

Crear Entornos Virtuales en Python

por [Albert Brugués](#) | Publicada [28/04/2019](#)

Un entorno virtual es un espacio donde podemos instalar paquetes específicos para un proyecto. Es decir, permite tener para un proyecto determinado un conjunto de paquetes/librerías aislados de la instalación principal de Python en nuestro sistema. Una razón para usar entornos virtuales es si por ejemplo tenemos muchos proyectos que utilizan una versión específica de una librería, pero queremos probar una versión más reciente de dicha librería sin crear errores en nuestros proyectos existentes.

Para la creación de entornos virtuales se recomienda utilizar el módulo [venv](#), el cual viene instalado por defecto con la librería estándar de Python desde la versión 3.3. Por otro lado, la instalación de paquetes se realiza mediante la herramienta [pip](#), la cual viene incluida por defecto a partir de Python 3.4. Como los comandos que vamos a ver en este post utilizan estas dos herramientas, se asume que se trabaja como mínimo con la versión 3.4.

Índice

1. Lista de comandos

1.1. Comandos relacionados con la gestión de entornos virtuales

1.2. Comandos relacionados con la gestión de paquetes

2. Buenas prácticas con los entornos virtuales

Lista de comandos

Los siguientes comandos son válidos para el símbolo del sistema (CMD) de Windows, y la aplicación de terminal de macOS y Linux.

Comandos relacionados con la gestión de entornos virtuales

- Crear un entorno virtual nuevo

En Windows, y asumiendo que Python está incluido dentro de las variables del sistema:

```
C:\>python -m venv c:\ruta\al\entorno\virtual
```

En macOS y Linux:

```
$ python -m venv ruta/al/entorno/virtual
```

Es recomendado que la carpeta para el entorno virtual sea una subcarpeta del proyecto Python al que está asociado.

- Activar un entorno virtual

En Windows:

```
C:\>c:\ruta\al\entorno\virtual\scripts\activate.bat
```

Este sitio web utiliza cookies para mejorar tu experiencia de navegación. Si sigues navegando estás aceptando su uso.

■ Desactivar un entorno virtual

Este comando es idéntico para Windows, macOS y Linux:

```
$ deactivate
```

■ Eliminar un entorno virtual

En Windows:

```
C:\>rmdir c:\ruta\al\entorno\virtual /s
```

En macOS y Linux:

```
$ rm -rf ruta/al/entorno/virtual
```

Eliminar un entorno virtual es tan sencillo como eliminar la carpeta que lo contiene. Por ello, esta operación también se puede realizar desde el correspondiente administrador de archivos.

Comandos relacionados con la gestión de paquetes

La gestión de los paquetes instalados, sea o no en un entorno virtual, se realiza mediante la herramienta pip. Esta herramienta obtiene los paquetes que le mandamos instalar del [Python Package Index](#). Para que los siguientes comandos funcionen bajo Windows, se asume que Python está incluido en las variables del sistema.

■ Instalar una nuevo paquete

```
$ pip install paquete
```

Siendo paquete el nombre del paquete/librería a instalar. Este comando instala también automáticamente las dependencias del paquete que deseamos instalar.

■ Eliminar un paquete

```
$ pip uninstall paquete
```

Siendo paquete el nombre del paquete/librería a desinstalar. Este comando no desinstala cualquier dependencia del paquete que desinstalamos.

■ Listar los paquetes instalados

```
$ pip list
```

Existe otro comando que lista los paquetes en formato del archivo *requirements.txt*. Este archivo contiene un listado de las versiones de los paquetes necesarios para ejecutar un proyecto en Python.

```
$ pip freeze
```

En sistemas Windows podemos copiar la salida de este comando, pegarla en un editor de notas como [Atom](#) o [Sublime Text](#) (que también podemos usar como nuestro entorno de programación), y guardarla manualmente con el nombre *requirements.txt*. Sin embargo, si usamos macOS o Linux podemos guardar la salida directamente en un fichero de texto.

```
$ pip freeze > requirements.txt
```

Este sitio web utiliza cookies para mejorar tu experiencia de navegación. Si sigues navegando estás aceptando su uso.

[Configuración](#)

ACEPTO

Existen dos consideraciones adicionales a tener en cuenta respecto a los entornos virtuales:

- No se debe añadir manualmente ningún fichero dentro de una carpeta que almacena un entorno virtual. Esta carpeta sólo debe contener los ficheros que se crean por defecto al crear el entorno virtual y los paquetes adicionales que hayamos instalado con el comando *pip*.
- La carpeta del entorno virtual no debe incluirse a nuestra herramienta de control de versiones. En lugar de ello es recomendado añadir un fichero *requirements.txt* con el listado de los paquetes necesarios para ejecutar el proyecto.00

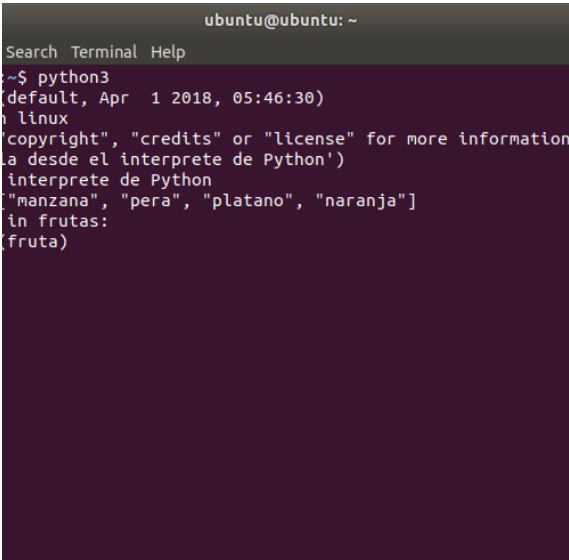
AUTOR



Albert Brugués

Soy ingeniero y un apasionado de la tecnología y las nuevas oportunidades que brinda. Más en particular me encanta la programación y la inteligencia artificial. En este blog pretendo compartir los conocimientos de Python que he ido adquiriendo a lo largo de los años.
16 ENTRADAS

TAMBIÉN TE PUEDE INTERESAR



Publicada [01/04/2019](#)
[Expresiones Regulares](#)

Publicada [02/02/2019](#)
[Entornos de desarrollo para programar en Python](#)

Este sitio web utiliza cookies para mejorar tu experiencia de navegación. Si sigues navegando estás aceptando su uso.

[Configuración](#) ACEPTO

Programa en Python



Deja un comentario

Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con *

COMENTARIO

*NOMBRE

*CORREO ELECTRÓNICO

WEB

POR FAVOR, INTRODUCE UNA RESPUESTA EN DÍGITOS:

cinco + veinte =

PUBLICAR EL COMENTARIO

< [SENTENCIAS CONDICIONALES EN PYTHON](#)



[BUCLES FOR EN PYTHON](#) >

© 2020 [Programa en Python](#) – Todos los derechos reservados

Funciona con – Diseñado con el [Tema Customizr](#)

- Aviso Legal
- Política de Cookies
- Política de Privacidad

Este sitio web utiliza cookies para mejorar tu experiencia de navegación. Si sigues navegando estás aceptando su uso.

[Configuración](#)

ACEPTO