**Ngôn Ngữ Lập Trình Python**

1: Lịch sử ra đời và phát triển :

Python là một ngôn ngữ lập trình được phát triển bởi Guido van Rossum vào cuối thập niên 1980, với phiên bản đầu tiên ra mắt vào năm 1991. Ban đầu, Guido phát triển Python như một dự án phụ vào kỳ nghỉ Giáng sinh của mình khi làm việc tại Centrum Wiskunde & Informatica (CWI) ở Hà Lan. Ngôn ngữ này lấy cảm hứng từ nhiều ngôn ngữ khác, đặc biệt là ABC – một ngôn ngữ lập trình giáo dục nhằm giúp người mới học lập trình dễ dàng hơn – nhưng cũng chịu ảnh hưởng của C, C++, và Modula-3. Guido đã chọn cái tên "Python" vì yêu thích chương trình hài kịch của nhóm Monty Python, điều này thể hiện phần nào tính chất nhẹ nhàng, hài hước trong triết lý thiết kế của ngôn ngữ này.



Hình 1. 1: Chân dung Guido van Rossum

Trong suốt thập niên 1990, Python đã dần phát triển thành một ngôn ngữ lập trình có cộng đồng người dùng tích cực. Một trong những yếu tố quan trọng giúp Python phổ biến hơn là sự tập trung vào tính dễ đọc của mã nguồn, cú pháp rõ ràng và nhất quán. Ngay từ đầu, Guido van Rossum đã đặt ra triết lý “bắt buộc phải dễ đọc”, và điều này đã giúp Python khác biệt so với nhiều ngôn ngữ khác thời bấy giờ. Python phiên bản 2.0 ra mắt vào năm 2000, mang lại nhiều cải tiến đáng kể như thu gom rác tự động (garbage collection) và hỗ trợ Unicode, giúp Python dễ dàng mở rộng ra các thị trường quốc tế. Tuy nhiên, Python 2 vẫn tồn tại một số hạn chế về hiệu suất và khả năng mở rộng, khiến các nhà phát triển bắt đầu nghĩ đến một bước phát triển vượt bậc.

A diagram of a diagram

Description automatically generated with medium confidence

Hình 1. 2: Sơ đồ tóm tắt quá trình phát triển của python tới phiên bản 3.7

Năm 2008, Python 3.0 được phát hành, đánh dấu một thay đổi lớn với nhiều cải tiến trong cú pháp và cấu trúc dữ liệu. Python 3 được thiết kế để khắc phục những yếu điểm của Python 2 và mang đến một nền tảng mạnh mẽ hơn cho các ứng dụng phát triển hiện đại. Tuy nhiên, sự khác biệt lớn giữa Python 2 và Python 3 đã khiến cho quá trình chuyển đổi trở nên khó khăn, đặc biệt đối với các dự án và thư viện lớn đã được viết bằng Python 2. Kể từ đó, Python tiếp tục phát triển với các phiên bản 3.x, không ngừng cập nhật và bổ sung các tính năng mới. Phiên bản Python 2 cuối cùng (Python 2.7) được duy trì đến ngày 1 tháng 1 năm 2020, khi Python 2 chính thức bị ngừng hỗ trợ, đánh dấu một bước chuyển đổi hoàn toàn sang Python 3.

Trong thập niên 2010, Python đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất thế giới nhờ vào sự phát triển của khoa học dữ liệu, học máy, và trí tuệ nhân tạo. Các thư viện như NumPy, Pandas, TensorFlow và PyTorch đã giúp Python trở thành một công cụ không thể thiếu trong các lĩnh vực này. Được biết đến với sự linh hoạt và tính đơn giản, Python không chỉ phổ biến trong cộng đồng khoa học dữ liệu mà còn trở thành ngôn ngữ lập trình chính trong nhiều lĩnh vực khác như phát triển web, phát triển ứng dụng di động, và tự động hóa.

Tính đến nay, Python vẫn tiếp tục phát triển mạnh mẽ với cộng đồng đông đảo và sự đóng góp của hàng nghìn nhà phát triển trên toàn thế giới. Với triết lý "Simple is better than complex" (Đơn giản hơn phức tạp) từ Zen of Python, Python luôn duy trì được tính nhất quán trong cú pháp và dễ học, dễ hiểu. Python Foundation, tổ chức phi lợi nhuận quản lý và phát triển Python, vẫn tiếp tục thúc đẩy ngôn ngữ này đi xa hơn, tạo điều kiện cho sự ra đời của các tính năng và cải tiến mới trong tương lai. Điều này giúp Python không chỉ duy trì được vị trí hàng đầu trong thế giới lập trình mà còn tiếp tục phát triển, thích nghi với những thay đổi của ngành công nghệ.

**2 : Cấu trúc dòng lệnh**

Cấu trúc của Python được thiết kế với mục tiêu đơn giản và dễ đọc, giúp người học dễ dàng tiếp cận và các nhà phát triển có thể nhanh chóng xây dựng và bảo trì các ứng dụng. Python là một ngôn ngữ lập trình dạng thông dịch, có nghĩa là mã nguồn được thực thi trực tiếp mà không cần biên dịch trước thành mã máy, điều này giúp rút ngắn thời gian phát triển và thử nghiệm. Một trong những đặc điểm nổi bật của Python là cấu trúc khối được xác định bởi thụt lề (indentation), chứ không phải dấu ngoặc như trong các ngôn ngữ khác. Quy tắc này không chỉ giúp mã nguồn dễ đọc mà còn đảm bảo tính nhất quán và tránh lỗi cú pháp do lạm dụng dấu ngoặc.

Python hỗ trợ nhiều kiểu dữ liệu tích hợp sẵn, từ kiểu cơ bản như số nguyên (integer), số thực (float), chuỗi (string), đến các kiểu cấu trúc dữ liệu phức tạp hơn như danh sách (list), từ điển (dictionary), tập hợp (set), và tuple. Các kiểu dữ liệu này được quản lý động, tức là Python tự động gán kiểu dữ liệu mà không cần khai báo, giúp tối ưu hóa việc quản lý và sử dụng bộ nhớ. Python cũng có cơ chế thu gom rác (garbage collection) tự động, giúp quản lý bộ nhớ và giảm thiểu rò rỉ tài nguyên, làm cho ngôn ngữ này trở nên hiệu quả hơn trong các dự án lớn.

Python còn hỗ trợ lập trình hướng đối tượng (OOP), lập trình thủ tục và lập trình hàm, cho phép người dùng lựa chọn phong cách phù hợp nhất cho dự án của mình. Lập trình hướng đối tượng trong Python hỗ trợ các khái niệm cơ bản như lớp (class), đối tượng (object), kế thừa (inheritance), đóng gói (encapsulation), và đa hình (polymorphism), giúp tổ chức và quản lý mã nguồn theo mô hình rõ ràng và dễ bảo trì. Đối với các yêu cầu xử lý nhanh và tối ưu hiệu suất, Python cung cấp các mô-đun và thư viện tích hợp giúp viết các hàm xử lý phức tạp và thực hiện các phép tính toán một cách nhanh chóng.

A diagram of a program

Description automatically generated with medium confidence

Hình 1. 3: Cấu trúc cơ bản của một chương trình python đơn giản.

Một điểm đáng chú ý khác trong cấu trúc của Python là khả năng mở rộng thông qua các thư viện và mô-đun bên ngoài. Python có một kho thư viện phong phú, được phát triển bởi cộng đồng lớn mạnh, từ các thư viện chuẩn cho đến các thư viện nâng cao cho các lĩnh vực như khoa học dữ liệu, học máy, phát triển web, và phân tích dữ liệu lớn. Người dùng có thể cài đặt thêm các thư viện từ Python Package Index (PyPI) chỉ với một dòng lệnh, giúp tiết kiệm thời gian và mở rộng khả năng của ngôn ngữ. Điều này cũng đồng nghĩa với việc Python có thể tích hợp dễ dàng với các công nghệ khác, tạo điều kiện thuận lợi cho các dự án đa nền tảng và đa ngôn ngữ.

Python sử dụng một trình thông dịch đơn (interpreter) trong quá trình thực thi mã, cho phép phát hiện lỗi dễ dàng trong từng dòng mã và tăng khả năng phản hồi nhanh cho nhà phát triển khi kiểm thử. Trình thông dịch này cũng hỗ trợ nhiều hệ điều hành khác nhau như Windows, macOS, và Linux, giúp mã Python có tính di động cao và dễ dàng chạy trên nhiều nền tảng mà không cần điều chỉnh nhiều. Python cũng hỗ trợ các công cụ để đóng gói mã nguồn thành các tệp thực thi, giúp việc triển khai và phân phối ứng dụng trở nên thuận tiện hơn.

Với cấu trúc đơn giản và tính linh hoạt cao, Python không chỉ dễ học mà còn là một công cụ mạnh mẽ cho các nhà phát triển chuyên nghiệp. Sự kết hợp giữa tính năng đa dạng, cú pháp rõ ràng, và khả năng mở rộng mạnh mẽ đã giúp Python trở thành một trong những ngôn ngữ lập trình được ưa chuộng nhất trong nhiều lĩnh vực công nghệ.

**3 : Cài đặt Python**

Python hỗ trợ nhiều nền tảng, nên các bước này sẽ giúp bạn cài đặt Python nhanh chóng và sẵn sàng sử dụng. Để cài đặt Python, có thể truy cập trang web chính thức của Python tại <https://www.python.org/downloads/>. Nơi đây chứa lịch sử các phiên bản cũng như các tệp tin cài đặt. Người dùng có thể tùy chọn các phiên bản phù hợp với cầu hình và hệ điền hành của mình.