1. **Ngôn ngữ PYTHON là gì ?**

Ngôn ngữ Python là một ngôn ngữ lập trình bậc cao được tạo ra bởi Guido van Rossum vào năm 1991. Các điểm nổi bật của Python bao gồm:

Cú pháp đơn giản, dễ đọc, gần gũi với ngôn ngữ tự nhiên.

Hỗ trợ thư viện cho nhiều lĩnh vực khác nhau như AI, Machine Learning, Web,…

Chạy được trên nhiều hệ điều hành như Windows, macOS, Linux.

Có cơ chế Garbage Collection giúp tối ưu việc sử dụng bộ nhớ.

Mã nguồn mở và cộng đồng người sử dụng lớn.

Ứng dụng của Python:

Phát triển web

Trí tuệ nhân tạo (AI) & Machine Learning (ML)

Khoa học dữ liệu & Phân tích dữ liệu

Tự động hóa (Automation) & Scripting

Phát triển Game

An ninh mạng & Kiểm thử bảo mật

Internet of Things (IoT) & Robotics

Lập trình ứng dụng Desktop

Xử lý văn bản & Xử lý ngôn ngữ tự nhiên (NLP)

Các điểm hạn chế của Python:

Tốc độ thực thi của Python chậm hơn so với các ngữ biên dịch như Java hay C++.

Python tiêu tốn bộ nhớ nhiều hơn so với C++ hoặc Java do có cơ chế quản lý bộ nhớ tự động (Garbage Collection).

Không phù hợp cho lập trình di động vì hiệu suất không cao và thiếu sự hỗ trợ chính thức từ Android và IOS.

Không phù hợp cho lập trình đa luồng vì Python có Global Interpreter Lock (GIL), khiến các luồng (threads) trong Python không thể thực thi song song thực sự trên CPU đa nhân.

Dễ gặp lỗi runtime do kiểu dữ liệu động vì Python sử dụng dynamic typing giúp cho người viết code không cần khai báo kiểu dữ liệu trước khi dùng khiến quá trình viết code nhanh hơn nhưng dễ gây lỗi trong quá trình truyền tham số.

**2. Cách học ngôn ngữ PYTHON dành cho người không biết gì về lập trình?**

Bước 1: Cài đặt Python và môi trường lập trình.

Bước 2: Học những kiến thức cơ bản của Python như:

* Biến và kiểu dữ liệu.

Biến: Lưu trữ dữ liệu.

ten = "Hoa"

tuoi = 30

Kiểu dữ liệu: Số (int, float), chuỗi (string), danh sách (list), tuple, dictionary, boolean...

diem = 10 # float

danh\_sach = [1, 2, 3] # list

thong\_tin = {"ten": "Hoa", "tuoi": 30} # dictionary

* Các phép toán và toán tử.

Phép cộng, trừ, nhân, chia, và các toán tử logic.

a = 1

b = 2

print(a \* b) # In ra 2

print(a > b) # In ra False

* Cấu trúc điều kiện

diem = 10

if diem >= 8.5:

print("A")

elif diem >= 7.8:

print("B+")

else:

print("Điểm của bạn là B hoặc thấp hơn")

* Vòng lặp

Vòng lặp for:

for i in range(6): # range(6) tạo ra dãy số 0,1,2,3,4,5

print(i)

Vòng lặp while

dem = 0

while dem < 5:

print("Đếm:", dem)

dem += 1

* Hàm

Tổ chức code thành các khối có thể tái sử dụng.

def chao\_mung(ten):

print(f"Xin chào, {ten}!")

chao\_mung("Hoa") # In ra: Xin chào, Hoa!

Bước 3: Làm các bài tập thực hành đơn giản như:

Viết chương trình cộng, trừ, nhân, chia hai số.

Viết chương trình nhập và in ra một mảng các phần tử.

Bước 4: Làm quen với Lập trình Hướng đối tượng trên ngôn ngữ Python.

Bước 5: Tìm hiểu về các thư viện của ngôn ngữ Python và công dụng của chúng.

Bước 6: Luyện tập các bài tập và thực hiện các dự án nhỏ.

Bước 7: Tham gia các cộng đồng, diễn đàn về Python trên các trang Facebook, Reddit để học hỏi thêm kiến thức, kinh nghiệm.

**3. Những điều cơ bản cần nhớ khi học PYTHON?**

Các điều cơ bản cần nhớ khi học Python:

Sử dụng thụt lềđể xác định khối lệnh thay vì dùng dấu ngoặc {} như một số ngôn ngữ khác.

Không cần đặt dấu chấm phẩy (;) ở cuối dòng.

Không cần khai báo kiểu dữ liệu của biến

**4. Ngôn ngữ PYTHON có gì khác với ngôn ngữ lập trình C và JAVA? (liệt kê theo dạng bảng)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Đặc điểm** | **Python** | **C** | **Java** |
| **Loại ngôn ngữ** | Thông dịch, đa mục đích | Biên dịch, thủ tục(procedudural) | Biên dịch sang bytecode chạy trên JVM , hướng đối tượng hoàn toàn |
| **Kiểu dữ liệu** | Dynamic typing (kiểu động) | Static typing (kiểu tĩnh) | Static typing (kiểu tĩnh |
| **Quản lý bộ nhớ** | Tự động (Garbage Collection) | Thủ công (sử dụng malloc, free) | Tự động (Garbage Collection) |
| **Cú pháp** | Sử dụng thụt lề để xác định khối lệnh | Sử dụng dấu {} và ; để xác định khối lệnh | Tương tự C (sử dụng {} và ;) nhưng có nhiều quy tắc hướng đối tượng |
| **Hỗ trợ OOP** | Có, nhưng không bắt buộc (đa paradigma: OOP, thủ tục, hàm) | Không hỗ trợ OOP bản địa (phải sử dụng C++ hoặc các cấu trúc tự xây dựng) | Hỗ trợ OOP toàn diện với các khái niệm như class, inheritance, interface |
| **Hiệu suất** | Thường chậm hơn do là ngôn ngữ thông dịch | Rất nhanh, gần sát mã máy vì được biên dịch trực tiếp | Tốc độ tốt, nhưng có thể chậm hơn C do chạy trên JVM |
| **Đa nền tảng** | Chạy trên nhiều hệ điều hành với cùng mã nguồn | Phụ thuộc vào trình biên dịch và hệ điều hành | "Write once, run anywhere" (chạy trên mọi nền tảng có JVM) |
| **Ứng dụng phổ biến** | Data Science, AI, phát triển web, tự động hóa, scripting | Hệ thống nhúng, lập trình hệ thống, phát triển phần mềm hệ thống | Ứng dụng doanh nghiệp, phát triển ứng dụng Android, web, server-side |
| **Thư viện** | Phong phú | Hạn chế | Phong phú |
| **Tính hướng đối tượng** | Có | Không | Có |