**Templates**

**What is template?**

Template is an html file or web page

**Why template?**

Inserting html code inside view is complex to maintain

There is no separation of business logic and presentation logic

In order separate business logic and presentation logic we use templates. Template contains presentation logic or output logic and input.

Following steps which are used to work with templates.

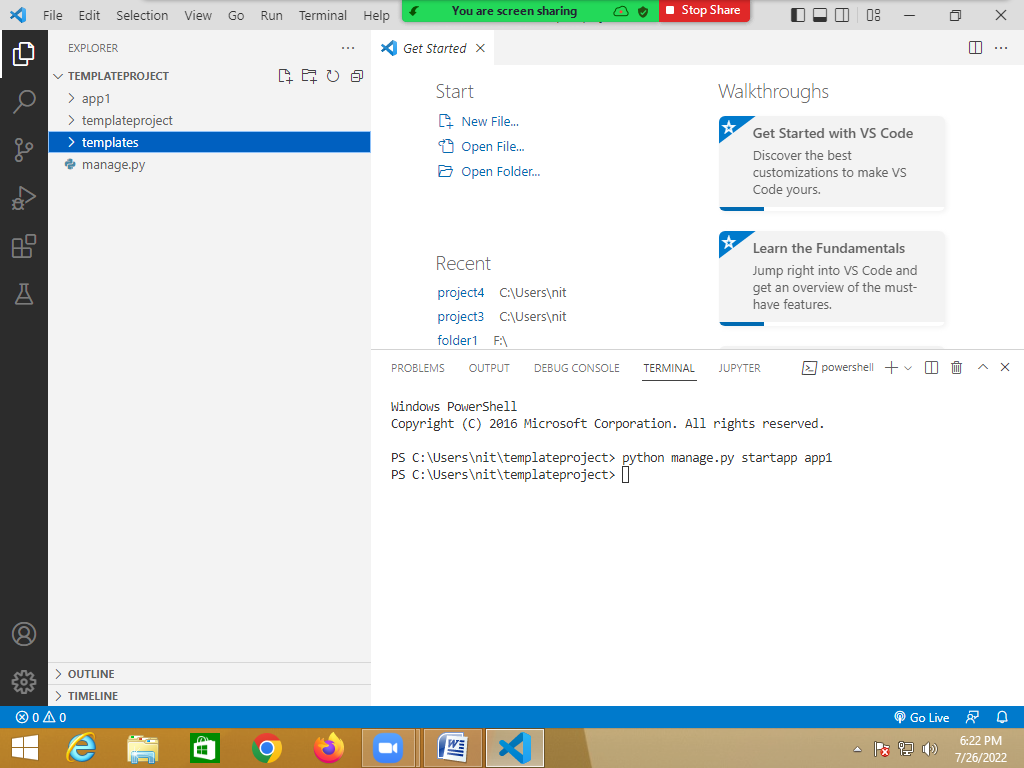
1. Create project

django-admin startproject templateproject

1. Create application within project

Python manage.py startapp app1

1. Create a folder with name templates in project folder



1. Add one html file inside templates folder

**welcome.html**

<html>

<body>

    <h1> Welcome to Templates </h1>

</body>

</html>

1. Create a view inside views.py

from django.shortcuts import render

# Create your views here.

def display(request):

    resp=render(request,"welcome.html")

    return resp

1. Configuration of templates

Open settings.py

import os.path

TEMPLATE\_DIR=os.path.join(BASE\_DIR,"templates")

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [TEMPLATE\_DIR],

        'APP\_DIRS': True,

        'OPTIONS': {

            'context\_processors': [

                'django.template.context\_processors.debug',

                'django.template.context\_processors.request',

                'django.contrib.auth.context\_processors.auth',

                'django.contrib.messages.context\_processors.messages',

            ],

        },

    },

]

1. Install application

Open settings.py

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'app1'

]

1. Define urlpattern

Open urls.py

from django.contrib import admin

from django.urls import path

from app1 import views

urlpatterns = [

    path('admin/', admin.site.urls),

    path('first/',views.display),

]

**render()**

it is a predefined function django.shortcuts this function returns HttpResponse object.this function receives request,template and arguments.

Combines a given template with a given context dictionary and returns an **[HttpResponse](https://docs.djangoproject.com/en/4.0/ref/request-response/" \l "django.http.HttpResponse" \o "django.http.HttpResponse)** object with that rendered text.

**Required arguments**

**request**

The request object used to generate this response.

**template\_name**

The full name of a template to use or sequence of template names.

**Optional arguments**

**context**

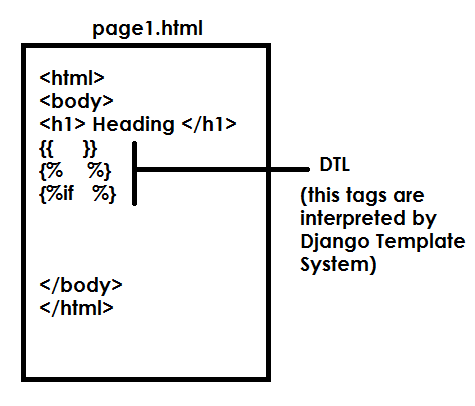
A dictionary of values to add to the template context. By default, this is an empty dictionary. If a value in the dictionary is callable, the view will call it just before rendering the template.

**Django Template Language (DTL)**

**What is Django template language?**

A Django template is **a text document or a Python string marked-up using the Django template language**. Some constructs are recognized and interpreted by the template engine. The main ones are variables and tags. A template is rendered with a context.

A template contains **variables**, which get replaced with values when the template is evaluated, and **tags**, which control the logic of the template.

****

**Variables**

Variables look like this: **{{ variable }}**. When the template engine encounters a variable, it evaluates that variable and replaces it with the result.

**Views.py**

from django.shortcuts import render

# Create your views here.

def display(request):

    dict1={'name':'naresh'}

    resp=render(request,"welcome.html",context=dict1)

    return resp

**welcome.html**

<html>

<body>

    <h1> Welcome to Templates </h1>

    <h2> Hello {{name}}</h2>

</body>

</html>

**{%for%} loop**

Loop over each item in an array.

**Syntax:**

{% for variable-name in collection %}

<html code>

{%endfor%}

**Views.py**

from django.shortcuts import render

# Create your views here.

def display(request):

    dict1={'name':['naresh','suresh']}

    resp=render(request,"welcome.html",context=dict1)

    return resp

**welcome.html**

<html>

<body>

    <h1> Welcome to Templates </h1>

    {% for n in name %}

    <h2> Hello {{n}}</h2>

    {%endfor%}

</body>

</html>

[**if**](https://docs.djangoproject.com/en/4.0/ref/templates/builtins/#std-templatetag-if)**, elif, and else**

Evaluates a variable, and if that variable is “true” the contents of the block are displayed:

|  |  |  |
| --- | --- | --- |
| **Syntax-1**  **{% if condition %}**  **Block**  **{% endif %}** | **Syntax-2**  **{% if condition %}**  **Block**  **{% else %}**  **Block**  **{%endif%}** | **Syntax-3**  **{% if condition %}**  **Block**  **{% elif condition %}**  **Block**  **{% else %}**  **Block**  **{% endif %}** |