 실습해오기 (Unix 기반 터미널 명령어: cd, pwd, rm, mkdir, cd.. 등)
 - Windows: Bash shell (CMD, PowerShell 안됩니다)
 - Linux, Mac: 기본 터미널 또는 Bash

6차세션 사전 과제

```
youngkuon-MACBOOK ~ pip install django
DEPRECATION: Configuring installation scheme with distu
or Linuxbrew Python, please see discussion at https://
Requirement already satisfied: django in /usr/local/lib
Requirement already satisfied: sqlparse>=0.2.2 in /usr/
Requirement already satisfied: asgiref<4,>=3.4.1 in /us
DEPRECATION: Configuring installation scheme with distu
or Linuxbrew Python, please see discussion at https://
```

Django 설치를 위해 **pip install django** 명령어를 실행합니다.

왼쪽처럼 이미 설치되었다는 메시지, 또는 성공적으로 설치되었다는 메시지가 콘솔에 출력되면 성공!

```
youngkuon-MACBOOK ~ pip install virtualenv
DEPRECATION: Configuring installation scheme with distu
or Linuxbrew Python, please see discussion at https://gi
Requirement already satisfied: virtualenv in /usr/local/l
Requirement already satisfied: backports.entry-points-sel
Requirement already satisfied: six<2,>=1.9.0 in /usr/loc
Requirement already satisfied: filelock<4,>=3.0.0 in /usr
Requirement already satisfied: platformdirs<3,>=2 in /usr
Requirement already satisfied: distlib<1,>=0.3.1 in /usr/
```

가상환경 설치를 위해 **pip install virtualenv** 명령어를 실행합니다.

왼쪽처럼 이미 설치되었다는 메시지, 또는 성공적으로 설치되었다는 메시지가 콘솔에 출력되면 성공!

가상환경 (Virtual Environment)

Python과 Django는 다양한 모듈을 제공하고 있으며, 기본 모듈을 제외한 추가적인 모듈은 pip install 명령어를 통해 설치해서 사용해야 합니다.

가상환경을 켜놓고 모듈을 설치하면, 모듈은 가상환경 내에서만 사용할 수 있습니다.

```
youngkuon-MACBOOK ✨ ➔ 🗂 ~/workspace ➔ cd ~  
youngkuon-MACBOOK ✨ ➔ 🏠 ~ ➔ cd ~/workspace  
youngkuon-MACBOOK ✨ ➔ 🗂 ~/workspace ➔ cd LikeLion_workspace/10th  
SESSION           assignment-review js-session-server pair-coding-00  
youngkuon-MACBOOK ✨ ➔ 🗂 ~/w/L/10th ➔ ls  
SESSION/session-06  
youngkuon-MACBOOK ✨ ➔ 🗂 ~/w/L/10th/SESSION/session-06 ➔ cd SESSION/session-06  
/Users/jake-alden/workspace/LikeLion_workspace/10th/SESSION/session-06
```

본인이 작업할 workspace로 이동합니다
pwd 명령어를 통해 올바른 위치에 도달했는지 확인할 수 있습니다.

```
youngkuon-MACBOOK ~ ~/w/L/10th/SESSION/session-06 main *1 virtualenv myvenv
created virtual environment CPython3.9.8.final.0-64 in 494ms
  creator CPython3Posix(dest=/Users/jake-alden/workspace/LikeLion_workspace/10th/SESSION/session-06, no_vcs_ignore=False, global=False)
    seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, appdirs=True, pythonpath=True)
      added seed packages: pip==22.0.4, setuptools==62.1.0, wheel==0.37.1
      activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator
myvenv
```

virtualenv myvenv 명령어를 실행합니다.
이는 **myvenv**라는 이름의 가상환경을 생성하는
명령어입니다.

ls 명령어로 **myvenv**라는 폴더(가상환경)가
잘 생성된 것을 확인할 수 있습니다.

```
youngkuon-MACBOOK ~ ~/w/L/10th/SESSION/session-06 main *1 source myvenv/bin/activate
myvenv
```

source myvenv/bin/activate 명령어를 실행하여 가상환경을 켜줍니다.
가상환경이 켜있을 경우 사진과 같이 **myvenv**라는 표시가 나타납니다.

```
youngkuon-MACBOOK ~ ~/w/L/10th/SESSION/session-06 main *1 django-admin startproject session06
youngkuon-MACBOOK ~ ~/w/L/10th/SESSION/session-06 main *1 ?1 ls
myvenv session06
youngkuon-MACBOOK ~ ~/w/L/10th/SESSION/session-06 main *1 ?1 cd session06
youngkuon-MACBOOK ~ ~/w/L/10th/SESSION/session-06/main/session06 main *1 ?1 ls
manage.py session06
```

django-admin startproject session06 명령어를 실행합니다.
이는 **session06**이라는 이름의 django 프로젝트를 생성하는 명령어입니다.

cd session06 명령어를 실행한 후, **ls** 명령어로 폴더 내부를 조회해보면
manage.py, 프로젝트명과 동일한 이름을 가진 **session06** 폴더가 있는 것을 확인
할 수 있습니다.

```
youngkuon-MACBOOK ~ /w/L/10th/SESSION/session-06/session06 main *1 ?1 pip install django
Collecting django
  Using cached Django-4.0.4-py3-none-any.whl (8.0 MB)
Collecting asgiref<4,>=3.4.1
  Downloading asgiref-3.5.1-py3-none-any.whl (22 kB)
Collecting sqlparse>=0.2.2
  Using cached sqlparse-0.4.2-py3-none-any.whl (42 kB)
Installing collected packages: sqlparse, asgiref, django
Successfully installed asgiref-3.5.1 django-4.0.4 sqlparse-0.4.2
```

가상환경 내에서 django를 다시 한번 설치해줍니다.

```
youngkuon-MACBOOK ~ /w/L/10th/SESSION/session-06/session06 ls
db.sqlite3 manage.py session06 myenv 1.63G 15:18

youngkuon-MACBOOK ~ /w/L/10th/SESSION/session-06/session06 python manage.py runserver myenv 1.63G 15:18
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
May 11, 2022 - 06:18:53
Django version 4.0.4, using settings 'session06.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

ls 명령어로 manage.py가 존재하는 위치에서,
python manage.py runserver 명령어를 실행하여 서버를 켜줍니다.

```
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work
Run 'python manage.py migrate' to apply them.
May 11, 2022 - 06:18:53
Django version 4.0.4, using settings 'session06.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

이렇게 <http://127.0.0.1:8000>에 서버가 열렸다는 메시지가 출력되면
장고 서버 생성하기 성공!!!

쉬는시간

06

Django Server 구축

```
youngkuon-MACBOOK ✚ ~/w/L/10th/SESSION/session-06/session06 ✚ main *1 ✚ django-admin startapp footprint  
youngkuon-MACBOOK ✚ ~/w/L/10th/SESSION/session-06/session06 ✚ main *1 ?1 ✚ ls  
db.sqlite3 footprint manage.py session06
```

manage.py가 있는 폴더에서 **django-admin startapp footprint** 명령어를 실행하여 **footprint**라는 app을 생성합니다.

Django에서 app이란 프로젝트 하위에서 기능별로 구분하는 단위입니다.

```
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'footprint', You, 6초 전 • Un
41 ]
42
```

**app을 생성했다면 django project 설정파일인 settings.py에
생성한 app을 등록해줘야 합니다!**

urls.py

```
1  from django.contrib import admin  
2  from django.urls import path  
3  
4  urlpatterns = [  
5      path('admin/', admin.site.urls),  
6  ]
```

클라이언트에서 보낸 요청의 Endpoint (URL)에
알맞은 기능을 수행할 수 있도록 길을 안내하는 역할을 합니다.

views.py

```
1  from account.models import Team
2  from post.models import Post
3  from django.http import JsonResponse
4  import hashlib
5  import json
6
7
8  def user_POST(request):
9      try:
10          body_unicode = request.body.decode('utf-8')
11          body = json.loads(body_unicode)
12
13      try:
14          team_name = body['team_name']
15          project_name = body['project_name']
16          new_id = body['id']
17          new_pw = body['pw']
18          hashed_pw = hashlib.sha256(str(new_pw).encode()).hexdigest()
19      except:
20          return JsonResponse({
21              'status': 400,
22              'success': False,
23              'message': '필요한 값이 전달되지 않았습니다'
24          })
25
```

서버의 **logical code**가 **views.py**에 존재합니다.
views.py에 어떤 기능을 하는 코드를 작성하면,
urls.py에서 요청의 URL에 알맞은 **views.py**의 함수로 연결 해줍니다.

models.py

```
1  from django.db import models  
2  
3  
4  class Team(models.Model):  
5      team_id = models.AutoField(primary_key=True)  
6      team_name = models.TextField(null=False)  
7      project_name = models.TextField(null=False)  
8      description = models.TextField(null=True)  
9      account_id = models.TextField(null=False)  
10     account_pw = models.TextField(null=False)  
11     profile_url = models.TextField(null=True)
```

Database Model을 정의합니다.
class 이름은 Database의 Table이 되고,
attribute는 Table의 Column이 됩니다.

```
1  from django.contrib import admin
2  from django.urls import path
3  from django.conf.urls import include
4
5  urlpatterns = [
6      path('admin/', admin.site.urls),
7      path('footprint/', include('footprint.urls')),
8  ]
```

session06/session06/urls.py

**session06 폴더의 urls.py에
footprint app의 urls.py를 연결해줍니다.**

```
1  from . import views  
2  from django.urls import path  
3  
4  urlpatterns = [  
5      path("", views.footprint_GET),  
6      path("send", views.footprint_POST),  
7  ]
```

session06/footprint/urls.py

**footprint 폴더의 urls.py에
footprint views 함수를 연결해줍니다.**

```
1 from django.shortcuts import render
2
3
4 def footprint_GET(request):
5     pass
6
7
8 def footprint_POST(request):
9     pass
```

session06/footprint/views.py

**footprint 폴더의 views.py에
footprint_GET, footprint_POST 함수를
정의해줍니다.**

```
1  from django.db import models
2
3
4      You, 15초 전 | 1 author (You)
5  class Footprint(models.Model):
6      footprint_id = models.AutoField(primary_key=True)
7      receiver = models.TextField(null=False)
8      message = models.TextField(null=False)
9      sent_at = models.DateTimeField(auto_now_add=True, blank=True)
```

session06/footprint/models.py

**footprint 폴더의 models.py에
Footprint Database 모델을 정의해줍니다.**

```
youngkuon-MACBOOK ✚ ➜ ~/w/L/10th/SESSION/session-06/session06 ➜ ⌂ main ↑5 *1 ➤ python manage.py makemigrations
No changes detected

youngkuon-MACBOOK ✚ ➜ ~/w/L/10th/SESSION/session-06/session06 ➜ ⌂ main ↑5 *1 ➤ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, footprint, sessions
Running migrations:
  No migrations to apply.
```

Database Migration

python manage.py makemigrations
python manage.py migrate
명령어를 순차적으로 실행합니다.

```
1  from footprint.models import Footprint
2  from django.http import JsonResponse
3  import json
4
5
6  def footprint_GET(request):
7      footprints = Footprint.objects.filter(receiver="김영권").order_by('-sent_at')
8      messages = []
9      for i in range(len(footprints)):
10          messages.append(footprints[i].message)
11
12      return JsonResponse({
13          'status': 200,
14          'message': 'Footprint 조회 성공',
15          'data': {
16              'messages': messages
17          }
18      })
```

views.py/footprint_GET 함수

```
21     def footprint_POST(request):
22         body_unicode = request.body.decode('utf-8')
23         body = json.loads(body_unicode)
24         Footprint.objects.create(
25             message=body['content'], receiver=body['receiverName'])
26
27     return JsonResponse({
28         'status': 200,
29         'message': '메시지 전송 성공'
30     })
```

views.py/footprint_POST 함수

Cross-Origin Resource Sharing (CORS)

단어가 내포한 의미 그대로, Origin(출처)이 다른 곳으로부터 리소스 공유

이때 ‘출처’라 함은 프로토콜, 서버 Host, 포트번호까지 합친 것을 의미함

Cross-Site Request Forgery (CSRF)

해커가 특정 사이트에 로그인 된 사용자에게 강제로 작업을 수행하게 하는 공격



```
<img src = "http://auction.com/changeUserAcoount?id=youngkwon02&password=내꺼지롱" />
```

```
43 MIDDLEWARE = [
44     # 'django.middleware.security.SecurityMiddleware',
45     'django.contrib.sessions.middleware.SessionMiddleware',
46     'django.middleware.common.CommonMiddleware',
47     # 'django.middleware.csrf.CsrfViewMiddleware',
48     'django.contrib.auth.middleware.AuthenticationMiddleware',
49     'django.contrib.messages.middleware.MessageMiddleware',
50     'django.middleware.clickjacking.XFrameOptionsMiddleware',
51 ]
```

session06/settings.py

개발 환경에서 학습을 위한 서버를 구현하는 중이므로
CSRF 관련 미들웨어의 보안처리를 주석처리 합니다.

```
asset > js > js main.js > .onload
You, 25초 전 | 1 author (You)
1 import { fetchGET, fetchPOST } from "./dataFetching.js";
2
3 // const SERVER_HOST =
4 //   "https://asia-northeast3-likelion-js-practice.cloudfunctions.net/api";
5
6 const SERVER_HOST = "http://localhost:8000";
7
8 const NAME = "김영권";
```

Frontend - main.js

기존 SERVER_HOST를 주석처리하고,
방금 구현한 Django server의 host로 변경합니다

```
50  async function sendFootprint() {  
51    const path = /footprint/send;  
52  
53    const message = document.querySelector(selectors: ".message-content").value;  
54    const sender = document.querySelector(selectors: ".message-sender").value;  
55  
56    const res = await fetchPOST(host: SERVER_HOST, path: path, body: {  
57      content: message,  
58      receiverName: sender,  
59    });  
60  }
```

Frontend - main.js

Footprint 전송 API는 Endpoint가 기존과 다릅니다!!!!
새로운 Endpoint 주소로 변경해주세요

```
69  window.onload = function () {
70      // 프로필 정보를 서버에서 받아와서 DOM을 업데이트 합니다.
71      // const isGetProfileSuccess = getProfileData(NAME);
72
73      // 메시지를 서버에서 받아와서 DOM을 업데이트 합니다.
74      const isGetFootprintSuccess = getFootprint( name: NAME );
75
76      // if (!isGetProfileSuccess || !isGetFootprintSuccess) {
77      //     alert("데이터 로딩에 실패하였습니다.");
78      // }
79
80      const sendButton = document.querySelector( selectors: ".send-btn" );
81      sendButton.addEventListener( type: "click", listener: () => {
82          sendFootprint();
83      });
84  };
```

Frontend - main.js

프로필 조회 관련 API는 구현하지 않았습니다.
프로필 조회 요청 코드는 주석처리합니다.

```
youngkuon-MACBOOK ✚ ➜ ~/w/L/10th/SESSION/session-06/session06 ➜ ↵ main ✘ 6 *1 !1 ➤ pip install django-cors-headers
Collecting django-cors-headers
  Downloading django_cors_headers-3.12.0-py3-none-any.whl (13 kB)
Requirement already satisfied: Django>=3.2 in /Users/jake-alden/workspace/LikeLion_workspace/10th/SESSION/session-06/myvenv/lib/python3.9/site-packages (from django-cors-headers) (4.0.4)
Requirement already satisfied: sqlparse>=0.2.2 in /Users/jake-alden/workspace/LikeLion_workspace/10th/SESSION/session-06/myvenv/lib/python3.9/site-packages (from Django>=3.2->django-cors-headers) (0.4.2)
Requirement already satisfied: asgiref<4,>=3.4.1 in /Users/jake-alden/workspace/LikeLion_workspace/10th/SESSION/session-06/myvenv/lib/python3.9/site-packages (from Django>=3.2->django-cors-headers) (3.5.1)
Installing collected packages: django-cors-headers
Successfully installed django-cors-headers-3.12.0
```

pip install django-cors-headers 명령어로 라이브러리 설치

```
33     INSTALLED_APPS = [
34         'django.contrib.admin',
35         'django.contrib.auth',
36         'django.contrib.contenttypes',
37         'django.contrib.sessions',
38         'django.contrib.messages',
39         'django.contrib.staticfiles',
40         'footprint',
41         'corsheaders',
42     ]
43
44     MIDDLEWARE = [
45         'corsheaders.middleware.CorsMiddleware',
46         # 'django.middleware.security.SecurityMiddleware',
47         'django.contrib.sessions.middleware.SessionMiddleware',
48         'django.middleware.common.CommonMiddleware',
49         # 'django.middleware.csrf.CsrfViewMiddleware',
50         'django.contrib.auth.middleware.AuthenticationMiddleware',
51         'django.contrib.messages.middleware.MessageMiddleware',
52         'django.middleware.clickjacking.XFrameOptionsMiddleware',
53     ]
```

session06/settings.py

corsheaders 등록 및 corsheaders 관련 미들웨어 추가

```
56     CORS_ALLOW_CREDENTIALS = True  
57  
58     CORS_ORIGIN_WHITELIST = [  
59         'http://127.0.0.1:5500',  
60     ]
```

session06/settings.py

CORS 관련 White-List에 Live Server HOST 등록

10TH LION'S FOOTPRINT

메시지 전송 테스트

안녕하세요ㅋㅋ

안녕하세요!!!

SEND YOUR MESSAGE

메시지를 입력하세요.

받는사람

SEND

여러분들이 구현한 서버로부터 데이터를 잘 받아오나요~?

필수 과제

오늘 세션내용 복습, 깃헙에 올린 후 스크린샷을 첨부하여 리드미 작성
완료 후, 각 조별 운영진에게 깃헙 링크 제출

README.md 파일을 6주차 폴더 내에 생성하세요!!

선택 과제

1. account 앱 생성
2. account.models에 사용자 정보 관련 model 정의, migration
3. account.urls 생성 및 프로젝트 root urls와 연결
4. account DB에 데이터 삽입하는 API 구현 및 요청을 통해 삽입
5. account DB에서 데이터를 읽어오는 API 구현 및 요청을 통해 랜더링

어려운 내용 따라 오시느라
고생 많으셨습니다!! 😊