

¿Qué es Python?

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo.

¿Qué beneficios ofrece Python?

Los beneficios de Python incluyen los siguientes:

- Los desarrolladores pueden leer y comprender fácilmente los programas de Python debido a su sintaxis básica similar a la del inglés.
- Python permite que los desarrolladores sean más productivos, ya que pueden escribir un programa de Python con menos líneas de código en comparación con muchos otros lenguajes.
- Python cuenta con una gran biblioteca estándar que contiene códigos reutilizables para casi cualquier tarea. De esta manera, los desarrolladores no tienen que escribir el código desde cero.
- Los desarrolladores pueden utilizar Python fácilmente con otros lenguajes de programación conocidos, como Java, C y C++.
- La comunidad activa de Python incluye millones de desarrolladores alrededor del mundo que prestan su apoyo. Si se presenta un problema, puede obtener soporte rápido de la comunidad.
- Hay muchos recursos útiles disponibles en Internet si desea aprender Python. Por ejemplo, puede encontrar con facilidad videos, tutoriales, documentación y guías para desarrolladores.
- Python se puede trasladar a través de diferentes sistemas operativos de computadora, como Windows, macOS, Linux y Unix.

¿Cómo se utiliza Python?

El lenguaje Python se aplica a varios casos de uso en el desarrollo de aplicaciones, incluidos los ejemplos siguientes:

Desarrollo web del lado del servidor

El desarrollo web del lado del servidor incluye las funciones complejas de *backend* que los sitios web llevan a cabo para mostrar información al usuario. Por ejemplo, los sitios web deben interactuar con las bases de datos, comunicarse con otros sitios web y proteger los datos cuando se los envía a través de la red.

Python es útil para escribir código del lado del servidor debido a que ofrece muchas bibliotecas que constan de código preescrito para crear funciones de *backend* complejas. Los desarrolladores también utilizan un amplio rango de marcos de Python que proporcionan todas las herramientas necesarias para crear aplicaciones web con mayor rapidez y facilidad. Por ejemplo, los desarrolladores pueden crear la aplicación web esqueleto en segundos porque no deben escribirla desde cero. Pueden probarla por medio de las herramientas de prueba del marco, sin depender de herramientas de prueba externas.

Automatización con *scripts* de Python

Un lenguaje de *scripting* es un lenguaje de programación que automatiza las tareas que suelen llevar a cabo las personas. Los programadores utilizan ampliamente los *scripts* de Python para automatizar muchas tareas diarias, como las siguientes:

- Cambiar el nombre de una gran cantidad de archivos a la vez
- Convertir un archivo en otro tipo de archivo

- Eliminar palabras duplicadas de un archivo de texto
- Llevar a cabo operaciones matemáticas básicas
- Enviar mensajes por *email*
- Descargar contenido
- Efectuar análisis básicos de registros
- Encontrar errores en varios archivos

Realizar tareas de ciencia de datos y *machine learning*

La ciencia de datos consiste en extraer conocimientos valiosos a partir de los datos, mientras que el *machine learning* enseña a las computadoras a aprender automáticamente de los datos y a efectuar predicciones precisas. Los científicos de datos utilizan Python para realizar tareas de ciencia de datos, como las que se indican a continuación:

- Corregir y eliminar datos incorrectos, lo que se conoce como limpieza de datos
- Extraer y seleccionar varias características de los datos
- Reducción de dimensionalidad, que consiste en agregar nombres significativos a los datos
- Buscar diferentes estadísticas a partir de los datos
- Visualizar los datos mediante el uso de tablas y gráficos, como los gráficos de líneas, los de barras, los circulares y los histogramas

Los científicos de datos utilizan las bibliotecas de ML de Python para entrenar los modelos de ML y crear clasificadores que clasifiquen los datos con precisión. Las personas que trabajan en diferentes campos utilizan clasificadores basados en Python para efectuar tareas de clasificación, como la clasificación de imágenes, texto y tráfico de red; el reconocimiento de habla; y el reconocimiento facial. Los científicos de datos también utilizan Python para las tareas de aprendizaje profundo, una técnica avanzada de ML.

Desarrollo de *software*

Los desarrolladores de *software* suelen utilizar Python para realizar distintas tareas de desarrollo y aplicaciones de *software*, como las que se indican a continuación:

- Realizar un seguimiento de los errores en el código del *software*
- Crear el *software* de forma automática
- Administrar los proyectos de *software*
- Desarrollar prototipos de *software*
- Desarrollar aplicaciones de escritorio por medio de las bibliotecas de interfaz gráfica de usuario (GUI)
- Desarrollar juegos simples basados en texto a videojuegos más complejos

Automatización de pruebas de *software*

La prueba de *software* es el proceso de verificar si los resultados reales del *software* coinciden con los resultados esperados, para garantizar que el *software* esté libre de errores.

- Los desarrolladores utilizan marcos de prueba de unidad de Python, como unittest, Robot y PyUnit, para probar las funciones que escriben.
- Los encargados de probar el *software* utilizan Python para escribir casos de prueba para diversos escenarios de prueba. Por ejemplo, lo utilizan para probar la interfaz de usuario de una aplicación web, los diversos componentes de *software* y las nuevas características.

Los desarrolladores pueden utilizar varias herramientas para ejecutar *scripts* de prueba de manera automática. Estas herramientas se conocen como herramientas de integración e implementación continuas (CI/CD). Los encargados de probar el *software* y sus desarrolladores utilizan las herramientas de CI/CD, como Travis CI y Jenkins, para automatizar las pruebas. La herramienta de CI/CD ejecuta automáticamente

los scripts de prueba de Python e informa los resultados de las pruebas cada vez que los desarrolladores presentan nuevos cambios de código.

¿Cuál es la historia de Python?

Guido Van Rossum, un programador de computación de los Países Bajos, creó Python. Python comenzó en 1989 en el Centrum Wiskunde & Informatica (CWI), en principio como un proyecto de afición para mantenerse ocupado durante las vacaciones de Navidad. El nombre del lenguaje se inspiró en el programa de televisión de la BBC “Monty Python’s Flying Circus” debido a que Guido Van Rossum era un gran aficionado del programa.

Historial de lanzamientos de Python

- Guido Van Rossum publicó la primera versión del código Python (versión 0.9.0) en 1991. Dicha versión ya incluía buenas características, como algunos tipos de datos y funciones para la gestión de errores.
- Python 1.0 se lanzó en 1994 con nuevas funciones para procesar fácilmente una lista de datos, como la asignación, el filtrado y la reducción.
- Python 2.0 se lanzó el 16 de octubre de 2000, con nuevas características útiles para los programadores, como la compatibilidad con los caracteres Unicode y una forma más corta de recorrer una lista.
- El 3 de diciembre de 2008, se lanzó Python 3.0. Incluía características como la función de impresión y más soporte para la división de números y la gestión de errores.

¿Cuáles son las características de Python?

Las características siguientes del lenguaje de programación Python lo hacen único:

Un lenguaje interpretado

Python es un lenguaje interpretado, lo que significa que ejecuta directamente el código línea por línea. Si existen errores en el código del programa, su ejecución se detiene. Así, los programadores pueden encontrar errores en el código con rapidez.

Un lenguaje fácil de utilizar

Python utiliza palabras similares a las del inglés. A diferencia de otros lenguajes de programación, Python no utiliza llaves. En su lugar, utiliza sangría.

Un lenguaje tipeado dinámicamente

Los programadores no tienen que anunciar tipos de variables cuando escriben código porque Python los determina en el tiempo de ejecución. Debido a esto, es posible escribir programas de Python con mayor rapidez.

Un lenguaje de alto nivel

Python es más cercano a los idiomas humanos que otros lenguajes de programación. Por lo tanto, los programadores no deben preocuparse sobre sus funcionalidades subyacentes, como la arquitectura y la administración de la memoria.

Un lenguaje orientado a los objetos

Python considera todo como un objeto, pero también admite otros tipos de programación, como la programación estructurada y la funcional.

¿Qué son las bibliotecas de Python?

Una biblioteca es una colección de códigos usados con frecuencia que los desarrolladores pueden incluir en sus programas de Python para evitar tener que escribir el código desde cero. De forma predeterminada, Python incluye la biblioteca estándar, que contiene una gran cantidad de funciones reutilizables. Además, más de 137 000 bibliotecas de Python están disponibles para diversas aplicaciones, incluidos el desarrollo web, la ciencia de datos y el *machine learning* (ML).

¿Cuáles son las bibliotecas de Python más populares?

Matplotlib

Los desarrolladores utilizan Matplotlib para trazar los datos en gráficos de dos y tres dimensiones (2D y 3D) de alta calidad. Por lo general, se utiliza en las aplicaciones científicas. Con Matplotlib, puede visualizar los datos mostrándolos en diferentes gráficos, como los gráficos de barras y los de líneas. También puede trazar varios gráficos de una sola vez, y estos se pueden trasladar a todas las plataformas.

Pandas

Pandas proporciona estructuras de datos optimizadas y flexibles que se pueden utilizar para manipular datos de serie temporal y datos estructurados, como las tablas y las matrices. Por ejemplo, puede utilizar Pandas para leer, escribir, combinar, filtrar y agrupar datos. Muchas personas lo utilizan para las tareas de ciencia de datos, análisis de datos y ML.

NumPy

NumPy es una conocida biblioteca que utilizan los desarrolladores para crear y administrar matrices, manipular formas lógicas y efectuar operaciones de álgebra lineal con facilidad. NumPy admite la integración a muchos lenguajes, como C y C++.

Requests

La biblioteca Requests proporciona funciones útiles que se necesitan para el desarrollo web. Puede usarla para enviar solicitudes HTTP; agregar encabezados, parámetros de URL y datos; y llevar a cabo muchas más tareas cuando se comunica con aplicaciones web.

OpenCV-Python

OpenCV-Python es una biblioteca que los desarrolladores utilizan para procesar imágenes para las aplicaciones de visión artificial. Proporciona muchas funciones para las tareas de procesamiento de imágenes, como la lectura y la escritura simultáneas de imágenes, la creación de un entorno 3D a partir de uno 2D y la captura y el análisis de las imágenes de video.

Keras

Keras es la biblioteca de red neuronal profunda de Python que cuenta con un excelente soporte para el procesamiento de datos, su visualización y mucho más. Keras admite muchas redes neuronales. Posee una estructura modular que ofrece flexibilidad en la escritura de aplicaciones innovadoras.

¿Qué son los marcos de Python?

Un marco de Python es una colección de paquetes y módulos. Un módulo es un conjunto de código relacionado, y un paquete es un conjunto de módulos. Los desarrolladores pueden usar los marcos de Python para crear aplicaciones de Python más rápido debido a que no tienen que preocuparse por los detalles de nivel inferior, como la forma en que se producen las comunicaciones en la aplicación web o el modo en que Python hará que el programa sea más rápido. Python tiene dos tipos de marcos:

- **El marco de pila completa** incluye casi todo lo que se necesita para crear una aplicación grande.
- **El micromarco** es un marco básico que proporciona funcionalidades mínimas para crear aplicaciones de Python simples. También proporciona extensiones si las aplicaciones necesitan funciones más sofisticadas.

¿Cuáles son los marcos de Python más conocidos?

Los desarrolladores pueden utilizar varios marcos de Python para que su desarrollo sea eficiente, incluidos los siguientes:

Django

Django es uno de los marcos web de Python de pila completa más utilizados para el desarrollo de aplicaciones web a gran escala. Proporciona varias características útiles, incluidos un servidor web para el desarrollo y las pruebas, un motor de plantillas para crear el sitio web de *frontend* y diversos mecanismos de seguridad.

Flask

Flask es un micromarco que se utiliza para el desarrollo de aplicaciones web pequeñas. Sus características incluyen un importante soporte de la comunidad, documentación bien escrita, un motor de plantillas, pruebas de unidad y un servidor web integrado. También proporciona extensiones para el soporte de validación, las capas de asignación de bases de datos y la seguridad web.

TurboGears

TurboGears es un marco diseñado para crear aplicaciones web con mayor rapidez y facilidad. Estas son algunas de sus características clave:

- Estructura específica de tabla de base de datos
- Herramientas para la creación y la administración de proyectos
- Motor de plantillas para crear las bases de datos
- Motor de plantillas para crear el frontend
- Mecanismos para manejar la seguridad web

Apache MXNet

Apache MXNet es un marco de aprendizaje profundo rápido, flexible y escalable que los desarrolladores utilizan para crear prototipos de investigación y aplicaciones de aprendizaje profundo. Admite múltiples lenguajes de programación, incluidos Java, C++, R y Perl. Proporciona un completo conjunto de herramientas y bibliotecas para brindar soporte al desarrollo. Por ejemplo, puede encontrar un libro interactivo de machine learning (ML), kits de herramientas de visión artificial y modelos de aprendizaje profundo para el procesamiento de lenguaje natural (NLP), que procesan este lenguaje, como el texto y el habla.

PyTorch

PyTorch es un marco para el machine learning que se ha creado sobre la biblioteca Torch, que es otra biblioteca de machine learning de código abierto. Los desarrolladores lo utilizan para aplicaciones como las

de NLP, robótica y visión artificial, para encontrar información significativa en las imágenes y los videos. También lo utilizan para ejecutar esas aplicaciones en las CPU y las GPU.

¿Qué son los IDE de Python?

Un entorno de desarrollo integrado (IDE) es un software que brinda a los desarrolladores las herramientas que necesitan para escribir, editar, probar y corregir código en un único lugar.

¿Cuáles son los IDE de Python más conocidos?

PyCharm

JetBrains, una empresa checa que desarrolla herramientas de software, creó PyCharm. Cuenta con una edición comunitaria gratuita que es adecuada para pequeñas aplicaciones de Python, así como con una edición profesional de pago que es adecuada para crear aplicaciones de Python a gran escala, con el siguiente conjunto completo de características:

- Compleción automática e inspección del código
- Gestión de errores y correcciones rápidas
- Limpieza del código sin necesidad de cambios de funcionalidad
- Compatibilidad con los marcos de aplicaciones web, como Django y Flask
- Compatibilidad con otros lenguajes de programación, como JavaScript, CoffeeScript, TypeScript, AngularJS y Node
- Herramientas y bibliotecas científicas, como Matplotlib and NumPy
- posibilidad de ejecutar, depurar, probar e implementar aplicaciones en máquinas virtuales remotas
- un depurador para encontrar errores en el código, un creador de perfiles para identificar problemas de rendimiento en el código y un ejecutor de pruebas para llevar a cabo pruebas de unidad
- Soporte para bases de datos

IDLE

El entorno integrado de desarrollo y aprendizaje (IDLE) es el entorno de desarrollo integrado (IDE) de Python instalado de forma predeterminada. Se ha desarrollado solo con Python por medio del kit de herramientas de GUI de Tkinter y ofrece las siguientes características:

- Funcionamiento en muchos sistemas operativos, como Windows, Unix y macOS
- Ventana de *shell* para ejecutar comandos y mostrar el resultado
- un editor de textos de varias ventanas que proporciona resaltado de la sintaxis del código y completación automática del código
- posesión de su propio depurador

Spyder

Spyder es un entorno de desarrollo integrado (IDE) de código abierto que utilizan numerosos científicos y analistas de datos. Proporciona una experiencia integral de desarrollo con características para el análisis avanzado de datos, su visualización y depuración. También incluye las siguientes características:

- Editor completo de código que admite varios lenguajes
- Consola interactiva de IPython
- Depurador básico
- Bibliotecas científicas, como Matplotlib, SciPy y NumPy
- Posibilidad de explorar variables en el código
- Posibilidad de visualizar documentación en tiempo real

Atom

Atom es un editor gratuito desarrollado por GitHub que admite la codificación en varios lenguajes de programación, incluido Python. Mediante el uso de Atom, los desarrolladores pueden trabajar directamente con GitHub, el sitio web en el que puede guardar su código de forma centralizada. Atom ofrece las siguientes características:

- Posibilidad de utilizarlo con muchos sistemas operativos
- Instalación o creación fáciles de nuevos paquetes
- Compleción automática de código más rápida
- Posibilidad de buscar archivos y proyectos
- Fácil personalización de la interfaz

¿Qué son los SDK de Python?

Un kit de desarrollo de software (SDK) es una colección de herramientas de software que los desarrolladores pueden utilizar para crear aplicaciones de software en un lenguaje particular. La mayoría de los SDK son específicos de diferentes plataformas de *hardware* y sistemas operativos. Los SDK de Python incluyen muchas herramientas, como bibliotecas, muestras de código y guías para desarrolladores, que estos encuentran útiles a la hora de escribir aplicaciones.

¿Qué es AWS PyCharm?

[AWS Toolkit for PyCharm](#) es el complemento para el entorno de desarrollo integrado (IDE) de PyCharm que facilita la creación, depuración e implementación de aplicaciones de Python en AWS. Mediante el uso del AWS Toolkit for PyCharm, los desarrolladores pueden comenzar fácilmente con el desarrollo de Python. Proporciona varias características útiles para los desarrolladores, incluidas guías de inicio, depuraciones paso a paso e implementaciones de IDE.

¿Qué es Boto3 en Python?

Boto3 es el AWS SDK para Python. Se puede utilizar para crear, configurar y administrar servicios de AWS, como , y . Boto3 también proporciona dos tipos de : las API de nivel inferior y las API de recursos para desarrolladores.