

Instituto Tecnológico de Zitácuaro

Framework Flask
Python

Asesor
Dr. Eduardo López Sandoval

Framework Python

Instalar Flask:

1. Abre una terminal y ejecuta el siguiente comando para instalar Flask:

```
pip install flask.
```

Framework Python

2. Crear una estructura de archivos básica:

Crea un directorio para tu proyecto web. Llamémoslo "mi_pagina_web".

Dentro de "mi_pagina_web", crea un archivo llamado app.py. Este será el archivo principal de tu aplicación Flask.

Crea una carpeta llamada templates en el mismo directorio que app.py. Flask buscará las plantillas HTML en esta carpeta.

Framework Python

3. Configurar la aplicación Flask:

- Abre el archivo `app.py` y agrega el siguiente código:

Framework Python

```
from flask import Flask, render_template
app = Flask(__name__)
@app.route('/')
def index():
    return render_template('index.html')
if __name__ == '__main__':
    app.run()
```

Framework Python

- Importamos la clase Flask y la función `render_template` de Flask.
- Creamos una instancia de la aplicación Flask llamada `app`.
- Definimos una ruta para la página principal ("/") utilizando el decorador `@app.route()`. Cuando un usuario visite la página principal, Flask llamará a la función `index()`.
- La función `index()` retorna el resultado de `render_template()`, que carga el archivo HTML `index.html` desde la carpeta `templates`.
- Por último, ejecutamos la aplicación Flask si se está ejecutando directamente (`if __name__ == '__main__': app.run()`).

Framework Python

4. Crear la plantilla HTML:

Abre el archivo index.html dentro de la carpeta templates.

Agrega el siguiente contenido básico para tu página web:

Framework Python

```
<!DOCTYPE html>
<html>
<head>
  <title>Mi página web</title>
</head>
<body>
  <h1>Bienvenido al framework de Python</h1>
</body>
</html>
```


Framework Python

5. Ejecutar la aplicación Flask:

En la terminal, navega hasta el directorio donde se encuentra app.py (es decir, "mi_pagina_web").

Ejecuta el siguiente comando para iniciar el servidor de desarrollo de Flask:

```
python app.py
```

Framework Python

Verás un mensaje indicando que la aplicación se está ejecutando en <http://localhost:5000>.

Abre tu navegador web y visita esa dirección. Verás tu página web con el título "Mi página web" y el encabezado "Bienvenido al framework de Python".

Framework Python

- Con estos pasos, has creado un framework utilizando Flask en Python. A partir de aquí, puedes agregar más rutas, personalizar tu página web con CSS, JavaScript, bootstrap, materialice, etc. integrar bases de datos y mucho más. Flask es un framework flexible y poderoso que te permite construir aplicaciones web de cualquier tamaño y complejidad.

Administración de Framework Python

Código de la página Index.html

ejemplo codigo0.txt

```
<!DOCTYPE html>
<html>
<head>
    <title>Mi página web</title>
</head>
<body>
    <h1>Bienvenido a mi página web</h1>
</body>
</html>
```

Administración de Framework Python

Mejorar agrega CDN de Bootstrap

ejemplo codigo1.txt

```
<!DOCTYPE html>
<html>
<head>
  <title>Mi Página</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container">
      <a class="navbar-brand" href="#">Mi Página</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="col
        aria-controls="navbarNav" aria-expanded="false" aria-label="
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link" href="#">Inicio</a>
```

Administración de Framework Python

- En este ejemplo, se ha agregado una barra de navegación utilizando la clase navbar de Bootstrap. Los enlaces del menú se encuentran dentro de la lista desordenada `` con la clase navbar-nav.
- Además, se ha agregado una sección de contenido principal dentro de un contenedor con la clase container.
- Recuerda que en este ejemplo se utiliza el enlace CDN de Bootstrap para cargar los estilos CSS y el script JavaScript de Bootstrap. Asegúrate de tener una conexión a Internet activa para que los archivos se carguen correctamente.

Mejora sustancial en el diseño de la página utilizando Bootstrap
Se ha agregado un diseño más estructurado y algunos
elementos adicionales
ejemplo codigo2.txt

```
<!DOCTYPE html>
<html>
<head>
  <title>Mi Página</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container">
      <a class="navbar-brand" href="#">Mi Página</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="col
        aria-controls="navbarNav" aria-expanded="false" aria-label="
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ms-auto">
          <li class="nav-item">
            <a class="nav-link" href="#">Inicio</a>
          </li>
```

Administración de Framework Python

agregar a materialize

ejemplo codigo3.txt

```
<head>
  <title>Mi Página</title>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/materialize.min.css">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
  <nav>
    <div class="nav-wrapper blue">
      <a href="#" class="brand-logo">Mi Página</a>
      <a href="#" data-target="mobile-menu" class="sidenav-trigger"><
      <ul class="right hide-on-med-and-down">
        <li><a href="#">Inicio</a></li>
        <li><a href="#">Acerca de</a></li>
        <li><a href="#">Servicios</a></li>
        <li><a href="#">Contacto</a></li>
      </ul>
    </div>
  </nav>

  <ul class="sidenav" id="mobile-menu">
```


Administración de Framework Python

agregar rutas

```
from flask import Flask, render_template

app = Flask(__name__)

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

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

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

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

Administración de Framework Python

En este ejemplo, se han agregado tres nuevas rutas además de la ruta raíz (/):

La ruta /about muestra la plantilla about.html.

La ruta /services muestra la plantilla services.html.

La ruta /contact muestra la plantilla contact.html.

Puedes crear las plantillas correspondientes para cada una de estas rutas en el directorio de plantillas de tu proyecto Flask (templates) y personalizar el contenido de cada una según tus necesidades.

Administración de Framework Python

Recuerda que cada ruta debe tener su propia función en Python decorada con `@app.route` y devolver la plantilla renderizada utilizando la función `render_template`.

Si deseas agregar más rutas, simplemente agrega más funciones decoradas con `@app.route` siguiendo el mismo patrón.

Ver código de cada hoja

- [about.html](#)
- [services.html](#)
- [contact.html](#)

Administración de Framework Python

Estos son ejemplos básicos de documentos HTML para las rutas adicionales. Puedes personalizar el contenido y el estilo de cada página según tus necesidades.

Recuerda guardar estos documentos HTML en el directorio de plantillas (templates) de tu proyecto Flask para que puedan ser renderizados correctamente por la función `render_template`.

Documentos con Materialize

Estos documentos HTML utilizan el framework Materialize CSS para agregar estilos y componentes a las páginas. Se han incluido la barra de navegación, los estilos de los encabezados y párrafos, y los estilos de las listas, entre otros.

Recuerda incluir los enlaces a los archivos CSS y JavaScript de Materialize CSS desde el enlace CDN en la sección head del documento HTML para aplicar los estilos y funcionalidades correctamente.