

Python (phát âm tiếng Anh: /'paɪθɑ:n/) là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển trí tuệ nhân tạo. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.[25] Vào tháng 7 năm 2018, van Rossum đã từ chức lãnh đạo trong cộng đồng ngôn ngữ Python sau 30 năm làm việc.[26][27]

Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.[28]

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, Python dần mở rộng sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

Python luôn được xếp hạng vào những ngôn ngữ lập trình phổ biến nhất.[29][30][31][32]

Guido van Rossum, nhà sáng lập ngôn ngữ Python.

Lịch sử

Bài chi tiết: Lịch sử Python

Python đã được Guido van Rossum thai nghén vào những năm 1980[33] tại Trung tâm Toán học – Tin học (Centrum Wiskunde & Informatica, CWI) ở Hà Lan như là một ngôn ngữ kế tục ngôn ngữ ABC – một ngôn ngữ được lấy cảm hứng từ SETL,[34] có khả năng xử lý ngoại lệ và giao tiếp với hệ điều hành Amoeba.[35] Nó bắt đầu được triển khai vào tháng 12 năm 1989.[36] Van Rossum đã tự mình gánh vác trách nhiệm cho dự án, với vai trò là nhà phát triển chính, cho đến ngày 12 tháng 7 năm 2018, khi ông thông báo rằng ông sẽ rời bỏ trách nhiệm của ông và cả danh hiệu "Nhà độc tài nhân từ cho cuộc sống" của Python, một danh hiệu mà cộng đồng Python đã trao tặng cho ông vì sự tận tụy lâu dài của ông với vai trò là người ra quyết định chính cho dự án.[37] Vào tháng 1 năm 2019, các nhà phát triển phần lõi Python đã bầu ra một "Hội đồng chèo lái" gồm năm thành viên để dẫn dắt dự án.[38][39]

Python 2.0 được ra mắt vào ngày 16 tháng 10 năm 2000, với nhiều tính năng mới mẻ, bao gồm một bộ dọn rác phát hiện theo chu kỳ và khả năng hỗ trợ Unicode.[40]

Python 3.0 được ra mắt vào ngày mùng 3 tháng 12 năm 2008. Đây là một phiên bản lớn của Python không tương thích ngược hoàn toàn.[41] Nhiều tính năng lớn của nó đã được chuyển mã ngược (backport) về loạt phiên bản Python 2.6.x và 2.7.x.[42] Các bản phát hành của Python 3 có đi kèm với công cụ 2to3, có tác dụng tự động hoá việc dịch mã Python 2 sang Python 3.[43]

Python 3.9.2 và 3.8.8 được xúc tiến[44] vì tất cả các phiên bản trước của Python (bao gồm cả 2.7)[45] gặp một số vấn đề bảo mật, có thể dẫn đến thực thi mã từ xa[46] và "đầu độc" bộ nhớ đệm.[47]

Triết lý thiết kế và tính năng

Python là một ngôn ngữ lập trình đa mẫu hình, lập trình hướng đối tượng và lập trình cấu trúc được hỗ trợ hoàn toàn, và nhiều tính năng của nó cũng hỗ trợ lập trình hàm và lập trình hướng khía cạnh (bao gồm siêu lập trình[48] và siêu đối tượng (phương thức thần kỳ)).[49] Các mẫu hình khác cũng được hỗ trợ thông qua các phần mở rộng, bao gồm thiết kế theo hợp đồng[50][51] và lập trình logic.[52]

Python sử dụng kiểu động và một dạng kết hợp giữa đếm tham chiếu và bộ dọn rác kiểm tra theo chu kỳ để quản lý bộ nhớ.[53] Nó cũng có tính năng phân giải tên động (liên kết muộn), cho phép liên kết các tên biến và phương thức trong quá trình thực thi chương trình.

Thiết kế của Python cung cấp một số tính năng cho lập trình hàm giống như trong ngôn ngữ Lisp. Python có các hàm filter, map và reduce; thông hiểu danh sách (list comprehension), từ điển (dictionary), tập hợp (set), và các biểu thức bộ sinh (generator).[54] Thư viện chuẩn cũng có hai mô đun (itertools và functools) triển khai các công cụ hàm được vay mượn từ Haskell và Standard ML.[55]

Triết lý căn bản của ngôn ngữ Python được trình bày trong tài liệu The Zen of Python (PEP 20), có dạng thơ Haiku, tóm gọn như sau:

Đẹp dễ tốt hơn xấu xí

Minh bạch tốt hơn ngầm định

Đơn giản tốt hơn phức tạp

Phức tạp tốt hơn rắc rối

Tính dễ đọc rất quan trọng.

Thay vì tích hợp hết tất cả các tính năng vào phần cốt lõi, Python được thiết kế để dễ dàng mở rộng (bằng các mô đun). Tính mô đun nhỏ gọn này đã làm cho Python trở nên phổ biến như là một cách thêm các giao diện lập trình được vào các ứng dụng hiện có. Tầm nhìn của Van Rossum về một ngôn ngữ có phần lõi nhỏ với một thư viện chuẩn rộng lớn và một trình thông dịch dễ dàng mở rộng bắt nguồn từ việc ông nản lòng trước ABC, một ngôn ngữ lập trình tán thành hướng tiếp cận ngược lại.[33] Python thường được mô tả là một ngôn ngữ "tặng kèm pin" nhờ vào thư viện chuẩn bao quát của nó.[56]

Python nỗ lực hướng đến một cú pháp đơn giản hơn, gọn gàng hơn trong khi vẫn cho các nhà phát triển lựa chọn phương pháp viết mã của họ. Đối lập với khẩu hiệu "có nhiều hơn một cách để làm việc này," triết lý thiết kế của Python lại nằm trong châm ngôn "chỉ nên có một— và tốt nhất là chỉ một—cách rõ ràng để làm việc này".[57] Alex Martelli, một Viện sĩ (Fellow) tại Tổ chức Phần mềm Python (Python Software Foundation) và là một tác giả viết sách Python, viết rằng "Mô tả một thứ gì đó là "tài tình" không được coi là một lời khen ngợi trong văn hoá Python."[58]

Các nhà phát triển Python nỗ lực tránh xa việc tối ưu hoá quá sớm và không chấp nhận các bản vá không cải thiện đáng kể tốc độ mà lại làm mất đi tính rõ ràng lên những phần không thiết yếu của bản thực hiện tham khảo CPython.[58] Khi cần đến tốc độ, một lập trình viên Python có thể di chuyển các hàm bị giới hạn về thời gian sang các mô đun mở rộng được viết bằng những ngôn ngữ như C, hoặc sử dụng PyPy, một trình biên dịch tức thời. Cython cũng có thể được dùng để biên dịch một tập lệnh Python sang C và tạo ra các lệnh gọi API ở cấp độ của C trực tiếp vào trình thông dịch Python.

Mục tiêu mà các nhà phát triển Python hướng đến là đem đến niềm vui khi sử dụng ngôn ngữ này. Điều này được thể hiện qua bản thân cái tên – một sự tôn vinh dành cho nhóm hài người Anh Monty Python[59] – và trong một số cách tiếp cận thi thoảng vui tươi trong hướng dẫn và các tài liệu tham khảo, chẳng hạn như một ví dụ có đề cập đến trứng và spam (gợi nhắc đến một tiểu phẩm trong Monty Python) thay cho foo và bar tiêu chuẩn.[60][61]

Một từ ngữ mới phổ biến trong cộng đồng Python là pythonic (đậm chất Python), một từ có thể có nhiều ý nghĩa liên quan đến phong cách lập trình. Nói rằng một phần mã nào đó là pythonic tức là phần mã đó sử dụng tốt các thành ngữ Python, trông tự nhiên hoặc trôi chảy về ngôn ngữ, phù hợp với triết lý tối giản của Python và nhấn mạnh vào tính dễ đọc. Ngược lại, những phần mã khó hiểu hoặc trông như một bản dịch thô từ một ngôn ngữ lập trình khác được gọi là unpythonic (không đậm chất Python).[62][63]

Những người sử dụng và say mê Python, nhất là những người được cho là am hiểu hay có nhiều kinh nghiệm, thường được gọi là các Pythonista