

Los objetivos de Python

En 1999, Guido van Rossum (creador de Python) definió sus objetivos para Python:

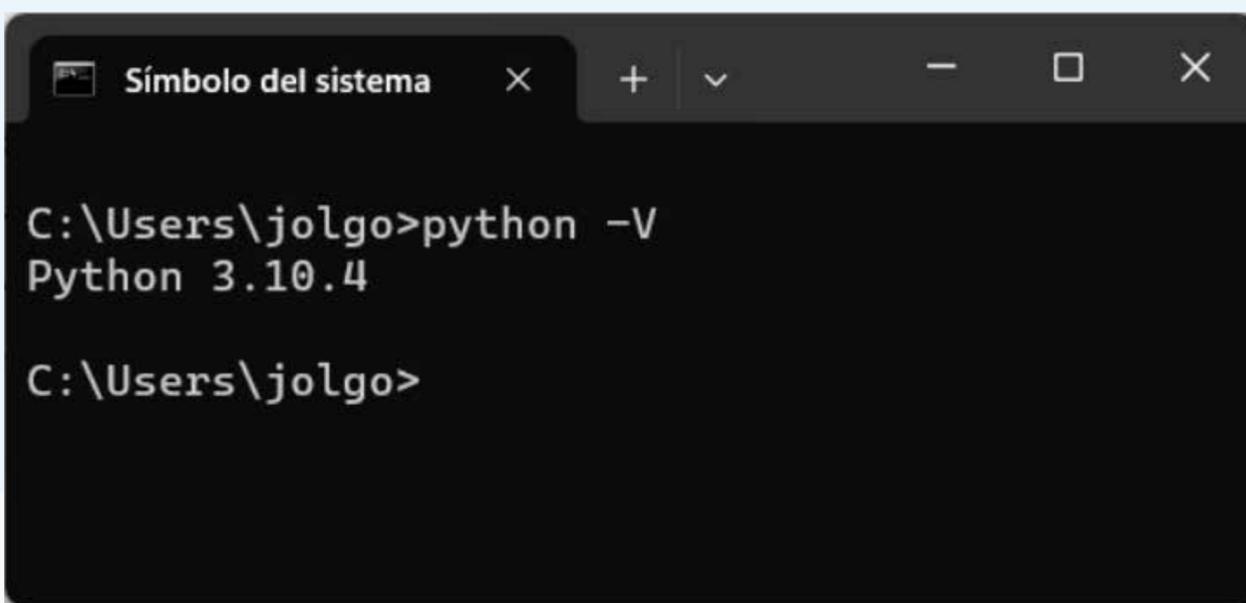
- Un lenguaje fácil e intuitivo tan poderoso como los de los principales competidores.
- De código abierto, para que cualquiera pueda contribuir a su desarrollo.
- El código es tan comprensible como el inglés simple.
- Adecuado para tareas cotidianas, permitiendo tiempos de desarrollo cortos.

¿Por qué Python?

- Es fácil de aprender - el tiempo necesario para aprender Python es más corto que en muchos otros lenguajes; esto significa que es posible comenzar la programación real más rápido.
- Es fácil de enseñar - la carga de trabajo de enseñanza es menor que la que necesitan otros lenguajes; esto significa que el profesor puede poner más énfasis en las técnicas de programación generales (independientes del lenguaje), no utilizando energía en complicados trucos, extrañas excepciones y reglas incomprensibles.
- Es fácil de utilizar para escribir software nuevo - a menudo es posible escribir código más rápido cuando se emplea Python.
- Es fácil de entender - a menudo, también es más fácil entender el código de otra persona más rápido si está escrito en Python.
- Es fácil de obtener, instalar y desplegar - Python es gratuito, abierto y multiplataforma; no todos los lenguajes pueden presumir de eso.

Lo primero es crear el manual de instalación, en el enlace [Instalar Python: instrucciones para Windows, Linux y macOS](#), está el título `Instalación y configuración de Python en Windows: ¿cómo funciona?`, que de una manera sencilla entrega el paso a paso para proceder con la instalación de la última versión disponible.

Luego de realizar la instalación, se debe verificar que el aplicativo funcione correctamente, en una terminal del sistema, power shell, git bash o cmd, ejecutar el comando **python -V**

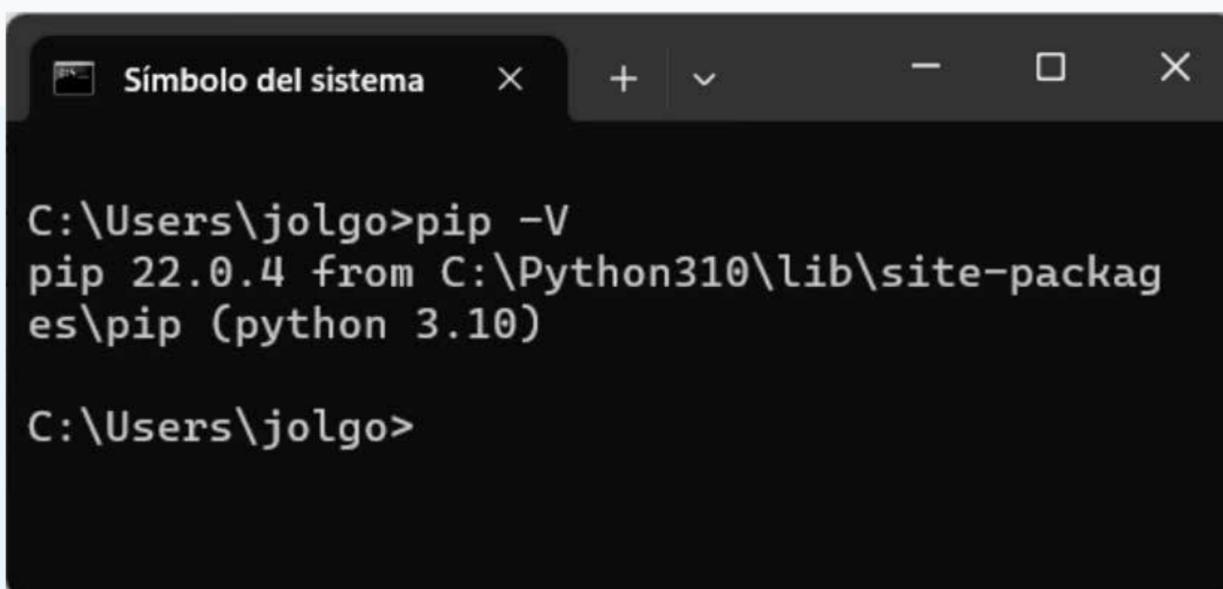


```
Símbolo del sistema
C:\Users\jolgo>python -V
Python 3.10.4
C:\Users\jolgo>
```

Fuente: Elaboración propia

La salida debe ser algo como la de la imagen.

Finalmente verificar que el manejador de paquetes de python también se encuentre instalado; en un módulo más adelante se explicará qué es y para qué se usa, por ahora solo es necesario verificar su instalación. Ejecutar el comando **pip -V**



```
Símbolo del sistema
C:\Users\jolgo>pip -V
pip 22.0.4 from C:\Python310\lib\site-packag
es\pip (python 3.10)
C:\Users\jolgo>
```

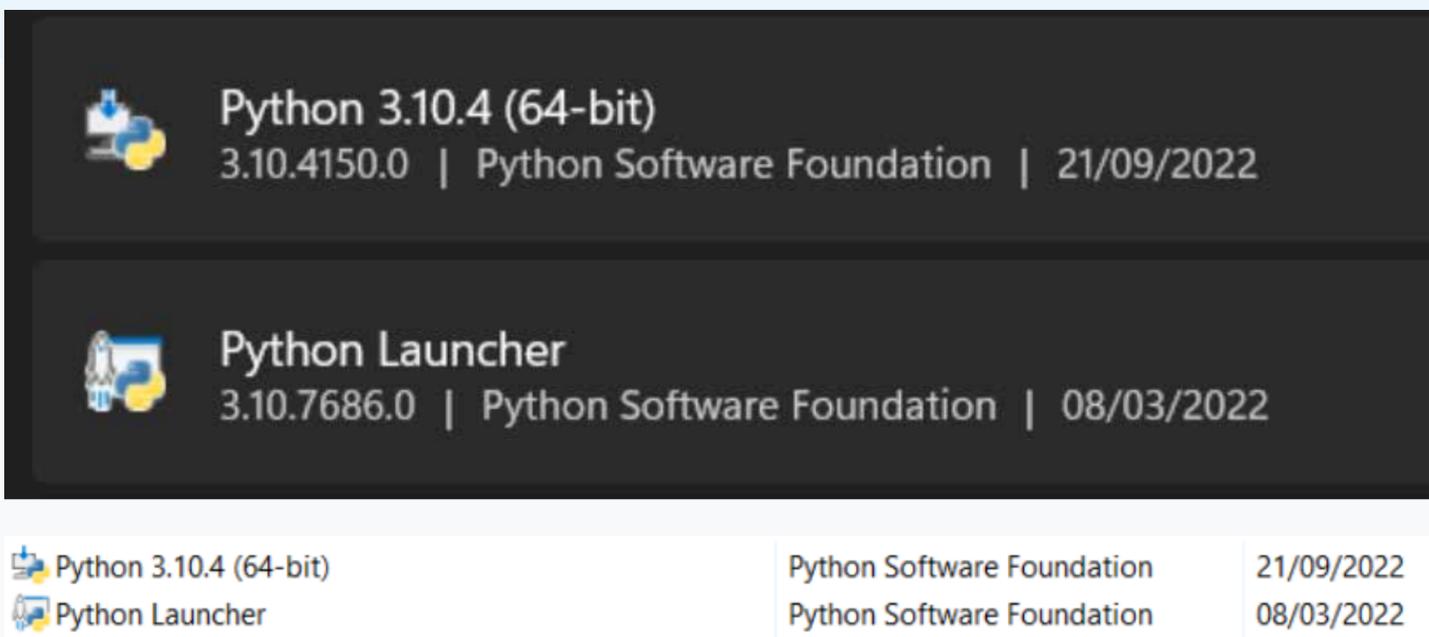
Fuente: Elaboración propia

Con esto se ha completado satisfactoriamente la instalación, si se hace en una fecha diferente es posible que las versiones de los aplicativos hayan cambiado o igual en caso que lo hayan instalado en un fecha posterior, el curso se realiza sobre una versión superior a Python 3.7, la recomendación es instalar la última versión disponible, para la fecha es la 3.12.

Si se presenta algún error a la hora de ejecutar el comando, es posible que no haya hecho la instalación todavía o que no haya agregado Python al PATH del sistema.

Para verificar el primero punto, revise en el listado de aplicativos instalados.

En la lista de programas por la P debería aparecer algo como lo de la imagen a continuación:



Fuente: Elaboración propia

Para el segundo caso, el PATH del sistema se pueden seguir los pasos del siguiente video

¿Cómo agregar Python al Path de Windows?

Se recomiendan los videos o hacer el paso a paso con los estudiantes debido a que pueden ser estas horas, las primeras del estudiante con temas de configuración del equipo y de programación.

En caso de que algún estudiante en el momento no pueda realizar la instalación de la herramienta en el equipo en el que se encuentre trabajando, se presentan un par de alternativas **TEMPORALES**, para que no se atrase en las primeras lecciones pero es muy importante que pueda instalar la herramienta.

Es hora de iniciar con el primer hola mundo utilizando Python, ya sea desde la terminal, colab, Python online, Visual Studio Code o cualquier otra herramienta, recalcando la importancia de instalarlo y ejecutarlo desde VS Code ya que todo se trabajará desde allí.

Desde VS Code es necesario que se cree un archivo, para ellos la recomendación es generar una carpeta para todo lo relacionado con el curso, dentro de ella la idea será crear archivos y/o carpetas para cada tema. Para este caso se creará un archivo `introduccionPython.py` y escribir las líneas.

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

Se guarda el archivo y se procede a ejecutar, es posible hacerlo de 3 maneras diferentes, la primera es presionando la tecla F5, la segunda desde el menú "Run" y la tercera haciendo clic derecho sobre el archivo en la opción "Run Python". Luego de ejecutado el código, se debe obtener como salida el mensaje "hola mundo" en la consola, como se muestra en la imagen.

```
honFiles/lib/python/debugpy/adapter/../../debugpy/launcher 53500 -- /Users/usuarioutp/Downloads/print\("hola mundo"\).py
hola mundo
```

Fuente: Elaboración propia

Ahora se ha finalizado el primer hola mundo, un programa que se conoce como el primero desarrollado en cualquier tecnología que se esté aprendiendo, en donde se verifica el correcto funcionamiento y el procedimiento para ejecutar el aplicativo.

Como complemento a este programa, podemos agregar comentarios dentro del código fuente, herramienta muy útil para indicar el resultado u objetivo de un bloque de código, los comentarios en Python inician con el símbolo numeral '#', los comentarios son anotaciones de texto que no se ejecutan al momento de correr el programa.

Una herramienta que combina la opción de realizar bloques de código y texto con el fin de explicar o de hacer anotaciones con formato o más extensas, sin dejar de lado los comentarios dentro de los bloques de código, son los Jupyter Notebook.

En este punto se puede presentar COLAB de google para mostrarles a los estudiantes lo que se logra hacer dentro de VS Code y también es de ayuda presentar Markdown como notación de escritura, ya que se utiliza para formatear los textos de los bloques del Notebook. Sintaxis Markdown al completo - Cheatsheet en español

Para crear un notebook, es necesario realizar algunas configuraciones, como instalar las extensiones de Jupyter para VS Code y crear un archivo con la extensión .ipynb, dentro del archivo es necesario seleccionar un kernel para que ejecute Python, es decir la versión de Python que se haya instalado y, al final se tendrá una interfaz similar a la de COLAB.