

MSP evangelism 2016

장고: 분노의 개발자

Session2

Django Framework
Tutorial

진행 | 하지윤





Index

- Design Pattern (MTV pattern)
- Start project
- settings.py 수정
- Start App
- Django admin page
- 제작할 서비스 소개 (3줄 요약 서비스)
- Model 구성 및 적용
- admin.py 수정
- Migration 및 app 등록
- Let's git



Design Pattern



Design Pattern

Design Pattern이란?

나보다 똑똑한 사람들이 먼저 겪은 문제를 해결하기 위해 만들어둔 정형화된 해결책

-> S/W의 개발 방법을 공식화한 것!

이를 이용하여 보다 편하게 개발이 가능함



Microsoft



Design Pattern

이런 MTV를 뜻하는게 아니라..





Design Pattern

MTV Pattern

Model

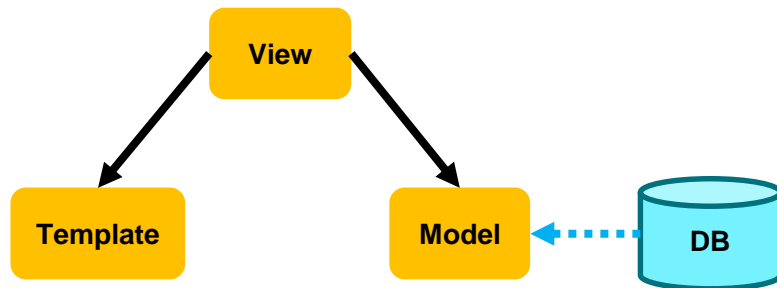
Database

Template

HTML / CSS / Javascript (Front-end)

View

Model과 Template의 연결





Design Pattern

Model Database

예를 들어 개인 블로그를 만든다고 했을 때..

글 제목 (CharField)	글 내용 (TextField)	작성 날짜 (DateTimeField)	수정 날짜 (DateTimeField)
Hello world	Hello World	2016-10-27	null
오늘의 할 일	밥 먹기 / 공부 하기 / 운동 하기	2016-10-30 10:20:13	2016-10-30 10:33:50



Design Pattern

Model Database

예를 들어 개인 블로그를 만든다고 했을 때..

글 제목 (CharField)	글 내용 (TextField)	작성 날짜 (DateTimeField)	수정 날짜 (DateTimeField)
Hello world	Hello World	2016-10-27	null
오늘의 할 일	밥 먹기 / 공부 하기 / 운동 하기	2016-10-30 10:20:13	2016-10-30 10:33:50



Design Pattern

Template

HTML

[Django Girls Blog](#)

published: June 30, 2014, 10:58 p.m.

[My second post](#)

Vestibulum id ligula porta felis euismod semper. Cum justo odio, dapibus ac facilisis in, egestas eget quam. L consectetur ac, vestibulum at eros. Sed posuere conseq mattis consectetur purus sit amet fermentum.

HTML+CSS

[Django Girls Blog](#)

published: June 30, 2014, 10:58 p.m.

[My second post](#)

Vestibulum id ligula porta felis euismod semper. Cum sociis natoq

[Django Girls Blog](#)

published: June 30, 2014, 10:58 p.m.

[My second post](#)

Vestibulum id ligula porta felis euismod semper. Cum sociis natoq sociis natoque penatibus et magnis dis parturient montes, nascetur Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecena

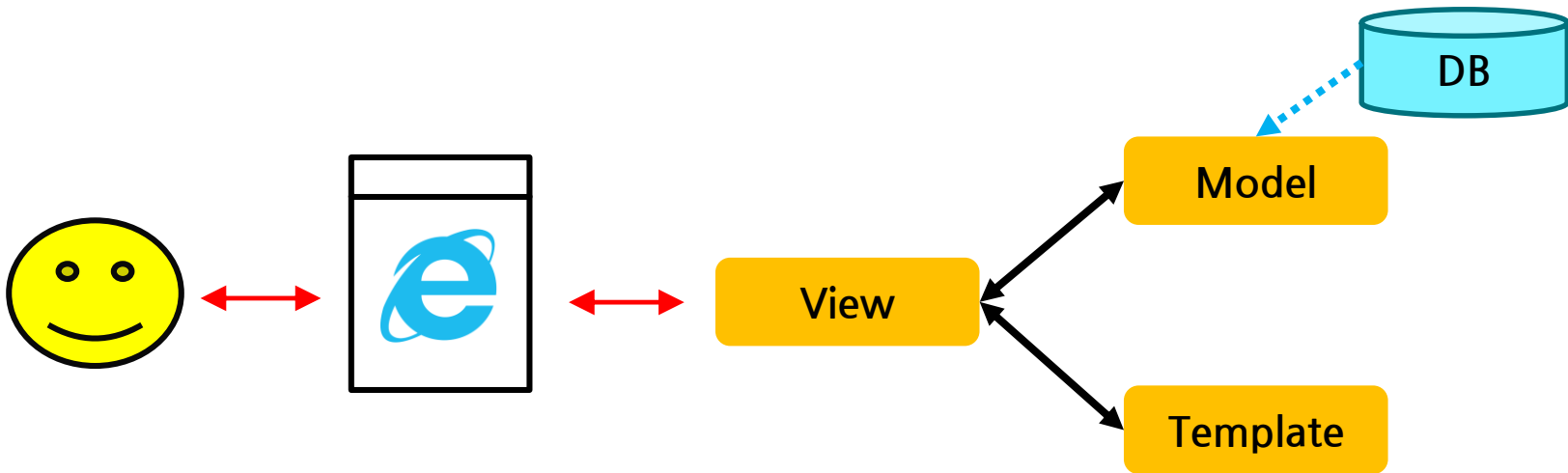
HTML+CSS+JS

www.microsoft.com/ko-kr/



Design Pattern

View Model과 Template를 이어주는 역할





Start Project



Start Project

`django-admin.py startproject today3line`

장고 프로젝트 생성명령

안된다면, `Django-admin startproject today3line` 명령 실행



Start Project

`django-admin.py startproject today3line`

장고 프로젝트 생성명령

프로젝트 이름

안된다면, `Django-admin startproject today3line` 명령 실행



Start Project

생성된 장고 프로젝트의 구성

today3line

manage.py

today3line

settings.py

urls.py

wsgi.py

__init__.py



Start Project

생성된 장고 프로젝트의 구성

[project_name]
manage.py
[project_name]
settings.py
urls.py
wsgi.py
__init__.py



settings.py 수정



settings.py 수정

LANGUAGE_CODE

TIME_ZONE

STATICFILES_DIRS



settings.py 수정

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

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

```
STATICFILES_DIRS = [os.path.join(BASE_DIR, "static"), ]
```



settings.py 수정

프로젝트의 최상위 폴더에서 'static' 폴더를 생성해주고 그 안에 default.png 파일 넣어주기!

today3line

manage.py

today3line

settings.py

urls.py

wsgi.py

__init__.py

static

default.png

default.png 파일 설치 링크

goo.gl/plQ9OQ



Django admin page



Django admin page

python manage.py migrate

- 기본 DB 테이블 생성
- settings.py의 INSTALLED_APPS에 설정된 app에 필요한 table 생성 - admin page관련

python manage.py createsuperuser

- admin page 관리자 계정 생성

python manage.py runserver

- 서버 실행
- 127.0.0.1:8000에 접속하여 서버 동작 페이지 확인
- 127.0.0.1:8000/admin에서 createsuperuser에서 만들어준 관리자 계정으로 접속해보기



Start App



Start App

```
python manage.py startapp main
```

새로운 application 생성 명령



Start App

`python manage.py startapp` **main**

새로운 application 생성 명령

application 이름



Start App

변경된 장고 프로젝트의 구성

today3line

manage.py

today3line

(생략)

static

default.png

main

models.py

views.py

admin.py

apps.py

__init__.py



Start App

변경된 장고 프로젝트의 구성

today3line

manage.py

today3line

(생략)

static

default.png

[app_name]

models.py

views.py

admin.py

apps.py

__init__.py



제작할 서비스 소개

- 네이버캐스트 3줄 요약 서비스



네이버캐스트 3줄 요약 서비스

navercast.naver.com/

상위 24개의 글을 가져오고 분석해서 3줄로 요약해주는 서비스

대표 이미지, 글 제목, 세 줄 요약된 내용, 원문으로 갈 수 있는 링크 정도로 모델 구성



Microsoft



Model 구성 및 적용



Model 구성 및 적용

SQL (Structured Query Language)

DB의 데이터를 관리하기 위해 설계된 특수 목적의 프로그래밍 언어

DB 테이블 생성

```
CREATE TABLE Blog (upload_date DATE, title CHAR(10), body VARCHAR2(100));
```

데이터 삽입

```
INSERT INTO Blog(upload_date, title, body)  
VALUES (2016-11-27-14:00:01, '분노의 개발자', '안녕하세요 여러분');
```

데이터 검색

```
SELECT title FROM Blog WHERE title='분노의 개발자' ;
```



Model 구성 및 적용

ORM (Object Relational Mapping)

- 데이터를 모두 객체로 보는 개념
- DB 테이블과 객체를 서로에 대해 대응
- SQL 작성 없이 DB 테이블 데이터를 OOP 객체로 전달받을 수 있는 것

ORM이 사용되는 이유

- SQL 작성 없이 기존 프로그래밍 언어를 사용하여 DB 개발을 편하게 해 줌



Model 구성 및 적용

ParsedData

title (TextField)	url (TextField)	image_url (TextField)	line1 (TextField)	line2 (TextField)	line3 (TextField)



Model 구성 및 적용

ParsedData *Primary Key*

title (TextField)	url (TextField)	image_url (TextField)	line1 (TextField)	line2 (TextField)	line3 (TextField)



Model 구성 및 적용

(today3line/main/models.py)

```
class ParsedData(models.Model):  
    url  
    title  
    image_url  
    line1  
    line2  
    line3
```



Model 구성 및 적용

(today3line/main/models.py)

```
class ParsedData(models.Model):  
    url = models.TextField(primary_key = True,)  
    title = models.TextField(default = '')  
    image_url = models.TextField(default = '/static/default.png')  
    line1  
    line2  
    line3
```



Model 구성 및 적용

(today3line/main/models.py)

```
class ParsedData(models.Model):  
    url = models.TextField(primary_key = True,)  
    title = models.TextField(default = '')  
    image_url = models.TextField(default = '/static/default.png')  
    line1 = models.TextField(default = '')  
    line2 = models.TextField(default = '')  
    line3 = models.TextField(default = '')
```



Model 구성 및 적용

(today3line/main/models.py)

```
class ParsedData(models.Model):  
    url = models.TextField(primary_key = True,)  
    title = models.TextField(default = '')  
    image_url = models.TextField(default = '/static/default.png')  
    line1 = models.TextField(default = '')  
    line2 = models.TextField(default = '')  
    line3 = models.TextField(default = '')  
  
    def __str__(self):  
        return str(self.url)
```



admin.py 수정



admin.py 수정

(today3line/main/admin.py)

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

admin page에서 ParsedData 테이블에 저장된 데이터 관리 가능



Migration 및 app 등록



Migration 및 app 등록

settings.py - INSTALLED_APPS 항목 추가

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    ''main'',  
]
```



Migration 및 app 등록

`python manage.py makemigrations`

model의 변경사항을 파일로 생성하는 명령



Migration 및 app 등록

python manage.py migrate

생성된 model의 변경내역을 DB에 반영하는 명령



Migration 및 app 등록

`python manage.py runserver`

- 127.0.0.1:8000/admin 접속 후 관리자 계정으로 로그인
- admin page에서 추가된 app관리 페이지 확인



Let's git



Let's git

```
git add .
```

```
git commit -m "make model and register admin"
```

```
git push
```