

[Atom 설치]

- <https://atom.io>
- File-Settings-Package 메뉴에서 다음 패키지 설치 : **autocomplete-python, script**

0. Django & MTV

- 브라우저를 통한 request -> HTTP Server -> urls.py (URL Dispatcher) : web server와 urls.py 를 연결해주는 것이 wsgi (예전 cgi)
- urls.py 는 URL 을 파싱(해석)하여 View 호출
- Views는 바로 response를 돌려주거나,
- Model을 통해 DB의 데이터를 가져와 Template(html 등)를 렌더링하여 response로 돌려줌
- 해당 response를 Browser가 해석하여 우리가 결과를 볼 수 있게 만들어줌

1. 장고 프로젝트 폴더 및 Python 가상환경 생성 (cmd 관리자권한으로 실행)

cd .. 명령어로 c drive까지 나가기

mkdir work_django -> work_django 폴더가 생깁니다.

cd work_django

mkdir django_mdl -> django_mdl 폴더가 생깁니다.

cd django_mdl

pip install virtualenv==16.7.7 -> 가상 환경을 만들어주는 라이브러리입니다.

virtualenv django_env -> django_env 라는 이름으로 가상 환경을 새로이 만듭니다.

django_env\Scripts\activate -> Scripts 폴더 내의 activate 파일을 실행해 가상 환경을 활성화합니다.

`pip install django==2.2.6` -> 가상 환경에는 최소의 라이브러리만 존재하므로 장고를 설치해줍니다.

2. 장고 프로젝트 생성 (C:\work_django\django_mldl)

: project는 하나의 웹사이트에 해당합니다.

`django-admin startproject site_1` -> site_1 폴더가 생깁니다.

`cd site_1`

Atom에서 프로젝트 폴더 열기

: File -> Add Project Folder... 에서 django_mldl > **site_1** 폴더 선택하기 (django_mldl 폴더가 아닙니다.)

- C:\work_django\django_mldl\site_1\site_1__init__.py : site_1이 하나의 python module임을 뜻함
- C:\work_django\django_mldl\site_1\site_1\settings.py : django project에 대한 세팅
- C:\work_django\django_mldl\site_1\site_1\urls.py : django project의 URL 패턴 정의 (router)

C:\work_django\django_mldl\site_1\site_1 -> **settings.py** 파일 수정

LANGUAGE_CODE = 'en-us' -> '**ko-kr**'

TIME_ZONE = 'UTC' -> '**Asia/Seoul**'

STATIC_ROOT = os.path.join(BASE_DIR, 'static')

3. 장고 app 생성 (C:\work_django\django_mldl\site_1)

: app 은 하나의 웹사이트 내에 있는 기능들에 해당합니다. (회원가입 기능, 상품 관련 기능 등)

python manage.py startapp lotto -> lotto 폴더가 생깁니다.

C:\work_django\django_mdl\site_1\site_1 -> **settings.py** 파일 수정

```
INSTALLED_APPS = [
```

```
    ... ,
```

```
    'lotto',
```

```
]
```

4. 장고 프로젝트 Server 실행 & 웹브라우저에서 확인

python manage.py runserver (포트번호 변경을 원할 시 : **python manage.py runserver 8888**)

브라우저에서 **127.0.0.1:8000** 접속

5. urls.py & views.py 이해하기

5-1) C:\work_django\django_mdl\site_1\site_1 -> **urls.py** 파일 수정

```
from lotto import views
```

```
urlpatterns = [
```

```
path('admin/', admin.site.urls),  
  
path("", views.index), # lotto > views.py 파일의 index() 함수 호출  
  
]
```

5-2) C:\work_django\django_mdl\site_1\lotto -> **views.py** 파일 수정

```
from django.http import HttpResponse
```

```
def index(request):  
  
    return HttpResponse('<h1>Hello, world!</h1>')
```

5-3) 웹브라우저에서 새로고침하여 변경 사항 확인

5-4) C:\work_django\django_mdl\site_1\site_1 -> **urls.py** 파일 수정

```
urlpatterns = [  
  
    path('admin/', admin.site.urls),  
  
    path("", views.index), # lotto > views.py 파일의 index() 함수 호출  
  
    path('hello/', views.hello, name='hello_main'), # lotto > views.py 파일의 hello() 함수 호출  
  
]
```

5-5) C:\work_django\django_mdl\site_1\lotto -> **views.py** 파일 수정

```
def hello(request):  
  
    return HttpResponse("<h1 style='color:red;'>Hello, world!</h1>")
```

5-6) 웹브라우저에서 <http://127.0.0.1:8000/hello/> 에 접속하여 변경 사항 확인

-> urls.py의 **path('hello/',** 에서 'hello/' 앞에는 'http://127.0.0.1:8000/'까지가 생략되어있는 것입니다.

6. models.py 변경하기 (DB Schema 변경하기)

* PDF 파일의 58 page 슬라이드에서 DB 테이블 확인

- Django 에서는 DB와 소통할 때 **SQL**을 직접 활용하는 대신 **Model**이라는 것을 활용

- SQL을 직접 작성하는 것보다 훨씬 더 쉽게 DB를 다룰 수 있으며, 이를 **ORM(Object-Relational Mapping)**을 활용한다고 말함.

6-1) C:\work_django\django_mdl\site_1\lotto -> **models.py** 파일 수정

```
from django.db import models
```

```
from django.utils import timezone
```

```
import random
```

```
# Create your models here.
```

```
class GuessNumbers(models.Model):
```

```
    name = models.CharField(max_length=24) # 로또 번호 리스트의 이름
```

```
    text = models.CharField(max_length=255) # 로또 번호 리스트에 대한 설명
```

```
    lottos = models.CharField(max_length=255,
```

```
        default='[1, 2, 3, 4, 5, 6]') # 로또 번호들이 담길 str
```

```
    num_lotto = models.IntegerField(default=5) # 6개 번호 set의 갯수
```

```
update_date = models.DateTimeField()
```

```
def generate(self): # 로또 번호를 자동으로 생성
```

```
    self.lottos = ""
```

```
    origin = list(range(1,46)) # 1~46의 숫자 리스트
```

```
    # 6개 번호 set 갯수만큼 1~46 뒤섞은 후 앞의 6개 골라내어 sorting
```

```
    for _ in range(0, self.num_lotto):
```

```
        random.shuffle(origin)
```

```
        guess = origin[:6]
```

```
        guess.sort()
```

```
        self.lottos += str(guess) + '\n' # 로또 번호 str에 6개 번호 set 추가
```

```
    self.update_date = timezone.now()
```

```
    self.save() # GuessNumbers object를 DB에 저장
```

```
def __str__(self): # Admin page에서 display되는 텍스트에 대한 변경
```

```
    return "pk {} : {} - {}".format(self.pk, self.name, self.text) # pk는 자동생성됨
```

6-2) models.py를 구성/변경하고 난 후에는 **DB에 반영**해줘야 함 (실제 물리적인 DB 파일 : **db.sqlite3**)

- **Migration** : models.py 에서의 변화를 **DB Schema에 반영**해주는 과정

- python manage.py **makemigrations** : models.py 에서의 **변화를 모아 migration file로 구성** (**commit**)

- python manage.py **migrate** : 구성된 migration file을 바탕으로 **실제 DB Schema 변경** (**push**)

(cmd @ C:\work_django\django_mdl\site_1)

Ctrl + C 로 서버 종료

python manage.py makemigrations

python manage.py migrate

7. admin site에 Model 객체 (DB에서의 record/row에 해당) 등록하기

7-1) admin site 접속하기

python manage.py runserver

브라우저에서 **127.0.0.1:8000/admin** 접속

7-2) admin site를 위한 사용자이름 & 비밀번호 생성하기

Ctrl + C 로 서버 종료

python manage.py createsuperuser

Username / Email / Password 입력하기 (Password는 입력 시 화면으로 출력되지 않음)

7-3) admin site 접속하기

python manage.py runserver

브라우저에서 **127.0.0.1:8000/admin** 접속하고 로그인하기

-> GuessNumbers 항목이 아예 나타나지 않음

7-4) admin site에 만들어두었던 Model 클래스 등록하기

* from .models import GuessNumbers : **현재 파일과 동일한 폴더**에 위치한 models.py 파일로부터 GuessNumbers class를 Import

* 위 import 문에서 "."은 현재 파일이 위치한 폴더 자체를 의미함.

from .models import ~ == from lotto.models import ~

C:\work_django\django_mdl\site_1\lotto -> **admin.py** 파일 수정

```
from django.contrib import admin
```

```
from .models import GuessNumbers
```

```
# from lotto.models import GuessNumbers
```

```
# Register your models here.
```

```
admin.site.register(GuessNumbers)
```

-> admin page 새로그침

7-5) admin site에 Model 객체 추가하기

- admin page에서 **Guess numberss** 클릭

- Guess numbers 추가

- **Name** : 번호 세트 1번 / **Text** : 1등 되라 압! / **Update date 날짜 & 시간** : 오늘 및 현재 클릭

- 저장 결과 확인

8. 장고 Test code 작성하기

: models.py 의 **generate()** 함수를 테스트하는 코드 작성

8-1) Test case 작성하기

C:\work_django\django_mdl\site_1\lotto -> **tests.py** 파일 수정

```
from django.test import TestCase
```

```
from .models import GuessNumbers
```

```
# Create your tests here.
```

```
class GuessNumbersTestCase(TestCase):
```

```
    def test_generate(self):
```

```
        g = GuessNumbers(name='Test numbers', text='selected numbers')
```

```
        g.generate() # GuessNumbers 클래스의 generate 함수를 실행
```

```
        print(g.update_date)
```

```
        print(g.lottos)
```

```
        # 실제 Test case (OK or FAILED)
```

```
        # default로 6개 숫자 x 5set = 30개의 숫자가 생성됨을 확인
```

```
        self.assertTrue(len(g.lottos) > 20)
```

8-2) Test case 실행하기

(cmd @ C:\work_django\django_mdl\site_1)

```
python manage.py test
```

9. Shell을 활용해 만들어 둔 장고 프로젝트 기능 활용하기

(cmd @ C:\work_django\django_mdl\site_1)

```
python manage.py shell
```

```
from lotto.models import GuessNumbers
```

```
GuessNumbers.objects.all()
```

```
GuessNumbers.objects.get(name='번호 세트 1번')
```

```
guess_object = GuessNumbers()
```

```
guess_object.generate()
```

```
guess_object.lottos
```

```
exit()
```

10. views.py와 Template 연동하기

10-1) C:\work_django\django_mdl\site_1\site_1 -> **urls.py** 파일 수정

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    # path("", views.index),  
    path('hello/', views.hello, name='hello_main'), # lotto > views.py 파일의 hello 함수 호출  
    path('lotto/', views.index, name='index'), # lotto > views.py 파일의 index 함수 호출  
]
```

10-2) C:\work_django\django_mdl\site_1\lotto -> **views.py** 파일 수정

def index(request):

브라우저로부터 넘어온 request를 그대로 template('default.html')에게 전달

{} 에는 추가로 함께 전달하려는 object들을 dict로 넣어줄 수 있음

return render(request, 'lotto/default.html', {})

10-3) TemplateDoesNotExist 에러 체크

python manage.py runserver

브라우저에서 **127.0.0.1:8000/lotto** 접속

10-4) Template 파일 만들어주기 ('lotto/default.html')

* Atom에서 'lotto' 폴더에 우클릭 후 'New File'을 클릭한 다음 파일명을 포함한 경로를 바로 적어주면 빠르게 폴더들을 한번에 생성할 수 있음 (슬라이드 73page 참고)

- C:\work_django\django_mldl\site_1\lotto <- 폴더 내에 **templates** 폴더 생성

- C:\work_django\django_mldl\site_1\lotto\templates <- 폴더 내에 **lotto** 폴더 생성

- C:\work_django\django_mldl\site_1\lotto\templates\lotto <- 폴더 내에 **default.html** 파일 생성

-> 나중에 서버에 웹 서비스를 올리게 되면, 전체 template 파일들이 한번에 모이게 됨.

그 때 lotto 라는 폴더로 따로 묶어주기 위한 과정

default.html 파일에 아래 내용 작성

<!DOCTYPE html>

<html lang="ko">

<head>

<title>My Little Lotto</title>

```

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">

<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">

<link href="//fonts.googleapis.com/css?family=Space+Mono" rel="stylesheet">

</head>


<body>

  <div class="page-header">

    <h1>Selected Lotto Numbers</h1>

  </div>

  <div class="container lotto">

    <h2>Good luck!</h2>

    <p> last update:2030/12/31</p>

    <p> 1, 10, 19, 15, 20, 35 </p>

  </div>

</body>

</html>

```

브라우저에서 새로고침하여 결과 확인 (@ 127.0.0.1:8000/lotto)

10-5) **css 파일 연결**하여 디자인 적용하기 (Static 파일 연결)

*** 파일 위치 참고 @ 슬라이드 76 page**

- C:\work_django\django_mldl\site_1\lotto <- 폴더 내에 **static** 폴더 생성
- C:\work_django\django_mldl\site_1\lotto\static <- 폴더 내에 **css** 폴더 생성
- C:\work_django\django_mldl\site_1\lotto\static\css <- 폴더 내에 **lotto.css** 파일 생성

lotto.css 파일에 아래 내용 작성

```
.page-header {  
    background-color: #652596;  
    margin-top: 0;  
    padding: 20px 20px 20px 40px;  
    font-family: 'Space Mono', monospace;  
}
```

```
.page-header h1 {  
    color: #FFFFFF;  
}
```

```
.container {  
    font-family: 'Space Mono', monospace;  
}
```

```
.container h2 {  
    color: #b9f442;  
}
```

C:\work_django\django_mld\site_1\lotto\templates\lotto 의 **default.html** 파일 head 태그 내부 수정

```
<!DOCTYPE html>  
  
{% load staticfiles %}  
  
<html lang="ko">  
  
<head>  
  
    <title>My Little Lotto</title>  
  
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">
<link href="//fonts.googleapis.com/css?family=Space+Mono" rel="stylesheet">
<link rel="stylesheet" href="{% static 'css/lotto.css' %}">
</head>
```

10-6) Static file 모으기 (Gathering static files)

(cmd @ C:\work_django\django_mldl\site_1)

Ctrl + C 로 서버 종료

python manage.py collectstatic

python manage.py runserver

브라우저에서 127.0.0.1:8000/lotto 접속

11. views.py, Template, Model 연동하기

11-1) C:\work_django\django_mldl\site_1\lotto -> views.py 파일 수정

```
from .models import GuessNumbers
```

```
def index(request):
```

```
    lottos = GuessNumbers.objects.all() # DB에 저장된 GuessNumbers 객체 모두를 가져온다.
```

```
    # 브라우저로부터 넘어온 request를 그대로 template('default.html')에게 전달
```

```
    # {} 에는 추가로 함께 전달하려는 object들을 dict로 넣어줄 수 있음
```

```
    return render(request, 'lotto/default.html', {'lottos':lottos})
```

11-2) C:\work_django\django_mdl\site_1\lotto\templates\lotto 의 **default.html** 파일 수정 & 웹사이트 체크

```
<body>

<div class="page-header">

<h1>Selected Lotto Numbers</h1>

</div>

<div class="container lotto">

    <p>{{ lottos }}</p>

    <h2> Good luck!</h2>

    <p> last update:2030/12/31</p>

    <p> 1, 10, 19, 15, 20, 35 </p>

</div>

</body>
```

11-3) C:\work_django\django_mdl\site_1\lotto\templates\lotto 의 **default.html** 파일 수정 & 웹사이트 체크

* | **linebreaksbr** 은 해당 데이터가 **str**일 경우 **str** 내의 줄바꿈 기호("\n")를 만날 때마다 줄을 바꿔준다.

```
<body>

<div class="page-header">

<h1>Selected Lotto Numbers</h1>

</div>

<div class="container lotto">

    {% for lotto in lottos %}

        <h2>{{ lotto.text }}</h2>

        <p>{{ lotto.update_date }}</p>

        <p>{{ lotto.lottos | linebreaksbr }}</p>

    {% endfor %}
```

</div>

</body>

12. admin 대신 사용자로부터 데이터를 입력받는 Form 만들기

12-1) C:\work_django\django_mdl\site_1\lotto <- 폴더 내에 **forms.py** 파일 만들기

12-2) **forms.py** 파일 수정

```
from django import forms
```

```
from .models import GuessNumbers
```

```
# Django에서 제공하는 ModelForm을 활용해 form 구성
```

```
class PostForm(forms.ModelForm):
```

```
# Form을 통해 받아들여야할 데이터가 명시되어 있는 메타 데이터 (DB 테이블을 연결)
```

```
class Meta:
```

```
    model = GuessNumbers
```

```
    fields = ('name', 'text',) # 사용자로부터 form 통해 입력받을 데이터
```

12-3) C:\work_django\django_mdl\site_1\site_1 -> **urls.py** 파일 수정

```
urlpatterns = [
```



```

path('admin/', admin.site.urls),

# path('', views.index),

path('hello/', views.hello, name='hello_main'), # lotto > views.py 파일의 hello 함수 호출

path('lotto/', views.index, name='index'), # lotto > views.py 파일의 index 함수 호출

path('lotto/new', views.post, name = "new_lotto"),

]

```

12-4) C:\work_django\django_mdl\site_1\lotto -> **views.py** 파일 수정

```

from .forms import PostForm

```

```

def post(request):

```

```

    form = PostForm() # 상단 from .forms import PostForm 추가

```

```

    return render(request, "lotto/form.html", {"form": form})

```

12-5) C:\work_django\django_mdl\site_1\lotto\templates\lotto 에 **form.html** 파일 생성

* **CSRF 공격 (Cross Site Request Forgery)** : 인터넷 사용자(희생자)가 자신의 의지와는 무관하게 공격자가 의도한 행위(수정, 삭제, 등록 등)를 특정 웹사이트에 요청하게 만드는 공격

* 클라이언트에서 해당 페이지를 접속하게 되면 Django에서 자동으로 **csrf_token**을 클라이언트로 보내어 **cookie**에 저장시키고, POST로 전송할 때 **cookie**의 **csrf_token** 이 함께 전송되어 인증하는 방식

```

<!DOCTYPE html>

```

```

{% load staticfiles %}

```

```

<html lang="ko">

```

```

<head>

```

```

    <title>My Little Lotto</title>

```

```
<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">

<link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">

<link href="//fonts.googleapis.com/css?family=Space+Mono" rel="stylesheet">

<link rel="stylesheet" href="{% static 'css/lotto.css' %}">

</head>


<body>

<div class="page-header">

  <h1>Generate your new lotto numbers!</h1>

</div>

<div class="container lotto">

  <form method="POST" class="post-form">

    {% csrf_token %} <!-- CSRF (Cross Site Request Forgery) 공격 방지를 위한 Token 생성 -->

    {{ form.as_p }} <!-- form을 p 태그로 감싸서 출력 -->

    <button type="submit" class="save btn btn-default">Save</button>

  </form>

</div>

</body>

</html>
```

12-6) <http://127.0.0.1:8000/lotto/new> 에 접속하여 form 체크

-> Ctrl+Shift+i 로 개발자도구 열고서 **CSRF & input** 태그 체크 후 다음 단계로 진행 (PDF 91 page)

13. Form을 통해 사용자로부터 입력받은 데이터를 실제로 POST 처리하기 (DB 반영)

13-1) C:\work_django\django_mdl\site_1\lotto -> **views.py** 파일 수정

```
from django.shortcuts import render, redirect
```

```
def post(request):
```

```
    if request.method == "POST":
```

```
        # print(request.POST) # 주석을 풀면 새로운 로또 번호 생성 후 cmd에서 이 값을 확인할 수 있음
```

```
        # print(request.method) # 주석을 풀면 새로운 로또 번호 생성 후 cmd에서 이 값을 확인할 수 있음
```

```
        # 사용자로부터 넘겨져 온 POST 요청 데이터를 담아 PostForm 객체 생성
```

```
        form = PostForm(request.POST) # filled form
```

```
        if form.is_valid():
```

```
            lotto = form.save(commit = False) # DB 저장은 아래 generate 함수의 .save()로 처리
```

```
            lotto.generate()
```

```
            return redirect('index') # urls.py의 name='index'에 해당
```

```
        # -> 상단 from django.shortcuts import render, redirect 수정
```

```
    else:
```

```
        form = PostForm() # empty form
```

```
        return render(request, "lotto/form.html", {"form": form})
```

13-2) http://127.0.0.1:8000/**lotto/new** 새로고침 후 대강 값 써넣고 **Save** 진행

Name : Form 입력

Text : Form 입력

14. 로또 번호에 대한 detail 페이지 추가하기

(url 입력을 통해 view 함수에 흘러들어갈 인자를 직접 넘겨받기)

14-1) C:\work_django\django_mdl\site_1\site_1 -> **urls.py** 파일 수정

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    # path("", views.index),  
    path('hello/', views.hello, name='hello_main'), # lotto > views.py 파일의 hello 함수 호출  
    path('lotto/', views.index, name='index'), # lotto > views.py 파일의 index 함수 호출  
    path('lotto/new', views.post, name = "new_lotto"),  
    # lottokey는 detail() 함수의 parameter name  
    path('lotto/<int:lottokey>/detail', views.detail, name='detail'),  
]
```

14-2) C:\work_django\django_mdl\site_1\lotto -> **views.py** 파일 수정

```
def detail(request, lottokey):  
    lotto = GuessNumbers.objects.get(pk = lottokey) # primary key  
    return render(request, "lotto/detail.html", {"lotto": lotto})
```

14-3) Template 파일 만들어주기 ('lotto/detail.html')

- C:\work_django\django_mdl\site_1\lotto\templates\lotto <- 폴더 내에 **detail.html** 파일 생성 & 아래 내용 작성

```
<!DOCTYPE html>
```

```
{% load staticfiles %}
```

```
<html lang="ko">
```

```
<head>
```

```
    <title>My Little Lotto</title>
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

```
    <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
```

```
    <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">
```

```
    <link href="//fonts.googleapis.com/css?family=Space+Mono" rel="stylesheet">
```

```
    <link rel="stylesheet" href="{% static 'css/lotto.css' %}">
```

```
</head>
```

```
<body>
```

```
    <div class="page-header">
```

```
        <h1>Information of the selected lotto num list</h1>
```

```
    </div>
```

```
    <div class="container lotto">
```

```
        <h2>{{ lotto.text }}</h2>
```

```
        <p> by {{ lotto.name }}</p>
```

```
        <p> {{ lotto.update_date }}</p>
```

```
        <p> {{ lotto.lottos | linebreaksbr }}</p>
```

```
    </div>
```

```
</body>
```

```
</html>
```

14-4) 이미 올라온 데이터에 대한 **Primary key** 체크

python manage.py runserver

- 브라우저에서 **http://127.0.0.1:8000/admin** 접속
- 'Guess numberss' 클릭
- '**pk**' 값 체크

14-5) 위 **admin page**에서 확인한 **Primary key** 값을 기준으로 상세 페이지 확인

브라우저에서 아래 URL 입력하여 상세페이지 접속

127.0.0.1:8000/lotto/<Primary key>/detail

14-6) 메인 페이지(default.html)에 상세페이지 링크 및 로또 번호 Form 페이지 링크 추가하기

- C:\work_django\django_mldl\site_1\lotto\templates\lotto 의 **default.html** 파일 수정 & 웹사이트 체크

<body>

<div class="page-header">

<h1>

Selected Lotto Numbers

</h1>

</div>

<div class="container lotto">

```

{% for lotto in lottos %}

    <!-- urls.py의 name='detail' 과 lottokey 에 해당 -->

    <h2><a href="{% url 'detail' lottokey=lotto.pk %}">{{lotto.text}}</a></h2>

    <p>{{ lotto.update_date }}</p>

    <p>{{ lotto.lottos | linebreaksbr }}</p>

{% endfor %}

</div>

</body>

</html>

```

14-7) 메인 페이지에서 수정 사항 확인

127.0.0.1:8000/lotto 접속

15. 장고 서비스 배포하기 (with Python anywhere) : 추후 진행할 [5. Django advanced] 에서 보다 확장된 배포 실습을 진행할 예정입니다.

15-1) Github 에 소스코드 업로드

* git bash 에서 403 에러 (permission 관련 에러) 발생 시 : git bash 터미널 계정 변경 방법 @ <http://j.mp/2CN1TtT>

- <https://github.com> 접속 후 로그인

- 우측 상단 **New repository** 클릭 후 새로운 저장소 만들기 (저장소 이름 : **django_lotto**)

- <https://git-scm.com/downloads> 접속하여 Git for windows 설치
- **Git bash** 실행
- **git config --list** 입력하여 **user.name & user.email** 입력 여부 확인
- 입력되지 않았을 경우 아래 2줄 실행하여 새로이 입력 후 **git config --list** 입력하여 다시 체크
- **git config --global user.name 사용자명**
- **git config --global user.email 메일주소**
- **cd ..** 및 **cd** 폴더명 명령어로 다음 경로까지 이동

C:\work_django\django_mldl\site_1

- 생성된 github repository 에 로컬 저장소 (로컬 폴더) 내의 파일 업로드

git init

git status

git add .

git commit -m "first commit"

git remote add origin https://github.com/repositivator/django_lotto.git <- "github.com/" 뒤의 사용자 계정명 ('repositivator')이 여러분의 Github 계정명으로 변경되어야 합니다. (저장소 생성 후 복사해서 가져오세요.)

git push -u origin master

- Github repository 새로고침하여 업로드된 파일 확인

15-2) Python anywhere 회원가입 & web app 추가하기

- <https://www.pythonanywhere.com/> 접속하여 회원가입 & 로그인 (여기서는 "**repluslotto**" 이라는 계정명을 예시로 활용하겠습니다. 각자 자신의 계정을 만들어주세요!)
- **Web** 탭 들어가기
- '**Add a new web app**' 클릭
- **Manual configuration (including virtualenvs)**

- Python 3.7

15-3) Python anywhere에 장고 프로젝트 소스코드 업로드하기

- <https://www.pythonanywhere.com/> 접속

- **Consoles** 탭 들어가기

- **Bash** 클릭

git clone https://github.com/repositivator/django_lotto.git <- ".git" 앞부분에 **여러분의 github 저장소 링크**를 적어주세요.

ls

tree django_lotto

virtualenv --python=python3.7 django_env

source django_env/bin/activate

pip install django==2.2.6

cd django_lotto

python manage.py collectstatic

python manage.py migrate

python manage.py createsuperuser (PythonAnywhere 계정명 추천)

15-4) 업로드된 **settings.py** 코드 수정하기 - **(1) Secret key 생성하기**

: **cmd** 혹은 **Python anywhere bash** 등에서 아래 파이썬 코드 복붙하여 실행하기

python <- python shell 실행

import string

```
import random
```

```
chars = ''.join([string.ascii_letters, string.digits, string.punctuation]).replace('W', '').replace(' ', '').replace('WW',  
')
```

```
SECRET_KEY = ''.join([random.SystemRandom().choice(chars) for i in range(50)])
```

```
print(SECRET_KEY)
```

-> 출력된 str 값 복사해두기 (ex : +:py@nL&|0ouVEVBhl<GO0m,1=:d)O;(3%;*_~47X!&#L*j=t)

15-4) 업로드된 **settings.py** 코드 수정하기 - (2) 위 복사해 둔 새로운 **Secret key**를 적용하고 기타 세팅 진행

- 새 탭에서 <https://www.pythonanywhere.com/> 접속
- **Files** 탭 들어가기
- **django_lotto/site_1** 의 **settings.py** 클릭
- 열어둔 **settings.py** 에서 아래와 같이 수정

```
SECRET_KEY = '+:py@nL&|0ouVEVBhl<GO0m,1=:d)O;(3%;*_~47X!&#L*j=t' # 위에서 미리 복사해 둔 str 값
```

```
DEBUG = False
```

```
ALLOWED_HOSTS = ['repluslotto.pythonanywhere.com'] # repluslotto : PythonAnywhere 계정명
```

-> 상단 **Save** 버튼 클릭

15-5) **Virtualenv** 경로 설정 (**Web** 탭)

- 새 탭에서 <https://www.pythonanywhere.com/> 접속
- **Web** 탭 들어가기
- **Virtualenv** 파트의 "**Enter path to a virtualenv, if desired**" 클릭
- 입력 : **/home/<PythonAnywhere계정명>/django_env** (ex. /home/repluslotto/django_env)

15-6) WSGI configuration file 수정 (Web 탭)

- "Code:" 파트의 "WSGI configuration file:"에 적혀 있는 파일명 클릭
- 전체 선택 & 삭제
- 아래 내용 붙여넣기 후 수정 진행 & 저장 (PythonAnywhere계정명을 수정해주셔야 합니다!)

```
import os
```

```
import sys
```

```
path = '/home/<PythonAnywhere계정명>/django_lotto_1' # "django_lotto_1"은 github 저장소명에 해당
```

```
if path not in sys.path:
```

```
    sys.path.append(path)
```

```
os.environ['DJANGO_SETTINGS_MODULE'] = 'site_1.settings' # "site_1"은 settings.py 파일 위치
```

```
from django.core.wsgi import get_wsgi_application
```

```
from django.contrib.staticfiles.handlers import StaticFilesHandler
```

```
application = StaticFilesHandler(get_wsgi_application())
```

15-7) 변경사항 반영 후 웹서비스 접속하기

- <https://www.pythonanywhere.com/> 접속
- Web 탭 들어가기
- Reload 버튼 클릭
- Reload 완료 후 웹브라우저에서 웹서비스 접속해서 테스트 진행 (URL 뒤에 /lotto 붙이기)
ex) <http://repluslotto.pythonanywhere.com/lotto/>