

TUTOR: LUIS GARCIA



**IEEE
COMPUTER
SOCIETY**

*Universidad de San Carlos de Guatemala
Student Branch Chapter*

CLASE 2

Taller de Flask/Django


Agenda

- Que es un Template
- Crear Templates
- Motor de template Jinja2
- Tags en Templates
- Ciclos y condicionales en Templates
- Archivos estaticos (CSS, JS, Imagenes)
- Ejemplo

TEMPLATE

Archivos que contienen datos estaticos, asi como marcadores de posicion para datos dinamicos.

Es una base prediseñada en la cual se puede personalizar la información básica, imagenes, colores, logotipo.



Sirve para separar la parte visual de la parte
logica de nuestra aplicacion

Para crear el diseño visual de nuestra aplicacion
que vera el usuario

**Para que nos
sirven?**

Como crear templates en Flask

1. Crear una carpeta llamada "templates"

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
from flask import Flask
from flask import render_template

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')


if __name__ == '__main__':
    app.run(debug = True, port= 8000)
```

2. Importar render_template



3. Retornamos el nombre del html





Es el proceso de mostrar tu pagina web en un navegador, debido a que transforma los documentos html y otros recursos de una pagina web en una representacion visual interactiva en el dispositivo del usuario.

Renderizacion





Con jinja2 podemos generar documentos HTML de
una forma rapida y sencilla

Facilita la escritura de vistas y permite separar la lógica
de un sistema de su presentacion

jinja2



```
IF condición THEN
    sentencias;
[ELSIF condición THEN
    sentencias;]
[ELSE
    sentencias;]
END IF;
```

CICLOS Y CONDICIONALES

El unico ciclo permitido por el motor Jinja2 es el ciclo **for** y tambien permite el uso de la condicional **if**

id	valor
edad	22
apellido	"Chang"
buscando	true
peso	9.35

USO DE VARIABLES

Podemos enviar objetos a nuestro templates

```
def sumar(number1, number2=20):
    print number1 # 15
    print number2 # 10
    print number1 + number2

sumar(number2=10, number1=15) # 25
```

FUNCIONES

Ademas de variables tambien es posible mandar funciones al template

QUE PODEMOS HACER EN JINJA2

1. Mandamos el valor de la variable

```
@app.route('/user/<name>')
def user(name='Eduardo'):
    return render_template('user.html', nombre=name)
```

nombre de la variable

Valor de la variable

Variables

Podemos mandar strings, enteros, funciones y listas

2. Podemos usar la variable

```
1 <html>
2 <head>
3   <title> Index </title>
4 </head>
5 <body>
6
7   <h4>Hola usuario</h4>
8   <h4> {{ nombre }} </h4>
9 </body>
10 </html>
```

3. Imprimimos el valor de la variable

Comentarios

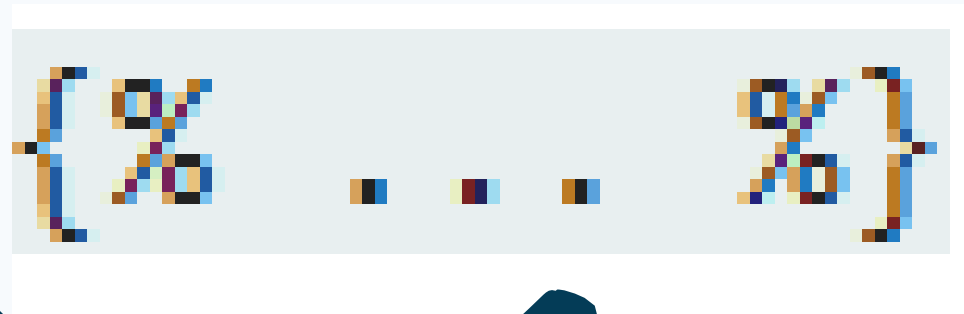
Para escribir comentarios se colocan entre :
{# comentario #}

```
{# note: commented-out template because we no longer use this  
{% for user in users %}  
...  
{% endfor %}  
#}
```

Estructuras de control

Una estructura de control se refiere a todas aquellas cosas que controlan el flujo de un programa, como condicionales y ciclos.

Dentro va el
condicional o
el ciclo



Ciclo for

Inicio del for

```
<ul>  
  {% for val in [1,2,3,4,5,6,7,8,9] %}  
    <li> {{ val }} </li>  
  {% endfor %}  
</ul>
```

Finalizacion
del for

Dentro del ciclo podemos acceder a diferentes atributos del objeto **loop**

- **index:** Iteración actual. El valor comienza en 1
- **index0:** Iteración actual. El valor comienza en 0 (ideal si deseamos replicar el comportamiento de la función enumerate)
- **first:** Valor verdadero si nos encontramos en la primera iteración.
- **last:** Valor verdadero si nos encontramos en la última iteración
- **length:** Número de iteraciones

```
{% for val in [1,2,3,4,5,6,7,8,9] %}  
    <li> {{ loop.index0 }} - {{ val }} </li>  
{% endfor %}
```

Condicional if

Es bastante facil de uiltizar ya que su sintaxis es parecido a la declaracion if de Python

```
{% if kenny.sick %}  
    Kenny is sick.  
{% elif kenny.dead %}  
    You killed Kenny!  You bastard!!!  
{% else %}  
    Kenny looks okay --- so far  
{% endif %}
```

Inicio del condicional if

condicional elif

condicional else

Finalización del
ciclo if

ARCHIVOS ESTATICOS

Este tipo de archivos tienen un trato diferente en Flask. Entre ellos estan los siguientes:



CSS

Sirven para diseñar la aplicacion



JavaScript

Agrega un comportamiento dinamico a la pagina



Imagenes

Como usarlos



```


```

 Para imagenes

```
<link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='css/style.css' ) }}">
```

 Para CSS

```
<script type="text/javascript" src="{{ url_for('static', filename='js/jquery.js' ) }}">  
</script>
```

 Para
JavaScript

1.Crear carpeta static

2.Dentro de esa carpeta puede agregar una carpeta para css, una para js y otra para imagenes

3.Colocar las rutas



Dudas

¡Muchas gracias!