

# Outputting Python to html Django

Jmar

In this [Python Django tutorial](#), we will understand **the implementation of outputting python to HTML Django**. And we'll also see different examples related to this. These are the following topics that we are going to discuss in this tutorial.

- Basic Steps for ouput to HTML
- How to include app in project
- How to access app url
- How to access view
- Template in Django
- Run Server in Django
- Outputting python to html Django

Table of Contents



## Basic Steps for output to HTML

Since we've learned how to represent program output in a

**≡ MENU****X**

## Step#1 Install Django

---

### MY LATEST VIDEOS

How to rotate image in Python Scipy



≡ MENU

We can use pip to install packages that aren't part of the Python distribution. The following command is executed at the command prompt to download and install Django.

pip install django

```
PS E:\MyProject> pip install django
Requirement already satisfied: django in c:\users\tanya\appdata\local\programs\python\python39\lib\site-packages (4.0.2)
Requirement already satisfied: sqlparse>=0.2.2 in c:\users\tanya\appdata\local\programs\python\python39\lib\site-packages (from django) (0.4.2)
Requirement already satisfied: tzdata in c:\users\tanya\appdata\local\programs\python\python39\lib\site-packages (from django) (2021.5)
Requirement already satisfied: asgiref<4,>=3.4.1 in c:\users\tanya\appdata\local\programs\python\python39\lib\site-packages (from django) (3.5.0)
PS E:\MyProject> 
```

Install Django

## Step#2 Test Django Installation

**≡ MENU**

After installing Django, use the command prompt to test whether the Django is installed successfully or not.

Command:

```
django-admin --version
```

---

Django Version

the terminal, execute the following command to create a Django project.

```
django-admin startproject project_name
```

You can call this project whatever you like, but make it brief and relevant because you'll be building big websites.

**NOTE:**

*No, you shouldn't have to reinstall Django every time you start a new project. Once you've installed it with pip, all you have to do now is run the django*

Here, I create a Django Project with the name PythonGuides.

```
PS E:\MyProject> django-admin startproject PythonGuides
```

```
PS E:\MyProject> _
```

Django Project

## Step#4 Django App

For every functionality, an app can be built as a completely self-contained module. To create a basic app in your Django project



≡ MENU

django-admin startapp app\_name

Here, I create a Django App with the name home.

Django App

## Step#5 Install App

To include the app in your project, add your app name to the INSTALLED APPS list in settings.py of the project.

Installed App

Read: [If statement in Django template](#)

## How to include app in project

In the previous section, we learned how to create a project and an app, as well as how to install an app. Now we'll learn how to include an app into our project in this section.

- Open Project Directory.
- Open the urls.py file under your project directory.
- Add the headers.
- You must now specify the app name in the list of URL patterns in order to include your app URLs.

Project Directory

Project's urls.py file

## Source Code of PythonGuides urls.py file:

```
from django.contrib import admin
from django.urls import path, include

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

Read: [Get URL parameters in Django](#)

**≡ MENU**

---

In this section, we'll learn how to access the app's URL. The project's urls.py file will not be able to access the app's URL file because it is not included in the app directory. The following steps must be completed for your Django Web application to execute properly.

- Create a urls.py file in the apps directory.
- Add the code.

App Directory

---

App's urls.py File

## Source Code of home urls.py file:

```
from django.urls import path, include  
from home import views  
  
urlpatterns = [  
    path("", views.home, name='home'),  
]
```

Read: [Python Django length filter](#)

**≡ MENU**

---

In this section, we'll learn how to access views in Django. A view function, also known as a “view,” is a Python function that receives a web request and provides a web response. This response could be the HTML content of a Web page, a redirect, a 404 error, an XML document, an image, or something else entirely.

In Django, views must be defined in the app `views.py` file. We may create views in Django in two ways: Basic views and MVT pattern views.

- **Basic view:** To render the HTML page in basic view, we utilize `HttpResponse`. However, this is not the most efficient means of rendering pages.
  - **MVT pattern view:** To create the precedent view, Django supports the MVT pattern. As a result, we render a
-

**≡ MENU**

App's view.py File

## Source Code of views.py file:

```
from django.shortcuts import render, HttpResponseRedirect

# Create your views here.

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

Read: [How to get data from get request in Django](#)

[Templates in Django](#)

≡ MENU

Django's template system makes it simple to create dynamic HTML pages. A template is made up of static sections of the HTML output that you want to create. The Django template engine allows us to develop dynamic web pages by separating the design from the python code.

The following are the steps to generate HTML pages using the template.

- Create Template Directory
- Template Configuration
- Create HTML page.

## Step#1 Create Template Directory

To begin, open the project app and create a directory template.

Template Directory

## Step#2 Template Configuration

We must provide some items in the settings.py file to configure the template system. The name of our template directory is **templates**, as we indicated earlier. DjangoTemplates searches each of the **INSTALLED APPS** subdirectories by default for a templates subdirectoru.



≡ MENU

Templates

### Step#3

Then, inside the newly formed template folder, create a template HTML page.



≡ MENU

HTML File

## Source Code of home.html file:

```
<!doctype html>
<html lang="en">

<head>
```



≡ MENU

```
/dist/css/bootstrap.min.css" rel="stylesheet">
```

```
<title>PythonGuides!</title>
</head>
```

```
<body>
```

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">PythonGuides</a>
    <button class="navbar-toggler" type="button" data-bs-
    toggle="collapse"
      data-bs-target="#navbarSupportedContent" aria-
    controls="navbarSupportedContent" aria-expanded="false"
      aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse"
      id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page"
            href="#">Home</a>
        </li>
      </ul>
      <form class="d-flex">
        <input class="form-control me-2" type="search"
          placeholder="Search" aria-label="Search">
        <button class="btn btn-outline-success"
```



≡ MENU

```
<div class="container my-5">
    <h1 style="text-align:center;">Welcome to Python
    Guides</h1>
</div>

<div class="card">
    <div class="card-body">
        <h3 style="text-align:center;">Thanks for landing on this
        page to know more about PythonGuides.com.</h3>
        <br>
        <h6>I welcome you to the website and hopefully, you got
        to learn something in Python. I started this website
        to share my finding and learnings in Python with
        you.</h6>
        <h6>To keep things simple, I am trying to write a lot of
        articles on Python. Feel free to give your valuable
        comments and also share the articles if you are liking
        and hoping it will be helpful to someone.</h6>
        <br>
        <h2 style="text-align:center;">Also, Subscribe to Our
        YouTube Channel for FREE Python Video Tutorials.</h2>
    </div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2
/dist/js/bootstrap.bundle.min.js"></script>
```



≡ MENU

## Run Server in Django

In this section, we'll learn to run the server in Django. The Django development server provides a great local development server that developers can use to preview code updates and test Django-powered website features.

Run the following command in the terminal to start the server.

```
python manage.py runserver
```

Read: [Python Django group by](#)

## Outputting python to HTML Django

In this section, we'll see the HTML Django output on the browser.

So, after running the development server, we need to open a browser and open the following URL <http://127.0.0.1:8000/>.

Django Output

Also, take a look at some more Django tutorials.

- [Python Django filter](#)
- [Python Django get](#)
- [Django for loop](#)

In this Django tutorial, we have understood the implementation of outputting python to HTML Django. Here is the complete list of topics that we have covered in this tutorial.

- Basic Steps for ouput to HTML
- How to include app in project
- How to access app url
- How to access view
- Template in Django
- Run Server in Django
- Outputting python to html django



Bijay Kumar

I am Bijay Kumar, a [Microsoft MVP](#) in SharePoint. Apart from SharePoint, I started working on Python, Machine learning, and artificial intelligence for the last 5 years. During this time



≡ MENU



- < [Python Dictionary of sets](#)
- > [PyTorch nn linear + Examples](#)

Search

---

Follow us in Twitter & Facebook

≡ MENU

## Recent Posts

[Try except in while loop Python](#)

[While loop with multiple conditions in Python](#)

[JWT Authentication Using Django Rest Framework](#)

[How to create an API in Python Django \(using Generic Class API\)](#)

[How to Create a User Profile Using Django](#)

**≡ MENU**[About PythonGuides.com](#)   [Contact](#)   [Privacy Policy](#)   [Sitemap](#)

© 2023 PythonGuides.com

[Update Privacy Preferences](#)

---

A RAPTIVE PARTNER SITE