Python (bahasa pemrograman)

Daftar isi

Tampilan
Teks

- Kecil
  Standar
  Besar

Lebar

- Standar
  Lebar

Python

Paradigma	Multi-paradigma: fungsional, imperatif, berorientasi objek, terstruktur, reflektif
Dirancang oleh	Guido van Rossum
Pengembang	Python Software Foundation
Rilis perdana	1990[1]
Rilis stabil	3.12.7[2] / 1 Oktober 2024
Tipe sistem	Duck, dynamic, gradual (sejak 3.5)[3]
Sistem operasi	Linux, macOS, Windows Vista (dan yang terbaru) dan banyak lagi
Lisensi	Python Software Foundation License
Ekstensi nama berkas	.py, .pyi, .pyc, .pyd, .pyo (sebelum 3.5),[4] .pyw, .pyz (sejak 3.5)[5]
Situs web	www.python.org
Repositori	www.python.org

Implementasi utama

Dialek

Cython, RPython, Starlark[6]

Terpengaruh oleh

ABC,[7] Ada,[8] ALGOL 68,[9] APL,[10] C,[11] C++,[12] CLU,[13] Dylan,[14] Haskell,[15] Icon,[16] Java,[17] Lisp,[18] Modula-3,[12] Perl, Standard ML[10]

Mempengaruhi

Apache Groovy, Boo, Cobra, CoffeeScript,[19] D, F#, Genie,[20] Go, JavaScript,[21][22] Julia,[23] Nim, Ring,[24] Ruby,[25] Swift[26]

Sunting kotak info • L • B

Bantuan penggunaan templat ini

Python adalah bahasa pemrograman tujuan umum yang ditafsirkan, tingkat tinggi. Dibuat oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, filosofi desain Python menekankan keterbacaan kode dengan penggunaan spasi putih yang signifikan. Konstruksi bahasanya dan pendekatan berorientasi objek bertujuan untuk membantu pemrogram menulis kode yang jelas dan logis untuk proyek skala kecil dan besar.[27]

Python diketik secara dinamis dan pengumpulan sampah. Ini mendukung beberapa paradigma pemrograman, termasuk pemrograman terstruktur (terutama, prosedural), berorientasi objek, dan fungsional. Python sering dideskripsikan sebagai bahasa "termasuk baterai" karena perpustakaan standarnya yang komprehensif.[28]

Python dibuat pada akhir 1980-an sebagai penerus bahasa ABC. Python 2.0, dirilis pada tahun 2000, memperkenalkan fitur-fitur seperti pemahaman daftar dan sistem pengumpulan sampah dengan penghitungan referensi.

Python 3.0, dirilis pada tahun 2008, adalah revisi utama dari bahasa yang tidak sepenuhnya kompatibel dengan versi sebelumnya, dan banyak kode Python 2 yang tidak berjalan tanpa modifikasi pada Python 3.

Penerjemah Python tersedia untuk banyak sistem operasi. Komunitas pemrogram global mengembangkan dan memelihara CPython, implementasi referensi[29] yang bebas dan sumber terbuka.

Sebuah organisasi nirlaba, Python Software Foundation, mengelola dan mengarahkan sumber daya untuk pengembangan Python dan CPython.

Python secara konsisten menempati peringkat sebagai salah satu bahasa pemrograman paling populer.[30][31][32][33]

Sejarah

Perancang Python, Guido van Rossum di OSCON 2006

Python dibuat pada akhir 1980-an[34] oleh Guido van Rossum di Centrum Wiskunde & Informatica (CWI) di Belanda sebagai penerus bahasa ABC (sendiri terinspirasi oleh SETL),[35] mampu menangani pengecualian dan berinteraksi dengan sistem operasi Amoeba.[36] Implementasinya dimulai pada bulan Desember 1989. Van Rossum memikul tanggung jawab penuh atas proyek tersebut, sebagai pengembang utama, hingga 12 Juli 2018, ketika ia mengumumkan "liburan permanen" dari tanggung jawabnya sebagai *Benevolent Dictator For Life* Python, sebuah gelar yang diberikan komunitas Python kepadanya untuk mencerminkan komitmen jangka panjangnya sebagai pengambil keputusan utama proyek.

Python 2.0 dirilis pada 16 Oktober 2000 dengan banyak fitur utama baru, termasuk pengumpul sampah pendeteksian siklus dan dukungan untuk Unicode.

Tanggal akhir masa pakai Python 2.7 yang awalnya ditetapkan pada tahun 2015 kemudian ditunda hingga tahun 2020 karena sejumlah besar kode yang tidak dapat dengan mudah dilanjutkan ke Python 3.

Python 3.6 (dan setiap perilisan lama), tidak lagi didukung per 2021.

Pada 2022, Python 3.10.4 dan 3.9.12 dipercepat dan begitu juga perilisan yang lebih lama termasuk 3.8.13, dan 3.7.13 dikarenakan banyak masalah keamanan pada 2022. Python 3.9.13 adalah versi 3.9 terbaru, dan mulai sekarang 3.9 (dan yang terlama; 3.8 dan 3.7) hanya akan mendapatkan pembaruan keamanan.[37]

Desain fitur dan filosofi

Python adalah bahasa pemrograman multi-paradigma. Pemrograman berorientasi objek dan pemrograman terstruktur juga didukung penuh, dan banyak fiturnya mendukung pemrograman fungsional dan pemrograman berorientasi aspek (termasuk dengan metaprogramming dan metaobjects

(metode ajaib)). Banyak paradigma lain yang didukung melalui ekstensi, termasuk desain berdasarkan kontrak dan pemrograman logika.

Desain Python menawarkan beberapa dukungan untuk pemrograman fungsional dalam tradisi Lisp. Memiliki fungsi `filter`, `map`, dan `reduce`;daftar pemahaman, kamus, set, dan ekspresi generator.[38] Pustaka standar memiliki dua modul (itertools dan functools) yang mengimplementasikan alat fungsional yang dipinjam dari Haskell dan Standard ML.[39]

Filosofi inti bahasa diringkas dalam dokumen *The Zen of Python* (*PEP* 20), yang mencakup kata-kata mutiara seperti:[40]

- Cantik itu lebih baik dari pada jelek.
- Eksplisit lebih baik daripada implisit.
- Sederhana lebih baik daripada kompleks.
- Kompleks lebih baik daripada rumit.
- Keterbacaan itu penting.

Daripada memiliki semua fungsionalitas yang dibangun ke dalam intinya, Python dirancang untuk menjadi sangat dapat dikembangkan. Modularitas yang ringkas ini membuatnya sangat populer sebagai sarana untuk menambahkan antarmuka yang dapat diprogram ke aplikasi yang ada. Visi Van Rossum tentang bahasa inti kecil dengan perpustakaan standar yang besar dan penerjemah yang mudah diperluas berasal dari rasa frustrasinya dengan ABC, yang mendukung pendekatan yang berlawanan.[34]

Pengembang Python menargetkan ini dapat menyenangkan untuk digunakan. Ini tercermin dalam namanya—sebuah penghargaan untuk grup komedi Inggris Monty Python—dan terkadang dengan pendekatan yang menyenangkan untuk tutorial dan materi referensi, seperti contoh yang merujuk pada spam dan telur (sebuah referensi ke sketsa Monty Python) alih-alih foo dan bar standar.

Pengguna dan peminat Python, terutama mereka yang dianggap berpengetahuan atau berpengalaman, sering disebut sebagai *Pythonistas*.

Sintaks dan semantik

Python dimaksudkan sebagai bahasa yang mudah dibaca. Pemformatannya tidak berantakan secara visual, dan sering kali menggunakan kata kunci bahasa Inggris di mana bahasa lain menggunakan tanda baca. Tidak seperti banyak bahasa lain, ia tidak menggunakan tanda kurung awal untuk membatasi blok, dan pernyataan titik koma setelahnya bersifat opsional. Ini memiliki lebih sedikit pengecualian sintaksis dan kasus khusus daripada C atau Pascal.

Indentasi

Python menggunakan indentasi spasi, daripada tanda kurung kurawal atau kata kunci, untuk membatasi blok. Peningkatan indentasi muncul setelah pernyataan tertentu; penurunan indentasi menandakan akhir dari blok saat ini. Dengan demikian, struktur visual program secara akurat mewakili struktur semantik program. Fitur ini terkadang disebut aturan off-side, yang dimiliki beberapa bahasa lain, tetapi di sebagian besar bahasa indentasi tidak memiliki arti semantik.

Pernyataan dan aliran kontrol

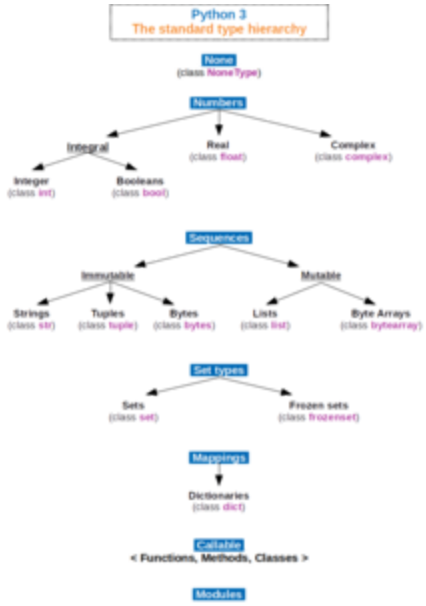
Penetapan nilai yang sama secara berurutan ke beberapa nama, misalnya, `x=2; y=2; z=2` menghasilkan pengalokasian penyimpanan ke (paling banyak) tiga nama dan satu objek numerik, yang ketiganya terikat. Karena nama adalah pemegang referensi umum, tidak masuk akal untuk mengasosiasikan tipe data tetap dengannya. Namun pada waktu tertentu sebuah nama akan terikat ke *suatu* objek, yang **akan** memiliki tipe; jadi ada pengetikan dinamis.

- Pernyataan `if`, yang secara kondisional mengeksekusi blok kode, bersama dengan `else` dan `elif` (sebuah kontraksi dari else-if).
- Pernyataan `for` yang melakukan iterasi pada objek yang dapat diulang, menangkap setiap elemen ke variabel lokal untuk digunakan oleh blok terlampir.
- Pernyataan `while` yang mengeksekusi sebuah blok kode selama kondisinya benar.
- Pernyataan `try` yang memungkinkan pengecualian yang dimunculkan dalam blok kode terlampir untuk ditangkap dan ditangani oleh `except` klausul; itu juga memastikan bahwa kode-pembersihan dalam file blok `finally` akan selalu berjalan terlepas dari bagaimana blok keluar.
- Pernyataan `raise` digunakan untuk memunculkan pengecualian tertentu atau memunculkan kembali pengecualian yang tertangkap.
- Pernyataan `class` yang mengeksekusi blok kode dan menempelkan namespace lokalnya ke class, untuk digunakan dalam pemrograman berorientasi objek.
- Pernyataan `def` yang mendefinisikan fungsi atau metode.
- Pernyataan `with` dari Python 2.5 dirilis pada September 2006, yang membungkus blok kode dalam manajer konteks (misalnya, memperoleh kunci sebelum blok kode dijalankan dan melepaskan kunci setelahnya, atau membuka file dan kemudian menutupnya), memungkinkan perilaku seperti Resource Acquisition Is Initialization (RAII) dan menggantikan idiom percobaan / akhirnya yang umum.[41]
- Pernyataan `break` keluar dari loop.
- Pernyataan `continue` melewati iterasi ini dan melanjutkan dengan item berikutnya.
- Pernyataan `pass` yang berfungsi sebagai NOP. Ini secara sintaksis diperlukan untuk membuat blok kode kosong.
- Pernyataan `assert` digunakan selama debugging untuk memeriksa kondisi yang seharusnya diterapkan.
- Pernyataan `yield` yang mengembalikan nilai dari fungsi generator. Dari Python 2.5, `yield` juga seorang operator. Formulir ini digunakan untuk mengimplementasikan coroutine.
- Pernyataan `import` , yang digunakan untuk mengimpor modul yang fungsi atau variabelnya dapat digunakan dalam program saat ini. Ada tiga cara menggunakan import: `import <nama modul> [sebagai <alias>]` atau `from <nama modul> import *` atau `from <nama modul> import <definisi 1> [sebagai <alias 1>], <definisi 2> [sebagai <alias 2>], ....`
- Pernyataan `print` diubah menjadi fungsi `print()` dengan Python 3.

Metode

Metode pada objek adalah fungsi yang dilampirkan ke kelas objek; sintaks `instance.method(argument)` adalah, untuk metode dan fungsi normal, gula sintaksis untuk `Class.method(instance, argument)`. Metode Python memiliki explisit `self` parameter untuk mengakses data instance, berbeda dengan yang tersirat `self` (atau `this`) dalam beberapa bahasa pemrograman berorientasi objek lainnya (mis., C++, Java, Objective-C, atau Ruby). Python juga menyediakan metode, sering dipanggil *dunder methods* (karena nama mereka dimulai dan diakhiri dengan garis bawah ganda), untuk mengizinkan kelas yang ditentukan pengguna untuk mengubah cara mereka ditangani oleh operasi asli termasuk panjang, perbandingan, dalam operasi aritmatika dan konversi penulisan.

Penulisan



Hierarki tipe standar di Python 3

Python menggunakan duck typing dan memiliki objek yang diketik tetapi nama variabel yang tidak diketik. Batasan jenis tidak diperiksa pada waktu kompilasi; sebaliknya, operasi pada suatu objek mungkin gagal, menandakan bahwa objek yang diberikan bukan tipe yang sesuai. Meskipun diketik secara dinamis, Python diketik dengan kuat, melarang operasi yang tidak terdefinisi dengan baik (misalnya, menambahkan angka ke string) daripada secara diam-diam mencoba memahaminya.

Python memungkinkan pemrogram untuk menentukan tipe mereka sendiri menggunakan kelas, yang paling sering digunakan untuk pemrograman berorientasi objek. Contoh instance baru dibangun dengan memanggil kelas (misalnya, `SpamClass()` atau`EggsClass()`), dan kelas-kelasnya adalah instance dari metaclass `type` (itu sendiri merupakan contoh dari dirinya sendiri), memungkinkan metaprogramming dan refleksi.

Operasi arimatika

Python memiliki simbol biasa untuk operator aritmatika (`+`, `-`, `*`, `/`), operator divisi lantai `//` dan operasi modulus `%` (dimana sisanya bisa negatif, misalnya, `4 % -3 == -2`). Ini juga memiliki `**` untuk eksponensial, misalnya, `5**3 == 125` dan `9**0.5 == 3.0`, dan sebuah matriks operator perkalian `@` .[42] Operator ini bekerja seperti dalam matematika tradisional; dengan aturan operasi yang sama, infiks operator (`+` dan `-` juga bisa unary untuk masing-masing mewakili bilangan positif dan negatif).

Contoh pemrograman Python

Program Halo Dunia:

```
print('Hello, world!')
```

Program untuk menghitung faktorial dari bilangan bulat positif:

```
n = int(input('Ketik sebuah angka, dan faktorialnya akan diprinted: '))

if n < 0:
    raise ValueError('Anda harus memasukkan bilangan bulat bukan negatif')
```

```
faktorial  = 1

for i in range(2, n + 1):
    faktorial *= i

print(faktorial)
```

Contoh kelas dalam bahasa pemrograman Python:

```
class Orang:
    def __init__(self, nama):
        self.nama = nama

    def tampilkan_nama(self):
        return self.nama

orang = Orang("Wikipedia") # Initialisasi object

print(orang.tampilkan_nama()) # Output: Wikipedia
```

Pustaka

Pustaka standar Python yang besar, biasanya disebut sebagai salah satu kekuatan terbesarnya,[43] menyediakan alat yang cocok untuk banyak tugas. Untuk aplikasi yang terhubung ke Internet, banyak format dan protokol standar seperti MIME dan HTTP didukung. Ini mencakup modul untuk membuat antarmuka pengguna grafis, menghubungkan ke database relasional, menghasilkan nomor pseudorandom, aritmatika dengan desimal presisi sewenang-wenang,[44] memanipulasi ekspresi reguler, dan pengujian unit.

Beberapa bagian dari pustaka standar dicakup oleh spesifikasi (misalnya, implementasi Web Server Gateway Interface (WSGI) `wsgiref` mengikuti PEP 333[45]), tetapi kebanyakan modul tidak. Mereka ditentukan oleh kode, dokumentasi internal, dan test suites mereka. Namun, karena sebagian besar pustaka standar adalah kode Python lintas platform, hanya beberapa modul yang perlu diubah atau ditulis ulang untuk implementasi varian.

Mulai Juni 2022, Python Package Index (PyPI), repositori resmi untuk perangkat lunak Python pihak ketiga, berisi lebih dari 380,000[46] paket dengan berbagai fungsi, termasuk:

- Otomatisasi
- Analisis data
- Database
- Dokumentasi
- Antarmuka pengguna grafis
- Pengolahan citra
- Machine learning
- Aplikasi Seluler
- Multimedia
- Jaringan komputer
- Komputasi ilmiah
- Sistem administrasi
- Kerangka uji
- Pemrosesan teks
- Kerangka web
- Web scraping[47]

Lingkungan pengembangan

Sebagian besar implementasi Python (termasuk CPython) menyertakan read–eval–print loop (REPL), yang memungkinkan mereka berfungsi sebagai penerjemah baris perintah di mana pengguna memasukkan pernyataan secara berurutan dan menerima hasil dengan segera.

Shell lain, termasuk IDLE dan IPython, menambahkan kemampuan lebih lanjut seperti penyelesaian otomatis yang ditingkatkan, retensi status sesi, dan Syntax highlighting.

Selain integrated development environments desktop standar, ada IDE berbasis peramban Web; SageMath (dimaksudkan untuk mengembangkan program Python yang berhubungan dengan sains dan matematika); PythonAnywhere, IDE berbasis browser dan lingkungan hosting; dan Canopy IDE, IDE Python komersial yang menekankan komputasi ilmiah.[48]

Konferensi Pengembang

Konferensi pengembang Python di Indonesia (PyCon Indonesia[49]) dilaksanakan sejak tahun 2017, berlangsung setiap tahun.

Berikut tema dan lokasi konferensi pengembang Python Indonesia:

1. Tahun 2017 di Surabaya dengan tema "A New Beginning"
2. Tahun 2018 di Jakarta dengan tema "Python for Everyone"
3. Tahun 2019 di Surabaya dengan tema "The Beauty of Python"
4. Tahun 2020 secara daring dengan tema "Connect - Collab - Contribute"

Ini adalah kegiatan yang dilaksanakan oleh komunitas Python Indonesia.

Generator dokumentasi API

Alat yang dapat digunakan untuk membuat dokumentasi API Python termasuk pydoc (tersedia sebagai bagian dari pustaka standar), Sphinx, Pdoc dan forknya, Doxygen dan Graphviz, diantara yang lain.[50]

Bahasa yang dipengaruhi oleh Python

Desain dan filosofi Python telah memengaruhi banyak bahasa pemrograman lainnya:

- Boo menggunakan indentasi, sebuah sintaks yang serupa, dan model objek yang serupa.[51]
- Cobra menggunakan indentasi dan sebuah sintaks yang serupa, dan dokumen *Pengakuannya* mencantumkan Python pertama di antara bahasa yang memengaruhinya[52]
- CoffeeScript, sebuah bahasa pemrograman yang dikompilasi silang ke JavaScript, memiliki sintaks yang terinspirasi Python.
- ECMAScript/JavaScript meminjam iterator dan generator dari Python.[53]
- GDScript, bahasa scripting yang sangat mirip dengan Python, built-in ke mesin permainan Godot.[54]
- Go didesain untuk "kecepatan bekerja dalam bahasa dinamis seperti Python"[55] dan berbagi sintaks yang sama untuk mengiris array.

- Swift, sebuah bahasa pemrograman yang dikembangkan oleh Apple, memiliki beberapa sintaks yang terinspirasi Python.[56]

Sertifikasi Python

Python Institute adalah lembaga yang kredibel yang menawarkan sertifikasi Python. Python Institute didirikan pada tahun 2003 oleh Guido van Rossum, pencipta bahasa pemrograman Python. Python Institute memiliki misi untuk mempromosikan penggunaan Python dan meningkatkan keterampilan pengembang Python. Python Institute telah menetapkan jalur sertifikasi global independen untuk bahasa pemrograman Python. Jalur tersebut terdiri dari enam track sertifikasi[57]:

- General-Purpose Programming (PCEP™, PCAP™, PCPP1™, and PCPP2™ exams)
- Data Science (PCED™ and PCAD™ exams)
- Testing (PCET™, PCAT™, and PCPT™ exams)
- Security (PCES™, and PCAS™ exams)
- Network Programming (PCEN™, and PCAN™ exams)
- Web Development (PCEW™, and PCAW™ exams)

Lihat pula

- Jython (java)
- PyPy
- JavaScript

Referensi

1. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama guttag
2. ^ *"Python 3.12.7 released" (dalam bahasa bahasa Inggris). 1 Oktober 2024. Diakses tanggal 1 Oktober 2024.*
3. ^ *"PEP 483 -- The Theory of Type Hints". Python.org.*
4. ^ File extension .pyo was removed in Python 3.5. See PEP 0488
5. ^ *Holth, Moore (30 March 2014). "PEP 0441 -- Improving Python ZIP Application Support". Diakses tanggal 12 November 2015.*
6. ^ *"Starlark Language". Diakses tanggal 25 May 2019.*
7. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama faq-created
8. ^ *"Ada 83 Reference Manual (raise statement)".*
9. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama 98-interview
10. ^ Lompat ke: *a b "itertools — Functions creating iterators for efficient looping — Python 3.7.1 documentation". docs.python.org.*
11. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama AutoNT-1
12. ^ Lompat ke: *a b* Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama classmix
13. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama effbot-call-by-object

14. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama AutoNT-2
15. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama AutoNT-3
16. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama AutoNT-4
17. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama AutoNT-5
18. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama AutoNT-6
19. ^ *"CoffeeScript"*. *coffeescript.org*.
20. ^ *"The Genie Programming Language Tutorial"*. *Diakses tanggal 28 February 2020*.
21. ^ *"Perl and Python influences in JavaScript"*. *www.2ality.com. 24 February 2013. Diakses tanggal 15 May 2015*.
22. ^ *Rauschmayer, Axel. "Chapter 3: The Nature of JavaScript; Influences". O'Reilly, Speaking JavaScript. Diarsipkan dari versi asli tanggal 2018-12-26. Diakses tanggal 15 May 2015*.
23. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama Julia
24. ^ *Ring Team (4 December 2017). "Ring and other languages". ring-lang.net. ring-lang*.
25. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama bini
26. ^ *Lattner, Chris (3 June 2014). "Chris Lattner's Homepage". Chris Lattner. Diakses tanggal 3 June 2014. "The Swift language is the product of tireless effort from a team of language experts, documentation gurus, compiler optimization ninjas, and an incredibly important internal dogfooding group who provided feedback to help refine and battle-test ideas. Of course, it also greatly benefited from the experiences hard-won by many other languages in the field, drawing ideas from Objective-C, Rust, Haskell, Ruby, Python, C#, CLU, and far too many others to list."*
27. ^ *"A Python Book: Beginning Python, Advanced Python, and Python Exercises". web.archive.org. 2012-06-23. Archived from the original on 2012-06-23. Diakses tanggal 2020-08-11*.
28. ^ *"Welcome to Python.org". Python.org (dalam bahasa Inggris). Diakses tanggal 2020-08-11*.
29. ^ *"History and License — Python 3.8.5 documentation". docs.python.org. Diakses tanggal 2020-08-11*.
30. ^ *"Stack Overflow Developer Survey 2020". Stack Overflow. Diarsipkan dari versi asli tanggal 2 March 2021. Diakses tanggal 2021-03-05*.
31. ^ *"The State of Developer Ecosystem in 2020 Infographic". JetBrains: Developer Tools for Professionals and Teams (dalam bahasa Inggris). Diarsipkan dari versi asli tanggal 1 March 2021. Diakses tanggal 2021-03-05*.
32. ^ *"index | TIOBE - The Software Quality Company". www.tiobe.com. Diarsipkan dari versi asli tanggal 25 February 2018. Diakses tanggal 2021-02-02. "Python has won the TIOBE programming language of the year award! This is for the fourth time in the history, which is a record! The title is awarded to the programming language that has gained most popularity in one year."*
33. ^ *"PYPL PopularitY of Programming Language index". pypl.github.io (dalam bahasa Inggris). Diarsipkan dari versi asli tanggal 14 March 2017. Diakses tanggal 2021-03-26*.
34. ^ Lompat ke: ***a b*** *"The Making of Python". www.artima.com. Diakses tanggal 2020-08-11*.
35. ^ *Rossum, Guido van (2000-08-29). "[Python-Dev] SETL (was: Lukewarm about range literals)". Diakses tanggal 2020-08-11*.
36. ^ *"General Python FAQ — Python 3.8.5 documentation". docs.python.org. Diakses tanggal 2020-08-11*.
37. ^ *Langa, Łukasz (2022-05-17). "Python Insider: Python 3.9.13 is now available". Python Insider. Diakses tanggal 2022-05-21*.

38. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama AutoNT-59
39. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama AutoNT-18
40. ^ *"PEP 20 -- The Zen of Python"*. *Python.org (dalam bahasa Inggris). Diakses tanggal 2020-09-24.*
41. ^ *"Highlights: Python 2.5"*. *Python.org.*
42. ^ *"PEP 465 -- A dedicated infix operator for matrix multiplication"*. *python.org. Diarsipkan dari versi asli tanggal 29 May 2020. Diakses tanggal 3 July 2018.*
43. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama AutoNT-86
44. ^ *"PEP 327 -- Decimal Data Type"*. *Python.org (dalam bahasa Inggris). Diakses tanggal 2020-08-11.*
45. ^ *"PEP 333 -- Python Web Server Gateway Interface v1.0"*. *Python.org (dalam bahasa Inggris). Diakses tanggal 2021-01-22.*
46. ^ *"Modulecounts"*. *www.modulecounts.com. Diakses tanggal 2020-08-19.*
47. ^ *Ebrahim, Mokhtar (2017-12-05). "Python web scraping tutorial (with examples)". Like Geeks (dalam bahasa Inggris). Diakses tanggal 2020-08-19.*
48. ^ *Enthought, Canopy. "Canopy". www.enthought.com. Diarsipkan dari versi asli tanggal 2017-07-15. Diakses tanggal 20 August 2016.*
49. ^ *"PyCon Indonesia 2019". Diakses tanggal 2019-10-17.*
50. ^ *"Documentation Tools". Python.org (dalam bahasa Inggris). Diarsipkan dari versi asli tanggal 11 November 2020. Diakses tanggal 2021-03-22.*
51. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama AutoNT-90
52. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama AutoNT-91
53. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama AutoNT-93
54. ^ *"Frequently asked questions". Godot Engine documentation. Diarsipkan dari versi asli tanggal 28 April 2021. Diakses tanggal 10 May 2021.*
55. ^ Kesalahan pengutipan: Tag <ref> tidak sah; tidak ditemukan teks untuk ref bernama AutoNT-94
56. ^ *Lattner, Chris (3 June 2014). "Chris Lattner's Homepage". Chris Lattner. Diarsipkan dari versi asli tanggal 22 December 2015. Diakses tanggal 3 June 2014. "I started work on the Swift Programming Language in July of 2010. I implemented much of the basic language structure, with only a few people knowing of its existence. A few other (amazing) people started contributing in earnest late in 2011, and it became a major focus for the Apple Developer Tools group in July 2013 [...] drawing ideas from Objective-C, Rust, Haskell, Ruby, Python, C#, CLU, and far too many others to list."*
57. ^ *"Certification Overview". pythoninstitute.org. Diakses tanggal 2024-06-05.*

Bacaan lanjutan

- *Downey, Allen B. (May 2012). Think Python: How to Think Like a Computer Scientist (edisi ke-Version 1.6.6). ISBN 978-0-521-72596-5.*
- *Hamilton, Naomi (5 August 2008). "The A-Z of Programming Languages: Python". Computerworld. Diarsipkan dari versi asli tanggal 29 December 2008. Diakses tanggal 31 March 2010.*
- *Lutz, Mark (2013). Learning Python (edisi ke-5th). O'Reilly Media. ISBN 978-0-596-15806-4.*
- *Pilgrim, Mark (2004). Dive into Python. Apress. ISBN 978-1-59059-356-1.*
- *Pilgrim, Mark (2009). Dive into Python 3. Apress. ISBN 978-1-4302-2415-0.*

- *Summerfield, Mark (2009). Programming in Python 3 (edisi ke-2nd). Addison-Wesley Professional. ISBN 978-0-321-68056-3.*

Pranala luar

- Situs web resmi