

Python

Índice

01 ¿Qué es Python?

02 ¿Para qué se usa Python?

03 Escalabilidad y compatibilidad

04 Ejemplos de uso



01 ¿Qué es Python?

¿Qué es Python?

Python es un *lenguaje de programación de alto nivel, interpretado* y de **propósito general** que se destaca por su **simplicidad** y **legibilidad** del código.

- Lenguaje de programación **versátil** y fácil de aprender.
- **Interpretado**, lo que significa que el código se ejecuta línea por línea.
- Ampliamente utilizado en **desarrollo web, científico, análisis de datos y automatización**.
- **Sintaxis clara y legible** que favorece la escritura de código limpio y mantenible.
- Posee una amplia comunidad de desarrolladores que contribuyen con **bibliotecas y herramientas**.

02

¿Para qué se usa
Python?



Python

Versátil y de propósito general que se utiliza en una variedad de aplicaciones en diferentes campos.

01

Ciencia de datos: Herramientas como NumPy, Pandas y Matplotlib son ampliamente utilizadas en análisis y visualización de datos.

02

Desarrollo web: Frameworks como Django y Flask facilitan la creación de aplicaciones web dinámicas.

03

Automatización: es ideal para crear scripts que automatizan tareas repetitivas en sistemas informáticos.

04

IA y machine learning: Bibliotecas como TensorFlow y PyTorch se usan en el desarrollo de modelos de machine learning y deep learning.

05

Desarrollo de software: Python se utiliza en el desarrollo de software de todo tipo, desde aplicaciones de escritorio hasta sistemas embebidos y juegos.

03

Escalabilidad y compatibilidad

Escalabilidad y compatibilidad

Python funciona de manera efectiva en diferentes entornos y es capaz de adaptarse a crecientes demandas de recursos y usuarios.

- ➔ **Compatibilidad multiplataforma:** Python es compatible con diversos sistemas operativos como Windows, macOS y Linux.
- ➔ **Integración con otros lenguajes:** Permite la integración con otros lenguajes como C/C++ y Java, lo que amplía su funcionalidad.
- ➔ **Escalabilidad horizontal:** Python ofrece opciones para escalar aplicaciones horizontalmente, distribuyendo la carga entre múltiples servidores o instancias.
- ➔ **Herramientas de optimización:** Existen herramientas como Cython y Numba que permiten optimizar el rendimiento de código Python para mejorar la escalabilidad.
- ➔ **Frameworks escalables:** Frameworks como Django y Flask proporcionan herramientas y prácticas para desarrollar aplicaciones web escalables y de alto rendimiento.

04 Ejemplos de uso

Ejemplos de uso

01

Leer, escribir, modificar y analizar **archivos CSV** para tareas como análisis de datos, generación de informes, limpieza de datos, etc.

02

Renombrar archivos en masa basándose en ciertos criterios como prefijos, sufijos, fechas, etc.

03

Convertir archivos de un formato a otro, como de CSV a Excel, de JSON a CSV, de PDF a texto plano, etc.

04

Procesar archivos de texto para tareas como búsqueda, reemplazo, extracción de información específica, análisis de logs, etc.

05

Extraer datos de múltiples fuentes, transformarlos según las necesidades y cargarlos en una base de datos u otro destino.

06

Utilizar datos de archivos para generar informes, **tablas y gráficos** que ayuden en la toma de decisiones.

07

Automatizar tareas como redimensionamiento, conversión de formatos, detección de objetos, etc., en archivos de **imágenes y videos**.

08

Implementar scripts que validen y limpien datos en archivos según reglas predefinidas, como la eliminación de duplicados, corrección de errores, etc.