

# Programación Concurrente y de Tiempo Real

## Seminario de Programación Concurrente con Python

### Manual de Instalación de Python

Antonio J. Tomeu Hardasmal  
*Escuela Superior de Ingeniería*

---

#### Abstract

El presente documento contiene un corto manual acerca de cómo instalar el intérprete de Python para Windows 10 y Linux, y debería permitirle una instalación funcional del mismo en pocos minutos, de manera que pueda ejecutar y probar los códigos que forman parte del seminario sobre Python. No obstante lo anterior, si cree necesitar una información más amplia que la recogida aquí, se le remite al sitio oficial del lenguaje Python, que encontrará en el url <https://www.python.org/>

*Keywords:* Instalación, Linux, Python, Ubuntu, Windows

---

#### 1. Instalación de Python para Windows 10

Resulta elemental. Todo lo que hay que hacer es navegar a la página de descargas de Python disponible en el url <https://www.python.org/downloads/>. Una vez en ella (Figura 1), escogemos la versión que nos interesa descargar, y que normalmente será la que tenga una numeración más alta. En la Figura 1, aparece recuadrada en verde.

El proceso de descarga será el habitual y, una vez concluido, no queda más que ejecutar el instalador con privilegios de administrador, lo cuál conduce a una ventana de interfaz como la de la Figura 2.

Es importante que verifique la casilla rotulada **Add Python 3.XX to Path** está marcada. Si no lo estuviera, o si esta casilla no aparece en el instalador que usted tiene, será necesario ajustar la variable de PATH de Windows con la ruta que identifica la carpeta de binarios de Python. De lo contrario, no será posible ejecutar el intérprete de Python desde cualquier punto de la jerarquía de ficheros del sistema. Una vez hecho lo anterior, se hace click sobre el tipo de instalación de Python deseado (Install now o Customize Installation) y el proceso comienza, mostrándonos el progreso mediante una barra similar a la que se observa en la Figura 3.

Concluido el proceso de instalación, se nos informa de ello con una imagen similar a la de la Figura 4. Quizás sea un buen momento para navegar al tutorial en línea que se le ofrece en esa pantalla.

---

*Email address:* [antonio.tomeu@uca.es](mailto:antonio.tomeu@uca.es) (Antonio J. Tomeu Hardasmal)

*Preprint submitted to Elsevier*

*30 de mayo de 2018*



Figura 1: Descargando el intérprete de Python

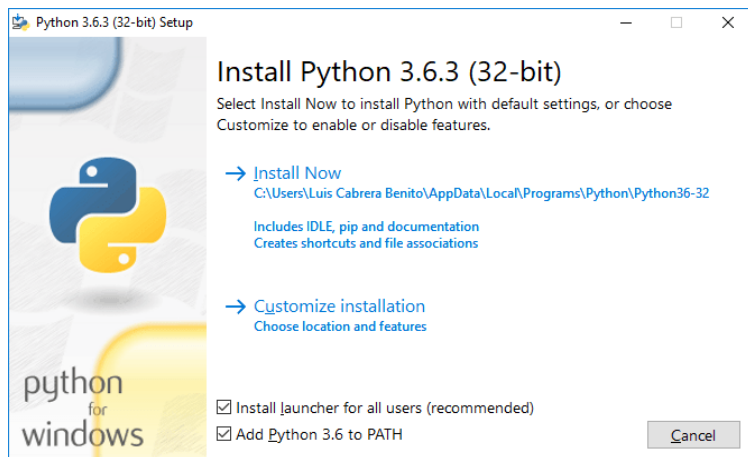


Figura 2: El instalador en ejecución

Para comprobar que la instalación se ha efectuado de manera adecuada, no queda más que abrir un terminal de consola, y ejecutar el binario Python. El resultado será la carga del intérprete del lenguaje, que entrará en modo de espera interactivo, según se muestra en la Figura 5.

### 1.1. Instalación de *pip*

Si necesita instalar módulos de Python adicionales, puede hacerlo mediante el binario *pip*, que debe ser instalado con carácter previo a su uso, tras la instalación de Python. *pip* permite la instalación, actualización y desinstalación de módulos Python. En las versiones recientes de Python para Windows, el binario ya estará instalado casi con absoluta seguridad. Para comprobarlo, pruebe a ejecutar desde el terminal lo siguiente:

```
C:\>pip
```

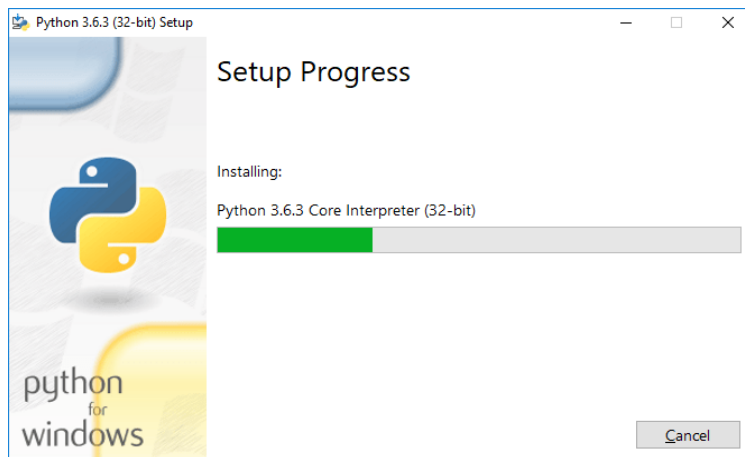


Figura 3: Progreso de la instalación

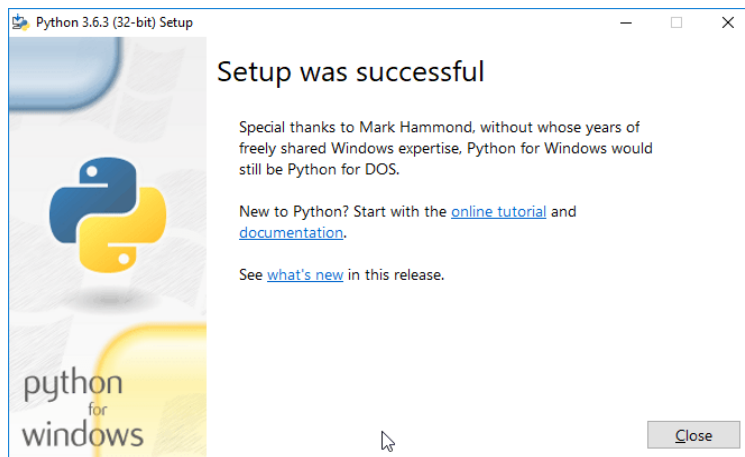


Figura 4: Fín de la ejecución

Si el binario esta disponible, obtendrá una salida impresa que muestra todos los *flags* que permiten parametrizarlo (Fig 6).

En otro caso, deberá instalarlo. Para ello, deberá descargar un corto programa Python y pedirle al intérprete que lo ejecute. Puede descargar ese programa desde el url siguiente: <https://pip.pypa.io/en/stable/installing/>. Una vez instalado, verifique como antes que puede ejecutarlo. En caso negativo, deberá actualizar la variable de PATH. Hecho esto, descargar e instalar módulos adicionales de Python es muy sencillo, ya que la sintaxis es siempre igual:

```
C:\>pip install nombre_modulo
```

La Figura 7 muestra un intento de instalar el paquete para cálculo científico de Pythom *numpy*. Como ya estaba instalado, Python informe de ello e indica cómo actualizarlo. De no haber estado ya instalado, el proceso de descarga e instalación habría tenido lugar.

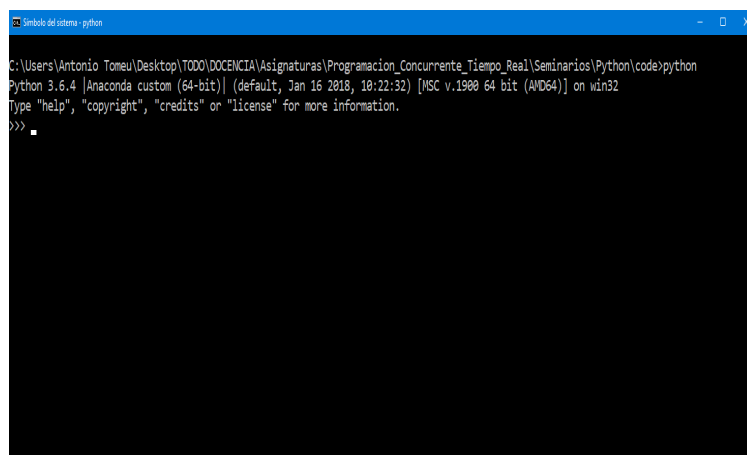


Figura 5: Intérprete de Python ejecutándose en modo interactivo

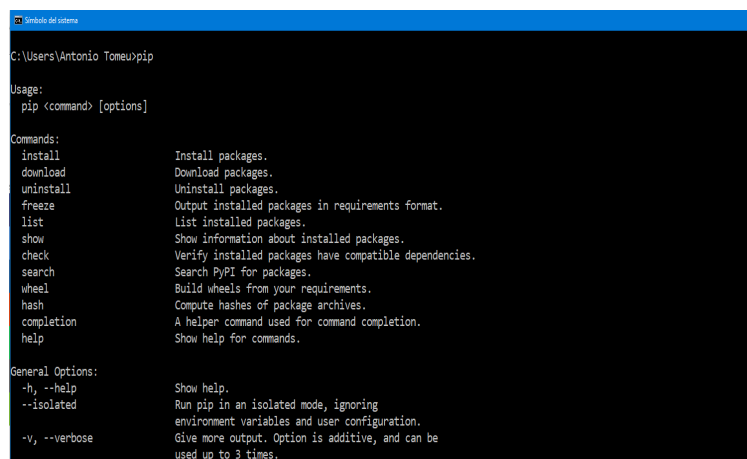


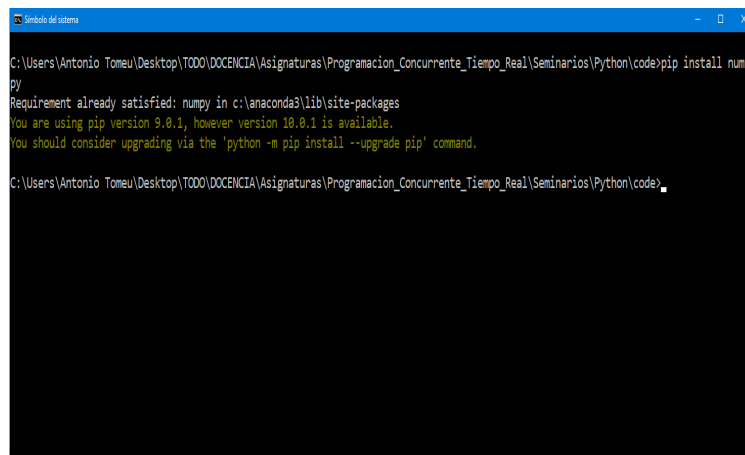
Figura 6: Comprobando si tenemos pip

## 2. Instalación de Python para Linux Ubuntu

La instalación de Python para los sistemas operativos de la familia Linux se hace siempre desde la línea de comandos del sistema, y requiere unos privilegios mínimos de administrador. En consecuencia, comience por abrir un *shell* de sistema en una ventana de terminal, bien a través de un icono que probablemente tendrá en su escritorio, bien de manera directa mediante la pulsación combinada de las teclas **Ctrl-Alt-T**. Si está utilizando versiones recientes (16.04 o superiores) de Linux Ubuntu, lo más probable es que disponga las versiones de Python 2 y 3 preinstaladas, así que comience por verificar que esto es así mediante la siguiente orden:

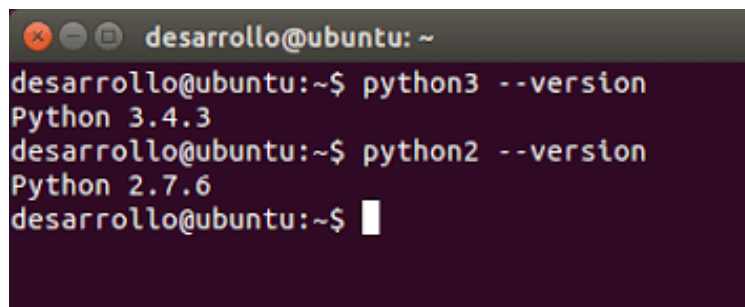
```
python3 --version
```

Si Python está instalado, obtendrá una salida parecida a la que se muestra en la Figura 8.



```
Símbolo del sistema
C:\Users\Antonio Tomeu\Desktop\TODO\DOCEENCIA\Asignaturas\Programacion_Concurrente_Tiempo_Real\Seminarios\Python\code>pip install num
py
Requirement already satisfied: numpy in c:\anaconda3\lib\site-packages
You are using pip version 9.0.1, however version 10.0.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
C:\Users\Antonio Tomeu\Desktop\TODO\DOCEENCIA\Asignaturas\Programacion_Concurrente_Tiempo_Real\Seminarios\Python\code>
```

Figura 7: Solicitud de instalación del módulo `numpy`



```
desarrollo@ubuntu: ~
desarrollo@ubuntu:~$ python3 --version
Python 3.4.3
desarrollo@ubuntu:~$ python2 --version
Python 2.7.6
desarrollo@ubuntu:~$
```

Figura 8: Verificando si Python está instalado con Linux Ubuntu

Si no está seguro de la versión de Linux que tiene, conviene que actualice su sistema mediante `apt-get` como sigue:

```
sudo apt-get update
sudo apt-get -y upgrade
```

Aunque el flag `-y` indica a Linux que está de acuerdo con la carga e instalación de todos los ítems de la actualización, es posible que Linux le pide confirmación acerca de algunas actualizaciones concretas. Una vez que su sistema esté actualizado, pruebe de nuevo a preguntar si Python está instalado tal y como se explicó anteriormente.

### 2.1. Instalación de *pip*

Nuevamente, es posible que queramos instalar módulos Python que no están disponibles con la instalación estándar. Para gestionar esto, necesito tener disponible el binario `pip` que a diferencia de Windows, necesita ser previamente instalado siempre, lo cuál se consigue de manera muy simple:

```
sudo apt-get install -y python3-pip
```

Una vez que el binario `pip` está instalado, aquellos módulos que necesitemos añadir se descargan e instalan como sigue:

```
pip3 install numpy
```

para el caso del módulo de cálculo científico `numpy`.

### 3. Depuración y Ejecución de Programa Python

A diferencia de Java, Python es un lenguaje interpretado puro; por lo tanto, no existe un compilador de Python como tal, entendiéndose como compilador a algún traductor del código fuente a código intermedio (bytecodes de Java) u objeto (C++). Para depurar un programa, lo que se hace es pedirle al intérprete de Python que lo ejecute. Si el programa presenta errores sintácticos, el intérprete nos indicará cuáles y en qué línea de programa los encuentra. Si el programa está libre de errores sintácticos (lo cual no quiere decir este también libre de errores semánticos) el intérprete ejecutará el programa línea a línea. Como ejemplo de todo esto, descargue de la carpeta de código el programa `discriminante.py` y ábralo con el editor que prefiera. Obtendrá el siguiente código, que es sintácticamente correcto:

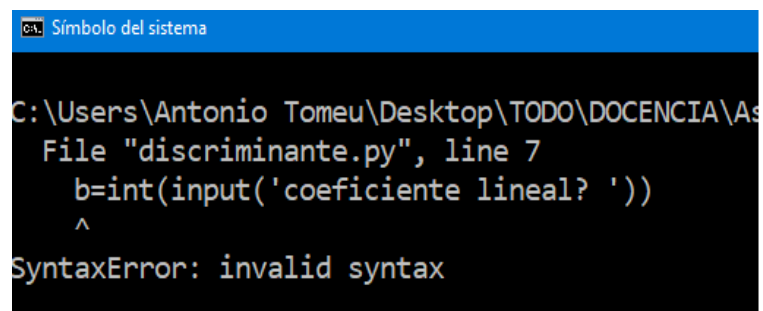
```
#esto es un comentario
#un modulo se importa asi:
import math

#el tipado de las variables es dinamico...
a=int(input('coeficiente_cuadratico?_'))
b=int(input('coeficiente_lineal?_'))
c=int(input('coeficiente_libre?_'))
discriminante=(b**2)-4*a*c
print('discriminante=', discriminante)
#la indentacion determina el ambito de las estructuras de control
if discriminante>0:
    raiz=math.sqrt(discriminante)
    print(raiz)
else:
    print('discriminante_negativo...')
```

Pruebe ahora a depurarlo:

```
C:\>python discriminante.py
```

Como el intérprete de Python no encuentra errores, el proceso de depuración es seguido inmediatamente de la carga y ejecución del programa, que le pide tres números por pantalla. Pruebe ahora a introducir algunos errores sintácticos (como por ejemplo borrar algún paréntesis en la lectura de la variable `a`) y trate de ejecutar el programa. Observará que el intérprete encuentra los errores y le informa de ello (Figura 9).



```
Símbolo del sistema
C:\Users\Antonio Tomeu\Desktop\TODO\DOCENCIA\As
File "discriminante.py", line 7
    b=int(input('coeficiente lineal? '))
        ^
SyntaxError: invalid syntax
```

Figura 9: Python informando de un error sintáctico