



## Tutorial de Python virtualenv

PROGRAMACION

LUNES, DICIEMBRE 07 2015



FOTO POR WILLIAM WARBY. ALGUNOS DERECHOS RESERVADOS.

Cuando se está desarrollando software con **Python**, es común utilizar **diferentes versiones de una mismo paquete**. Por ejemplo, imaginemos que se está desarrollando un juego con la versión 1.2 de *Pygame* y mientras eso pasa, se comienza el desarrollo de un nuevo juego que necesita las nuevas características presentes en la versión 1.3 del mismo paquete.

En este punto ya se tiene instalada una versión del paquete en el sistema que no puede ser eliminada en favor de la nueva versión. Así que el problema a



La solución consiste en crear *virtualenvs* o entornos virtuales. En palabras

simples, un entorno virtual de *Python* es **un espacio completamente independiente de otros entornos virtuales y de los paquetes instalados**

**Cómo instalar un paquete** Esto significa que es posible instalar la versión 1.2

**Python con requirements** de *Pygame* en un entorno virtual y la versión 1.3 en otro diferente o de forma

Compartir

**Cómo instalar un paquete**

**Python con pi**

**Python con setup.py**

Para poder utilizar este simple pero poderoso concepto es necesario instalar una

utilidad que permita gestionar la creación y utilización de dichos entornos

**Tutorial de Python**

**virtualenvwrap**

**Cómo instalar la utilidad virtualenv**

Se puede instalar la utilidad *virtualenv* utilizando el gestor de paquetes de las

distribuciones Linux:

*Los comandos mostrados en este tutorial aplican para Python 2.7.*



*Archlinux*

```
$ sudo pacman -S python2-virtualenv
```



*Fedora*

```
$ sudo dnf install python-virtualenv
```

```
# Debian, Ubuntu
```

```
# (esta opción instalará Python 3 y Python 2.7)
```

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

También es posible instalar *virtualenv* utilizando **pip**:

```
# Linux, OS X
```

```
$ sudo pip install virtualenv
```

```
# Archlinux
```

```
# (si pip fue instalado usando el gestor de paquetes)
```

```
$ sudo pip2 install virtualenv
```

## Cómo crear un Python virtualenv

Para crear un *virtualenv* simplemente se ejecuta el siguiente comando desde una terminal:

```
# Linux, OS X
```



```
# (si virtualenv fue instalado usando el gestor de paquetes)
```

## ➤ Como instalar paquete

### Python con requirements

Lo que crea el directorio `mi_proyecto/` con la siguiente estructura.

## ➤ Cómo instalar un paquete

### 34 Python con pip

Compartir

```
bin/
```

## ➤ Distribución de librerías

### Python con setup.py

## ➤ Tutorial de Python Django

En el directorio `bin/` se encuentran los ejecutables necesarios para interactuar

con `virtualenv`. En `include/` se encuentran algunos archivos de cabecera de C (cuya extensión es `.h`) necesarios para compilar algunas librerías de *Python*. Y

almente en `lib/` se encuentra una copia de *Python* así como un directorio

mado `site-packages/` en el cual se aloja el código fuente de los paquetes

*python* instalados en el *virtualenv*.

## ➤ Cómo activar un Python virtualenv

Para activar un *virtualenv* o entorno virtual, se procesa el archivo `bin/activate`

que se encuentra en la carpeta que se ha creado al ejecutar la utilidad *virtualenv*:



```
cd mi_proyecto
$ source bin/activate
(mi_proyecto)$
```

El prompt de la terminal indica que el *virtualenv* está activado. De esta manera ya es posible utilizar los paquetes *Python* instalados así como instalar paquetes adicionales.

## ➤ Cómo desactivar un Python virtualenv

Para desactivar un *virtualenv* porque se necesita trabajar en otro diferente, por ejemplo, se ejecuta el comando `deactivate`. No es necesario ir a la carpeta del *virtualenv* para realizar la operación:

```
(mi_proyecto)$ deactivate
$
```

El prompt de la terminal indica que el *virtualenv* ha sido desactivado con éxito.

## ➤ Cómo instalar paquetes en un Python virtualenv

Después de activarlo, lo único que resta es instalar los paquetes que sean



Por ejemplo, si se desea instalar **django**, se ejecuta el siguiente comando:

### ➤ Cómo instalar paquete

**Python con requirements**

### ➤ Cómo instalar un paquete

34

**Python con pip**

Compartir



**Distribución de librerías**



**¿en qué directorio ubico el código fuente de mi proyecto?**



**Tutorial de Python Django**



**La ubicación del código fuente del proyecto en el que se está trabajando no**

**importante. Puede ser colocado inclusive dentro del directorio del *virtualenv*.**



**virtualenvwrap** está activado, todas las librerías de Python que instalen



lo podrán ser usadas al activar ese *virtualenv* específico.



CRITO POR JOSE MIGUEL VENEGAS MENDOZA. ETIQUETADO CON PIP, PYTHON, VIRTUALENV.





Recomendar 4

Share

Ordenar por los mejores

## ▶ Cómo instalar paquete Python con requirements

## ▶ Cómo instalar un paquete Python con pip

34 **Ariel Gustavo Sebastián Benite** • hace 17 días

Compartir

## ▶ Distribución de librerías Python con Setup.py



Hola,  
Tengo una Consulta.  
Resulta que he creado, usando esta guía, un virtualenv para un proyecto de python + Django y he trabajado bien.  
Pues resulta que al actualizar la distro se ha actualizado la versión de python y ya no he podido ejecutar el proyecto ya que al tratar de hacerlo aparentemente no se encuentra python3 dentro del directorio.

Al Ejecutar python3 **manage.py** runserver

Obtengo el siguiente error:  
python3: error while loading shared libraries: libpython3.4m.so.1.0: cannot open shared object file: No such file or directory

Alguna Sugerencia

^ | v • Reply • Share >



**Jose Miguel Venegas Mendoza** Mod ➔ Ariel Gustavo Sebastián Benite  
• hace 4 días

Intenta agregando /usr/local/lib y posiblemente /usr/local/lib64 al archivo /etc/ld.so.conf y luego ejecuta ldconfig.

Fuentes:

<http://serverfault.com/questions/1000000/python3-error-while-loading-shared-libraries>

<http://stackoverflow.com/questions/1000000/python3-error-while-loading-shared-libraries>

^ | v • Reply • Share >



**Jose Miguel Venegas Mendoza** Mod ➔ Ariel Gustavo Sebastián Benite  
• hace 4 días

Hola Ariel!

¿Que versión de Python estabas utilizando antes de actualizar la distro?

^ | v • Reply • Share >



**Cueto re-cabron** • hace 6 meses

Excelente

^ | v • Reply • Share >



## Categorías

### ▶ Cómo instalar paquet

Python con requirements

Desarrollo de videojuegos



### ▶ Cómo instalar un paqu

34 Python con pi

Compartir: Desarrollo web



### ▶ Distribución de librer



Python con setup.py

Entos de tecnología



Tutorial de Python Dja



Ludum-Dare de Pythc

virtualenvwrapp



ogramación



## ▶ Etiquetas



DOTSTRAP

CAMPUS-PARTY

CSS

DJANGO

GRUNT-JS

HAXE

HAXEPUNK

HTML



TML5

JAVASCRIPT

LUDUM-DARE

NODE-JS

PIP

POSTGRESQL

POSTMORTEM

PYPI

PYTHON

REACT-JS

REQUIREMENTSTXT

VIDEOJUEGOS

VIRTUALENV

VIRTUALENVWRAPPER



## Acerca de

## 🔗 Cómo instalar paquetes Python con requirements

Rukbottoland es un espacio creado para que puedas aprender desarrollo web y desarrollo de videojuegos. Aquí podrás encontrar artículos sobre programación en Python, Haxe y JavaScript.

## 🔗 Cómo instalar un paquete

### 34 Python con pip

Compartir

#### 📄 Distribución de librerías

#### 📄 Python con setup.py

#### 📄 Tutorial de Python Django



#### Tutorial de Python con virtualenvwrapper



## Secciones

[🏠 Inicio](#)[📄 Blog](#)[🎮 Juegos](#)

## Síguenos

[G Google+](#)[🐦 Twitter](#)[f Facebook](#)

Con el poder de Wok, tema gracias a Bourbon.io.

[Política de privacidad](#) • [Términos y condiciones de uso](#)

Copyright © 2011 - 2016. Todos los derechos reservados.