



# Introducción a la Programación Segura

Estructuras de decisión, bucles y funciones del lenguaje Python.



# INTRODUCCIÓN A LA PROGRAMACIÓN SEGURA

UNIDAD 2: ESTRUCTURAS DE DECISIÓN, BUCLES Y FUNCIONES DEL LENGUAJE PYTHON.

## **Contenidos**

1. Elementos claves para el funcionamiento de un programa.
2. Ejecución de un programa.
3. Diferencias entre compilación e interpretación.
4. Características y funciones del lenguaje Python.
5. Descarga, instalación y configuración de Python.



# UNIDAD DE APRENDIZAJE

UNIDAD N°2: ESTRUCTURAS DE DECISIÓN, BUCLES Y FUNCIONES DEL LENGUAJE PYTHON.

Aprendizaje Esperado:

2.1. Desarrolla scripts de Python, para entregar solución a problemas de mediana complejidad, considerando los estándares internacionales de programación y la evaluación de los resultados obtenidos en base a la evidencia.





# Definición del lenguaje Python:

¿Qué es Python?

Es un **lenguaje de scripting** independiente de plataforma y **orientado a objetos**. Permite realizar casi cualquier tipo de programa, desde aplicaciones de servidores, hasta aplicaciones de IA. **Es un lenguaje interpretado**, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que **ofrece ventajas** como la **rapidez de desarrollo** e inconvenientes como una menor velocidad.



¿Cuándo y quién lo creó?

**Guido van Rossum** comenzó a trabajar en el desarrollo de Python en **diciembre de 1989** en el Centro para las Matemáticas y la Informática (CWI) en los Países Bajos. Python se inspiró en el lenguaje de programación ABC y se diseñó con la intención de crear un lenguaje más legible y con un enfoque en la simplicidad y facilidad de uso.



# Definición y construcción del lenguaje de programación Python:

## ¿Por qué se llama Python?

El nombre "Python" **no tiene relación con la serpiente** del mismo nombre. En realidad, el nombre **proviene** de la afición de Guido por la **serie de televisión británica "Monty Python's Flying Circus"**. Esto es evidente en algunos términos dentro del lenguaje, como la función `spam()` (en referencia a un famoso sketch de Monty Python).



## ¿Cuál es su tipo de Licencia?

Python es un **lenguaje de código abierto** con una licencia de software de Python (Python Software Foundation License). Esto significa que **Python es gratuito** y su código fuente está disponible para que cualquiera lo pueda examinar, modificar y distribuir.

# Elementos claves para el funcionamiento de un programa:

Para desarrollar programas en lenguaje Python deben existir algunos elementos básicos en el computador:

Elementos básicos del entorno de Programación Python:

Interprete de Python:

El intérprete de Python es un **programa especial que lee y ejecuta código fuente** escrito en lenguaje Python. En lugar de compilar el código fuente a lenguaje de máquina como lo hacen algunos lenguajes de programación, Python utiliza un enfoque de "interpretación".

Cuando escribes un programa en Python y lo ejecutas, **el intérprete lee línea por línea el código fuente y lo traduce a instrucciones que la máquina puede entender y ejecutar directamente**. Esto permite que el intérprete pueda **interpretar y ejecutar** el código **en tiempo real**.

# Diferencias entre compilación e interpretación:

La principal diferencia entre un lenguaje compilado y un lenguaje interpretado radica en cómo se procesa y ejecuta el código fuente de un programa.

## Lenguaje Interpretado:

- En un lenguaje interpretado, el **código** fuente **se traduce y ejecuta línea por línea en tiempo real** por un programa llamado "**intérprete**".
- El **intérprete lee y ejecuta el código** fuente directamente, **traduciendo cada instrucción** a lenguaje de máquina y ejecutándola en el momento en que se encuentra.
- Como el código se interpreta en tiempo real, el **proceso de ejecución puede ser más lento** en comparación con los lenguajes compilados, ya que el intérprete debe realizar la traducción y ejecución de cada instrucción en cada ejecución del programa.
- Los lenguajes interpretados **son más flexibles y portátiles**, ya que el mismo **código fuente puede ejecutarse en diferentes plataformas** siempre que haya un intérprete disponible para esa plataforma.



# Diferencias entre compilación e interpretación:

La principal diferencia entre un lenguaje compilado y un lenguaje interpretado radica en cómo se procesa y ejecuta el código fuente de un programa.

## Lenguaje Compilado:

- El **código** fuente del programa **se traduce completamente a lenguaje de máquina** antes de que el programa se ejecute. Este **proceso** de traducción **se lleva a cabo por** un programa llamado "**compilador**".
- El compilador **analiza** todo el **código** fuente y **lo traduce** en un archivo ejecutable específico para la plataforma objetivo.
- Una vez que el código se ha compilado, **no es necesario recompilarlo cada vez** que se ejecute el programa. El archivo ejecutable resultante puede ejecutarse directamente y no requiere la presencia del compilador.
- Los lenguajes compilados **tienden a ser más rápidos en la ejecución**, ya que el código **ya está traducido a lenguaje de máquina** y no necesita interpretarse línea por línea en tiempo real.





Programming  
with python



Programming  
with C++



Programming  
directly to  
binary



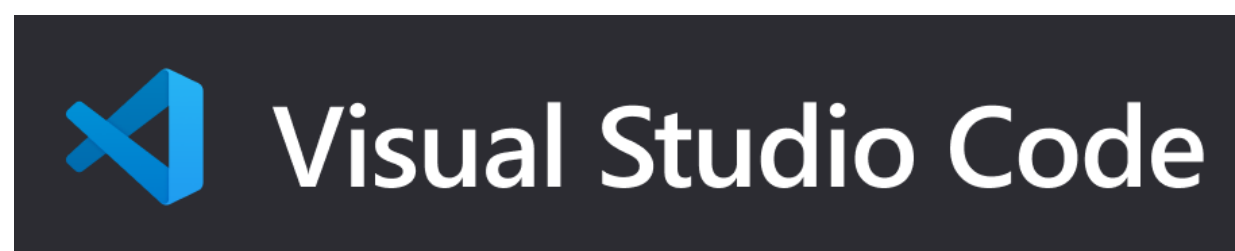
# Elementos claves para el funcionamiento de un programa:

Para desarrollar programas en lenguaje Python deben existir algunos elementos básicos en el computador:

Elementos básicos del entorno de Programación Python:

Editor de código:

Es una **aplicación que te permite escribir, editar y guardar tus programas en Python**. Hay varios editores de código disponibles, como **Visual Studio Code**, **PyCharm**, **Atom**, **Sublime Text**, entre otros. Estos editores suelen tener características útiles como resaltado de sintaxis, autocompletado y depuración.





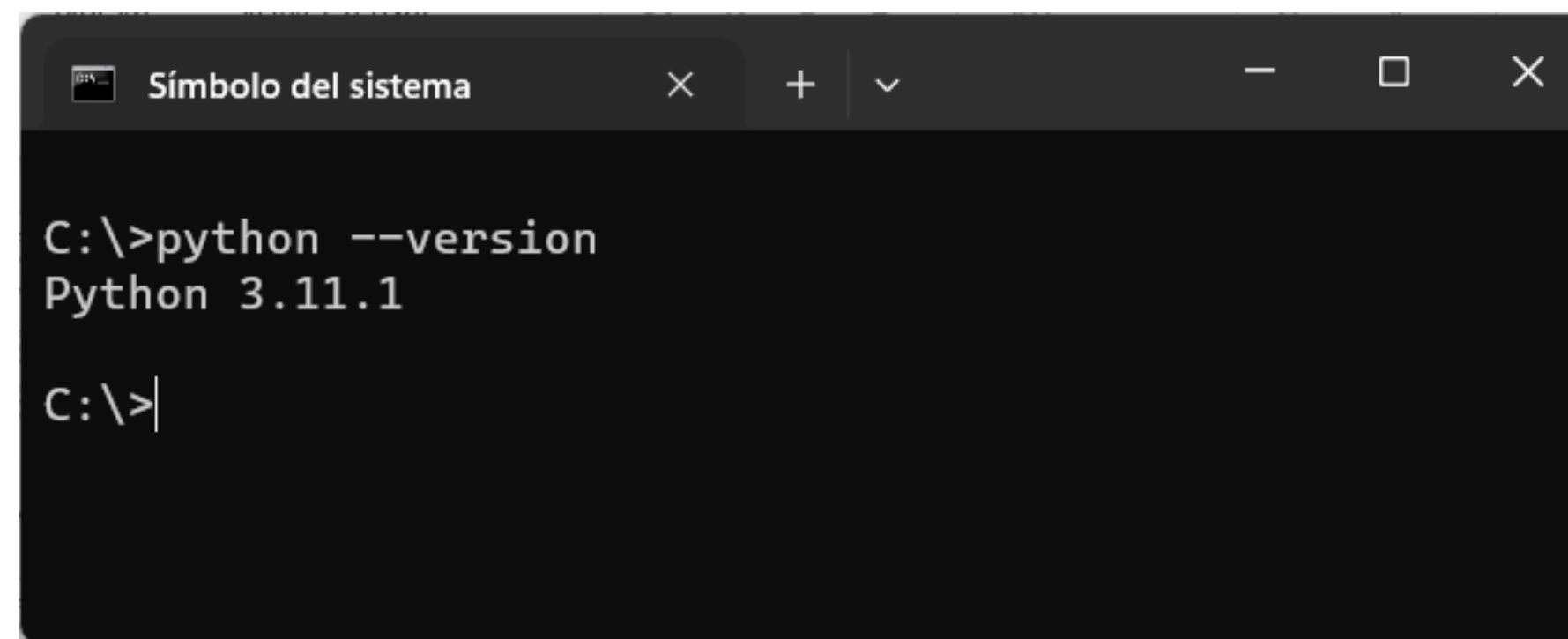
# Elementos claves para el funcionamiento de un programa:

Para desarrollar programas en lenguaje Python deben existir algunos elementos básicos en el computador:

Elementos básicos del entorno de Programación Python:

Terminal o consola:

Es una **interfaz** de texto que **permite interactuar con el sistema operativo y ejecutar comandos**. A través de la terminal, puedes ejecutar programas de Python, instalar paquetes, gestionar archivos y realizar otras tareas relacionadas con el desarrollo.



```
Símbolo del sistema
C:\>python --version
Python 3.11.1
C:\>
```

# Elementos claves para el funcionamiento de un programa:

Para desarrollar programas en lenguaje Python deben existir algunos elementos básicos en el computador:

Elementos básicos del entorno de Programación Python:

## Módulos y bibliotecas:

Python tiene una amplia variedad de módulos y bibliotecas que extienden su funcionalidad. Los **módulos son archivos de Python que contienen funciones y variables** que pueden ser importadas en otros programas. Las **bibliotecas son conjuntos de módulos** relacionados que ofrecen funcionalidades adicionales para tareas específicas, como matemáticas, manipulación de archivos, acceso a bases de datos, entre otros.

## Ambiente virtual (Virtualenv):

Es una herramienta que te **permite crear un entorno de desarrollo aislado para cada proyecto**. Esto es útil para **evitar conflictos entre las dependencias** de diferentes proyectos y mantener un entorno limpio y ordenado.



# Elementos claves para el funcionamiento de un programa:

Para desarrollar programas en lenguaje Python deben existir algunos elementos básicos en el computador:

Elementos básicos del entorno de Programación Python:

Gestor de paquetes (pip):

Es una herramienta que te [permite instalar, desinstalar y gestionar paquetes y bibliotecas de Python](#) desde el repositorio oficial PyPI (Python Package Index).

Documentación oficial:

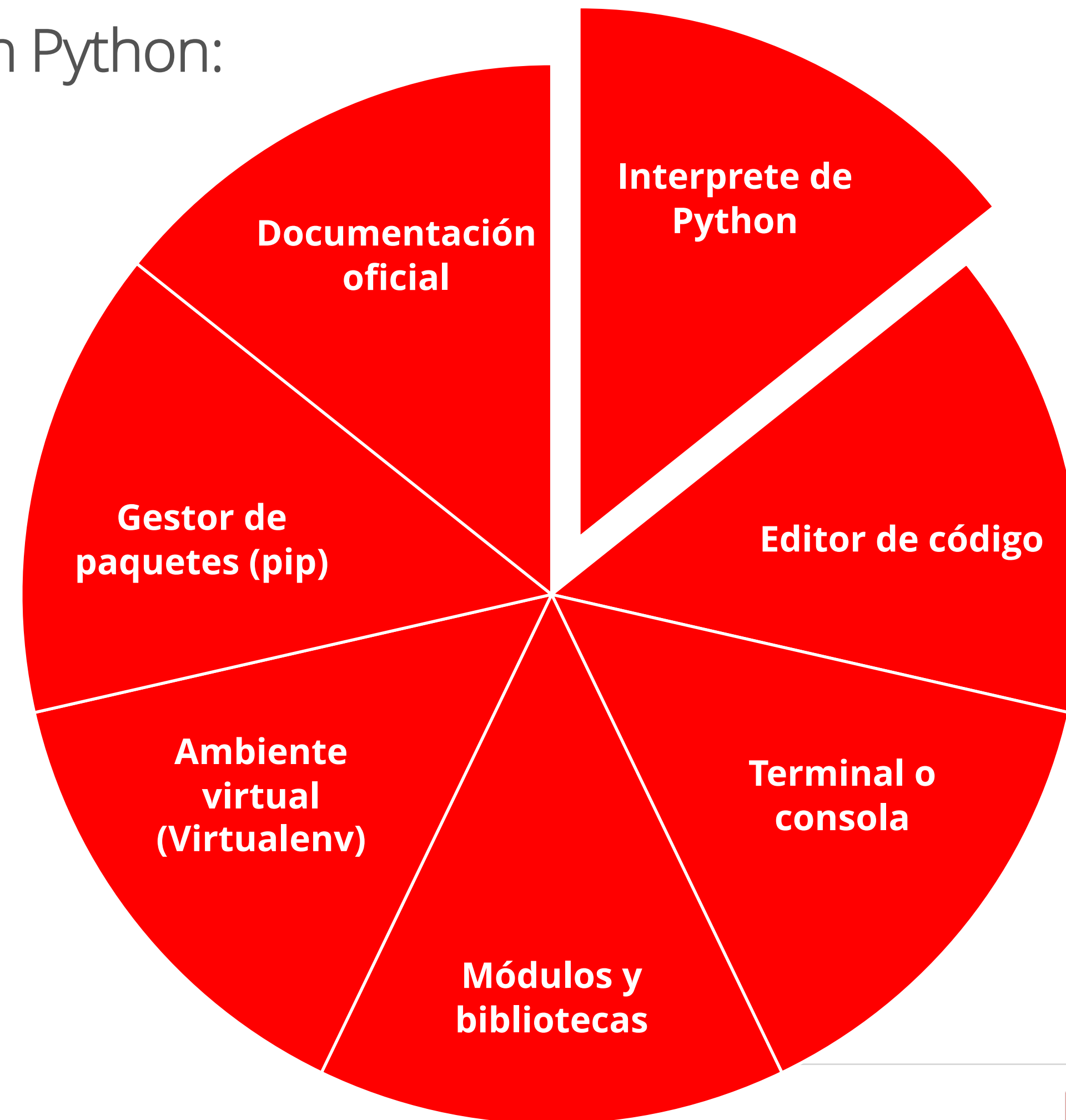
Python tiene una [extensa documentación oficial que incluye tutoriales, guías y la referencia completa](#) del lenguaje y sus bibliotecas. Es una fuente invaluable de información para los programadores de Python.

# Elementos claves para el funcionamiento de un programa:

Para desarrollar programas en lenguaje Python deben existir algunos elementos básicos en el computador:

## Elementos básicos del entorno de Programación Python:

Estos elementos trabajan juntos para proporcionar un **entorno de desarrollo completo y robusto para escribir, ejecutar y depurar programas en Python**. Con ellos, los desarrolladores pueden crear aplicaciones, scripts y proyectos de todo tipo, desde pequeñas herramientas hasta aplicaciones web complejas y proyectos de inteligencia artificial.





# Ejecución de un programa:

Los programas son secuencias de instrucciones escritas en un lenguaje de programación específico, como Python. Cuando ejecutas un programa, este pasa por una serie de etapas para que la computadora pueda entenderlo y realizar las tareas que le has indicado:

## Etapas principales del funcionamiento de un programa:

### Edición:

En esta etapa, utilizas un editor de código para **escribir el código fuente del programa**. El código fuente es el conjunto de instrucciones escritas en lenguaje de programación que define lo que el programa debe hacer.

### Análisis léxico y sintáctico:

Una vez que has escrito el código fuente, el intérprete o compilador (dependiendo del lenguaje) **analiza el código para identificar los componentes léxicos** (como palabras clave, identificadores, operadores) **y verifica que la sintaxis del programa sea correcta**, es decir, que las instrucciones estén escritas en el formato correcto.

# Ejecución de un programa:

Los programas son secuencias de instrucciones escritas en un lenguaje de programación específico, como Python. Cuando ejecutas un programa, este pasa por una serie de etapas para que la computadora pueda entenderlo y realizar las tareas que le has indicado:

Etapas principales del funcionamiento de un programa (lenguaje interpretado Python):

## Ejecución:

A diferencia de los lenguajes compilados, Python no genera un código ejecutable independiente previamente. En su lugar, el intérprete **ejecuta el código directamente línea por línea**. Cada instrucción se **traduce en tiempo real a instrucciones de bajo nivel que la computadora** puede entender y ejecutar.

## Interacción y E/S (Entrada/Salida):

Durante la ejecución, el programa puede **interactuar con el usuario a través de la consola** mediante la función **input()** para obtener **datos de entrada** y la función **print()** para **mostrar resultados** en pantalla. También puede realizar operaciones de entrada/salida con archivos en el disco duro utilizando las funciones de manejo de archivos de Python.



# Ejecución de un programa:

Los programas son secuencias de instrucciones escritas en un lenguaje de programación específico, como Python. Cuando ejecutas un programa, este pasa por una serie de etapas para que la computadora pueda entenderlo y realizar las tareas que le has indicado:

Etapas principales del funcionamiento de un programa (lenguaje interpretado Python):

## Finalización:

Cuando el programa ha completado su ejecución o encuentra una declaración return (si es una función), el **programa finaliza y puede mostrar el resultado final** o, en algunos casos, mantener la consola abierta para interactuar con el usuario.

# Lenguaje Python, cómo se posiciona entre otras programaciones:

Python se ha convertido en **uno de los lenguajes de programación más populares** debido a su facilidad de uso, versatilidad, robusta comunidad de desarrolladores y una amplia gama de aplicaciones. Es una excelente opción tanto para principiantes como para profesionales y continúa creciendo en adopción y relevancia en la industria de la tecnología.





# Lenguaje Python, cómo se posiciona entre otras programaciones:

## Aspectos que contribuyen al éxito del Lenguaje Python:

### Simplicidad y Legibilidad:

- ✓ **Sintaxis clara y sencilla** que **se asemeja mucho al lenguaje humano**. Esto facilita la lectura y escritura del código, lo que lo hace **ideal para principiantes y también para desarrolladores experimentados** que buscan una forma rápida y eficiente de expresar sus ideas.

### Amplia Comunidad y Soporte:

- ✓ Python tiene una **comunidad activa y colaborativa de desarrolladores** en todo el mundo. Esta comunidad **ha generado una gran cantidad de bibliotecas y módulos** para casi cualquier propósito imaginable. Además, existe una **amplia documentación y recursos de aprendizaje** disponibles en línea.

# Lenguaje Python, cómo se posiciona entre otras programaciones:

## Aspectos que contribuyen al éxito del Lenguaje Python:

### Versatilidad:

- ✓ Python es conocido como un "lenguaje multipropósito" porque se puede utilizar para una amplia variedad de aplicaciones, desde desarrollo web y automatización hasta inteligencia artificial y análisis de datos. Su flexibilidad lo convierte en una excelente elección para diferentes proyectos y tareas.

### Facilidad de Aprendizaje:

- ✓ La legibilidad y simplicidad de Python hacen que sea fácil de aprender, especialmente para aquellos que son nuevos en la programación. Esto ha llevado a que Python sea utilizado en muchos cursos introductorios de programación en universidades y escuelas.



# Lenguaje Python, cómo se posiciona entre otras programaciones:

## Aspectos que contribuyen al éxito del Lenguaje Python:

### Apoyo a Bibliotecas y Frameworks:

- ✓ Python cuenta con una **amplia gama de bibliotecas y frameworks que facilitan el desarrollo de aplicaciones web, ciencia de datos, aprendizaje automático, entre otros**. Algunos ejemplos populares son Django para desarrollo web, NumPy y Pandas para análisis de datos, y TensorFlow para aprendizaje automático.

### Comunidad de Aprendizaje:

- ✓ Python ha atraído a una **gran comunidad de aprendizaje y educación en línea**. Plataformas como Codecademy, Coursera y edX ofrecen cursos interactivos y proyectos para aprender Python de manera accesible y práctica.







# Versiones de Python :

Las principales versiones de Python que han sido lanzadas y son utilizadas ampliamente son Python 2 y Python 3. Cada una de estas versiones tiene varias versiones secundarias con correcciones de errores y mejoras.

## Principales versiones de Python:

**Python 1.x:** Las primeras versiones de Python se lanzaron en **1991**. Las versiones **1.0** y **1.5** fueron versiones iniciales del lenguaje, con limitaciones y características básicas.

**Python 2.x:** Se lanzó la primera versión de Python 2.0 en octubre de **2000**. Python 2.x trajo muchas mejoras y nuevas características al lenguaje. Las versiones **2.1, 2.2, 2.3**, y así sucesivamente, continuaron mejorando el lenguaje.

**Python 3.x:** La primera versión de Python 3.0 se lanzó en diciembre de **2008**. Python 3.x introdujo cambios significativos en el lenguaje para abordar ciertos problemas de diseño y mejorar la legibilidad y consistencia del código. Python 3.0 fue una versión disruptiva, y versiones posteriores, como Python 3.1, 3.2, 3.3, y así sucesivamente, han traído más mejoras y optimizaciones. **La versión actual es Python**

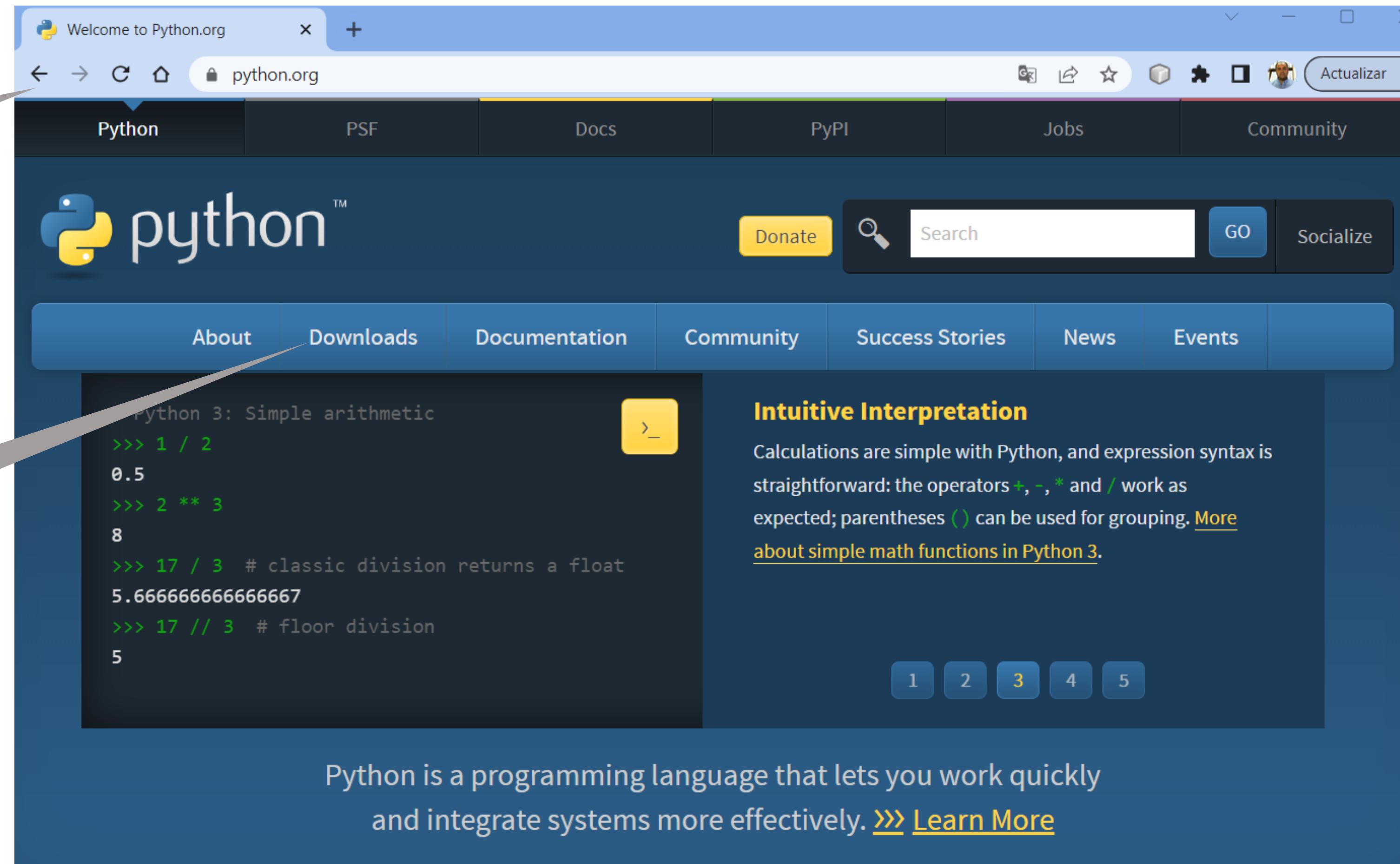


# Descarga, instalación y configuración de Python:

Descargando el Intérprete:

Ingresa a la url  
<https://www.python.org>

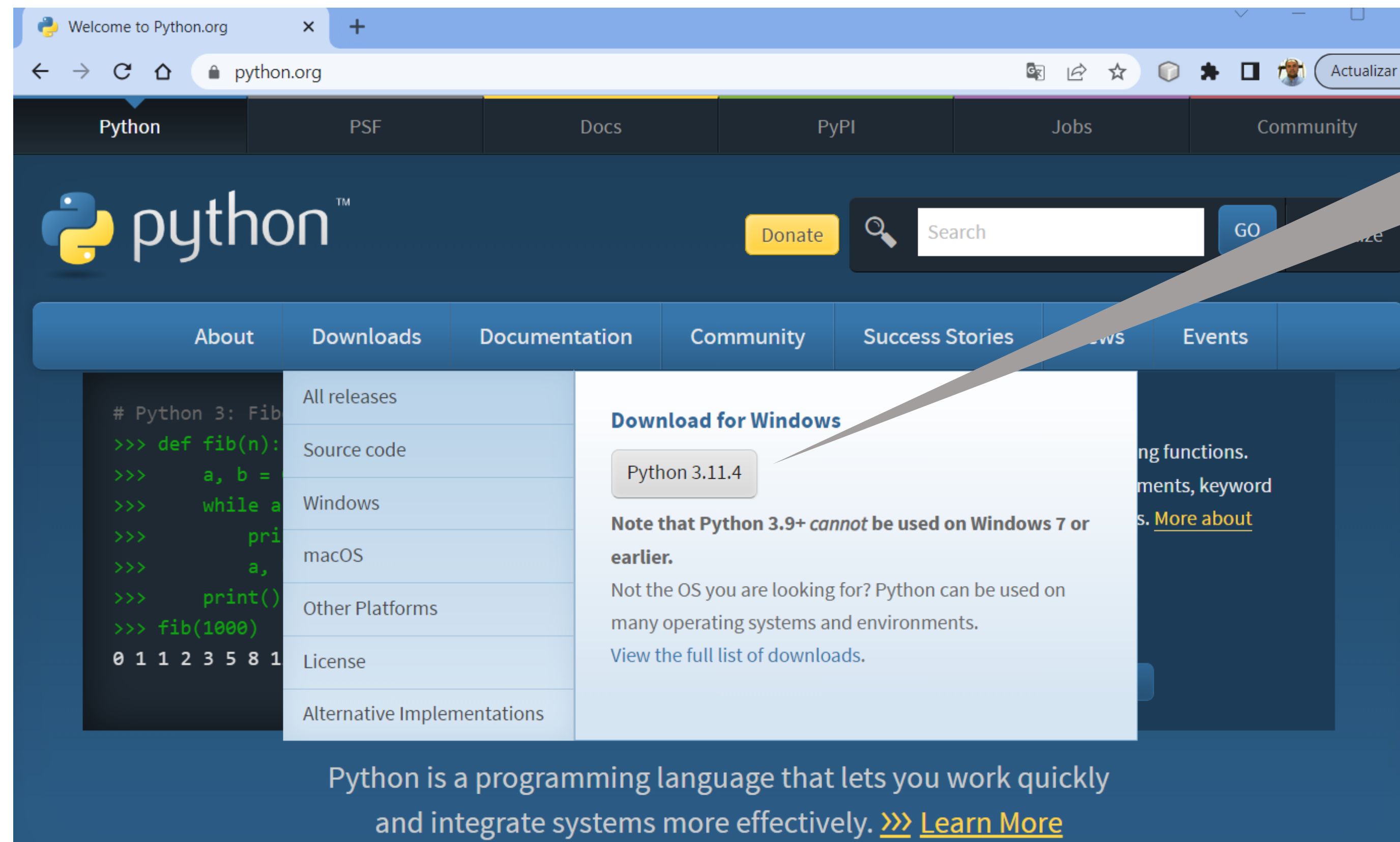
Hacemos click  
en Downloads





# Descarga, instalación y configuración de Python:

Descargando el Intérprete:



Hacemos click en Python 3.11.4 para descargar la última versión

Con esto obtendremos el archivo de instalación



python-3.11.4-a  
md64.exe

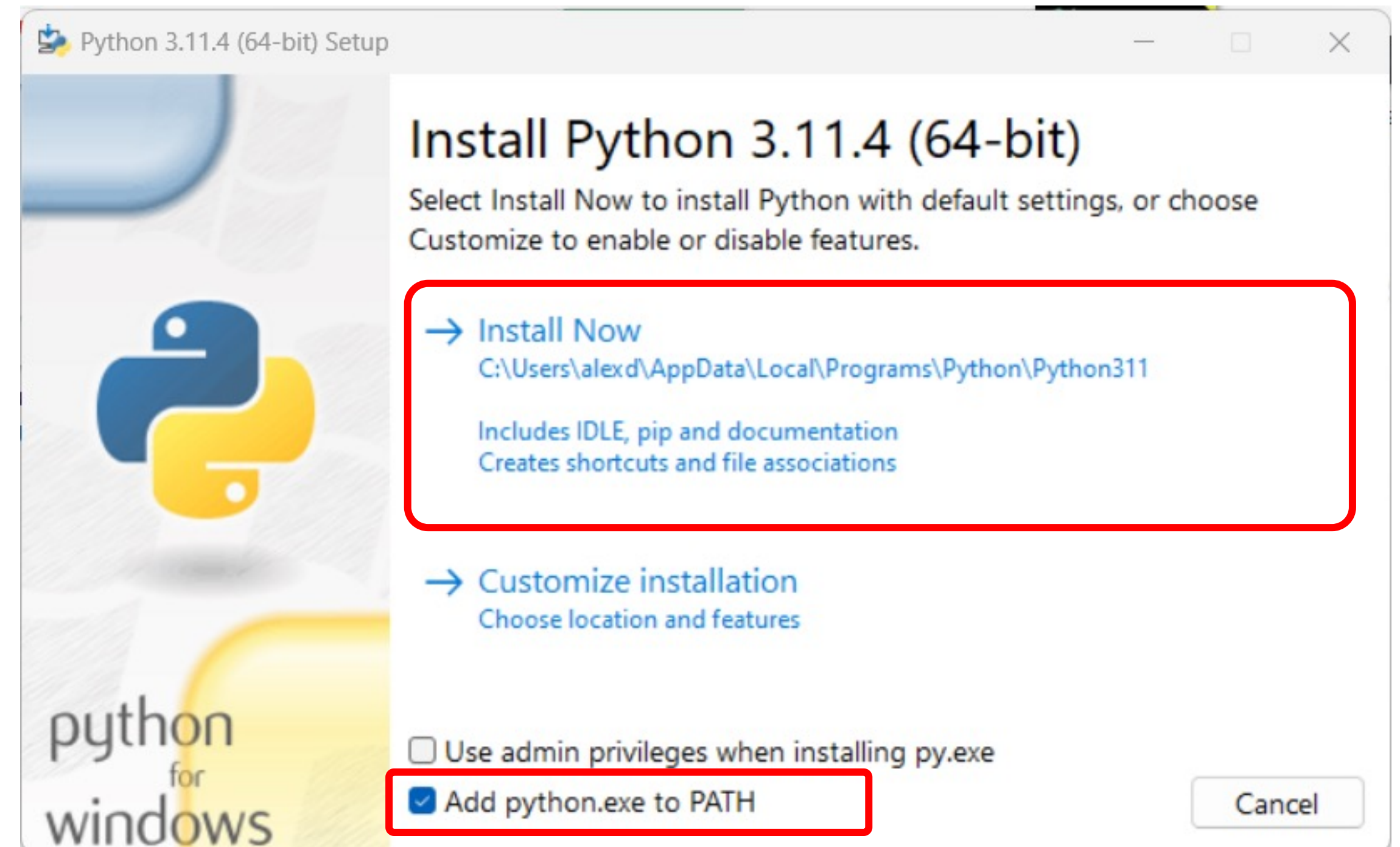
# Descarga, instalación y configuración de Python:

Instalando el Intérprete:



Ejecutamos el instalador

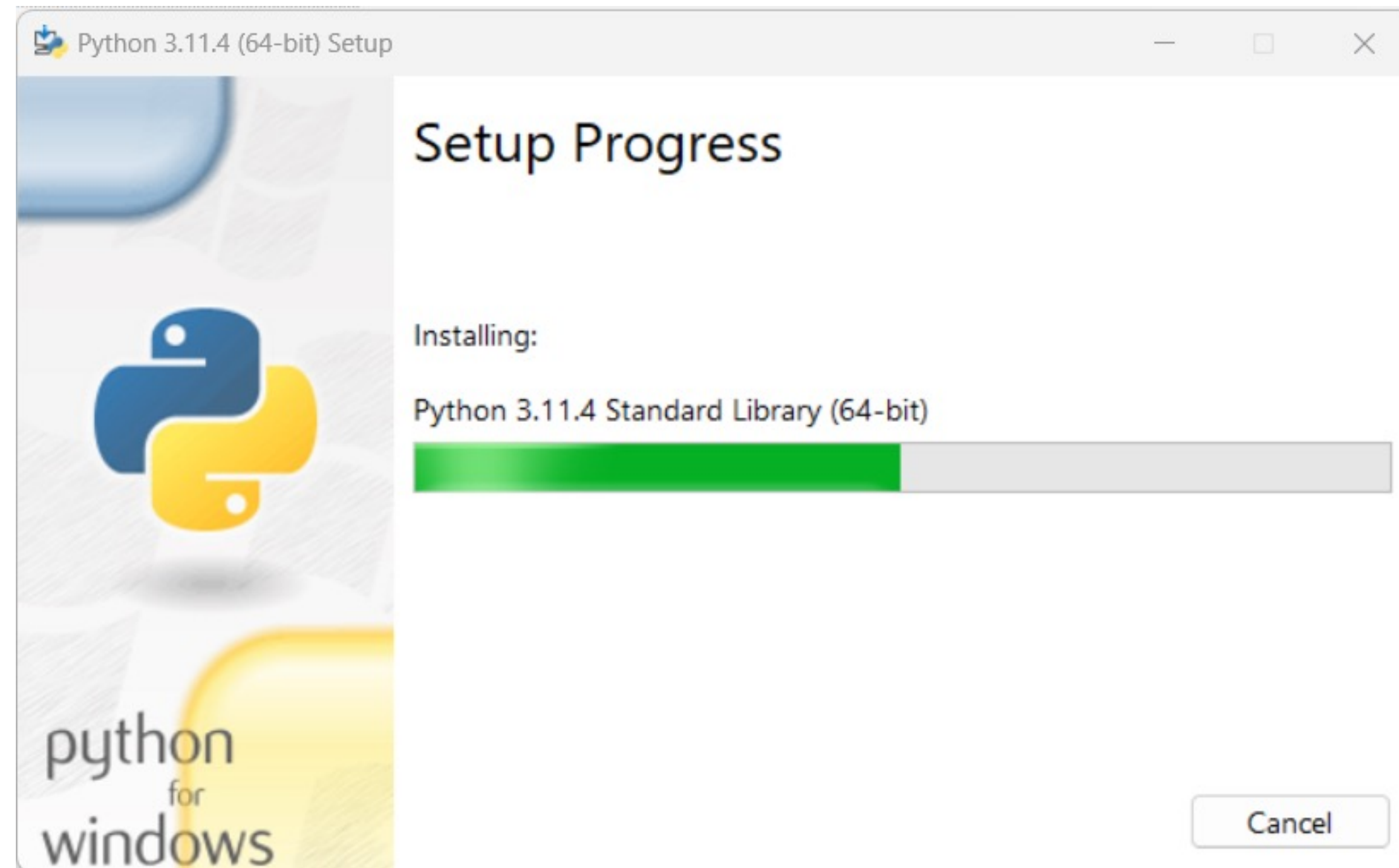
Debemos marcar la casilla **“Add Python.exe to PATH”** de lo contrario no podremos utilizar comandos de Python desde la Consola del Sistema Operativo



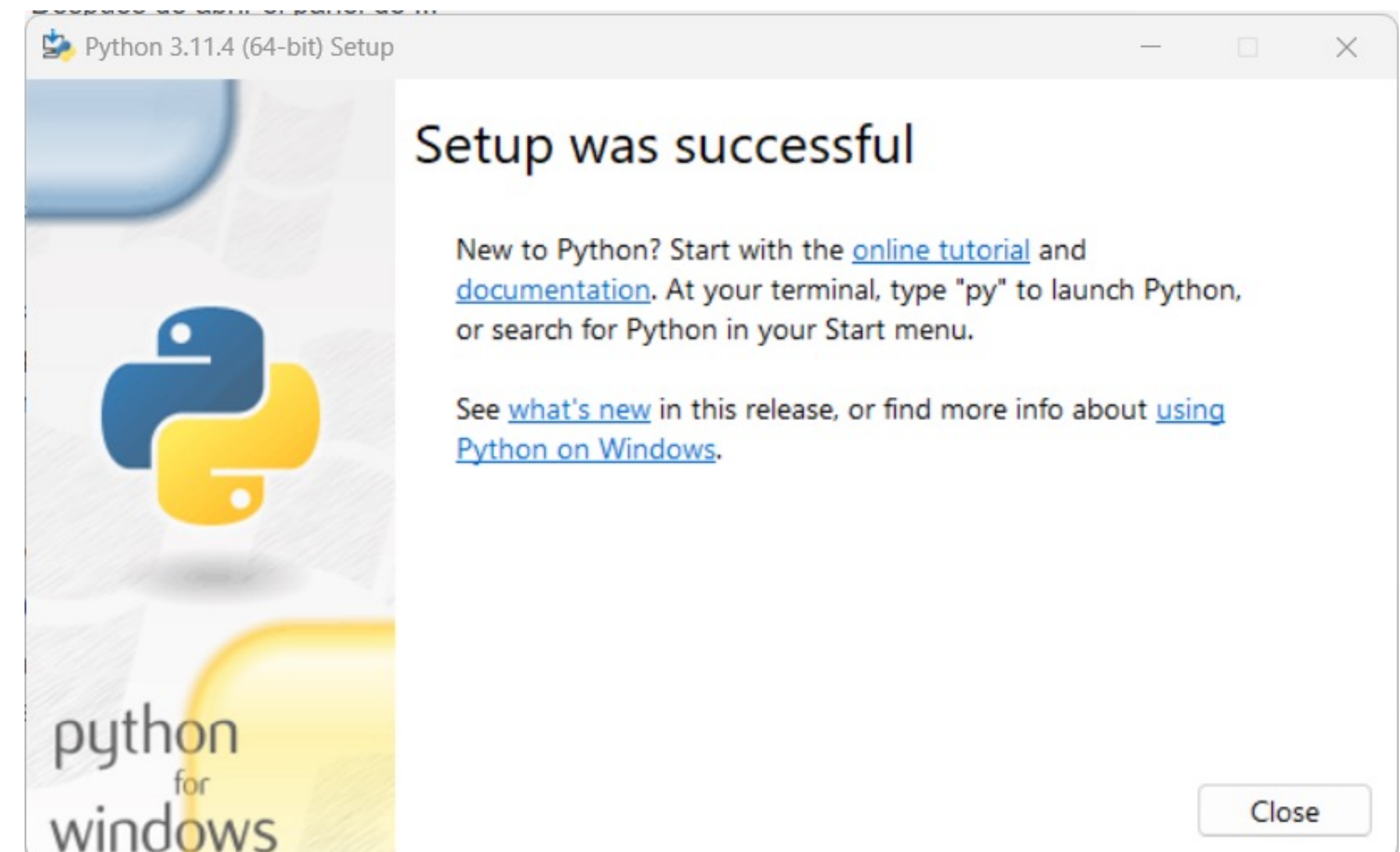


# Descarga, instalación y configuración de Python:

Instalando el Intérprete:



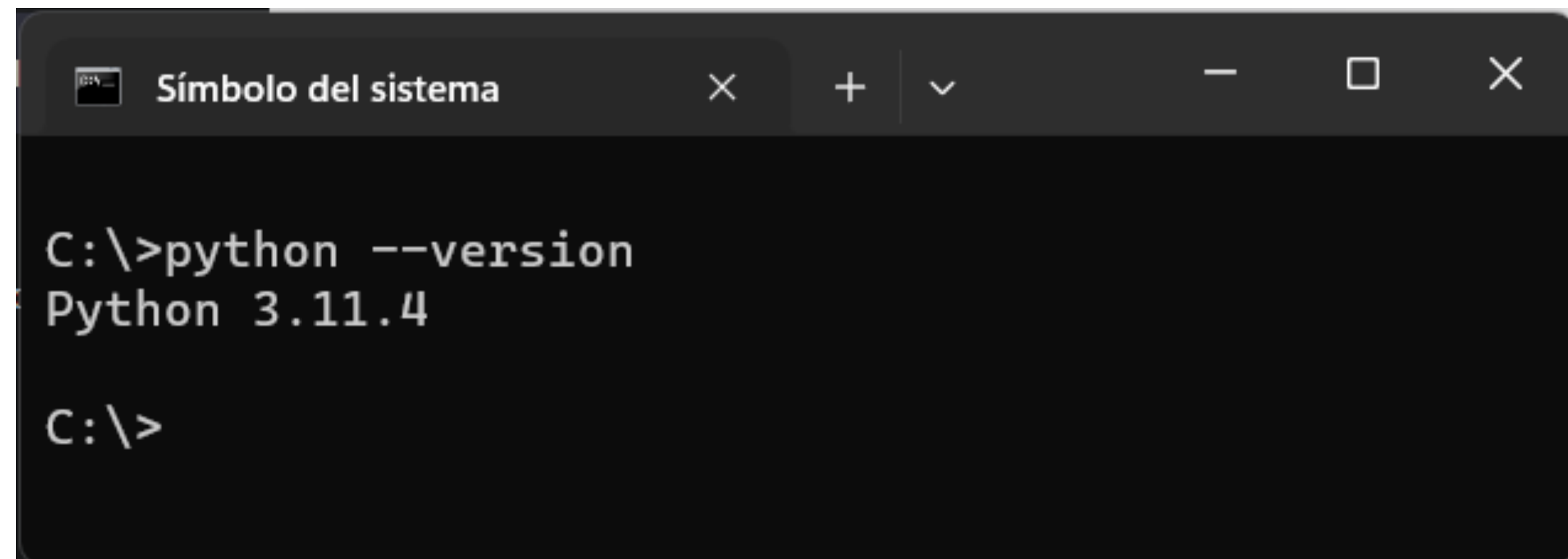
Finalmente tendremos la instalación completa



# Descarga, instalación y configuración de Python:

Instalando el Intérprete:

Ahora podemos comprobar la instalación desde la consola de comandos con el comando **Python --version**



```
C:\>python --version
Python 3.11.4

C:\>
```





**A reforzar lo aprendido ...**

# Ejercicios:

De acuerdo al contenido visto en clases, responde las siguientes preguntas:

Ejecución de un programa:

¿Qué es el código fuente de un programa y qué sucede con él durante la ejecución?

R: El código fuente de un programa es el conjunto de instrucciones escritas en un lenguaje de programación antes de ser traducido o interpretado durante la ejecución.

¿Cuál es la diferencia entre un programa compilado y un programa interpretado?

R: Un programa compilado se traduce completamente a lenguaje de máquina antes de ejecutarse, mientras que un programa interpretado se traduce línea por línea en tiempo real.

Menciona al menos tres lenguajes de programación que requieran de un intérprete para ejecutarse.

R: Ejemplos de lenguajes interpretados incluyen Python, JavaScript y Ruby.



# Ejercicios:

De acuerdo al contenido visto en clases, responde las siguientes preguntas:

Definición y construcción del lenguaje de programación Python:

¿Quién es el creador de Python y en qué año fue lanzado por primera vez?

R: Python fue creado por Guido van Rossum y se lanzó por primera vez en 1991.

¿Cuál es el enfoque principal de Python en términos de sintaxis y diseño?

R: El enfoque principal de Python es la legibilidad y la simplicidad de su sintaxis.

¿Qué es "The Zen of Python"? Menciona al menos dos principios de esta filosofía.

R: "The Zen of Python" es una colección de principios que guían el diseño del lenguaje. Ejemplos: "Beautiful is better than ugly" y "Readability counts".

# Ejercicios:

De acuerdo al contenido visto en clases, responde las siguientes preguntas:

Diferencias entre compilación e interpretación:

Explica en qué consiste el proceso de compilación de un programa.

R: El proceso de compilación de un programa implica traducir todo el código fuente a lenguaje de máquina en un archivo ejecutable.

Describe cómo funciona el proceso de interpretación de un programa.

R: La interpretación de un programa implica traducir y ejecutar el código fuente línea por línea en tiempo real mediante un intérprete.

¿Cuáles son las ventajas y desventajas de la compilación y la interpretación?

R: Ventajas de la compilación: Mayor rendimiento de ejecución. Ventajas de la interpretación: Mayor portabilidad y flexibilidad.



# Ejercicios:

De acuerdo al contenido visto en clases, responde las siguientes preguntas:

Lenguaje Python, cómo se posiciona entre otras programaciones:

Enumera al menos tres áreas en las que Python es ampliamente utilizado.

R: Python es ampliamente utilizado en desarrollo web, análisis de datos, inteligencia artificial y automatización, entre otros.

¿Qué características de Python lo hacen atractivo para principiantes y desarrolladores experimentados?

R: Python se destaca por su legibilidad y simplicidad, lo que lo hace atractivo para principiantes y desarrolladores experimentados.

Compara Python con otro lenguaje de programación que conozcas, resaltando sus diferencias y similitudes.

R: Comparación con otro lenguaje depende de la elección, podría compararse con Java, Ruby, C#, entre otros.

# Ejercicios:

De acuerdo al contenido visto en clases, responde las siguientes preguntas:

Versiones de Python:

¿Cuál es la versión principal de Python que ha llegado a su fin de vida y ya no recibe soporte oficial?

R: Python 2.x ha llegado a su fin de vida y ya no recibe soporte oficial.

Menciona al menos dos mejoras o características nuevas introducidas en Python 3.x en comparación con Python 2.x.

R: En Python 3.x se mejoró la gestión de bytes y cadenas, y se introdujo el operador de matriz (matrix multiplication operator).

¿Por qué es importante mantenerse actualizado con las últimas versiones de Python?

R: Mantenerse actualizado con las últimas versiones de Python es importante para obtener mejoras de seguridad, características y correcciones de errores.



# Ejercicios:

De acuerdo al contenido visto en clases, responde las siguientes preguntas:

Descarga, instalación y configuración de Python:

Explica cómo puedes descargar e instalar Python en un sistema operativo Windows.

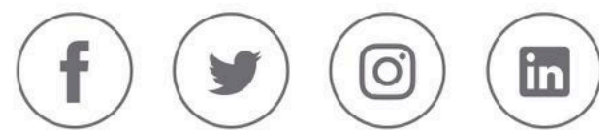
R: Para descargar e instalar Python en Windows, se debe visitar el sitio web oficial de Python, descargar el instalador correspondiente y ejecutarlo.

¿Qué es un entorno virtual (Virtualenv) y cómo se crea uno en Python?

R: Un entorno virtual (Virtualenv) es un ambiente aislado que permite instalar paquetes y bibliotecas específicos para un proyecto sin afectar otros proyectos.

¿Cuál es el propósito de configurar la variable de entorno PATH al instalar Python?

R: Configurar la variable de entorno PATH al instalar Python permite acceder a Python desde cualquier ubicación en la línea de comandos.



inacap.cl