

Django Shell - Model

- shell 이란?
- Model로 Database 사용

Shell - Model

1. shell 이란? (1/2)

Django Code를 구현하기 전 Code Test를 진행 할 수 있도록 지원해준다.

Django Code를 간단하게 작성하고 실행하여 결과를 확인함으로써 실제 Python Script 파일을 작성 할 때 Code의 결과를 명시적으로 확인하고 이해하면서 작성 할 수 있게 된다.

기본적으로는 CMD에서 python 명령어를 입력하여 동작하는 것과 차이가 없지만, Django Code가 동작 하기 위한 기본 Module과 Setting들이 활성화 되어 있다.

Shell - Model

1. shell 이란? (2/2)

```
(VirPy) D:\workspace\myWeb> manage.py shell
```

```
Python 3.5.3 (v3.5.3:1880cb95a742, Jan 16 2017, 16:02:32) [MSC v.1900 64 bit (AMD64)]  
on win32
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
(InteractiveConsole)
```

```
>>>
```

Shell - Model

2. Model로 Database 사용 (1/19)

DB 검색

```
>>> from sub_app.models import Research
>>>
>>> search = Research.objects.all()
>>> search
[<Research: Research object>, <Research: Research object>, ...]
>>>
```

Shell - Model

2. Model로 Database 사용 (2/19)

DB 검색

```
>>> from sub_app.models import Research
>>>
>>> search = Research.objects.get(id=10)
>>> search
<Research: Research object>
>>>
```

Shell - Model

2. Model로 Database 사용 (3/19)

DB 검색

```
>>> from sub_app.models import Research
>>>
>>> search = Research.objects.filter(gender='남자')
>>> search
[<Research: Research object>, <Research: Research object>, ...]
>>>
```

Shell - Model

2. Model로 Database 사용 (4/19)

DB 검색

```
>>> from sub_app.models import Research
>>>
>>> search = Research.objects.exclude(id=2)
>>> search
[<Research: Research object>, <Research: Research object>, ...]
>>>
```

Shell - Model

2. Model로 Database 사용 (5/19)

DB 검색

```
>>> from sub_app.models import Research
>>>
>>> search = Research.objects.exclude(id=2)
>>> search
[<Research: Research object>, <Research: Research object>, ...]
>>>
```


Shell - Model

2. Model로 Database 사용 (6/19)

DB 검색

```
>>> from sub_app.models import Research
>>>
>>> search = Research.objects.all().values('main_title', ...)
>>> search
[{'main_title': 'value1', ...}, {'main_title': 'value2', ...}, ...]
>>>
```

Shell - Model

2. Model로 Database 사용 (7/19)

DB 검색

```
>>> from sub_app.models import Research
>>>
>>> search = Research.objects.values('main_title', 'gender', ...).get(id=2)
>>> search
{'gender': '여자', 'main_title': 'value2', ...}
>>>
```

Shell - Model

2. Model로 Database 사용 (8/19)

DB 검색

```
>>> from sub_app.models import Research
>>>
>>> search = Research.objects.all().values('main_title', ...).order_by('-main_title')
>>> search
[{'main_title': 'value2', ...}, {'main_title': 'value1', ...}, ...]
>>>
```

Shell - Model

2. Model로 Database 사용 (9/19)

DB 검색

column__gt : >
column__gte : >=
column__lt : <
column__lte : <=
column__startswith : LIKE 절, 대소문자 구분 0, 특정 문자열로 시작하는 자료 검색
column__istartswith : LIKE 절, 대소문자 구분 x, 특정 문자열로 시작하는 자료 검색
column__endswith : LIKE 절, 대소문자 구분 0, 특정 문자열로 끝나는 자료 검색
column__iendswith : LIKE 절, 대소문자 구분 x, 특정 문자열로 끝나는 자료 검색
column__contains : LIKE 절, 대소문자 구분 0, 특정 문자열이 들어가는 자료 검색
column__icontains : LIKE 절, 대소문자 구분 x, 특정 문자열이 들어가는 자료 검색
column__range : BETWEEN 절, 튜플로 시작/끝 지정, 값 또는 날짜의 범위 지정
column__isnull : IS NULL 문, True/False로 사용

Shell - Model

2. Model로 Database 사용 (10/19)

DB 추가

```
>>> from sub_app.models import Research
>>>
>>> insert = Research()
>>> insert.main_title = '제목1'
>>> insert.gender = '남자'
...
...
>>> insert.save()
>>>
```

Shell - Model

2. Model로 Database 사용 (11/19)

DB 추가

```
>>> from sub_app.models import Research
>>>
>>> insert = Research(id=10)
>>> insert.main_title = '제목1'
>>> insert.gender = '남자'
...
...
>>> insert.save()
>>>
```

Shell - Model

2. Model로 Database 사용 (12/19)

DB 추가

```
>>> from sub_app.models import Research
>>>
>>> insert = Research(id=10, main_title='제목1', gender='남자', ...)
>>> insert.save()
>>>
```

Shell - Model

2. Model로 Database 사용 (13/19)

DB 추가

```
>>> from sub_app.models import Research
>>>
>>> Research(main_title='제목1', gender='남자', ...).save()
>>>
```


Shell - Model

2. Model로 Database 사용 (14/19)

DB 수정

```
>>> from sub_app.models import Research
>>>
>>> update = Research.objects.get(id=1)
>>> update.main_title = '제목 수정'
>>> update.answer1 = '응답 수정'
...
...
>>> update.save()
```

Shell - Model

2. Model로 Database 사용 (15/19)

DB 수정

```
>>> from sub_app.models import Research
>>>
>>> update = Research.objects.get(id=1)
>>> update.main_title = '제목 수정'
>>> update.answer1 = '응답 수정'
...
...
>>> update.save(update_fields=('main_title', 'answer1', ...))
```

Shell - Model

2. Model로 Database 사용 (16/19)

DB 수정

```
>>> from sub_app.models import Research
>>>
>>> update = Research.objects.filter(location='서울').update('location'='서울시')
>>>
```

Shell - Model

2. Model로 Database 사용 (17/19)

DB 삭제

```
>>> from sub_app.models import Research
>>>
>>> update = Research.objects.get(id=10).delete()
>>>
```

Shell - Model

2. Model로 Database 사용 (18/19)

DB 삭제

```
>>> from sub_app.models import Research
>>>
>>> update = Research.objects.filter(research_date='2017-10-26').delete()
>>>
```

Shell - Model

2. Model로 Database 사용 (19/19)

DB 삭제

```
>>> from sub_app.models import Research
>>>
>>> update = Research.objects.all().delete()
>>>
```