

django

Agenda

- About Me
- About Django Framework
- Project Overview
- Venv
- .gitignore
- Installing Django & Running the Server
- Pages App
- Templates
- About Page
- base.html and partials
- Static Files
- Update templates
- Link Highlighting
- Word Counter

About Me

Name: Luv-Krishna Soyjaudah

Email: krishnasoyjaudah@gmail.com

Experience:

WhatsApp: +230 5803 4656

2 years - Risk Management

LinkedIn: <https://www.linkedin.com/in/luv-krishna-soyjaudah-65617b91/>

~ 5 years – IT consultant (Oracle Fusion Cloud, EBS, EPM Cloud, ERPNext, etc.)

Recently learned (and still learning) and applied Django on a personal project.

About Django

Django is a free and open source web framework written in Python, made up of a collection of modules that make development easier.

When to use Django?

Django is built to encourage rapid development and clean, practical design.

It provides a concrete foundation of the basics, allowing developers to focus on parts of their project that require customisation, instead of wasting time with the fundamental stuff.

A few popular websites built with Django

The logo for Disqus, featuring the word "DISQUS" in a bold, blue, sans-serif font.

Source: <https://codecondo.com/popular-websites-django/>

[illegible]

The screenshot shows the 'WORD COUNTER' website. The header has a dark blue background with the site name in white. Below it is a light blue navigation bar with links for HOME, ABOUT, TEAM, and CONTACT. The main content area has a light blue background. A breadcrumb trail at the top left reads 'Home / Team'. There are three team member cards. Each card features a portrait photo, the person's name, their title, and a blue 'Read' button. The first card is for Stephen King, CEO, with a photo of him resting his chin on his hand. The second card is for Dan Brown, COO, with a photo of him smiling. The third card is for Lisa Jewell, CFO, with a photo of her sitting in an ornate chair. The background of the entire page is a collage of wooden blocks with letters and numbers.

CONTACT

HOMEABOUTBLOGCONTACT

Home / Contact

Send us a message...

First Name*

Last Name*

Email address*

Phone*

Message*

☐ I accept the Terms & Conditions

Send

Venv – Virtual Environment

Venv – is an isolated environment which does not contain the python packages that will not be used in the project.

Create a virtual environment

```
python -m venv ./venv/
```

Folder name



This will create a venv folder in the project folder with the name 'venv'.

Activate venv

<path of project folder>/venv/scripts/activate.bat ← Windows

E.g.

C:\Users\USER\OneDrive\Desktop\wordcounter\venv\scripts\activate

source ./venv/bin/activate ← Mac

Installing Django & Running the Server

With the virtual environment activated, run

pip freeze ← To make sure there are no packages installed

pip install django ← To install Django in the virtual environment

Run '*django-admin help*' to view commands available in Django.

<https://docs.djangoproject.com/en/3.0/ref/django-admin/>

Start Project

Run

'django-admin startproject wordcounter .' to start the project 'wordcounter'.

This will create the main files needed to start.

Run '*python manage.py runserver*' to start the server.

Open a new page in a browser and go to 'localhost:8000'.

.gitignore

.gitignore → File which specifies what will not be pushed to git repository

Go to <http://gitignore.io/> to generate gitignore file for django.

In Root folder, create file '.gitignore'

Copy and paste content from gitignore.io and save.

Add '/venv' to .gitignore

Pages App

Pages App – To manage homepage, about page and count page.

The About page shows employees, so models will have to be imported from the Employees app into the Pages app.

1. Create the Pages app.

python manage.py startapp pages

2. Add app to INSTALLED_APPS in settings.py file (wordcounter > settings.py > INSTALLED_APPS)

'pages.apps.PagesConfig'

3. Create 'urls.py' file in Pages app.

Add *'from django.urls import path'* to bring in the path package.

Add *'from . import views'* to be able to bring the url that is attached to a method in the view file.

Add

```
urlpatterns = [  
    path("", views.index, name='index')  
]
```

↑
For homepage

Pages App

4. In the pages > views.py file

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

```
def index(request):  
    return HttpResponse('<h1>Hello World</h1>')
```

5. Add the path to the project urls file.

In wordcounter > urls.py

```
from django.urls import path, include ← Add 'include'
```

To urlpatterns, add:

```
path("", include('pages.urls'))
```

6. Open 'localhost:8000' in a browser and check whether 'Hello World' is showing.

Templates

1. In Root folder, create a folder 'templates'.
2. Tell django where to look for templates

wordcounter > settings.py

```
TEMPLATES = [  
    {  
    ...  
        'DIRS' : [os.path.join(BASE_DIR, 'templates')]  
    ...  
]
```

3. In folder 'templates', create folder 'pages'. Create files 'index.html' and 'about.html'.
4. index.html → <h1>Hello World</h1>
5. about.html → <h1>About</h1>
6. count.html → <h1>Count</h1>

About & Count Pages

1. Pages App > urls.py

Add

```
path('about', views.about, name='about')  
in urlpatterns
```

2. Pages App > views.py

from django.shortcuts import render → To import the render shortcut to render html pages

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

```
def about(request):  
    return render(request, 'pages/about.html')
```

3. Open 'localhost:8000' and 'localhost:8000/about' to check if working.
4. Repeat the same steps for count.html

base.html and partials

1. base.html → code that must be extended on other templates. Makes use of the Jinja syntax `{% %}`.
2. partials are html files that contain content that will be displayed on every page of the website. For example, Navbar and Footer.
3. In folder 'templates', create file 'base.html'.
4. In folder 'templates', create folder 'partials'. In folder 'partials', create files '_navbar.html', '_site_header.html' and '_footer.html'.
5. Copy and paste navbar html code into '_navbar.html', site header html code into '_site_header.html' and footer html code into '_footer.html'.
6. Include these files in the 'base.html' file.
7. Use `{% extends 'base.html' %}` to extend the base.html file onto the index.html, about.html and count.html files.

base.html and partials

_navbar.html

```
<!-- Nav Bar -->
<nav class="navbar navbar-expand-lg navbar-dark bg-gold">
  <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav"
aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav mx-auto">
      <li class="nav-item active">
        <a class="nav-link mr-5" href="index.html">Home <span
class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link mr-5" href="about.html">About</a>
      </li>
      <li class="nav-item">
        <a class="nav-link mr-5" href="blog.html">Blog</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="contact.html">Contact</a>
      </li>
    </ul>
  </div>
</nav>
```

_site_header.html

```
<!-- Site Header -->
<section id="site-header">
  <div class="container mt-3">
    <p class="text-white text-center top-header-
big">WORD COUNTER</p>
  </div>
</section>
```

_footer.html

```
<!-- Footer -->
<footer id="main-footer" class="py-4 bg-gold text-white text-center">
  <span class="text-center">
    <a href="#" class="link">Terms & Conditions</a> |
    <a href="#" class="link">Privacy Policy</a>
  <br>

  ALL RIGHTS RESERVED. COPYRIGHT &copy; 2020
  WORD COUNTER
  </span>
</footer>
```

base.html and partials

base.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">

  <!-- Font Awesome -->
  <link rel="stylesheet" href="assets/css/all.css">

  <!-- Bootstrap Min CSS Local-->
  <link rel="stylesheet" href="assets/css/bootstrap.min.css">

  <!-- Custom -->
  <link rel="stylesheet" href="assets/css/style.css">

  <title> Word Counter</title>
</head>
```

```
<body>

<!-- Nav Bar -->
{% include 'partials/_navbar.html' %}

{% block content %}

{% endblock %}

<!-- Footer -->
{% include 'partials/_footer.html' %}

</body>
</html>
```

Static Files

Configuration in settings.py

Where to search for static files

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

Common practice is to have all static files in root directory.

Where static files will be copied to

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

This is where django will serve the static files in production.

1. Create static folder in root directory.
2. Copy and paste folders 'css', 'img', 'js' and 'webfonts' to the 'static' folder.
3. Run the command:

```
python manage.py collectstatic
```

This will create the static folder in 'wordcounter' project folder (not root) and copy all static files listed at Step 2 above.

4. Add '/static' to .gitignore.

Update templates

Now that static files have been copied to the project folder, the static files need to be called onto the templates.

`{% load static %}` → to successfully call static files onto templates

For example, in `base.html`

Add `{% load static %}` at the top.

For Font Awesome,

Update

```
<link rel="stylesheet" href="assets/css/all.css">
```

to

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

Repeat this for all static css, img and js files in `base.html`, `_navbar.html` and `_footer.html`

Update templates

index.html and **about.html**

Add `{% extends 'base.html' %}` at the top.

If static files, add `{% load static %}` at the top.

Copy and paste content from original html files in `{% block content %}` tags.

Update static files links as on previous page.

Run `'localhost:8000'` and `'localhost:8000/about'` to see if working properly.

Link Highlighting

In the Navbar, the links must be highlighted when switching pages.

Use django IF statements to accomplish this.

E.g. For homepage

```
<li
    {% if '/' == request.path %}
        class = "nav-item active mr-3"
    {% else %}
        class = "nav-item mr-3"
    {% endif %}
>
```

Word Counter



What do we want to do?

- When a string of text is entered in the text box and the Count button is clicked, our program counts the number of words in the string and returns the number of words.

In Pages app > views.py

```
def count(request):  
    textinput = request.GET['textinput']  
    wordlist = textinput.split()  
    count = len(wordlist)
```

Word Counter

```
context = {  
    'textinput': textinput,  
    'count': count,  
}  
  
return render(request, 'pages/count.html', context)
```

On index.html

Set `<form action="{% url 'count' %}">`