Django

M – Model

V – View

T – Template

View

- Function Based View
- Class Based View

Function Based View

A function based view, is a Python function that takes a Web request and returns a Web response.

This response can be the HTML contents of a Web page, or a redirect, or a 404 error, or an XML document, or an image or anything.

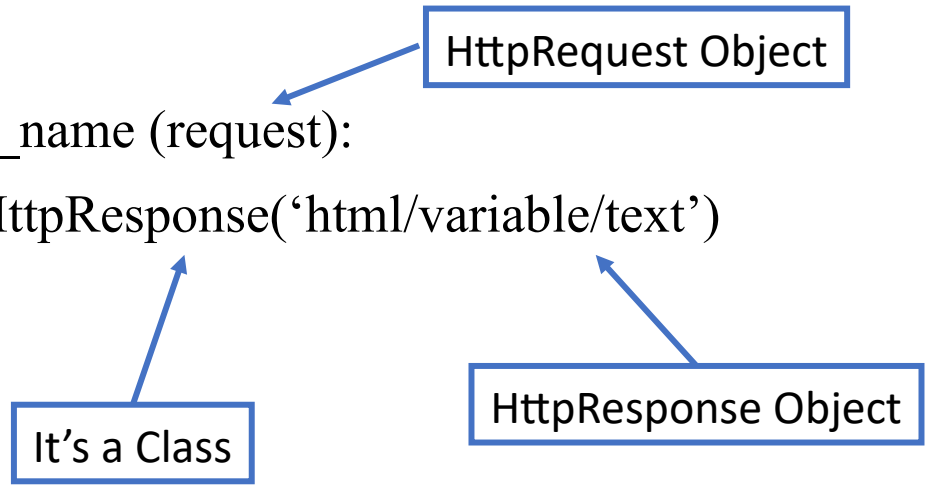
Each view function takes an HttpRequest object as its first parameter.

The view returns an HttpResponse object that contains the generated response. Each view function is responsible for returning an HttpResponse object.

We will call these functions as *view function* or *view function of application or view*.

Syntax:-

```
def function_name (request):
    return HttpResponse('html/variable/text')
```



Function Based Views

We use **views.py** file of the application to write functions which may contain business logic of application, later it required to define url name for this function in the **urls.py** file of the project.

Syntax:-

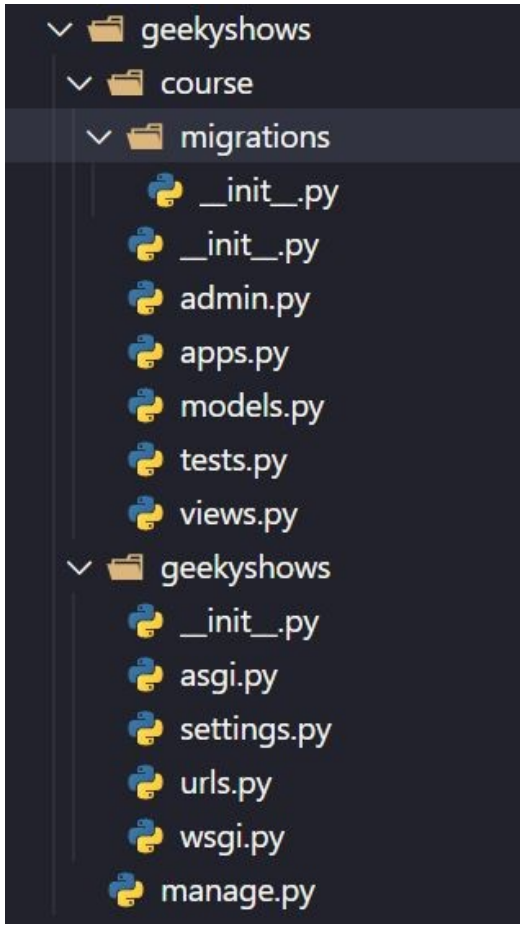
views.py

```
def function_name1 (request):
        return HttpResponse('html/variable/text')

def function_name2 (request):
        return HttpResponse('html/variable/text')
```

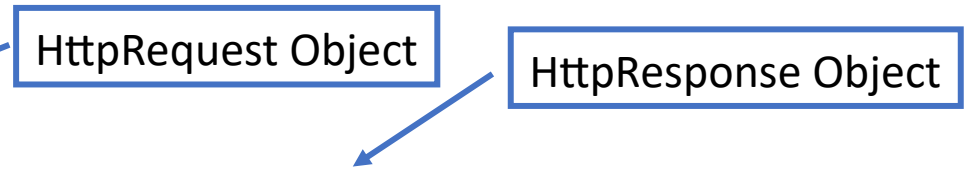
HttpRequest Object

HttpResponse Object



Function Based Views

Syntax:-




```
def function_name (request):
    return HttpResponse('html/variable/text')
```

Where **HttpResponse** is class which is in **django.http** module so we have to import it before using **HttpResponse**.

views.py

```
from django.http import HttpResponse
def learn_django(request):
    return HttpResponse('Hello Django')

def learn_python(request):
    return HttpResponse('<h1>Hello Python</h1>')
```

```
def learn_var(request):
    a = '<h1>Hello Variable</h1>'
    return HttpResponse(a)

def learn_math(request):
    a = 10 + 10
    return HttpResponse(a)
```

Function Based Views

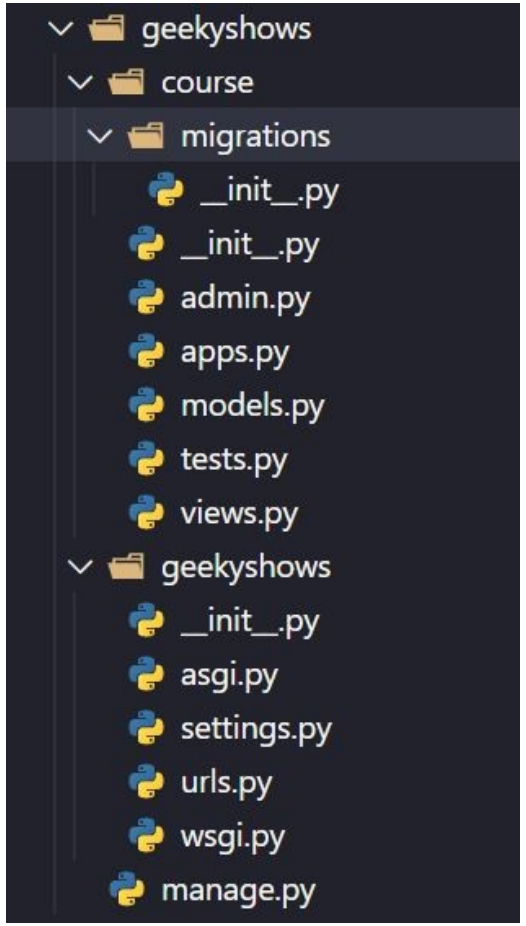
Single Application with Single view function

views.py

```
from django.http import HttpResponse
def learn_django(request):
    return HttpResponse('Hello Django')
```

urls.py

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('learndj/', views.learn_django),
]
```



Function Based Views

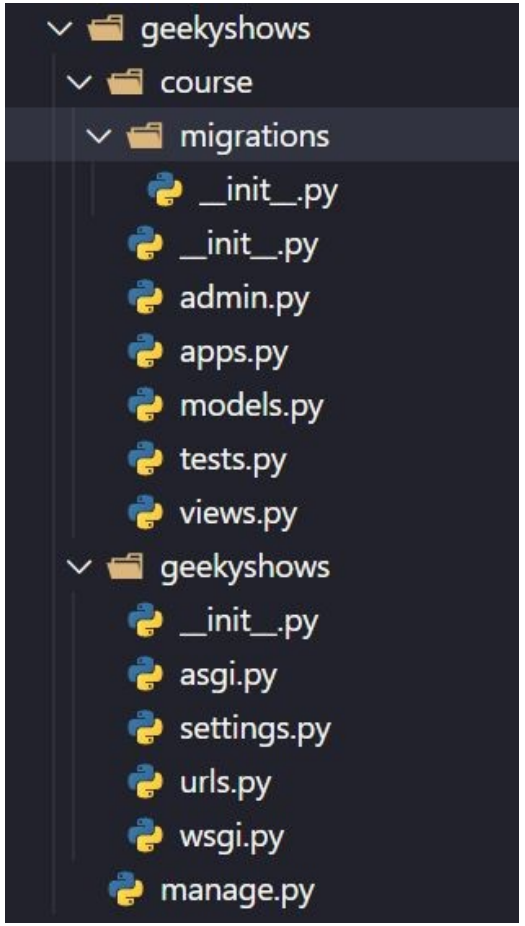
Single Application with multiple view functions.

views.py

```
from django.http import HttpResponse
def learn_django(request):
    return HttpResponse('Hello Django')
def learn_python(request):
    return HttpResponse('<h1>Hello Python</h1>')
```

urls.py

```
urlpatterns = [
    path('learndj/', views.learn_django),
    path('learnpy/', views.learn_python),
]
```



Geeky Steps

- Create Django Project: *django-admin startproject geekyshows*
- Change Directory to Django Project: *cd geekyshows*
- Create Django Application: *python manage.py startapp course*
- Add/Install Application to Django Project (course to geekyshows) using settings.py file
  *INSTALLED_APPS = [‘django.contrib.admin’,**‘course’**,]*
- Write View Function inside **views.py** file
  - Open **views.py**
  - Import HttpResponse class from django.http module

    from django.http import HttpResponse
  - Write view Function

  def learn_django(request):

  return HttpResponse(‘Hello Django’)
  - Save views.py