Plantillas II

En este módulo veremos cómo usar variables en plantillas, acceso a objetos y propiedades desde plantillas y uso de diccionarios en contexto.

from django.http import HttpResponse

from django.template import Template, Context

def saludo(request):

    doc\_externo = open("C:/Users/Alexis/Desktop/Programacion/Backend/Frameworks/Django/Archivos Django/ProyectoDjango5/ProyectoDjango5/templates/index.html")

    plt = Template(doc\_externo.read())

    doc\_externo.close()

    ctx = Context()

    documento = plt.render(ctx)

    return HttpResponse(documento)

Para este modulo trabajaremos con este codigo y su respecitva plantilla

La plantilla html es esta:

<!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>

    <h2 *style*="color: blueviolet;">

        Hola mundo de django, esta es mi primera pagina hmtl con django

    </h2>

</body>

</html>

Primero trabajaremos con variables desde la plantilla, podemos hacer referencia a valores almacenados en variables desde nuestro codigo Python.

Por ejemplo:

    nombre = "Alexis"

Desde nuestra vista en Python vamos a declarar la variable nombre con valor Juan, ahora como utilizamos dicho valor para que se vea dentro de la página web

Para tendremos que trabajar con el contexto.

    ctx = Context()

Este contexto puede recibir diccionarios que nos permiten hacer uso de variables y objetos.

    ctx = Context({

        'nombre\_persona' : nombre

    })

De esta forma le estamos diciendo al contexto, que bajo esta clave que hemos decidido llamar nombre\_persona está el valor almacenado en la variable nombre.

Como lo trasladamos a la plantilla

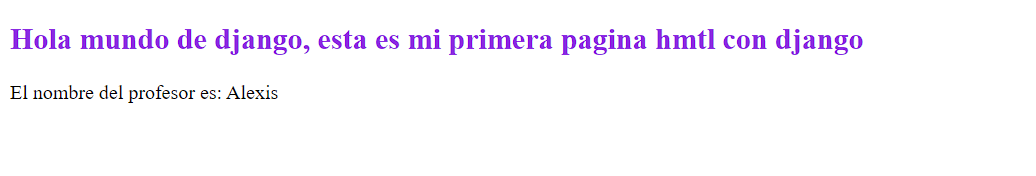
    </h2>

    <p>

        El nombre del profesor es: {{nombre\_persona}}

    </p>

En este caso almacenaremos la llave del diccionario a la cual hemos asociado el valor de la variable nombre entre dos llaves en la plantilla html, ahora si corremos el servidor y nos vamos a la página.



Como vemos está rescatando el valor de la variable.

Podemos rescatar varios valores:

def saludo(request):

    nombre = "Alexis"

    apellido = "Reynoso"

    doc\_externo = open("C:/Users/Alexis/Desktop/Programacion/Backend/Frameworks/Django/Archivos Django/ProyectoDjango5/ProyectoDjango5/templates/index.html")

    plt = Template(doc\_externo.read())

    doc\_externo.close()

    ctx = Context({

        'nombre\_persona' : nombre,

        'apellido\_persona' : apellido

    })

    documento = plt.render(ctx)

    return HttpResponse(documento)

En este caso definimos otra variable llamada apellido

    <p>

        El nombre del profesor es: {{nombre\_persona}} {{apellido\_persona}}

    </p>

En la plantilla agregamos la nueva clave dentro de doble llaves



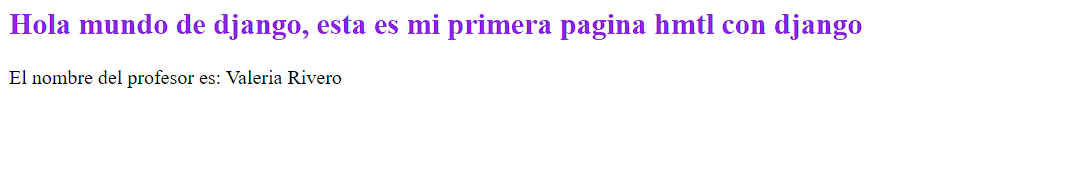
O claramente podríamos haber sustituido en el diccionario en vez de la variable haber puesto un valor directamente.

    ctx = Context({

        'nombre\_persona' : 'Valeria',

        'apellido\_persona' : 'Rivero',

    })



Esto dependerá del caso en el que nos encontremos.

Asi commo trabajamos con variables sencillas y simples, tambien podemos trabajar con clases de las librerías de Python, por ejemplo usaremos el modulo datetime.

from django.http import HttpResponse

from django.template import Template, Context

import datetime

def saludo(request):

    nombre = "Alexis"

    apellido = "Reynoso"

    fecha\_actual = datetime.datetime.now()

    doc\_externo = open("C:/Users/Alexis/Desktop/Programacion/Backend/Frameworks/Django/Archivos Django/ProyectoDjango5/ProyectoDjango5/templates/index.html")

    plt = Template(doc\_externo.read())

    doc\_externo.close()

    ctx = Context({

        'nombre\_persona' : nombre,

        'apellido\_persona' : apellido,

        'fecha' : fecha\_actual,

    })

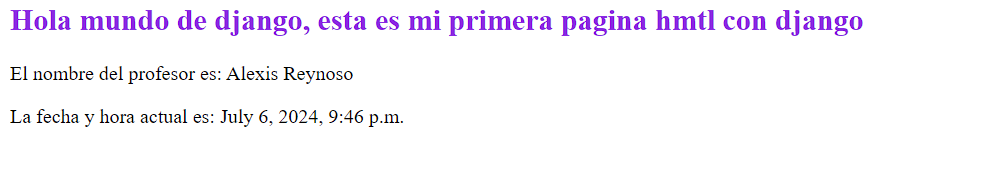
    documento = plt.render(ctx)

    return HttpResponse(documento)

    <p>

        La fecha y hora actual es: {{fecha}}

    </p>



Ahora modificaremos en este caso la fecha usando la nomenclatura de puntos y la POO.

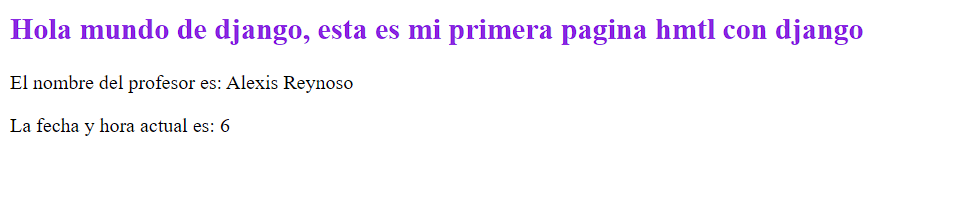
Desde la plantilla podemos usar la clave ***fecha*** seguida de un “.” Seguido de la propiedad del objeto fecha al que queremos acceder.

En este caso solo accederemos y mostraremos el día de hoy.

    <p>

        La fecha y hora actual es: {{fecha.day}}

    </p>

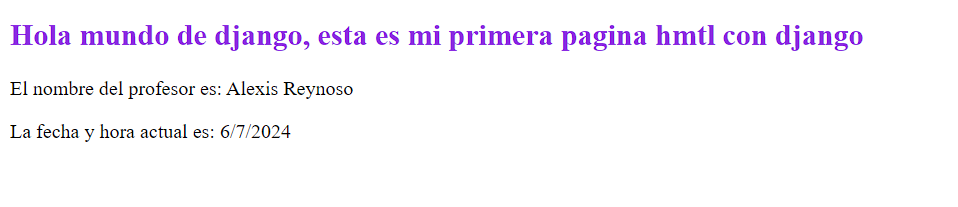


Para mostrar el dia, mes y año:

    <p>

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

    </p>



Podemos crear nuestras propias clases ya acceder a esas instancias de clase que podamos crear a futuro.

from django.http import HttpResponse

from django.template import Template, Context

import datetime

class Persona(object):

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

        self.**nombre** = nombre

        self.**apellido** = apellido

    def saludo(request):

        p1 = Persona("Valeria", "Rivero")

*# nombre = "Alexis"*

*# apellido = "Reynoso"*

        fecha\_actual = datetime.datetime.now()

        doc\_externo = open("C:/Users/Alexis/Desktop/Programacion/Backend/Frameworks/Django/Archivos Django/ProyectoDjango5/ProyectoDjango5/templates/index.html")

        plt = Template(doc\_externo.read())

        doc\_externo.close()

        ctx = Context({

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

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

            'fecha' : fecha\_actual,

        })

        documento = plt.render(ctx)

        return HttpResponse(documento)

Modificamos nuestra url

urlpatterns = [

    path('admin/', admin.**site**.**urls**),

    path('persona/', Persona.saludo)

]

