

# Primeros pasos con Flask

---

Loreto Pelegrín Castillo



## FLASK

**Flask** es un microframework web para el lenguaje de programación Python. Esto significa que te proporciona las herramientas básicas para construir aplicaciones web de manera rápida y sencilla, sin la complejidad de frameworks más grandes.

<https://flask.palletsprojects.com/en/stable/>

Comprobar la versión de Python que se tiene.

- `python --version`

Si es superior a la versión 3.9, se puede usar Flask con Python.

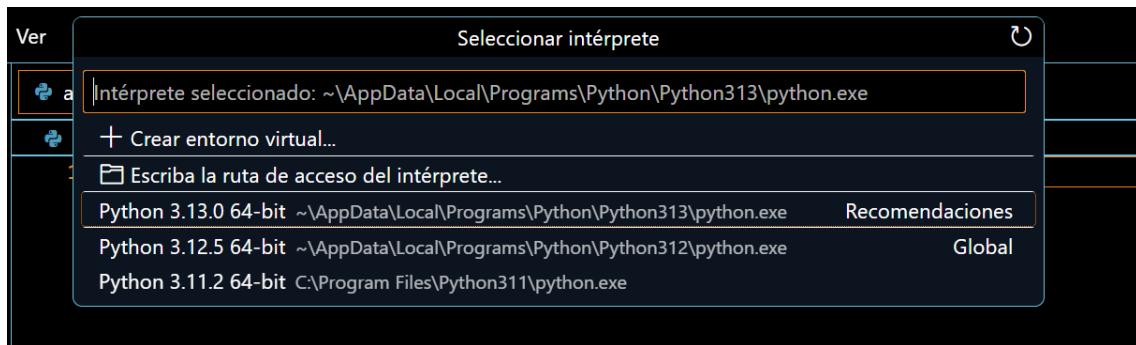
Cambiar la versión de Python.

Control + shift + p -> abre el buscador

Python: Select Interpreter o Seleccionar Intérprete.



Y seleccionamos la versión más moderna.



## Entornos virtuales

Son herramientas que se usan para gestionar las dependencias de los proyectos. Cuantos más proyectos de Python tengas, más probable es que necesites trabajar con diferentes versiones de librerías de Python, Las nuevas versiones de las librerías para un proyecto pueden romper la compatibilidad en otro proyecto.

Los entornos virtuales son grupos independientes de bibliotecas Python, uno para cada proyecto. Los paquetes instalados para un proyecto no afectarán a otros proyectos o a los paquetes del sistema operativo.

Python viene con el módulo **venv** para crear entornos virtuales.

### Crear un entorno:

```
py -3 -m venv .venv
```

### Activar un entorno

```
.venv\Scripts\activate
```

Si nos da error al activar:

```
Set-ExecutionPolicy RemoteSigned
```

Activar Script En PowerShell, entrar en modo administrador.

Para ejecutar la aplicación web asociada al entorno, esta debe de estar siempre activa.

### Desactivar un entorno

```
deactivate
```

## Instalar Flask

```
pip install Flask
```

Pip es el manejador de paquetes de Python para instalar frameworks

## Primera app

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hola():
    return "Hola Mundo"
```

## Manejo de múltiples rutas

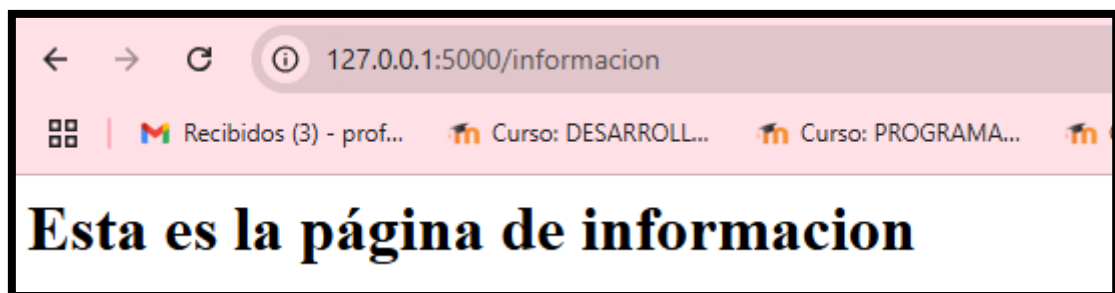
```
from flask import Flask

app = Flask(__name__)

@app.route('/')
@app.route('/index')
def hola():
    return "Hola Mundo"

@app.route('/informacion')
def informacion():
    return "<h1>Esta es la página de información</h1>"

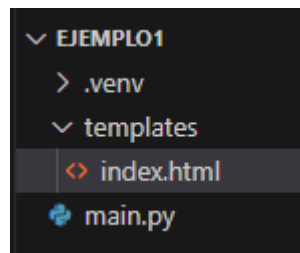
if __name__ == "__main__":
    app.run()
```



## Manejo de templates

Un template es una plantilla web prediseñada a la cual se le pueden modificar. Estas plantillas contienen código HTML, CSS y las modificamos según nuestras necesidades.

Para trabajar con templates crearemos un directorio llamado templates dentro de nuestro entorno de trabajo. En él, se almacenarán los archivos HTML. La página principal de una página web siempre se llama **index**, en este caso **index.html**.



```
templates > <> index.html > ...
1  <!DOCTYPE html>
2
3  <html lang="es">
4
5  <head>
6      <title>
7          Página de Inicio
8      </title>
9  </head>
10
11 <body>
12     <h1>Este es el primer nivel de encabezado</h1>
13     <h2>Segundo nivel</h2>
14     <h3>Tercero</h3>
15     <h4>Cuarto</h4>
16     <h5>Quinto</h5>
17     <h6>Sexto</h6>
18
19     <p>Esto es un parrafo</p>
20
21     <ul>
22         <li>Primer elemento</li>
23         <li>Segundo elemento</li>
24     </ul>
25 </body>
26
27 </html>
28
```

En el main, se importa `render_template` y en la función donde se quiera llamar al archivo `html` se le llama desde el `return`.

```
return render_template('index.html')
```

```
from flask import Flask, render_template

app = Flask(__name__)

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

@app.route('/informacion')
def informacion():
    return "<h1>Esta es la página de información</h1>"

if __name__ == "__main__":
    app.run()
```

## Enlazando a otros templates

Imaginemos que queremos crear un menú de navegación o redirigir desde la página principal a otra parte de nuestro sitio web.

Crearemos una página llamada `informacion.html`

```
from flask import Flask, render_template

app = Flask(__name__)

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

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

if __name__ == "__main__":
    app.run()
```

Para poder enlazar la pagina de inicio con la pagina de información hay que utilizar:

```
{{url_for('pagina')}}
```

```
<!DOCTYPE html>

<html lang="es">

<head>
  <title>
    Página de Inicio
  </title>
</head>

<body>
  <h1>Pagina de inicio</h1>

  <ul>
    <li>Contacto</li>
    <li><a href="{{url_for('informacion')}}">Información</a></li>
  </ul>
</body>

</html>
```

## Pagina de inicio

- Contacto
- [Información](#)

## La Información de contacto es:

**Loreto Pelegrin Castillo**

Calle Avanza 123