

Introducción a Python

Desde el origen hasta el análisis de datos

Mecánica y Termodinámica - Laboratorio

8 de Febrero de 2024

El lenguaje de programación permite crear programas que controlen el comportamiento **físico** y **lógico** de una máquina, para expresar algoritmos con precisión o como modo de comunicación humana.

¿Qué es Python?

Python es un lenguaje de programación de alto nivel, interpretado, multiplataforma y de código abierto. Fue creado por Guido van Rossum a fines de la década de los 80 y su nombre se inspira en el grupo británico de comedia Monty Python.



Figura: Los Python, animales inclasificables del humor

¿Qué necesito para ejecutar el lenguaje?

Depende del tipo de lenguaje (interpretado o compilado) y del entorno en el que estés trabajando.

- Intérprete o Compilador:
 - 1 ¿Tengo el intérprete adecuado instalado para lenguajes interpretados?
 - 2 ¿Está disponible un compilador para lenguajes compilados?
- Entorno de Ejecución:
 - 1 ¿El sistema operativo es compatible con el lenguaje?
 - 2 ¿Existen restricciones de hardware o arquitectura?
- Configuración del Entorno de Desarrollo:
 - 1 ¿Cuál es el entorno de desarrollo utilizado (IDE, editor de texto)?
 - 2 ¿Está configurado correctamente para el lenguaje?

- Variables de Entorno
 - 1 ¿Se requiere configurar variables de entorno específicas para el lenguaje?
 - 2 ¿El entorno incluye las bibliotecas estándar necesarias para el lenguaje?
- Gestión de Versiones:
 - 1 ¿Se necesita una versión específica del lenguaje?
 - 2 ¿Cómo se manejan las versiones del lenguaje en el sistema?
- Permisos de Ejecución:
 - 1 ¿Se tienen los permisos necesarios para ejecutar programas escritos en ese lenguaje?
- Configuración de Rutas (si es aplicable):
 - 1 ¿Las rutas del ejecutable o del intérprete están configuradas correctamente?
- Dependencias del Sistema Operativo:
 - 1 ¿Existen dependencias específicas del sistema operativo que deben instalarse?

¿Qué se necesita para ejecutar un código entonces?

- 1 **Entorno de Desarrollo (IDE):** Un IDE es una herramienta que proporciona un entorno integrado para escribir, depurar y ejecutar código. Ejemplos incluyen PyCharm para Python, Visual Studio para C#, o Eclipse para Java.
- 2 **Intérprete o Compilador:** Dependiendo del lenguaje de programación, necesitarás un intérprete o un compilador. Los lenguajes interpretados, como Python, ejecutan el código directamente, mientras que los lenguajes compilados, como C++, necesitan ser compilados antes de ejecutarse.
- 3 **Instalación del Lenguaje de Programación:** Asegúrate de tener el lenguaje de programación instalado en tu sistema. Muchos lenguajes tienen instaladores específicos que puedes descargar e instalar.
- 4 **Bibliotecas y Dependencias:** Para programas más complejos, es posible que necesites bibliotecas o dependencias adicionales. Las herramientas de gestión de paquetes (pip para Python, npm para JavaScript, etc.) pueden ser útiles para gestionar estas dependencias.

¿Qué se necesita para ejecutar un código entonces?

- 1 **Conocimientos Básicos del Lenguaje:** Tener conocimientos básicos del lenguaje de programación que estás utilizando es esencial. Esto incluye comprender la sintaxis, las estructuras de control de flujo y las características fundamentales del lenguaje.
- 2 **Sistema Operativo Compatible:** Asegúrate de que el lenguaje de programación que estás utilizando sea compatible con tu sistema operativo.
- 3 **Hardware Suficiente:** Asegúrate de que tu hardware cumple con los requisitos del lenguaje y del IDE que estás utilizando.
- 4 **Depurador (Debugger):** Una herramienta para identificar y corregir errores en el código.

Lenguaje Interpretado vs no Interpretado

La principal diferencia es en cómo se ejecuta el código fuente y se convierte en instrucciones ejecutables para la máquina.

Lenguaje Interpretado	Lenguaje No Interpretado (Compilado)
<p>Ejecución en tiempo real: En un lenguaje interpretado, el código fuente es ejecutado directamente por el intérprete sin la necesidad de una fase de compilación previa. Cada línea de código se traduce y ejecuta a medida que se encuentra, en tiempo real.</p>	<p>Compilación previa: En un lenguaje no interpretado, el código fuente pasa por un proceso de compilación antes de ser ejecutado. Durante la compilación, el código fuente se traduce a un código de máquina específico para la plataforma de destino, generando un archivo ejecutable.</p>
<p>Portabilidad: El código fuente puede ser directamente ejecutado en diferentes plataformas siempre que exista un intérprete para esa plataforma. Esto hace que los programas escritos en lenguajes interpretados sean más portátiles.</p>	<p>Ejecución del código compilado: Después de la compilación, el programa se ejecuta a través del archivo ejecutable resultante, sin necesidad de tener el código fuente disponible. Esto puede aumentar la eficiencia de ejecución en comparación con la interpretación en tiempo real.</p>

Cada enfoque tiene sus ventajas y desventajas en términos de portabilidad, velocidad de ejecución y facilidad de desarrollo

Ejecución de Código en Python

Para ejecutar código en Python, necesitas el intérprete de Python instalado en tu sistema. Puedes escribir código en un archivo de texto y ejecutarlo desde la línea de comandos o utilizar entornos de desarrollo como IDLE, Spyder, o Jupyter Notebook.

Compilador vs. Intérprete

A diferencia de los lenguajes compilados, Python es un lenguaje interpretado. Esto significa que el código se ejecuta directamente por el intérprete sin necesidad de compilación previa. Esta característica facilita el desarrollo y la ejecución rápida de programas.

Entornos de Desarrollo Integrado, **Integrated Development Environment IDE**

Un IDE Es un conjunto de herramientas que facilita el desarrollo de software. Incluye un editor de código, un compilador o intérprete, herramientas de depuración y otras utilidades. Los IDEs son diseñados para aumentar la productividad del desarrollador y mejorar la calidad del código.

Entorno de Desarrollo para Python:

- **PyCharme:**
- **VScode:** Visual Studio Code
- **Spyder:** IDE diseñado para científicos de datos con herramientas integradas.
- **Jupyter Notebook:** Entorno interactivo que permite combinar código, texto y visualizaciones en un solo documento.
- **IDLE:** Entorno de desarrollo integrado de Python

¿Qué es Anaconda?

Tiene como objetivo simplificar simplificar la gestión de entornos y la administración e implementación de paquetes y librerías

¿Qué es gestión de entornos?

- **Propósito:** La gestión de entornos se centra en crear y gestionar ambientes aislados para proyectos específicos, asegurando que las dependencias y bibliotecas necesarias estén controladas y sean coherentes.
- **Herramientas:** Utiliza herramientas como `virtualenv` o `conda` en el caso de Python para crear y administrar entornos virtuales.
- **Beneficios:** Permite la reproducibilidad del entorno de desarrollo, evita conflictos entre versiones de bibliotecas y facilita la colaboración en proyectos.

Algunos lenguajes interpretados

- Python: Es un lenguaje de programación de alto nivel interpretado, ampliamente utilizado en desarrollo web, ciencia de datos y automatización de tareas.
- JavaScript: Lenguaje interpretado utilizado principalmente en desarrollo web para agregar interactividad a las páginas.
- Ruby: Un lenguaje interpretado conocido por su elegancia y facilidad de uso, comúnmente utilizado en desarrollo web y automatización de tareas.
- PHP: Un lenguaje interpretado diseñado específicamente para el desarrollo web del lado del servidor.
- Bash: Lenguaje de scripting interpretado utilizado en sistemas Unix y Linux para automatizar tareas de línea de comandos.

Algunos lenguajes Compilados:

- C: Un lenguaje de programación compilado ampliamente utilizado que proporciona un control de bajo nivel sobre el hardware de la computadora.
- C++: Una extensión del lenguaje C que agrega características de programación orientada a objetos. Se compila a código máquina.
- Java: Aunque es compilado a un bytecode que se ejecuta en la Máquina Virtual Java (JVM), se considera un lenguaje compilado en el sentido de que la compilación y ejecución son pasos separados.
- C (C Sharp): Desarrollado por Microsoft, se compila a código intermedio de la Common Intermediate Language (CIL) que se ejecuta en la Máquina Virtual de Microsoft (CLR).
- Rust: Un lenguaje compilado diseñado para ser seguro, concurrente y práctico, con un rendimiento cercano al lenguaje C.

Otros tipos de lenguaje (Just - in - time)

Julia:

Tipo: Julia es un lenguaje de programación de alto nivel y de alto rendimiento. Enfoque: Diseñado para ser rápido y eficiente, Julia se utiliza especialmente en cómputo técnico y científico.

Interpretación y Compilación: Julia utiliza un enfoque único llamado "Just-In-Time" (JIT), lo que significa que realiza la compilación justo antes de la ejecución. Combina características de lenguajes interpretados y compilados. Mientras escribe código en Julia, puede sentirse como si estuviera escribiendo en un lenguaje interpretado, pero el compilador JIT puede generar código de máquina optimizado para la ejecución eficiente durante el tiempo de ejecución.

Ejemplos Básicos en Python

```
# Ejemplo de código en Python  
print("Hola , -mundo!")
```

```
# Operaciones matemáticas simples  
suma = 3 + 5  
multiplicacion = 2 * 4
```

```
# Importar bibliotecas  
import pandas as pd  
import numpy as np  
  
# Importar datos desde un archivo CSV  
datos = pd.read_csv('archivo.csv')  
  
# Calcular la moda y el promedio  
moda = np.mode(datos['columna'])  
promedio = np.mean(datos['columna'])
```