

Capítulo 3

Presentación del entorno Python

1. Python, la tercera parte del nombre

Si descubre ahora el entorno de Python o lo ha hecho después del 1 de enero de 2020, es probable que ni siquiera se pregunte acerca de elegir entre Python 2 y Python 3: naturalmente, utilizará la versión más reciente. Pero aquellos cuyo conocimiento del lenguaje es anterior a esta fecha pueden haber notado esta dualidad, esta convivencia que podría haber sido fuente de confusión.

Con el tiempo, Python acumuló algunos errores que se tenían que corregir. Entre estos defectos, citemos la redundancia de información (varias formas de obtener un determinado resultado), la gestión de las cadenas de caracteres no muy óptima, una forma obsoleta de declarar clases, etc. Así, en diciembre de 2008, una versión Python 3.0.0 se lanzó, junto con una versión 2.6.

Esta nueva versión principal de Python no es compatible con Python 2. Es decir, un programa escrito con la versión 3.0.0 o superior de Python, no puede funcionar con un intérprete de Python versión 2 (y viceversa). Esto puede ser problemático si alguna vez un proyecto de Python 3 tiene que usar una librería externa escrita en Python 2. Debido a la incompatibilidad de versiones, esta dependencia no se puede usar, lo que puede añadir un freno al avance del proyecto. Donde antes la elección de la versión principal de Python era una cuestión espinosa, desde 2020 ya no se produce: Python 3 es ahora la única versión mantenida de Python.

Cuando se lanzó Python 3, la comunidad de Python anunció la muerte programada de Python 2 para 2015 y comunicó ampliamente la importancia de migrar todos los proyectos de Python a esta nueva versión principal. Esto realmente no se produjo, y en 2014, dada la extensión de los importantes y populares proyectos de Python que todavía estaban en la versión 2, se estableció una prórroga hasta 2020. En la actualidad, los sistemas operativos Linux y macOS ya no ofrecen Python 2 en su instalación base y Windows nunca lo hizo.

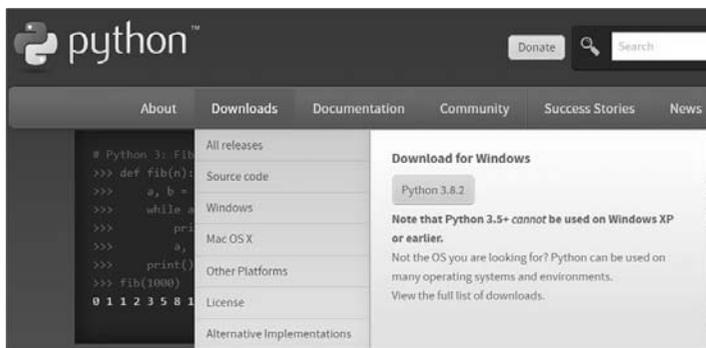
■ Observación

Estos hechos pueden parecer una anécdota histórica, pero dada la enorme popularidad de Python antes de la versión 3, se escribieron un gran número de librerías en esta versión, que no son compatibles con la versión 3, la que actualmente se mantiene. Conocer esta parte de la historia de Python, significa estar alerta cuando desee incluir una librería en su proyecto, lo que supone verificar que sea compatible con Python 3.

2. Instalación

2.1 python.org

Vaya a <https://www.python.org/downloads/>. El sitio de Python debería ofrecerle automáticamente un archivo correspondiente a su sistema operativo.

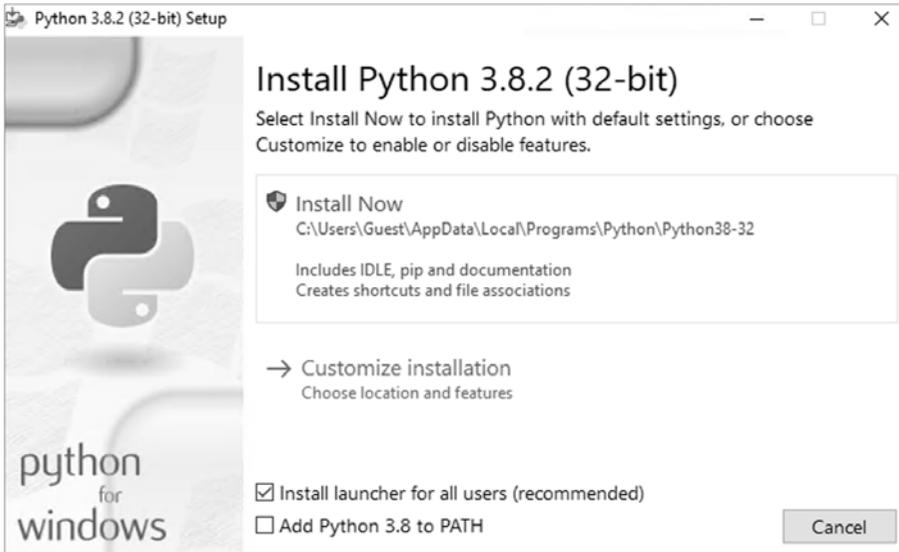


2.2 Windows

De forma predeterminada, el sitio le pide que descargue un archivo de "instalación web", que requiere una conexión a Internet. Al navegar a la página de descarga de una versión específica de Python (generalmente en la dirección <https://www.python.org/downloads/release/python-3125/> para la versión 3.12.5)), podrá elegir entre:

- un archivo comprimido que contiene una versión de Python, que se puede usar sin tener que instalarlo (esto es práctico si no tiene suficientes permisos para realizar una instalación estándar);
- un instalador "clásico";
- el instalador a través de la Web.

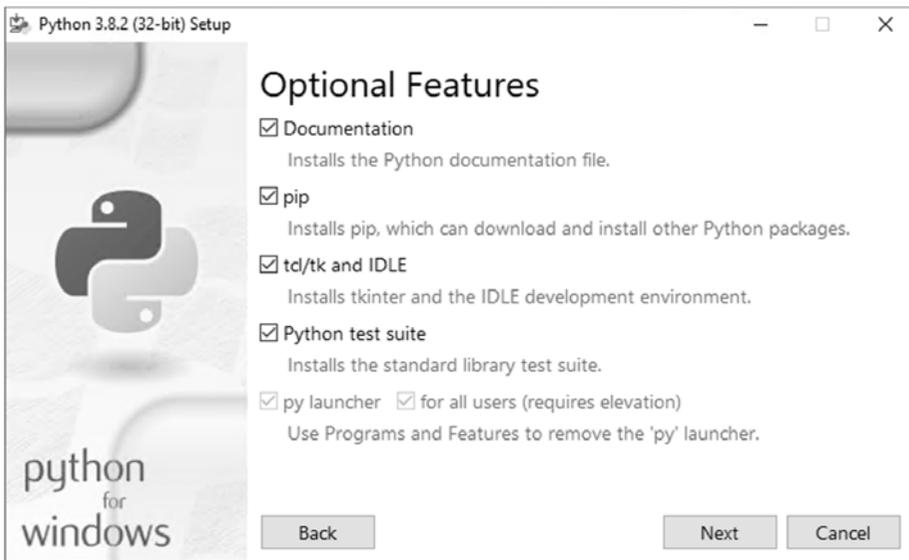
Durante la instalación, tendrá que decidir sobre algunas opciones de las que se muestran los detalles aquí:



En la parte inferior de esta pantalla, tiene dos opciones disponibles:

- **Use admin privileges when installing py.exe** (usar privilegios de administrador al instalar py.exe) determina si el programa **Python (py.exe)** es accesible para todos los usuarios del sistema o solo para el usuario actual.
- **Add Python.exe to PATH** (añadir o no el ejecutable en el PATH). El PATH es una variable de entorno que generalmente contiene varios directorios. Cuando un usuario quiere ejecutar un comando (normalmente python o incluso dir), el sistema operativo buscará en los directorios del PATH si el ejecutable deseado está presente. Si es así, el comando funcionará. De lo contrario, aparecerá un mensaje de error indicando que no se encontró el ejecutable. Si planea usar Python desde la línea de comandos, debe marcar esta opción. Si piensa usar solo IDEs para trabajar, puede dejarla sin marcar.

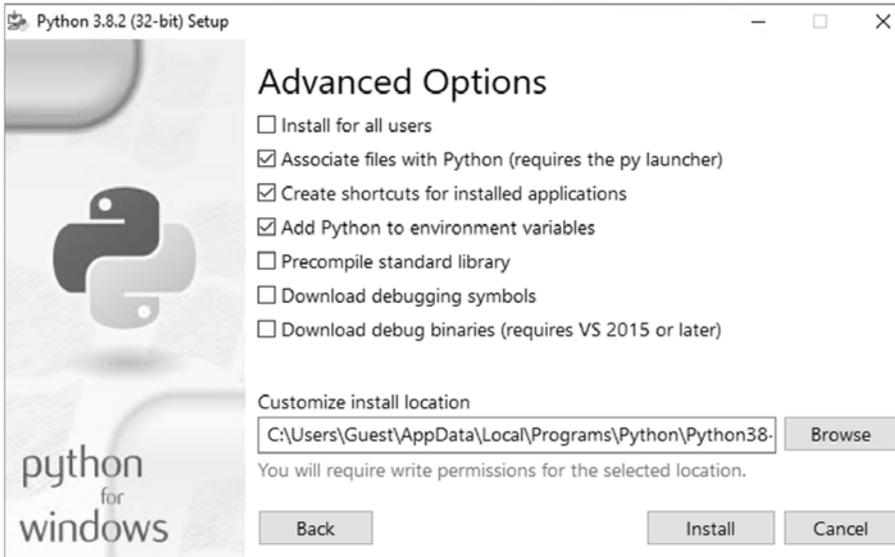
Si está satisfecho con la elección del destino y el contenido que se va a instalar, puede ejecutar la instalación. De lo contrario, es necesaria la personalización de las opciones de instalación:



- **Documentation:** instalar la documentación puede ser interesante si a menudo no tiene conexión a Internet (de viaje, por ejemplo).

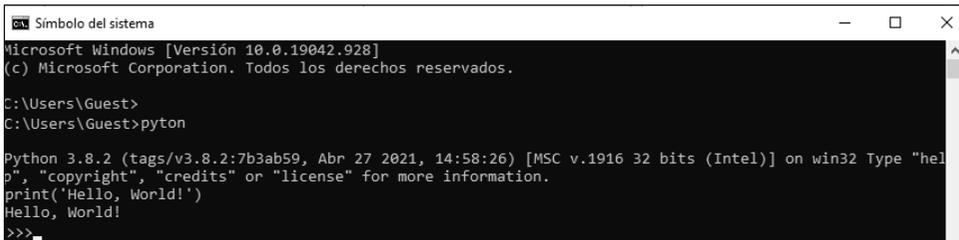
- **pip**: pip es una herramienta fácil de usar para instalar bibliotecas en lugar de recuperarlas e instalarlas manualmente.
- **tcl/tk and IDLE**: tcl/tk es un módulo para generar interfaces gráficas; IDLE es un entorno de desarrollo gráfico. Marque esta casilla según las necesidades de su proyecto y su método de trabajo.
- **Python test suite**: la suite de pruebas de Python es un conjunto de módulos para probar eficazmente su aplicación y se debe instalar sin dudar.
- **py launcher**: el lanzador de Python se explicó anteriormente.
- **for all users (requires admin privileges)** permite instalar Python para todos los usuarios de la máquina, pero se requieren derechos de administrador.

Después del contenido de la instalación, aquí están los parámetros:



Para las pocas opciones que no son explícitas:

- **Add Python to environment variables** (añadir Python a las variables de entorno), permite que el sistema sepa dónde están los ejecutables de Python, las librerías, etc.
 - Cuando depura su aplicación, Python genera archivos de depuración que contienen su código, así como información adicional para que sea más fácil para usted. Para poder navegar en los módulos básicos de Python durante la depuración, es necesario instalar los símbolos de depuración de estos módulos, a través de la opción descargar los símbolos de depuración (**Download debugging symbols**) y descargar los binarios de depuración (**Download debug binaries (requires VS2017 or later)**).
- Haga clic en **Install** y, después de unos segundos, se instalará Python. Para probar su instalación, abra una ventana de comandos (en Windows: tecla **[Windows] + R**, y luego `cmd`) y escriba el comando `python`. Entra en el entorno del intérprete de Python. Puede escribir todo el código Python que desee, pero para esta prueba de instalación, será suficiente un texto sencillo que se muestra en la pantalla: `print('Hello, World!')`.



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19042.928]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Guest>
C:\Users\Guest>python

Python 3.8.2 (tags/v3.8.2:7b3ab59, Abr 27 2021, 14:58:26) [MSC v.1916 32 bits (Intel)] on win32 Type "help()", "copyright()", "credits()" or "license()" for more information.
print('Hello, World!')
Hello, World!
>>>
```

2.3 mac OS X

A partir de macOS Monterey (12.3), lanzado en octubre de 2021, Python 3 se instala de forma predeterminada con el programa `python3`. El sufijo 3 es un remanente de los días en que ambas versiones de Python se instalaban una al lado de la otra.

■ Para verificar su instalación, abra la aplicación "Terminal" y escriba:

```
$> python3 --version Python 3.12.5

$> python3
Python 3.12.5 (main, Aug 6 2024, 19:08:49) [Clang 15.0.0
(clang-1500.3.9.4)] on darwin
Type "help", "copyright", "credits" or "licence" for more information.
>>> print("Hola Python!") Hola Python!
```

Tener que agregar la versión al nombre del ejecutable puede ser problemático si alguna vez tiene scripts que usan el comando `python` (sin el sufijo). Por lo tanto, puede ser necesario indicarle a su shell que el comando `python` apunta al ejecutable `python3`. Para hacer esto, necesita crear un alias en su archivo de configuración de shell, que es `.bashrc` por defecto y se encuentra en la raíz de su directorio de inicio:

```
■ $> echo "alias python=/usr/local/bin/python3.12.5" >> ~/.bashrc
```

Después de ejecutar el comando anterior, debe verificar que el alias funcione:

```
■ $> python -version
Python 3.12.5
```

2.4 Unix/Linux

La mayoría de las distribuciones populares, ya tienen por defecto Python 3. Por lo tanto, no es necesario ningún proceso de instalación. Si tiene que instalar Python el sitio de Python no ofrece un archivo de instalación. Dependiendo de la distribución en la que esté trabajando, es muy probable que exista una herramienta de administración de software, que le permitirá instalar Python 3.

```
■ $> apt install python3 # En Ubuntu
$> emerge --ask dev-lang/python # En Gentoo
```

3. Herramientas

3.1 pip

El ejecutable `pip` se usa para instalar librerías de Python desde la línea de comandos. Esto es muy útil, ya que puede ser bastante tedioso ir al sitio web que aloja la librería, descargarla y colocarla en el lugar correcto para que Python pueda acceder a ella.

Todo esto lo hace automáticamente `pip`, que también es la herramienta oficial del grupo de trabajo responsable de mantener los principales proyectos de la librería de Python (PyPA: *Python Package Authority*). Este grupo de trabajo alimenta un índice oficial de módulos disponibles al que se conecta `pip`. Por lo tanto, es la mejor manera de asegurarse de que tiene dependencias perfectamente estables y actualizadas.

Al igual que con el ejecutable de Python, es posible que deba usar el comando `pip3` y no `pip` para llamar al instalador dedicado a Python 3 (principalmente en macOS o Ubuntu). Verificar la versión llamando al comando `pip -version` puede ahorrarle algunos problemas.

```
■ $> pip install pyside6 # Instala la librería Qt para Python
```

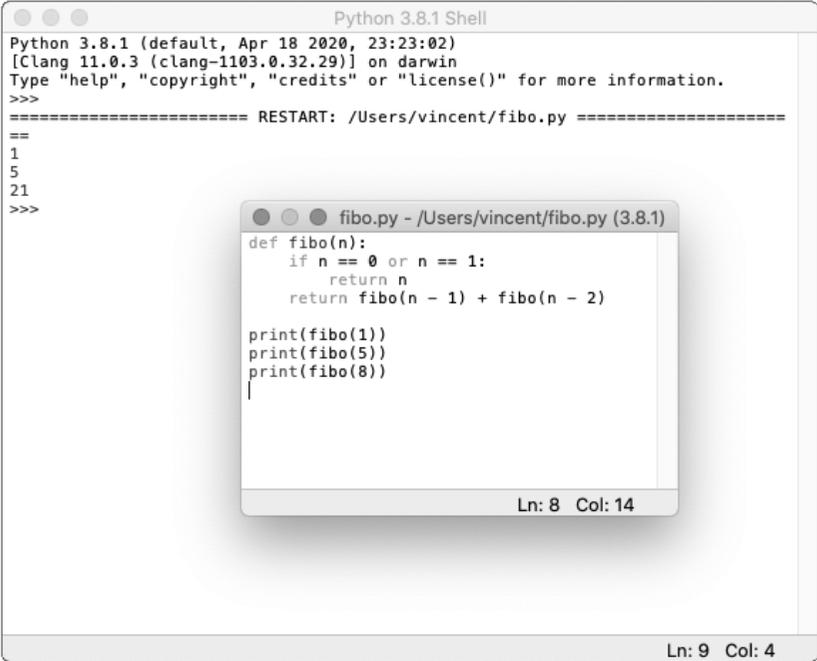
`pip` también ofrece otros comandos prácticos para administrar las librerías instaladas en su sistema:

- `pip uninstall` para desinstalar las librerías que ya no necesita.
- `pip list` para obtener la lista de librerías instaladas.
- `pip help` para mostrar todas las demás opciones disponibles.

3.2 IDLE

IDLE (*Integrated Development and Learning Environment*: entorno integrado de aprendizaje y desarrollo) es un IDE básico para Python y disponible en todos los sistemas operativos. Ofrece varias funciones que incluyen:

- la coloración sintáctica;
- el autocompletado;
- un depurador;
- una indentación "inteligente".



The image shows a screenshot of a Python 3.8.1 Shell window. The main window displays the following text:

```
Python 3.8.1 (default, Apr 18 2020, 23:23:02)
[Clang 11.0.3 (clang-1103.0.32.29)] on darwin
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /Users/vincent/fibo.py =====
==
1
5
21
>>>
```

A smaller window titled "fibo.py - /Users/vincent/fibo.py (3.8.1)" is overlaid on the main window, showing the following Python code:

```
def fibo(n):
    if n == 0 or n == 1:
        return n
    return fibo(n - 1) + fibo(n - 2)

print(fibo(1))
print(fibo(5))
print(fibo(8))
|
```

The status bar at the bottom of the main window shows "Ln: 9 Col: 4". The status bar at the bottom of the smaller window shows "Ln: 8 Col: 14".

Sin embargo, aunque esta herramienta es suficiente para iniciarse en Python, está claramente limitada para un uso más avanzado. Es muy deficiente en términos de ergonomía, faltan funcionalidades importantes en la interfaz gráfica (numeración de líneas, navegación más fácil entre archivos, personalización de colores y fuentes de letra, etc.).