

Programación WEB con Python

Debido a su popularidad, Python se usa ampliamente en la Web. Python se utiliza en los mapas de Yahoo, Linux Weekly News, Shopzilla y Ultraseek, estos son solo algunos ejemplos.

Python mejorará sus habilidades generales de programación. Su consistencia, código limpio, y filosofías tomadas de la programación funcional hacen que sea un placer trabajar con ellos. También, porque Python tiene una biblioteca totalmente funcional y una comunidad activa que no tendrá ningún problema en trabajar con cosas que no entiendes. Python necesita un alojamiento algo especial, que puede costar un poco más.

Entre los frameworks que existen para desarrollar aplicaciones web en Python se encuentran:

- BFG
- CherryPy
- Django
- Pylons
- Pyramid
- Turbo-gears
- Webpy
- Web2py
- Zope

Para éste tutorial vamos a utilizar Pyramid, una fusión bastante reciente de los frameworks BFG y Pylons. Éste es uno de los más flexibles del grupo, ya que no impone restricciones a la hora de escoger componentes, y tiene muchas maneras de expandirse para proyectos grandes.

Pre-requisitos

Éste tutorial necesita para ser comprendido por lo menos saber como cambiar de directorio en la línea de comandos y algo de experiencia con algún tipo de lenguaje de programación. Éste no necesariamente tiene que ser Python, porque su sintaxis básica es lo suficientemente sencilla como para aprenderse en el momento si se conocen los fundamentos (condicionales, variables, funciones, etc) de otro lenguaje. Como no se asume experiencia previa en Python se explicarán algunas convenciones del lenguaje a medida que vayan apareciendo.

Para realizar éste tutorial se recomienda disponer de un ordenador con algún tipo de sistema UNIX/Linux instalado, dado que éstos ya vienen con Python preinstalado y su consola es más versátil. Sin embargo también es posible realizar éste tutorial en Windows, sólo que para llamar a los programas se haría necesario escribir su ruta completa para poder ejecutarlos por el shell de MS-DOS.

Pasos a seguir

Antes de empezar a programar vamos a tomar un paso extra que es recomendado antes de desarrollar aplicaciones web en Python: Vamos a crear primero un entorno virtual para trabajar, para que los cambios y programas que descarguemos no afecten a nuestra instalación principal de Python. El programa para hacerlo se llama “virtualenv”, y puede ser obtenido de dos maneras: Descargando el archivo de internet a la carpeta donde vayamos a trabajar, o instalándolo de la manera “convencional” de nuestro sistema (apt-get en las distros de Linux derivadas de Debian, por ejemplo).

```
$ sudo apt-get install python-virtualenv
```

Sin embargo, también se puede descargar un único archivo desde Internet y copiarlo en el directorio donde vayas a trabajar. Para hacerlo, se puede bajar de:

pylonsbook.com/virtualenv.py

Éste archivo lo que hace es copiar todo el contenido de nuestra instalación de Python en otra carpeta, para que utilizemos esa en vez de la convencional. Para utilizar éste archivo se usaría el comando:

```
$ python virtualenv.py --no-site-packages env
```

Ésto creará en la carpeta donde estés un directorio “env” donde se encontrará el entorno virtual. Dentro estará una copia del intérprete de Python, que será la que utilizaremos de ahora en adelante para ejecutar los comandos. Para hacerlo, sin embargo, hay que llamar a éste intérprete en vez del convencional, y para no tener que estar escribiendo todo el camino completo hacia el ejecutable vamos a hacer que el entorno virtual se integre con nuestra línea de comandos ejecutando:

```
$ source env/bin/activate
```

De ahora en adelante, verás que tu línea de comandos tiene anexado “(env)” al principio, indicando que estás utilizando el entorno virtual. Todos los comandos

que ejecutes de ahora en adelante, y sobre todo, los programas que instales desde el Python Package Index, usarán el entorno virtual y no tu instalación global de Python. Para salir del entorno virtual basta con que ejecutes:

```
$ deactivate
```

Una vez con el entorno virtual instalado y activado, una de las maneras más rápida y efectiva de instalar Pyramid es recurriendo al [Python Package Index](#) (PyPI). Éste es el sitio web donde se encuentra hospedado prácticamente todo el software que se ha escrito en éste lenguaje, y que puede ser instalado en nuestra PC con el comando `easy_install`. Es una especie de repositorio exclusivo para las librerías de Python, que permite instalarlas independientemente del sistema operativo que se posea. Para instalar pyramid basta con escribir entonces:

```
(env)$ easy_install pyramid
```

El programa `easy_install` entonces buscará automáticamente en el Python Package Index el programa que tenga el nombre indicado (pyramid en éste caso) y lo descargará e instalará en nuestro sistema junto con todos los paquetes que necesite. Aquí estamos asumiendo que se hará dentro del entorno virtual `env` creado anteriormente, si quieres ejecutar el comando `easy_install` fuera del entorno virtual en tu instalación principal de Python necesitarás permisos de super usuario con `sudo` o con `su`.

La descarga desde Internet y su instalación no debería durar mucho, como máximo tendrás que esperar 5 minutos aproximadamente con una conexión banda ancha. Con Pyramid una vez instalado, ahora en la próxima sección vamos a hacer una aplicación web de “hola mundo”, y posteriormente un sistema básico para ver marcadores y sus etiquetas guardados en una base de datos.

Una vez instalado Pyramid ya podemos pasar a probarlo. Creamos un archivo “`holamundo.py`”, y en él escribimos el siguiente código:

```
holamundo.py
```

```
from paste.httpserver import serve
from pyramid.config import Configurator
from pyramid.response import Response

def hello_world(request):
    return Response("Hola mundo!")

if __name__ == "__main__":
```

```
config = Configurator()
config.add_view(hello_world)
app = config.make_wsgi_app()

serve(app, host='0.0.0.0')
```

Descripción

- Las tres líneas del principio son los “imports”, las llamadas que hace el código para poder utilizar las distintas partes de pyramid. En la primera línea se importa la función “serve” del módulo “paste”, que es la que se encarga de ejecutar el servidor web. Los otros imports son la clase “Configurator”, que nos permite realizar las configuraciones y producir el objeto final de nuestra aplicación, y la clase “Response”, que encapsula el envío de información desde Pyramid hasta el navegador del usuario.
- La función “hello_world” es una “vista” sencilla, que muestra una página con el texto plano “Hola Mundo”. Al igual que toda vista recibe como parámetro un objeto request, que más adelante se verá que contiene la información referida a la petición (GET, POST) que el usuario realizó a nuestra aplicación web. Su única línea retorna una cadena de texto dentro de un objeto Response, que mostrará al usuario en su navegador el texto “Hola mundo!”
- La tercera y última parte del “hola mundo” inicia con `if __name__ == "__main__"`, una técnica de Python para hacer que el código que esté debajo del if sólo se ejecute si se está llamando directamente a ese archivo. Luego se crea el objeto configurator, y se le añade a través de su método `add_view` la vista “hello_world”, la que hicimos antes. Éste método sólo sirve para establecer direcciones que no reciben variables, por lo que más adelante usaremos serán las rutas. Ya que no indicamos más detalles para la vista, ésta pasa a ser la página index principal del sitio web. El siguiente paso es crear nuestra aplicación (en la variable `app`) a través del método `make_wsgi_app` del configurador, que luego pasamos a la función `serve` como primer parámetro, seguido de la cadena `'0.0.0.0'` para inicializar el servidor con nuestro sitio web.

Referencias

[La documentación oficial de Pyramid](#) (tutorial de routes-sqlalchemy)

[La documentación oficial de SQLAlchemy](#) (fundamentalmente la sección ORM)

[El tutorial de ejemplo](#) del sistema de plantillas Genshi.

<http://tecnoyucas.blogspot.mx/2011/05/programacion-web-con-python.html>

<https://docs.python.org/2/howto/webrowsers.html>

<https://www.djangoproject.com/>

<http://www.forsdelweb.com/f130/como-utilizo-python-para-hacer-pagina-web-1087037/>