Plantillas V.

En este módulo veremos la simplificación de codigo con un paquete de shortcuts y las plantillas incrustadas (plantillas anidadas).

Para saber un poco mas sobre los shortcuts podemos visitar la documentación oficial de Django:

<https://docs.djangoproject.com/en/5.0/topics/http/shortcuts/>

En este modulo trabajaremos con el siguiente archivo view.py

from django.http import HttpResponse

from django.template import Template, Context

from django.template import loader

import datetime

class Persona(object):

    def \_\_init\_\_(self, nombre, apellido):

        self.**nombre** = nombre

        self.**apellido** = apellido

    def saludo(request):

        p1 = Persona("Alexis", "Reynoso")

        fecha\_actual = datetime.datetime.now()

        doc\_externo = loader.get\_template('index.html')

        ctx = {

            'nombre\_persona' : p1.**nombre**,

            'apellido\_persona' : p1.**apellido**,

            'fecha' : fecha\_actual,

            'temas' : ["Plantillas", "Modelos", "Formularios", "Vistas", "BBDD", "Despliegue aplicación"] *#"Plantillas", "Modelos", "Formularios", "Vistas", "BBDD", "Despliegue aplicación"*

        }

        documento = doc\_externo.render(ctx)

        return HttpResponse(documento)

Lo que modificaremos dentro de este archivo será añadir una nueva importación

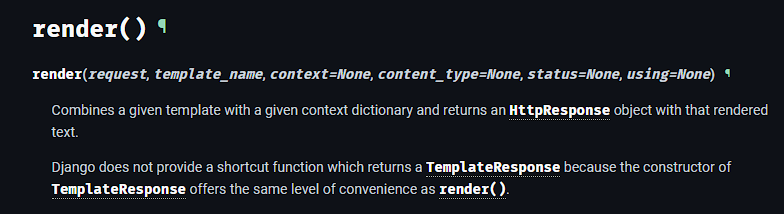
from django.shortcuts import render

Aqui importaremos el método render.

Ahora modificaremos el archivo eliminando las siguientes lineas  
- La linea que tiene el render

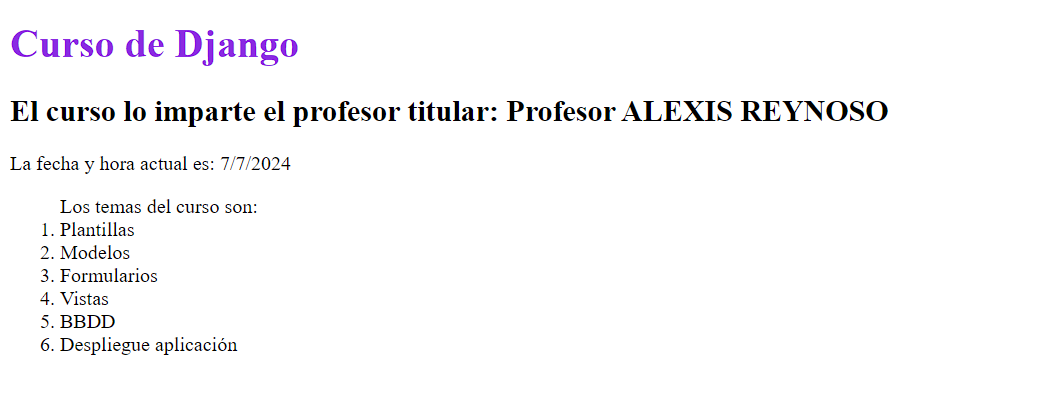
- La linea que tiene la plantilla cargada

Y modificaremos el return por medio de la función render como nos indica la documentación oficial.



        return render(request, 'index.html', ctx)

De esta manera Podemos simplificara el render de nuestro archivo.



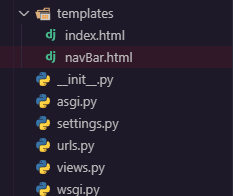
Vemos que la plantilla carga igualmente.

Ahora veremos lo que son las plantillas incrustadas o anidadas.

Trabajando con documentos web es normal que haya varios documentos html uno dentro del otro.

En este caso trabajaremos con una plantilla principal (index.html y una segunda plantilla llamada navBar.html

.



Trabajaremos dentro del archivo navBar.

<!DOCTYPE html>

<html>

    <head>

        <meta *charset*="utf-8">

        <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">

        <title></title>

        <meta *name*="description" *content*="">

        <meta *name*="viewport" *content*="width=device-width, initial-scale=1">

        <link *rel*="stylesheet" *href*="">

    </head>

    <body>

        <nav>

            <ul>

                <li>Inicio</li>

                <li>Servicios</li>

                <li>Quienes somos</li>

                <li>Contacto</li>

                <li>Acerca de</li>

            </ul>

        </nav>

    </body>

</html>

Usaremos este codigo por lo mientras.

Ahora como metemos el navBar a nuestro index, en este caso pondremos la barra de navegación al principio de todo encima del titulo.

<!DOCTYPE html>

<html *lang*="en">

<head>

    <meta *charset*="UTF-8">

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    {% include "navBar.html" %}

    <h1 *style*="color: blueviolet;">

        Curso de Django

    </h1>

    {% if nombre\_persona == 'Alexis' %}

        <h2>El curso lo imparte el profesor titular: Profesor {{nombre\_persona|upper}} {{apellido\_persona|upper}}</h2>

    {% else %}

        <h2>El profesor titular no esta presente, lo sustitura un profesor suplente</h2>

    {% endif %}

    <p>

        La fecha y hora actual es: {{fecha.day}}/{{fecha.month}}/{{fecha.year}}

    </p>

    <ol>

        Los temas del curso son:

        {% if  temas%}

            {% for tema in temas %}

                <li>{{tema}}</li>

            {% endfor %}

        {% else %}

            <p>No hay elementos que mostrar, vuelva mas tarde</p>

        {% endif %}

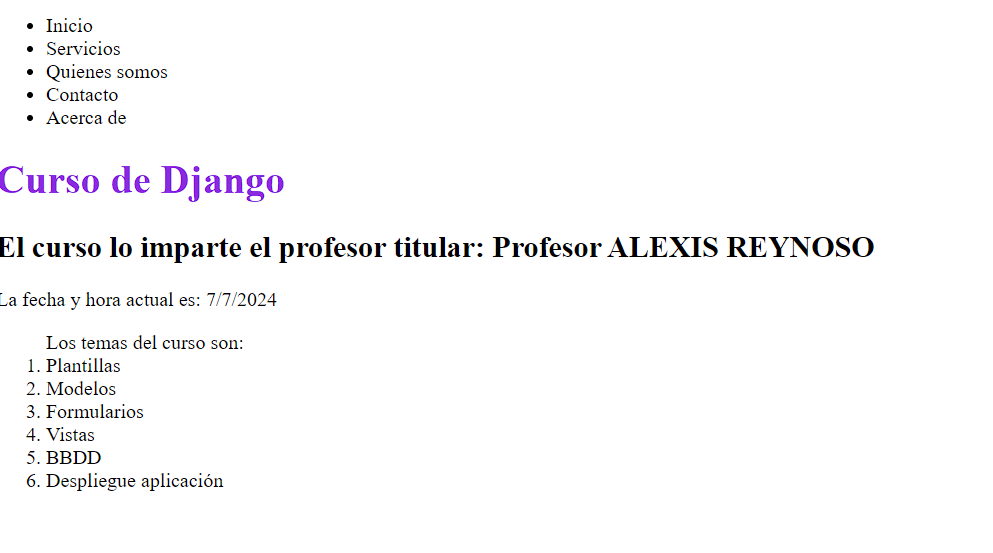
    </ol>

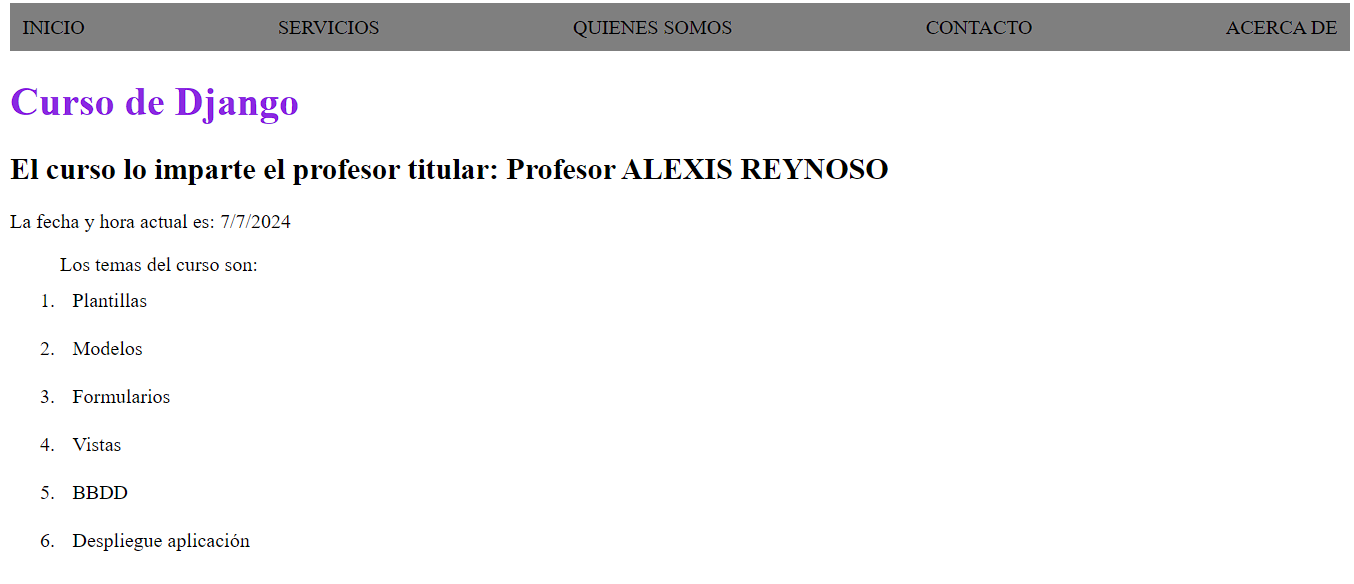
</body>

</html>

Nos apoyaremos usando la linea

{% include "navBar.html" %}



Aquí le agregamos un poco mas de estilo muy básico

Pero como agregamos archivos css, para modificar nuestras plantillas.

Primero que nada pasaremos la linea load static de django en nuestra plantilla en este caso navBar.html:

{% load static %}

<!DOCTYPE html>

<html>

    <head>

        <meta *charset*="utf-8">

        <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">

        <title></title>

        <meta *name*="description" *content*="">

        <meta *name*="viewport" *content*="width=device-width, initial-scale=1">

        <link *rel*="stylesheet" *href*="">

    </head>

    <body>

        <header>

            <nav>

                <h1>LOGO</h1>

                <ul>

                    <li><a *href*="#">INICIO</a></li>

                    <li><a *href*="#">CONTACTO</a></li>

                    <li><a *href*="#">SOBRE MI</a></li>

                    <li><a *href*="#">CURSOS</a></li>

                </ul>

            </nav>

        </header>

    </body>

</html>

Y Tambien si estamos trabajando con plantillas incrustadas como es este caso deberemos tambien ingresar esa misma linea.

{% load static %}

<!DOCTYPE html>

<html *lang*="en">

<head>

    <meta *charset*="UTF-8">

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    {% include "navBar.html" %}

    <h1 *style*="color: blueviolet;">

        Curso de Django

    </h1>

    {% if nombre\_persona == 'Alexis' %}

        <h2>El curso lo imparte el profesor titular: Profesor {{nombre\_persona|upper}} {{apellido\_persona|upper}}</h2>

    {% else %}

        <h2>El profesor titular no esta presente, lo sustitura un profesor suplente</h2>

    {% endif %}

    <p>

        La fecha y hora actual es: {{fecha.day}}/{{fecha.month}}/{{fecha.year}}

    </p>

    <ol>

        Los temas del curso son:

        {% if  temas%}

            {% for tema in temas %}

                <li>{{tema}}</li>

            {% endfor %}

        {% else %}

            <p>No hay elementos que mostrar, vuelva mas tarde</p>

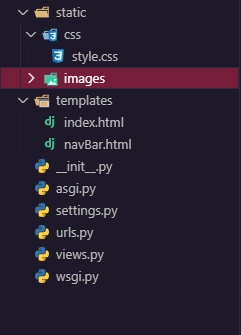
        {% endif %}

    </ol>

</body>

</html>

Ahora deberemos de crear una carpeta llama “static” donde guardaremos todos nuestros archivos estáticos, dentro de esa carpeta crearemos una subcarpeta donde guardaremos los archivos css.



Ahora para que esto funcione vamos a realizar cambios en nuestro archivo settings.py

*# Static files (CSS, JavaScript, Images)*

*# https://docs.djangoproject.com/en/5.0/howto/static-files/*

STATIC\_URL = '/static/'

Bajaremos hasta encontrar la seccion de STATIC, ahora debemos de crear la constante en caso de que no este creada STATICFILES\_DIRS con la ruta de nuestra carpeta static.

STATICFILES\_DIRS = [

    'C:/Users/Alexis/Desktop/Programacion/Backend/Frameworks/Django/Archivos Django/ProyectoDjango8/ProyectoDjango8/static'

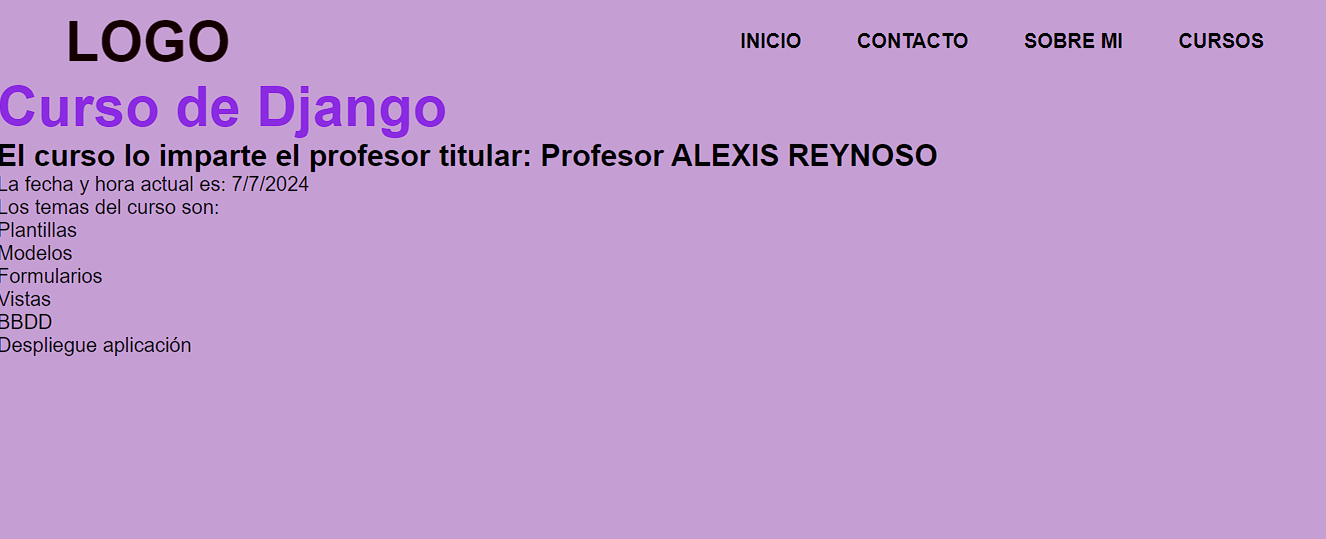
    ]

Y de esta manera al aplicar un estilo css y querer cargarlo en nuestra plantilla, basta con usar lo siguiente.

        <link *rel*="stylesheet" *type*="text/css" *href*="{% static 'css/style.css' %}">

Esta linea dentro de la sección meta que vinculara el css a nuestro archivo.

Lo mismo tendríamos que hacer si queremos en este caso trabajar con imágenes o con js.



Aquí el css

\*{

    margin: 0;

    padding: 0;

}

body{

    background-color: rgb(197, 158, 212);

    font-family: Arial, Helvetica, sans-serif;

    margin: 0;

    padding: 0;

}

nav{

    display: flex;

    align-items: center;

    justify-content: space-between;

    padding-top: 20px;

    padding-left: 5%;

    padding-right: 5%;

    font-weight: bold;

}

h1{

    color: rgb(20, 0, 0);

    font-size: 45px;

}

ul li{

    display: inline-block;

    padding: 0px 20px;

}

a{

    color: rgb(0, 0, 0);

    text-decoration: none;

}

a*:hover*{

    color: rgb(209, 41, 41);

}