

Entornos de Python: común y Anaconda (base)

Este artículo está enfocado tanto en quien empieza a programar en Python desde lo básico como para los profesionales que deseen ponerse al día en entornos de Python. Está enfocado como si fuera un curso **para entender y para terminar sabiendo usar el entorno de Python tanto el común como base de Anaconda para comprender claramente la diferencia entre ambos** (los entornos virtuales de ambos los veremos en este otro artículo (<https://jarroba.com/entornos-virtuales-de-python-comun-y-anaconda/>)), y como se trabaja con los entornos de Python actualmente en el ámbito profesional.

Aunque instalar Python y ponerse a programar es muy sencillo, **hoy día** considero de expresa importancia **tener muy claro los diferentes entornos de Python** (comunes, base, virtuales, etc.) **tanto para trabajar holgadamente con los diferentes entornos de Python sin que nos pille nada de sorpresa** (por ejemplo ¿Qué es “base”? o ¿Qué es “venv”? se responderá a esto más adelante) **como para saber el alcance de cada uno** (Por ejemplo ¿Por qué no funciona un paquete que he instalado? Puede que sea porque está en otro entorno diferente dentro del mismo ordenador; o ¿Por qué se ejecuta con un Python antiguo si tengo instalado el Python más moderno? Por la configuración de los diferentes entornos; etc.).

Entornos de Python

Cuando instalemos Python (explico cómo se instala más adelante) de alguna de las maneras disponibles, **tendremos en nuestro ordenador el ejecutable “python.exe”** (pese a que en este ejemplo pongo “.exe” de Windows, si utilizas Linux o Mac funciona de manera semejante), que **es el famoso “Interprete de Python”**.

El intérprete de Python se puede llamar directamente desde la consola escribiendo “python” (ya que al instalar Python se habrá creado una variable de entorno del sistema apuntando hasta este ejecutable “python.exe”). Este ejecutable ejecutará Python en alguna de sus versiones (Python 2.8, Python 3.7, etc.); es decir, dependiendo de la versión de Python instalada, el mismo método puede estar más o menos optimizado, y se permitirá una sintaxis u otra.



Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR

Por otro lado, dentro de la misma instalación de Python (dentro de la carpeta donde están todos los archivos de Python) tendremos las carpetas con un montón de archivos que hacen funcionar a Python, siendo la más interesante la carpeta **“Lib”** (en Linux las bibliotecas se guardan de otra manera, pero el concepto es el mismo; esto lo veremos más adelante para cada uno de los sistemas operativos); en esta carpeta es **donde estarán todas las bibliotecas preinstaladas de Python disponibles listas para que las usemos** (como por ejemplo la biblioteca Math, que sirve para poder realizar operaciones matemáticas, como la raíz cuadrada, de una manera muy sencilla; entre otras muchas bibliotecas) y aquí se guardarán otras bibliotecas que podremos instalar en un futuro.

Si quisiéramos ampliar el contenido de la carpeta **“Lib”** podríamos descargar a mano alguna de las bibliotecas (llamadas “paquetes” en Python) y pegarlas dentro de esta carpeta **“Lib”**; sin embargo, mejor **utilizar “un sistema de gestión de paquetes”** que funcionará algo así como “una tienda de aplicaciones” gratuitas y nos ahorrará mucho trabajo ya que si una biblioteca tiene dependencias (que necesita de otras bibliotecas para funcionar), automáticamente instalará todo lo que necesite para trabajar directamente con la biblioteca sin más preocupaciones. El “sistema de gestión de paquetes” más utilizado en Python -y que suele venir preinstalado- es el llamado Pip y la página donde podremos consultar todas las bibliotecas es <https://pypi.org/> (<https://pypi.org/>); por lo que, **Pip se encargará de gestionar por nosotros el contenido de la carpeta “Lib” a la que esté apuntando** (más adelante explico su instalación y uso completo con ejemplos).

Este es el **“entorno de Python”**, es decir, todo aquello que está al alcance de nuestro proyecto/código Python, **como las bibliotecas que necesiten nuestro código de Python** (de la carpeta **“Lib”**) **y el intérprete de Python que ejecute a nuestro código de Python** (el ejecutable **“python.exe”**) en concreto.

Por ejemplo, en nuestro ordenador escribimos un código de Python que usa la biblioteca Math ya que realizamos operaciones matemáticas; en el futuro, queremos copiar nuestro script a otro ordenador (que podrá tener el mismo entorno u otro diferente) que tiene instalado Python y lo queremos ejecutar, pero resulta que justo la biblioteca Math no la tiene ya que quien gestionaba ese ordenador decidió eliminarla por motivos de espacio (esto puede ser un servidor que tengan un “entorno de Python” peculiar), ocurrirá que nuestro código no se podrá ejecutar en este otro ordenador y nos devolverá un error (tendremos varias opciones: instalar la biblioteca Math si se puede; o muy a las malas, tendremos que crear/copiar el código de la operaciones que usemos de Math a mano).

Hoy día hay que diferenciar entre los **tipos de “entorno de Python”** que podremos utilizar en un mismo ordenador:

- **Global (base):** es el entorno base que afecta a todos los proyectos (aunque no tiene porqué ser utilizado directamente por un proyecto).
- **Particular por conjunto de proyectos:** se puede crear un entorno (virtual) que pueden usar uno o varios proyectos a la vez.
- **Particular por proyecto:** entorno al que solo tiene acceso un único proyecto.

Como hemos dicho y como resumen rápido, dependerá del ejecutable Python (**“python.exe”**) y las bibliotecas (**“Lib”**) a las que tiene acceso un proyecto.

Entorno global (base)

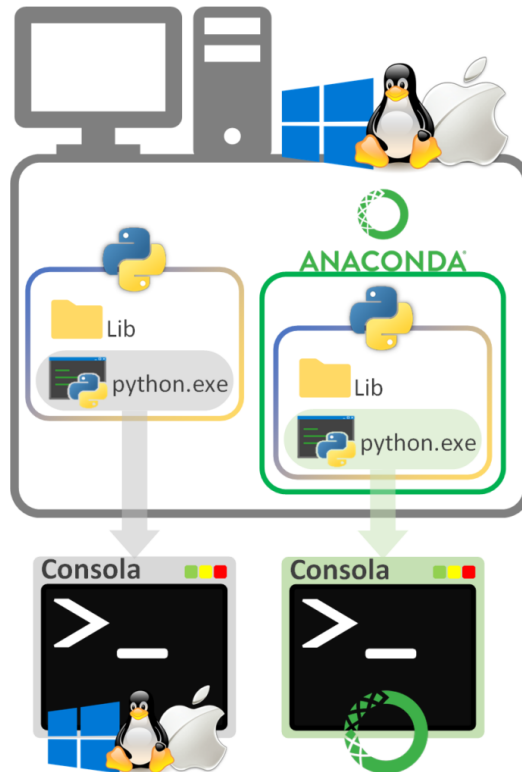
Hoy día es muy común encontrarnos con estos **tipos de entornos globales de Python**:

- **Python común** (en inglés **“stock Python”**): Es el Python que se instala directamente en el sistema operativo y se puede usar el intérprete de Python directamente desde la consola del mismo sistema operativo.

(<http://jarroba.com/faq/>)

ACEPTAR

- **Python gestionado por un “sistema de gestión de paquetes”** (como Anaconda): Es un Python gestionado por el gestor de paquetes y cuyo intérprete de Python no está accesible desde la consola normal del sistema; requiere acceder desde la consola del gestor de paquetes (por ejemplo, desde la consola de Anaconda y que además se llama “base”).



Se pueden tener ambos instalados a la vez (“Python común” y Python gestionado por un “sistema de gestión de paquetes”), pero cada intérprete de Python estará por separado (son dos programas distintos, aunque los dos sean Python e incluso puedan coincidir la misma versión) y deberemos tener claro con qué entorno queremos ejecutar nuestro código cada vez si tenemos ambos instalados.

También **se pueden tener varios “Python comunes”** a la vez en diferentes rutas del sistema operativo (esto se suele hacer para tener varias versiones de Python en el mismo ordenador).

A continuación, cuento como se instala Python común y Python gestionado por un “sistema de gestión de paquetes” Anaconda (Si ya lo tienes instalado te puedes saltar estos puntos). Es importante instalar el que necesites, si dudas recomiendo instalar Anaconda, y si quieres entenderlo todo bien es mejor que instales ambos para aprender de los dos.

Instalar un IDE

Si instalas algún IDE (como Visual Studio, PyCharm, Spyder, Jupyter, etc.) por defecto apuntará a un entorno global, que será el que ejecute, aunque se podrá cambiar desde el propio IDE y hasta nos permitirá configurar diferentes entornos. E incluso algunos pueden traer su propio entorno de Python (que suelen instalar en la misma carpeta de instalación del IDE).

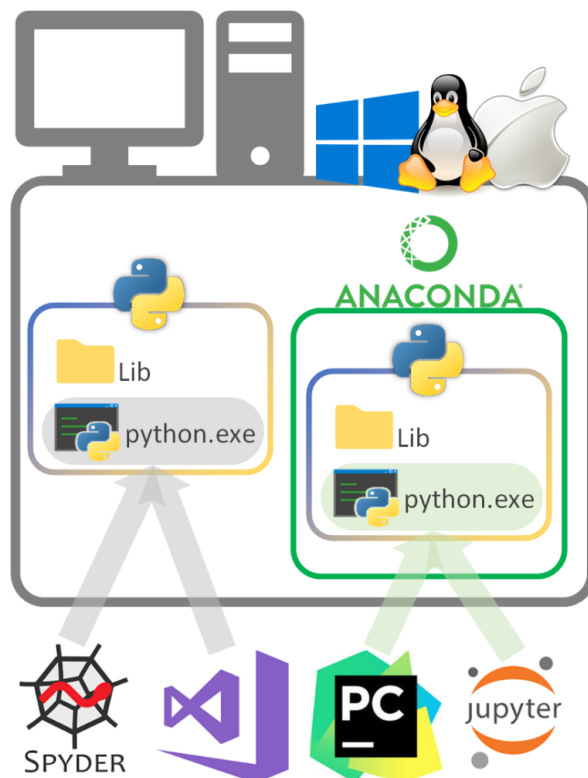
Realmente cada IDE por debajo abre una consola y escribe los comandos que se ven en este artículo, así que dependiendo de la consola que haya abierto el IDE ejecutará un entorno u otro (e igualmente se aplica a los entornos virtuales que ya veremos).

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR



Si quieres saber más y cómo se instala uno (PyCharm en este caso) te recomiendo que eches un vistazo al vídeo que te hemos preparado en:

0.2-Instalacion IDE PyCharm con Anaconda



Instalar Python Común

En **Windows desde la Microsoft Store**: <https://www.microsoft.com/en-us/p/python-38/9msszt1n39l>
 (<https://www.microsoft.com/en-us/p/python-38/9msszt1n39l>) (para la versión 3.8, puede que haya salido otra más moderna que puedes consultar directamente en la búsqueda: <https://www.microsoft.com/en-us/search?q=python> (<https://www.microsoft.com/en-us/search?q=python>) o desde la **web oficial** <https://www.python.org/downloads/> (<https://www.python.org/downloads/>)

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR

Nota sobre los instaladores 32 bits y 64 bits: Asegúrate si estás trabajando en un sistema de 32 bits en instalar Python de 32 bits; lo mismo si estás trabajando con la versión del sistema operativo de 64 bits, que instales Python de 64 bits (confundir o mezclar las instalaciones nos dará error cuando queramos que funcione Python).

En **Linux** (**suele venir preinstalado** al menos en Ubuntu; puedes comprobarlo con el comando: `python3 -V`) se puede **instalar con APT** mediante el comando (comprueba la versión antes de instalar en <https://www.python.org/downloads/> (<https://www.python.org/downloads/>)):

```
sudo apt install python3.8
```

En **Mac** suele venir preinstalado o directamente es más fácil **usar el “sistema de gestión de paquetes” Anaconda**.

Se puede **comprobar la instalación de Python** en cualquier sistema operativo preguntando por la versión de Python desde la consola:

```
python3 -V
```

O bien:

```
python -V
```

Nota sobre estos dos comandos: depende de la variable de entorno creada será “python3” o “python” a secas. Tiende a confundir, aunque hoy día es lo mismo (aunque seguramente solo existirá una en tu ordenador, ojo con esto). Como curiosidad cuando convivían Python2 y Python3 se utilizaba el comando “python” para Python2 (también se veía por ahí como “python2”) y “python3” para Python3 ya que estaban los dos instalados a la vez sobre el mismo ordenador (el motivo era para que no dejaran de funcionar scripts de Python antiguos y poder programar con el Python nuevo); hoy día con Python3 siendo único (Python2 ya no tiene soporte) es indiferente como se llame, pero hay que saber cómo se llama (y puede que todavía te encuentres con algún sistema que tenga las dos variables), suele tender a ser otra vez “python” a secas ya que es más fácil.

Rutas por defecto donde se instala el **ejecutable de Python** (“VV” es la versión de Python instalada):

- **Windows:** C:\Users\<usuario>\AppData\Local\Programs\Python\PythonVV-VV\python.exe
- **Linux/Mac:** /usr/bin/pythonVV

Rutas por defecto donde está la carpeta “**Lib**”:

- **Windows:** C:\Users\<usuario>\AppData\Local\Programs\Python\PythonVV-VV\Lib\
- **Linux/Mac:** /usr/lib/pythonVV

Información adicional sobre la instalación:

- **Windows:** <https://docs.python.org/3/using/windows.html> (<https://docs.python.org/3/using/windows.html>)
- **Unix:** <https://docs.python.org/3/using/unix.html> (<https://docs.python.org/3/using/unix.html>)
- **Mac:** <https://docs.python.org/3/using/mac.html> (<https://docs.python.org/3/using/mac.html>)

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

Instalar “sistema de gestión de paquetes” para Python: Anaconda

Descargar desde <https://www.anaconda.com/products/individual> (<https://www.anaconda.com/products/individual>) e instalar.

Instalación paso a paso de Anaconda en vídeo en:



Rutas por defecto donde se instala el **ejecutable de Python** ("VV" es la versión de Python instalada):

- **Windows:** C:\Users\<usuario>\Anaconda3\python.exe
- **Linux:** /home/<usuario>/anaconda/bin/python
- **Mac:** /Users/<usuario>/anaconda/bin/Python

Rutas por defecto donde está la carpeta "**Lib**":

- **Windows:** C:\Users\<usuario>\Anaconda3\Lib\
- **Linux:** /home/<usuario>/anaconda/lib/pythonVV
- **Mac:** /Users/<usuario>/anaconda/lib/pythonVV

Información adicional sobre la instalación: <https://docs.anaconda.com/anaconda/install/>

(<https://docs.anaconda.com/anaconda/install/>)

Probar un entorno de Python programando con Python

Maneras de programar en Python:

- **Con el modo interactivo:** Nos permitirá escribir código directamente y ejecutar cada una de las líneas. La limitación que tiene es que está en memoria RAM, es decir, que cuando cerremos la consola el código se perderá.
- **Por fichero:** Creamos un fichero con código Python dentro y lo ejecutamos. Normalmente se pone la extensión ".py" al fichero. Como es un fichero guardado en disco no perderemos el código de Python que escribamos si se cierra la consola (o si apagamos el ordenador) y podremos copiarlo a otros sitios (por ejemplo, a otros ordenadores).

Para ambas, lo primero que necesitamos es abrir una consola

Uso de cookies

Este sitio web utiliza cookies para mejorar tu experiencia. Al continuar, aceptas el uso de cookies. Para más información, consulta nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

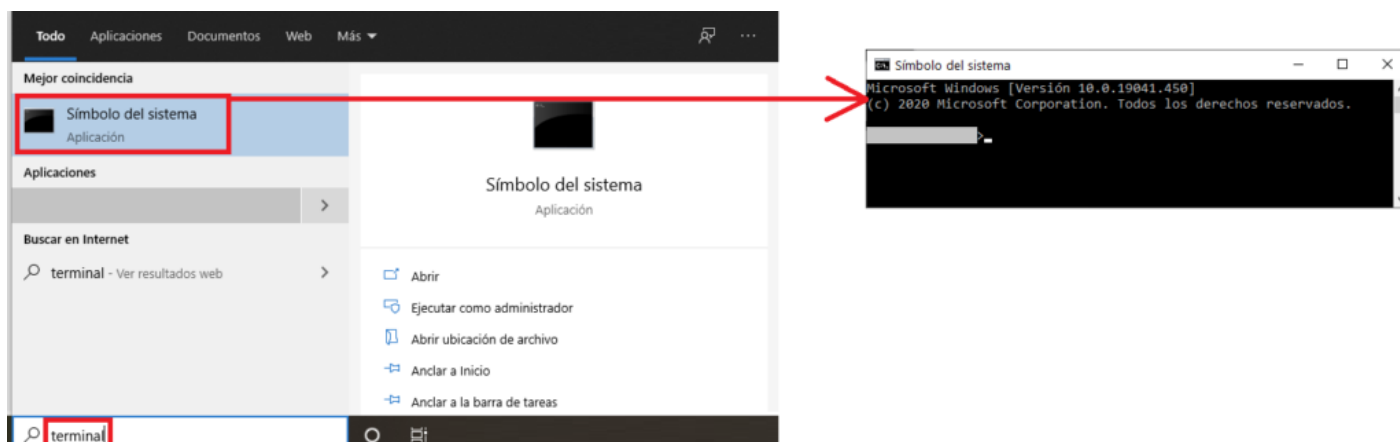
(<http://jarroba.com/faq/>)

ACEPTAR

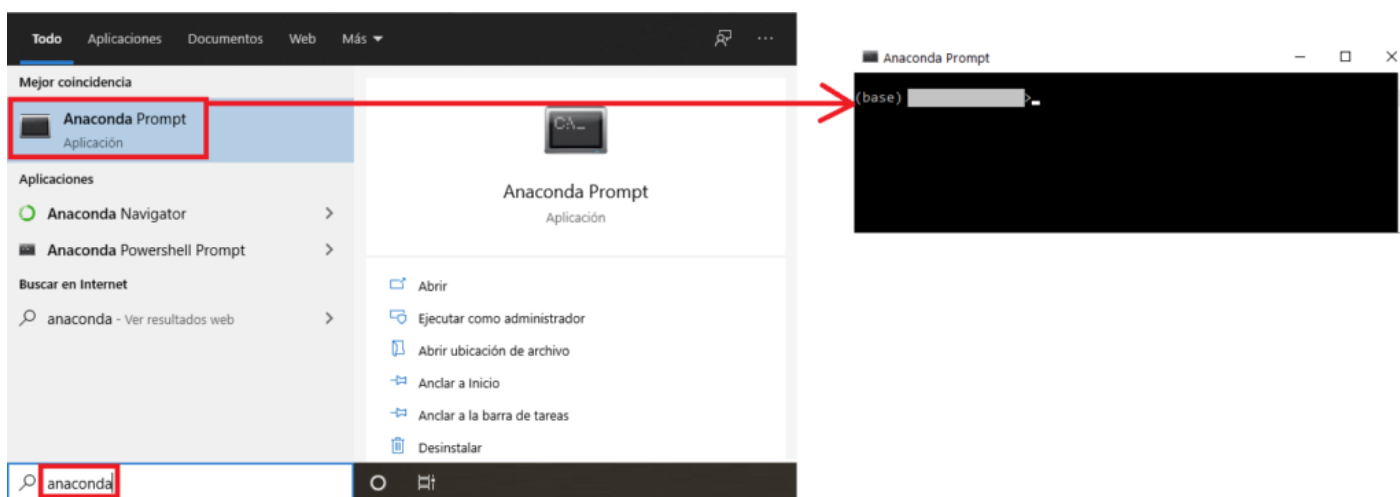
1. Abrir una consola

Abrimos la consola del sistema o la de Anaconda.

Para **abrir la consola del sistema operativo simplemente abrimos una nueva terminal**. En cualquier sistema operativo **buscamos en su buscador la palabra “terminal”** y la ejecutamos para abrirla (la siguiente imagen es del buscador de Windows; se aplica igual al resto de sistemas operativos con interfaz visual: Mac, Ubuntu, etc.).



O si queremos **abrir la terminal de anaconda**, escribiendo “Anaconda” en el buscador del sistema operativo nos mostrará uno que se llama “Anaconda prompt”.



Nota sobre la terminal de Anconda: verás que antepone a la ruta entre paréntesis la palabra “(base)”, esto indica que el código Python que escribamos en esta consola serán interpretadas por el Python base (Python gestionado por Anaconda); como continuación a este artículo tienes el artículo sobre Entornos virtuales de Python (<https://jarroba.com/entornos-virtuales-de-python-comun-y-anaconda/>) para ver cómo cambiar desde este entorno a uno virtual y aprender a gestionarlo. Quiere decir que Anaconda ya nos gestiona el entorno de Python y por defecto no nos dejará ejecutar código directamente sobre su instalación global de Python.

Recomendación para aprender: recomiendo probar ambas consolas («Anaconda prompt» y la consola del sistema).

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

2. “modo interactivo” de Python

ACEPTAR

En cualquiera de las terminales entra en el **“modo interactivo” de Python** escribiendo en la consola “python” y pulsando la tecla “Intro/Enter”:

```
python
```

Si no te funciona, prueba a escribir (como se comentó anteriormente, dependerá de la variable de entorno creada):

```
python3
```

Habrás entrado en el intérprete de Python (en este momento la consola ya no estará controlando al sistema; es decir, podrás escribir código Python, pero no podrás navegar por los directorios de tu ordenador hasta que salgas del intérprete de Python). Detrás de los tres símbolos mayor que “>>>” podrás escribir código Python línea a línea.

Prueba a escribir un par de líneas de código en Python como imprimir un texto y ejecuta la línea de código pulsando la tecla “Intro/Enter”:

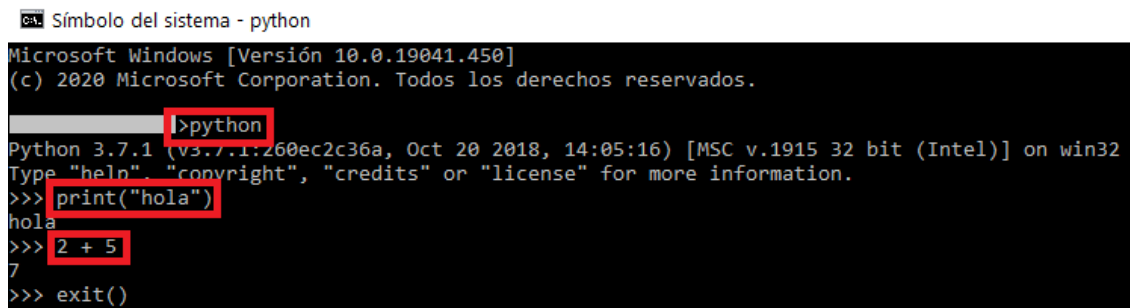
```
print("hola")
```

O suma un par de números:

```
2 + 5
```

Para salir del intérprete de Python escribe:

```
exit()
```



CA. Símbolo del sistema - python

```
Microsoft Windows [Versión 10.0.19041.450]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\>python
Python 3.7.1 (tags/v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("hola")
hola
>>> 2 + 5
7
>>> exit()
```

Para probar las bibliotecas globales (las que están en la carpeta “Lib”) podemos importar por ejemplo “Math”. Entra otra vez en el intérprete de Python y escribe:

```
from math import sqrt
```

Y prueba a realizar la raíz cuadrada de 9 (en la imagen siguiente te muestro que obtendrás un error si intentas utilizar este método antes de importar, y después de importar funcionará correctamente):

```
sqrt(9)
```



```

C:\> python
Python 3.7.1 (tags/v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> sqrt(9)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'sqrt' is not defined
>>> from math import sqrt
>>> sqrt(9)
3.0
>>> exit()

```

3. Ejecutar un fichero con código Python

El modo interactivo de Python anterior está muy bien para pequeñas pruebas en memoria RAM, pero lo más cómodo para mucho código y que se guarde en disco duro es crear un fichero con código de Python y ejecutarlo.

Un proyecto de Python es justo una carpeta que guarda ficheros con código Python listo para ejecutar (es un resumen rápido de proyecto Python, ya veremos sus partes).

Crea un fichero en tu sistema operativo que tenga un par de líneas de Python, por ejemplo, mostrar un texto y realizar una suma de números (para ver el resultado por consola utiliza el método "print"):

```
print("Ejecutando desde un fichero")
print(2+3)
```

Por ejemplo, yo he creado en el escritorio de mi ordenador un fichero con un bloc de notas en blanco (si usas la consola de Linux puedes crear este fichero con "Vim"; tienes una explicación sencilla de cómo se usa Vim con el ejemplo de crear un fichero con código en el artículo <https://jarroba.com/usar-servidores-y-crear-una-web/> (<https://jarroba.com/usar-servidores-y-crear-una-web/>) en el cuadro "Aprende a usar VIM en un par de minutos") que he renombrado como "mi_script.py" (sin las comillas). Y un detalle muy importante, asegúrate de guardar el fichero.

Para ejecutar este fichero con código de Python simplemente escribimos en la terminal la palabra "python", un espacio, seguido de la ruta completa hasta el fichero y pulsamos "Intro/Enter"; esto ejecutará el código Python guardado en el fichero. El ejemplo con la ruta hasta mi fichero:

```
python C:\Users\<usuario>\Desktop\mi_script.py
```

```

C:\> python C:\Users\<usuario>\Desktop\mi_script.py
Ejecutando desde un fichero
5

```

4. Gestor de paquetes PIP

Uso de cookies

Es importante saber cómo se utiliza este "gestor de paquetes" pues nos facilitará mucho gestionar los paquetes/bibliotecas en cada entorno de Python.

ACEPTAR

Abre la consola del sistema o la de Anaconda, en ambas funciona igual, pero en cada consola gestionará una carpeta "Lib" diferente (o la de "Python común" o la de "Anaconda").

Para ello utilizaremos el **gestor de paquetes PIP** que suele venir pre-instalado.

Comprobar e instalar PIP

Prueba si tienes instalado PIP con el comando (si da error es que no está bien configurado o no está instalado; si te aparece la versión es que está bien configurado y listo para usar):

```
pip --version
```

Al comando «pip» le pasa igual que los comando «python» y «python3», que depende del sistema puede que tengas que escribir «pip3» en vez de «pip» (normalmente van parejos, si tienes que escribir «python3» seguramente tengas que escribir «pip3»):

```
pip3 --version
```

En el caso de que no esté instalado PIP, te he creado un vídeo de cómo se instala en: (<https://youtu.be/LLsczhqYUxY>)

6.3- PIP - Administrador de Paquetes



Nota si tienes instalado Anaconda: Anaconda por defecto trae PIP configurado, si estás trabajando con la consola de Anaconda te tiene que funcionar (no confundas la consola del sistema operativo con la consola de Anaconda).

Nota si tienes instalado tanto Python Común como Anaconda: puede que te funcione en una consola y no en la otra; al fin y al cabo, son entornos completamente diferentes.

Se pueden consultar todos los **paquetes de PIP en el repositorio oficial**: <https://pypi.org/> (<https://pypi.org/>)

O la **instalación rápida de PIP** paso a paso:

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>) **En Linux/Mac descarga el fichero get-pip.py:**

ACEPTAR

```
curl https://bootstrap.pypa.io/get-pip.py > get-pip.py
```

Para **instalar** abre la consola y escribe (apunta a la ruta donde tengas “get-pip.py”, aquí estoy suponiendo que está en la misma ruta):

```
python get-pip.py
```

Puede que te haya aparecido en la consola el siguiente “WARNING”:

```
WARNING: The scripts pip, pip3 and pip3.8 are installed in '/home/<USUARIO>/local/bin' which is not on PATH.
```

Copia la ruta sin las comillas para **añadir PIP a la variable de entorno “PATH”** y que el uso de PIP sea más sencillo. Para ello utiliza el siguiente comando reemplazando la ruta por la que has copiado (pégala entre los dos puntos y la doble comilla de cierre):

```
export PATH="$PATH:/home/<USUARIO>/local/bin"
```

En **Windows** (o desde un navegador en cualquier sistema operativo) **descargar el fichero get-pip.py** desde: <https://bootstrap.pypa.io/get-pip.py> (<https://bootstrap.pypa.io/get-pip.py>)

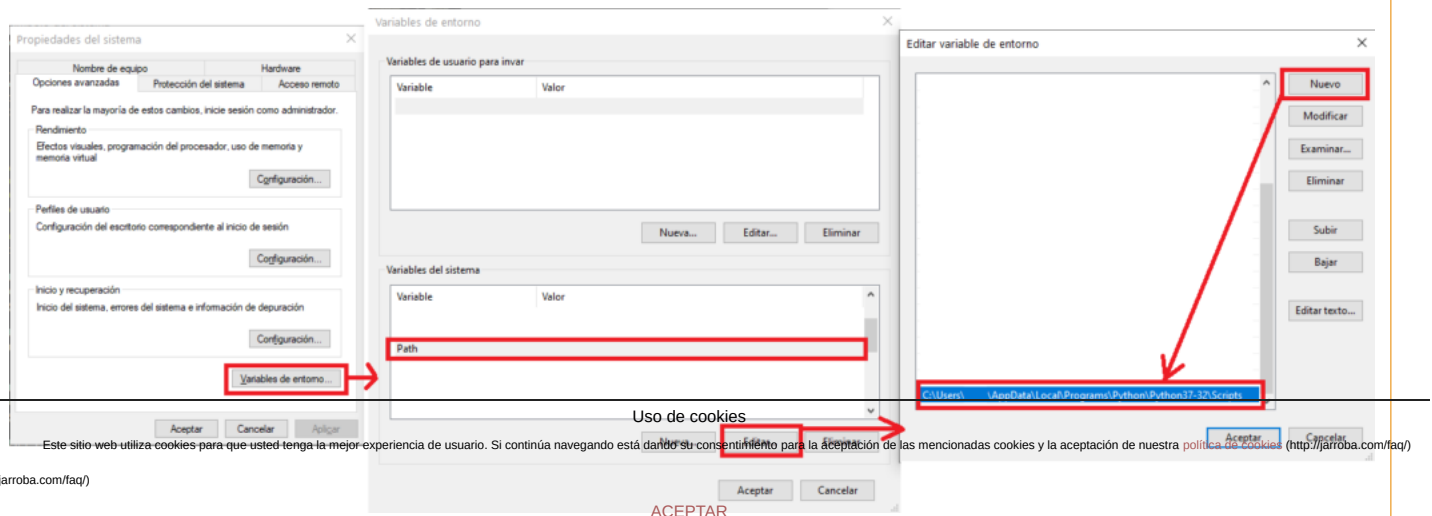
Instalar con el comando:

```
python get-pip.py
```

Y puede que te haya aparecido en la consola el siguiente “WARNING”:

```
WARNING: The scripts pip.exe, pip3.7.exe and pip3.exe are installed in 'C:\Users\<USUARIO>\AppData\Local\Programs\Python\Python37-32\Scripts' which is not on PATH.
```

Copia la ruta sin las comillas para **añadir PIP a la variable de entorno “PATH”** y que el uso de PIP sea más sencillo. Para ello busca en el buscador de Windows “Editar las variables de entorno del sistema”. En la ventana llamada “Propiedades del sistema” pulsa en el botón “Variables de entorno...”; en la siguiente ventana que se abre llamada “Variables de entorno” selecciona “Path” abajo en la sección “Variables del sistema” y pulsa el botón “Editar”; en la siguiente ventana llamada “Editar variable de entorno” pulsa el botón “Nuevo” y pega la ruta que has copiado “C:\Users\<USUARIO>\AppData\Local\Programs\Python\Python37-32\Scripts”; termina cerrando todas las ventanas pulsando “Aceptar”.



Es importante cerrar la consola de Windows y volver a abrirla para que use la nueva variable de entorno.

Tanto en Linux/Mac como en Windows, si ya tienes instalado PIP puedes eliminar el fichero get-pip.py, ya no nos servirá para nada más.

Vamos a **consultar qué paquetes/bibliotecas hay instaladas** dentro del entorno actual con el comando de PIP:

```
pip list --local
```

En la siguiente imagen te dejo un ejemplo y la comparativa de “Python común” a la izquierda y “la consola de Anaconda” a la derecha. Donde podrás ver que Anaconda tiene muchísimos paquetes en comparación, pues para eso sirve, para facilitar el uso rápido de Python al tener incorporadas un montón de bibliotecas/paquetes; al contrario del “Python común” que viene más vacío, para que lo configuremos a nuestro gusto.

The image shows two side-by-side terminal windows. The left window is titled 'Símbolo del sistema' and the right is 'Anaconda Prompt'. Both show the command 'pip list --local' highlighted with a red box. The left terminal lists a smaller set of packages, while the right terminal lists a much larger set, including many Anaconda-specific tools.

Package	Version
bleach	3.1.5
chardet	3.0.4
colorama	0.4.3
docutils	0.16
idna	2.10
importlib-metadata	1.7.0
keyring	21.3.0
packaging	20.4
pip	20.2.3
pkginfo	1.5.0.1
Pygments	2.6.1
pyparsing	2.4.7
pywin32-ctypes	0.2.0
readme-renderer	26.0
requests	2.24.0
requests-toolbelt	0.9.1
rfc3986	1.4.0
setuptools	49.2.0
six	1.15.0
tqdm	4.48.2
twine	3.2.0
urllib3	1.25.10
webencodings	0.5.1
wheel	0.35.1
zipp	3.1.0

Package	Version
alabaster	0.7.12
anaconda-client	1.7.2
anaconda-navigator	1.9.12
anaconda-project	0.8.3
argh	0.26.2
asn1crypto	1.4.0
astroid	2.4.2
astropy	4.0
atomicwrites	1.4.0
attrs	20.1.0
autopep8	1.5.4
Babel	2.8.0
backcall	0.2.0
backports.shutil-get-terminal-size	1.0.0
bcrypt	3.2.0
beautifulsoup4	4.9.1
bitarray	1.5.3
bkcharts	0.2
bleach	3.1.5
bokeh	1.4.0
boto	2.49.0
Bottleneck	1.3.2
brotlipy	0.7.0
certifi	2020.6.20
cffi	1.14.2
chardet	3.0.4

Para este ejemplo voy a **instalar el paquete** “quick-queue” (es un desarrollo de código libre que hice para acelerar cerca de un 1800% respecto a la cola multiproceso -multiprocessing.queue- que trae Python incorporada, tienes más información en <https://pypi.org/project/quick-queue/> (<https://pypi.org/project/quick-queue/>); para probar puedes instalar el paquete que quieras):

```
pip install quick-queue
```

Éste y sus dependencias (si las tiene) se instalarán gracias a PIP:

```

>pip install quick-queue
Collecting quick-queue
  Using cached quick_queue-1.0.5-py3-none-any.whl (8.9 kB)
Installing collected packages: quick-queue
Successfully installed quick-queue-1.0.5

```

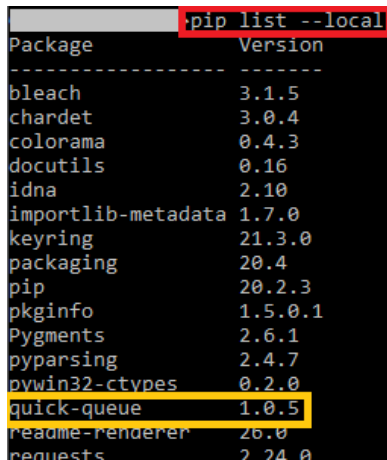
Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

Volvemos a consultar las bibliotecas instaladas en el entorno actual:

ACEPTAR

```
pip list --local
```



```

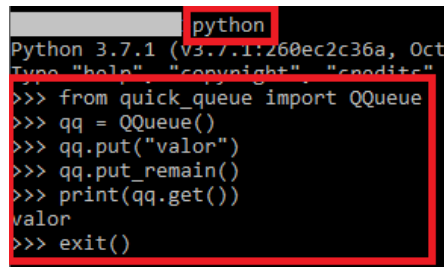
Package            Version
-----
bleach             3.1.5
chardet            3.0.4
colorama           0.4.3
docutils           0.16
idna               2.10
importlib-metadata 1.7.0
keyring            21.3.0
packaging          20.4
pip               20.2.3
pkginfo            1.5.0.1
Pygments           2.6.1
pyparsing          2.4.7
pywin32-ctypes     0.2.0
quick-queue        1.0.5
readme-renderer    26.0
requests           2.24.0
  
```

Y ya solo nos queda utilizar el paquete que acabamos de instalar. Abrimos el intérprete de Python escribiendo (o crea un fichero con código Python como vimos antes):

```
python
```

E importamos el paquete que acabamos de instalar:

```
from quick_queue import QQueue
```



```

Python 3.7.1 (v3.7.1:260ec2c36a, Oct
Type "help", "copyright", "credits"
>>> from quick_queue import QQueue
>>> qq = QQueue()
>>> qq.put("valor")
>>> qq.put_remain()
>>> print(qq.get())
valor
>>> exit()
  
```

Si no da ningún error al importar es que funciona correctamente. Aunque si queremos podemos probarla y al terminar salir de la consola interactiva de Python escribiendo "exit()", por ejemplo basta con obtener la clase (en la imagen pongo un ejemplo completo del uso de "quick_queue" que sin multitarea no tiene mucho más sentido de usar; la pongo más para que se vea que podemos usar el paquete que hemos importado y funcionará):

```
qq = QQueue()
exit()
```

Y, para terminar, si quieres **desinstalar un paquete que hayamos instalado en el entorno de Python**:

```
pip uninstall quick-queue
```

Continúa con los entornos virtuales de Python tanto en Python común como en Anaconda

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR

Entornos virtuales de Python: común y Anaconda



Los entornos virtuales de Python nos van las siguientes ventajas: Aíslan la ejecución de los proyectos (poder ejecutar diferentes proyectos que trabajen con versiones tanto de Python como de cierta biblioteca/paquete diferente) Uso particular de entornos (virtuales) preparados (con ciertas bibliotecas/paquetes instalados); es decir, cada proyecto se ejecutará en el entorno virtual que es bueno ... Sigue leyendo



0

Bibliografía

- <https://www.python.org/> (<https://www.python.org/>)
- <https://docs.python.org/3/> (<https://docs.python.org/3/>)
- <https://www.anaconda.com/> (<https://www.anaconda.com/>)
- <https://docs.anaconda.com/> (<https://docs.anaconda.com/>)
- <https://pip.pypa.io/en/stable/> (<https://pip.pypa.io/en/stable/>)
- <https://pypi.org/> (<https://pypi.org/>)
- <https://docs.python.org/3/using/windows.html> (<https://docs.python.org/3/using/windows.html>)
- <https://docs.python.org/3/using/unix.html> (<https://docs.python.org/3/using/unix.html>)
- <https://docs.python.org/3/using/mac.html> (<https://docs.python.org/3/using/mac.html>)
- <https://docs.anaconda.com/anaconda/install/> (<https://docs.anaconda.com/anaconda/install/>)
- Experiencia propia

Comparte esta entrada en:



ps:// ps:// ps:// ps://

plus ww ww twitt

.goo w.lin w.fa er.c

gle. kedi ceb om/

com n.co ook. hom

/sha m/s com e?

re? har /sha stat

url= eArt re.p us=

http icle hp? Ent

s%3 9 qm t orn

A% mini tps os+

95 - 8/4/23 - Jarroba

Usa de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR

Los socios ahorran

Anuncio Expedia.mx



(<https://creativecommons.org/licenses/by-nc-sa/3.0/es/>) (<https://www.safecreative.org/userfeed/1401310112503>)

Entornos de Python: común y Anaconda (base) (<https://jarroba.com/entornos-de-python-anaconda-y-comun/>) por "www.jarroba.com" esta
bajo una licencia Creative Commons
Reconocimiento-NoComercial-CompartirIgual 3.0 Unported License.
Creado a partir de la obra en www.jarroba.com (<https://jarroba.com/entornos-de-python-anaconda-y-comun/>)

One thought on “Entornos de Python: común y Anaconda (base)”



Daniel F dice:

28/04/2022 a las 01:31 (<https://jarroba.com/entornos-de-python-anaconda-y-comun/#comment-293986>)

Muchas gracias, con esto entendi muchas conceptos que no eran claros !!

Responder

Deja una respuesta

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

Comentario *

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

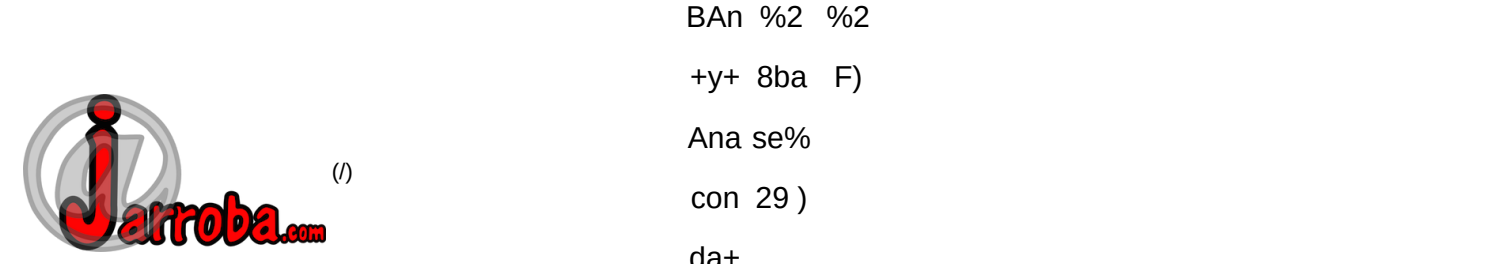
Nombre *
(<http://jarroba.com/faq/>)

Correo electrónico *


Web


Publicar el comentario


Este sitio usa Akismet para reducir el spam. Aprende cómo se procesan los datos de tus comentarios (<https://akismet.com/privacy/>).





Síguenos en:

<https://www.youtube.com/user/JarrobaWeb>

<https://twitter.com/WwwJarrobaCom>

<https://www.facebook.com/jarrobaWeb>

<https://github.com/jarroba?tab=repositories>

<https://plus.google.com/103352032205227460108/>

- Contato: jarrobaweb@gmail.com (<mailto:jarrobaweb@gmail.com>)
- [FAQ \(HTTPS://JARROBA.COM/FAQ/\)](https://JARROBA.COM/FAQ/)
- [VISITAS \(HTTPS://JARROBA.COM/VISITAS/\)](https://JARROBA.COM/VISITAS/)
- [MAPA DEL SITIO \(HTTPS://JARROBA.COM/SITEMAP_INDEX.XML\)](https://JARROBA.COM/SITEMAP_INDEX.XML)

Copyright © 2023 Jarroba.com (<https://www.jarroba.com>) - Todos los derechos reservados

By Reimon & Richard