



LIKE LION

at

JBNU

전체 세션

# CONTENTS

1. 30분 안에 Django로  
포트폴리오 사이트 만들기
2. GUI로 해보는 더 쉬운 헤로쿠 배포하기
3. pythonAnywhere에 더 쉽게 배포하기

# 1-1. 개발환경 설정하기



**python -m venv <폴더명>** 가상환경을 만드는 명령어

**. <폴더명>/script/activate** 만든 가상환경을 실행하는 명령어 (tab으로 자동완성)

**pip install django** django를 설치하는 명령어

**django-admin startproject <폴더명>** django 프로젝트 시작 명령어

**python manage.py startapp <앱이름>** 앱을 추가하는 명령어

[startbootstrap.com/themes/](https://startbootstrap.com/themes/) 로 들어가서 원하는 테마 다운로드

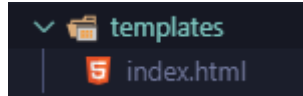
실습에서는 [Freelancer 테마](#) 사용

## 1-2. 앱 등록



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

1. 만든 app을 setting.py에 등록



2. 만든 app 폴더 안에 templates 폴더와,  
index.html 파일 생성

```
def homeView(request):  
    return render(request, 'index.html')
```

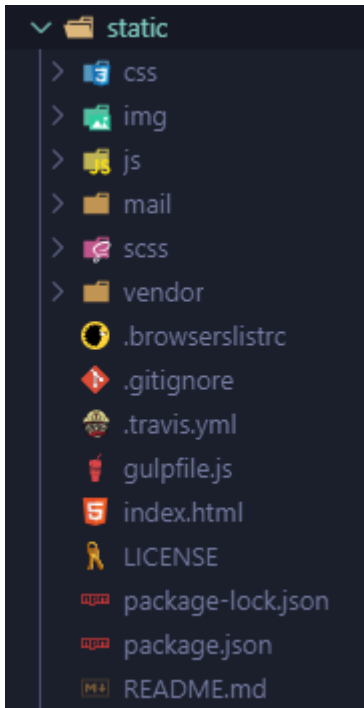
3. views.py에서 해당 코드 작성  
=> index.html을 렌더링해줌

# 1-3. 앱 구현



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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.homeView, name="home")
]
```



4. urls.py에 다음과 같이 작성  
=> url 맵핑

5. python manage.py runserver 명령어를 통해  
로컬 서버 실행  
=> index.html의 결과가 나옴

6. static 폴더를 만들고,  
다운 받은 부트스트랩 테마를 모두 복붙

7. static 폴더의 index.html의 내용을 복사,  
templates 폴더의 index.html로 붙여넣기

# 1-4. static 등록([설명링크](#))



```
STATIC_URL = '/static/'
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "main", "static"),
]
STATIC_ROOT = os.path.join(BASE_DIR, "static")
```

```
{% load static %}
<!DOCTYPE html>
<html lang="ko">
```

```

```

```
<!-- Bootstrap core JavaScript -->
<script src="{% static 'vendor/jquery/jquery.min.js' %}"></script>
<script src="{% static 'vendor/bootstrap/js/bootstrap.bundle.min.js' %}"></script>

<!-- Plugin JavaScript -->
<script src="{% static 'vendor/jquery-easing/jquery.easing.min.js' %}"></script>

<!-- Contact Form JavaScript -->
<script src="{% static 'js/jqBootstrapValidation.js' %}"></script>
<script src="{% static 'js/contact_me.js' %}"></script>

<!-- Custom scripts for this template -->
<script src="{% static 'js/freelancer.min.js' %}"></script>
```

1. setting.py에 static path들을 적어 줌  
=> “main”은 static 폴더가 들어간 앱 이름

2. index.html의 맨 위에 다음과 같이 static을 불러오는 코드를 작성

3. 다음과 같이 index.html의 src 속성을 찾아, 모두 Django static 태그로 바꾸어줌

# 1-5. 모델 만들기



```
class Portfolio(models.Model):
    title = models.CharField(max_length=100)
    body = models.TextField()
    img = models.ImageField()
    created_at = models.DateTimeField(auto_now=True)
    updated_at = models.DateTimeField(auto_now_add=True)

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

1. models.py에  
다음과 같이 모델 클래스를 만듦

2. ImageField 사용을 위해 pillow 패키지 설치  
⇒ pip install pillow

```
# Media
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
MEDIA_URL = '/media/'
```

3. settings.py에 media 관련 작성해줌

```
"""
from django.contrib import admin
from django.urls import path
from main import views
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.homeView, name="home")
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

4. urls.py에 media 파일 url 링크 추가

# 1-5. admin 사이트 적용

A screenshot of the Django admin login page. It has a dark blue header with the text 'Django 관리'. Below the header, there are two input fields: '사용자 이름:' (Username) and '비밀번호:' (Password). At the bottom of the form is a blue button labeled '로그인' (Login).

```
from django.contrib import admin
from .models import Portfolio

# Register your models here.
admin.site.register(Portfolio)
```

현재: 한글로고.png

변경:  선택된 파일 없음

1. <http://127.0.0.1:8000/admin>로 접속

2. 관리자 계정 생성을 위한 명령어 실행  
⇒ `python manage.py createsuperuser`

3. admin.py 에 모델 등록

4. 글 작성하고, 이미지 링크 클릭해서 확인



# 1-6. media 연결



```
from django.shortcuts import render
from .models import Portfolio
# Create your views here.

def homeView(request):
    portfolios = Portfolio.objects.all()

    return render(request, 'index.html', {"portfolios": portfolios})
```

```
<!-- Portfolio Item 1 -->
{% for post in portfolios %}
```

```

```

```
{% endfor %}
```

1. 모델의 변경사항을 적용하기 위해  
명령어 실행

⇒ `python manage.py makemigrations`  
⇒ `python manage.py migrate`

2. `views.py`에 모델 등록 후, 넘겨줌

3. 원하는 부분의 html 태그들을 하나만 남긴 뒤,

이런 식으로 템플릿 태그를 열어 반복해서 호출

⇒ `img` 필드만 `.url`을 붙여줌

⇒ `src` 속성은 링크이기 때문

# 1-7. 추가(테마에 따라 다름)



```
<!-- Portfolio Modals -->
{% for post in portfolios %}
<!-- Portfolio Modal 1 -->
```

1. 모달 창 또한 똑같이 해준다.
2. 허나 데이터 타겟이 정적이라 제대로 동작 안 함

```
<div
  class="portfolio-modal modal fade"
  id="portfolioModal{{post.id}}"
```

3. 모달 창의 id가 동적으로 변할 수 있게 바꿔 줌

```
<div
  class="portfolio-item mx-auto"
  data-toggle="modal"
  data-target="#portfolioModal{{post.id}}"
```

4. 포트폴리오 태그로 가서 데이터 타겟을 동적으로 바꿔 줌

5. 나머지 글들은 html 문서 안에서 자유롭게 작성

# 1-8. 완성샷



## PORTFOLIO



admin에서 등록하면 위와 같이 뜨게 됨

## 2-1. 깃허브 등록



Owner: CaesiumY / Repository name:

Great repository names are short and memorable. Need inspiration? How about [special-rotary-phone?](#)

Description (optional):

☐ **Public**  
Anyone can see this repository. You choose who can commit.

☒ **Private**  
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README  
This will let you immediately clone the repository to your computer.

Add .gitignore:  | Add a license:

[Create repository](#)

...or create a new repository on the command line

```
echo "# mutsa-7th-portfolio" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/CaesiumY/mutsa-7th-portfolio.git
git push -u origin master
```

1. 이미 존재하는 폴더를 업로드할 것이기 때문에 맨 아래는 체크 해제하고 레포지토리 생성

2. manage.py 가 있는 path에서 아래 명령어들 똑같이 입력해 실행

## 2-1. 깃허브 등록



 **gitignore.io**

자신의 프로젝트에 꼭 맞는 .gitignore 파일을 만드세요

생성

[소스 코드](#) | [커맨드라인 문서](#) | [동영상 튜토리얼 보기](#)

```
.gitignore x
.gitignore
You, 14 days ago | 2 a
1
2 # Created by htt
3 # Edit at https:
4
5 ### Django ###
6 *.log
7 *.pot

# Virtual Environment
myenv/
```

1. [gitignore.io](https://gitignore.io)로 들어가서 django를 검색 후 생성

2. .gitignore 파일을 만들고, 내용을 복붙

3. .gitignore 파일에 자신의 가상환경 폴더명을 작성

4. 변경 사항을 모두 깃으로 푸시

## 2-2. 사전 작업



### 1. Heroku deploy에 필요한 패키지들 설치

- ⇒ `pip install gunicorn`
- ⇒ `pip install dj-database-url`
- ⇒ `pip install psycopg2-binary`
- ⇒ `pip install whitenoise`

A screenshot of a code editor showing a file named 'Procfile'. The content of the file is: `web: gunicorn portfolio.wsgi --log-file -`. The editor has a dark theme and a tab labeled 'Procfile' with a close button.

### 2. Procfile을 생성(대문자 주의)하고, 'portfolio' 부분에 자신의 프로젝트명 작성

A screenshot of a code editor showing a file named 'runtime.txt'. The content of the file is: `python-3.7.2`. The editor has a dark theme and a tab labeled 'runtime.txt' with a close button.

### 3. runtime.txt 파일을 생성, 필요한 python 버전을 작성

## 2-2. 사전 작업

```
ALLOWED_HOSTS = ['mutsa-portfolio.herokuapp.com']
```

```
MIDDLEWARE = [  
    'whitenoise.middleware.WhiteNoiseMiddleware',  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',
```

```
# Heroku: Update database configuration from $DATABASE_URL.  
db_from_env = dj_database_url.config(conn_max_age=500)  
DATABASES['default'].update(db_from_env)
```



4. setting.py로 가서 다음과 같이 작성.  
ALLOWED\_HOSTS 에는 헤로쿠 앱 주소를 적거나,  
‘\*’ 를 작성

5. 마지막에도 다음과 같은 코드 작성

6. pip freeze > requirements.txt  
명령어를 통해 pip 패키지 목록을 파일로 자동 생성

7. 모두 완료했으면 깃으로 푸시해서 업로드

```
requirements.txt  
You, 6 days ago | 1 author (You)  
1 astroid==2.3.3  
2 autopep8==1.4.4  
3 colorama==0.4.1  
4 dj-database-url==0.5.0  
5 Django==2.2.7  
6 django-rest-framework==3.10.3  
7 gunicorn==20.0.0  
8 isort==4.3.21  
9 lazy-object-proxy==1.4.3  
10 mccabe==0.6.1  
11 Pillow==6.2.1  
12 psycpg2-binary==2.8.4  
13 pycodestyle==2.5.0  
14 pylint==2.4.3  
15 pytz==2019.3  
16 six==1.13.0  
17 sqlparse==0.3.0  
18 typed-ast==1.4.0  
19 whitenoise==4.1.4  
20 wrapt==1.11.2
```

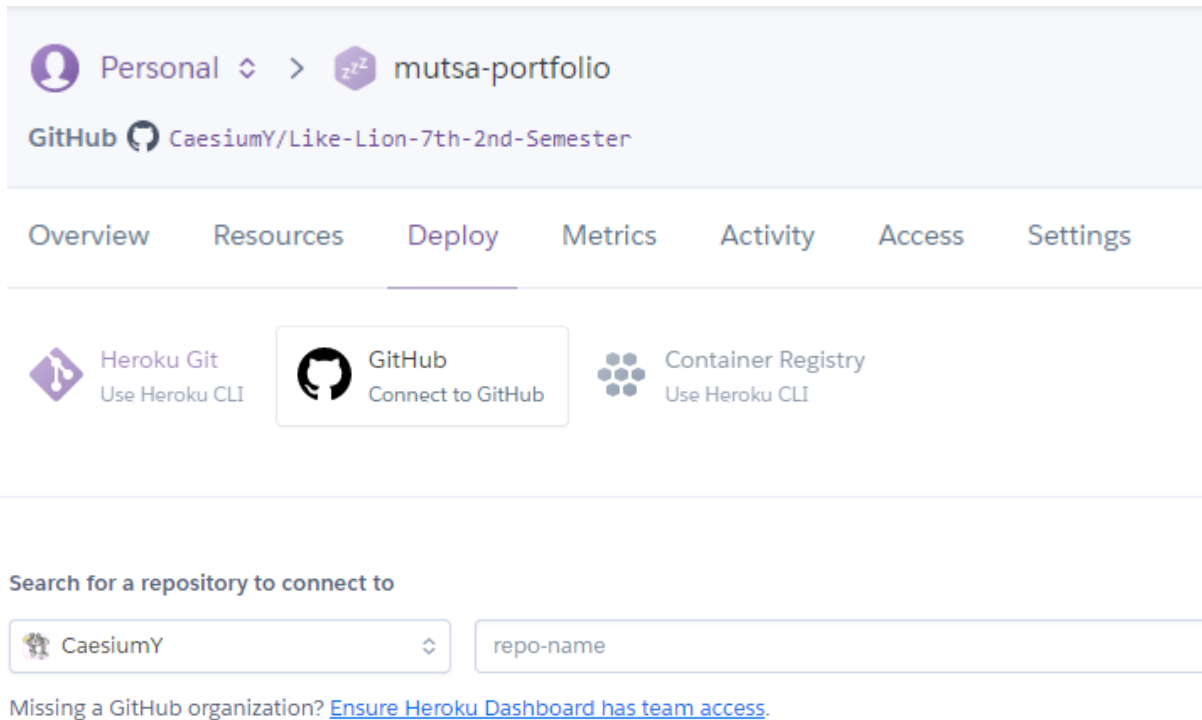
## 2-3. 헤로쿠 대시보드



1. [헤로쿠 대시보드](#)로 이동하여, 새로운 앱을 생성
2. 앱 생성 후 앱 관리 페이지로 접속

3. deploy 탭으로 접속

4. github 항목을 선택하고,  
원하는 레포지토리 검색 후 등록





## 2-3. 헤로쿠 대시보드



Deploy a GitHub branch

This will deploy the current state of the branch you specify below. [Learn more.](#)

Choose a branch to deploy

🔑 master

Deploy Branch

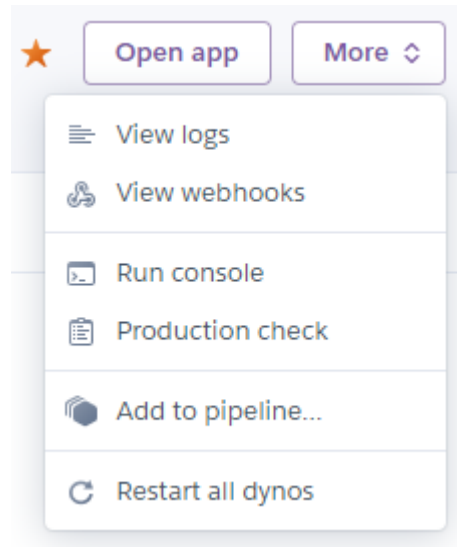
- 5. 원하는 브랜치를 선택 후 수동 배포
- 6. 필요하다면 위의 자동 배포 활성화

7. 지금까지 완벽하게 했다면 한 번에 성공!

8. 하지만 그럴 리가 없다.

9. 그러니 오류 나면 다시 되짚어보자

## 2-4. 헤로쿠 명령어



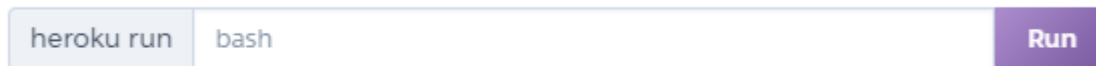
1. DB가 바뀌었으니 관리자 계정을 다시 생성해야 한다.

2. More 버튼을 클릭하고, Run console 선택

3. 이곳에 평소에 쓰던 명령어들을 작성하면 된다.

⇒ `python manage.py createsuperuser` 라든지

⇒ `python manage.py migrate` 라든지



Try a `heroku run` command, as you would from the command line e.g. [console](#), [bash](#)

# 3-0. 깃허브 등록



Owner: CaesiumY / Repository name: mutsa-7th-portfolio

Great repository names are short and memorable. Need inspiration? How about [special-rotary-phone](#)?

Description (optional): mutsa-7th-portfolio

☐ Public  
Anyone can see this repository. You choose who can commit.

☒ Private  
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README  
This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None ⓘ

Create repository

...or create a new repository on the command line

```
echo "# mutsa-7th-portfolio" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/CaesiumY/mutsa-7th-portfolio.git
git push -u origin master
```

1. 이미 존재하는 폴더를 업로드할 것이기 때문에  
맨 아래는 체크 해제하고 레포지토리 생성

⇒ 옆은 private로 생성한 예제

⇒ 보통은 Public

2. manage.py 가 있는 path에서  
아래 명령어들 똑같이 입력해 실행

⇒ git init 과 git remote~ 까지만 하자

## 3-0. 깃허브 등록



 **gitignore.io**

자신의 프로젝트에 꼭 맞는 .gitignore 파일을 만드세요

 생성

[소스 코드](#) | [커맨드라인 문서](#) | [동영상 튜토리얼 보기](#)

```
.gitignore x
.gitignore
You, 14 days ago | 2 a
1
2 # Created by htt
3 # Edit at https:
4
5 ### Django ###
6 *.log
7 *.py

# Virtual Environment
myenv/
```

3. [gitignore.io](https://gitignore.io)로 들어가서 django를 검색 후 생성

4. .gitignore 파일을 만들고, 내용을 복붙

5. .gitignore 파일에 자신의 가상환경 폴더명을 작성

6. 변경 사항을 모두 깃으로 푸시

## 3-1. 사전 작업



```
ALLOWED_HOSTS = ['*']
```

7. setting.py로 가서 다음과 같이 작성.  
ALLOWED\_HOSTS 에는  
'pythonanywhere.com'를 적거나, '\*' 를 작성

```
requirements.txt X
requirements.txt
You, 6 days ago | 1 author (You)
1 astroid==2.3.3
2 autopep8==1.4.4
3 colorama==0.4.1
4 dj-database-url==0.5.0
5 Django==2.2.7
6 djangorestframework==3.10.3
7 gunicorn==20.0.0
8 isort==4.3.21
9 lazy-object-proxy==1.4.3
10 mccabe==0.6.1
11 Pillow==6.2.1
12 psycpg2-binary==2.8.4
13 pycodestyle==2.5.0
14 pylint==2.4.3
15 pytz==2019.3
16 six==1.13.0
17 sqlparse==0.3.0
18 typed-ast==1.4.0
19 whitenoise==4.1.4
20 wrapt==1.11.2
```

8. `pip freeze > requirements.txt`  
명령어를 통해 pip 패키지 목록을 파일로 자동 생성  
⇒ 옆 사진과 같지 않아도 됨

9. 모두 완료했으면 깃으로 푸시해서 업로드

## 3-2. pythonAnywhere 시작하기



### Beginner: Free!

A limited account with one web app at `your-username.pythonanywhere.com`, restricted outbound Internet access from your apps, low CPU/bandwidth, no IPython/Jupyter notebook support.  
It works and it's a great way to get started!

Recent  
Consoles

+ 5 -

Create a Beginner account

### 1. PythonAnywhere에 가입

Bash console 14175845

View all

New console:

\$ Bash

>>> Python ▾

More...

02:39 ~ \$

### 2. 대시보드에서 Bash를 통해 커맨드라인 접속

### 3. 성공!

## 3-3. CLI 다루기



```
02:39 ~ $ git clone https://github.com/CaesiumY/mutsa-7th-portfolio.git
```

1. git clone으로 내 저장소를 가져옴

```
02:39 ~ $ virtualenv --python=python3.6 myenv
```

2. 가상환경을 만들어준다.  
⇒ 윈도우와는 다른 명령어

3. 이후로는 개발환경 세팅하듯이 하면 끝  
⇒ `. myenv/bin/activate`  
⇒ `cd 프로젝트 폴더 이름`  
⇒ `pip install -r requirements.txt`  
⇒ `python manage.py migrate`  
⇒ 등등...

## 3-4. 웹 앱 등록하기



1. 대시보드로 돌아와 web 탭을 클릭하고, Add a new web app 선택
2. 선택지에서 **django가 아닌 manual** 선택

pythonanywhere

Send feedback

Consoles Files **Web** Schedule Databases

✓ edith.pythonanywhere.com

⊕ Add a new web app

### Configuration for edith.pythonanywhere.com

**Actions:**

Reload edith.pythonanywhere.com Delete edith.pythonanyw

**Code:**

What your site is running.

Source code: You can use the [Files tab](#) to navigate to your app's source code.

WSGI configuration file: [/var/www/edith\\_pythonanywhere\\_com\\_wsgi.py](#)

Python version: 3.4

**Virtualenv:**

Use a virtualenv to get different versions of flask, django etc from our default system ones. More info [here](#). You need to Reload your web app to activate it; NB - we do nothing if the virtualenv does not exist.

/home/edith/my-first-blog/myvenv

3. 그림과 같이 가상환경 주소를 적어 줌



# 3-5. wsgi 설정



Code:

What your site is running.

Source code: [Enter the path to your web app source code](#)

Working directory: [/home/mn0658/](#)

[Go to directory](#)

WSGI configuration file: [/var/www/mn0658\\_pythonanywhere\\_com\\_wsgi.py](#)

Python version: 3.6

## 1. wsgi 파일 링크 클릭

## 2. 전부 삭제하고,

옆 코드 복사 뒤

SAVE!

```
1 import os
2 import sys
3
4 path = '/home/<your-PythonAnywhere-username>/<your-project>' # PythonAnywhere 계정으로 바꾸세요.
5 if path not in sys.path:
6     sys.path.append(path)
7
8 os.environ['DJANGO_SETTINGS_MODULE'] = 'mysite.settings' # settings.py가 있는 폴더명으로 바꾸세요.
9
10 from django.core.wsgi import get_wsgi_application
11 from django.contrib.staticfiles.handlers import StaticFilesHandler
12 application = StaticFilesHandler(get_wsgi_application())
```


*Colored by Color Scripter*

## 3-6. 새로 고침



Configuration for [mn0658.pythonanywhere.com](https://mn0658.pythonanywhere.com)

Reload:

 Reload [mn0658.pythonanywhere.com](https://mn0658.pythonanywhere.com)

**1. Reload를 눌러 새로 고침 한 뒤,  
위의 링크로 접속!**

# + Heroku VS pythonAnywhere



## 헤로쿠의 특징

- 깃과 연동되어 자동 업데이트
- 소셜 로그인 지원
- 한 계정에 여러 앱 등록 가능
- 느림
- 상대적으로 복잡한 배포 과정
- 미디어 파일 지원 X

## PythonAnywhere 의 특징

- 매우 빠른 속도
- 쉬운 배포과정
- 미디어 파일 지원 O
- 업데이트마다 수동
- 소셜 로그인 미지원
- 한 계정에 하나의 웹 앱