# Tutorial Instalación De Python

Para instalar Python, lo primero que debía hacerse era visitar el sitio web oficial, [**www.python.org**](https://www.python.org). Desde la página principal, el usuario podía hacer clic en la pestaña “Downloads”, aunque también tenía la opción de ir directamente al enlace [**www.python.org/downloads/**](https://www.python.org/downloads/), donde el sitio detectaba automáticamente el sistema operativo y mostraba el instalador correspondiente, ya fuera para Windows, macOS o Linux, además de identificar si el sistema era de 32 o 64 bits. Una vez allí, debía hacer clic en el botón de descarga para obtener el archivo de instalación de la última versión estable de Python. En el momento en que se escribió este tutorial, la versión más reciente era Python 3.11.3. Si el usuario necesitaba una versión anterior, podía desplazarse hacia abajo en la misma página y seleccionar la que deseaba mediante los enlaces correspondientes.

Después de completar la descarga, se procedía a ejecutar el archivo de instalación. En el caso de Windows, este era un archivo .exe. Al abrir el instalador, se mostraban dos opciones: **“Install Now”** y **“Customize Installation**”. Si se seleccionaba **“Install Now”**, Python se instalaba de forma automática con las configuraciones predeterminadas. Esta era la opción más recomendada para principiantes. No obstante, si se necesitaban ajustes más específicos como elegir una carpeta diferente, instalar componentes opcionales o modificar opciones avanzadas, se podía optar por **“Customize Installation”**. Antes de hacer clic en cualquiera de estas opciones, era fundamental marcar la casilla que decía **“Add python.exe to PATH”**. Esta acción añadía Python a la variable de entorno del sistema, lo que permitía ejecutarlo desde cualquier terminal o símbolo del sistema sin tener que navegar manualmente hasta su carpeta de instalación.

Una vez iniciada la instalación, el sistema podía solicitar permiso para realizar cambios en el dispositivo. Tras conceder dicho permiso, comenzaba el proceso de instalación propiamente dicho, durante el cual se copiaban todos los archivos necesarios, incluida la biblioteca estándar de Python. Al finalizar, aparecía un cuadro de diálogo confirmando que Python se había instalado correctamente. En ese momento, el usuario solo tenía que hacer clic en **“Close”** para cerrar el instalador.

Para verificar que la instalación fue exitosa, el usuario debía abrir la terminal o el símbolo del sistema escribiendo **“cmd”** en el menú de inicio. Dentro de la terminal, al escribir el comando **“Python –version”**, debía mostrarse la versión instalada. Al escribir simplemente Python, se accedía a la consola interactiva de Python, conocida como el **“shell”.** Allí aparecía un mensaje con tres signos de mayor que (>>>), indicando que el intérprete estaba listo para recibir comandos. Como prueba de funcionamiento, el usuario podía escribir 8 + 2 y pulsar Enter. El resultado, 10, se mostraba en la línea siguiente. Luego, al escribir **print("Hola mundo")**, debía aparecer el mensaje **“Hola mundo”** en pantalla. Para salir de esta consola interactiva, era posible escribir exit() o quit(), o también usar la combinación de teclas Ctrl + Z seguida de Enter.

Python incluía un entorno de desarrollo llamado IDLE, que se instalaba automáticamente junto con el lenguaje. Para abrirlo, se escribía **“IDLE”** en el buscador de Windows. Una vez iniciado, IDLE mostraba una ventana similar a la consola de Python, donde también aparecía el prompt >>>. Este entorno ofrecía funciones adicionales como autocompletado, sangría automática, resaltado de sintaxis y un depurador básico, lo que lo convertía en una buena opción para comenzar a escribir y probar programas de Python sin necesidad de instalar herramientas externas.

En sistemas macOS, el proceso era muy similar. Muchos equipos venían con una versión de Python preinstalada, generalmente Python 2.x, que ya no era recomendada para nuevos desarrollos. Para verificar qué versión estaba disponible, el usuario debía abrir la aplicación Terminal y escribir python --version. Si quería comprobar la presencia de Python 3, debía usar el comando python3 --version. Si Python 3 no estaba instalado, era necesario descargar el archivo instalador .pkg desde el sitio web oficial. Luego de completar la descarga, el usuario debía hacer doble clic en el archivo para iniciar el instalador, el cual guiaba paso a paso a través del proceso. Durante la instalación, era posible que se solicitara la contraseña del administrador del sistema. Una vez finalizada la instalación, se abría automáticamente la carpeta de instalación, donde podía encontrarse la aplicación IDLE. Para confirmar que todo funcionaba correctamente, bastaba con abrir IDLE y escribir en su consola **print("Hola mundo").** También se podía volver a la terminal y ejecutar python3 para abrir la consola interactiva, desde donde se podían ingresar instrucciones directamente. Para salir de esta consola, se usaba el comando exit() o la combinación Ctrl + D.

En los sistemas Linux, Python casi siempre venía preinstalado. Para confirmar si estaba disponible y qué versión se encontraba instalada, se podía usar el comando python3 --version o, si existía, python --version. En caso de que Python 3 no estuviera presente, su instalación se realizaba a través del gestor de paquetes de la distribución correspondiente. En sistemas basados en Debian como Ubuntu, se podía ejecutar sudo apt update seguido de sudo apt install python3. En Fedora, el comando adecuado era sudo dnf install python3, y en Arch Linux, se usaba sudo pacman -S python.

Una vez instalado Python, el usuario podía escribir sus programas en diversos entornos. Aunque IDLE estaba disponible de manera predeterminada en Windows y macOS, también existían otras opciones más potentes. Por ejemplo, PyCharm, desarrollado por JetBrains, era un entorno de desarrollo profesional con versiones gratuitas y de pago, ideal para proyectos de mayor envergadura. Visual Studio Code, conocido como VS Code, era una opción muy popular gracias a su ligereza y extensibilidad mediante extensiones como el plugin de Python. Sublime Text, aunque más simple, era rápido y compatible con múltiples lenguajes. Los usuarios más avanzados también podían integrar Python en editores como Vim o Emacs mediante plugins especializados. Otra opción era Eclipse, que admitía desarrollo en Python mediante el complemento PyDev. Para quienes preferían no instalar nada en su equipo, plataformas como [Replit](https://replit.com) o Google Colab ofrecían intérpretes de Python directamente en línea, accesibles desde cualquier navegador.

Independientemente del entorno elegido, el paso más importante era asegurarse de que Python estuviera instalado correctamente y que el usuario se sintiera cómodo ejecutando instrucciones básicas. Con ello, quedaba listo para comenzar a desarrollar sus primeros scripts y explorar el mundo de la programación en Python.