python (编程语言)

python 是一种用于通用编程的高级编程语言。python 由 guido van rossum 创建，于 1991 年首次发布，其设计理念强调代码可读性，尤其是使用重要的空白。它提供了在小型和大型规模上实现清晰编程的构造。2018 年 7 月，van rossum 辞去了语言社区的领导者职务。[27] [28]

Python

范式	面向对象、命令性、功能性、程序性、反光
设计人员	guido van rossum
开发 人员	python 软件基金会
第一次出现	1990 年；28 年前[6]
稳定的释放	3.7.1/10 月 20 日；44 天前[3] 2.7.15；6 个月前[3]
打字纪律	鸭子,动感,强壮;并且**从 3.5 版:** 逐渐 [4]
许可证	python 软件基础许可证
文件名扩展	. py,. pyc,. pyd,. pyo (在 3.5 之前),[5] . pyw,. pyz (因为 3.5) [6]
网站	www. python. org
主要实现	
cpython、 ironpython、 CPython、 micropython、 numba、 pypi、 stackless python	
方言	
cython, rpython	
受	
abc,[7] algol 68,[8] apl[9] c,[10] c++,[11] clu,[12] 迪伦,[13] 哈斯凯尔,[14] 图标,[15] 爪哇,[16] lisp,[17] mo 杜拉-3,[11] perl, standard ml[9]	

影响
boo,眼镜蛇,咖啡,[18] d, f #, 猎鹰,精灵, [19]去, apache groovy, javascript,[20][21] julia,[22] nim, ring,[23] ruby,[24] 斯威夫特[25]
• 维基百科的 python 编程

python 具有动态类型系统和自动内存管理功能。它支持多种编程范式，包括面向对象、命令式、功能和过程，并具有大型且全面的标准库。[29]

python 解释器可用于许多操作系统。cpython 是 python 的参考实现，它是开源软件[30]，具有基于社区的开发模型，python 的几乎所有其他实现也是如此。python 和 cpython 由非营利的 python 软件基金会管理.

历史

guido van rossum oscon 2006 裁剪

主要文章: python 的历史

python 是由荷兰 wiskunde & amp; informatica 中心 (cwi) 的 guido van rossum 在 20 世纪 80 年代末构想的 [31], 作为 abc 语言的接班人 (本身灵感来自 setl)[32], 能够异常处理和接口与 Amoeba 操作系统。[7] 它的实施在 1989 年 12 月开始了。[33] van rossum 对 python 的长期影响反映在 python 社区给他的标题中: *仁慈的生活独裁者* (bdfl)——7 月 12 日, 他从这个帖子中给了自己永久的假期,2018.[34]

python 2.0 于 2000 年 10 月 16 日发布，具有许多主要的新功能，包括循环检测垃圾收集器和对 unicode 的支持。[35]

python 3.0 于 2008 年 12 月 3 日发布。这是对语言的重大修订，并不完全向后兼容。[36] 它的许多主要特征被备份到 python 2.6. x[37] 和 2.7. x 版本系列。python 3 的版本包括 2to3 实用程序，它自动（至少部分）将 python 2 代码转换为 python 3。[38]

python 2.7 的终止日期最初定为 2015 年，然后推迟到 2020 年，因为担心大量现有代码不容易移植到 python 3。[39] [40]2017 年 1 月，google 宣布开发 python 2.7 到 go 转编译器，以提高并发工作负载下的性能。[41]

特点和理念

python 是一种多范式编程语言。完全支持面向对象的编程和结构化编程,它的许多功能支持函数式编程和面向方面的编程(包括通过元编程) [42] 和元对象(魔术方法)。[43] 许多其他范例通过引伸支持，包括设计由合同 [44][45] 和逻辑编程。[46]

python 使用动态类型，以及引用计数和循环检测垃圾收集器的组合来进行内存管理。它还具有动态名称解析(后期绑定)，它在程序执行过程中绑定方法和变量名。

python 的设计为 lisp 传统中的函数式编程提供了一些支持。它具有 filter()、map()和 reduce()函数;列出理解、字典和集合;和生成器表达式。[47] 标准库有两个模块（迭代工具和功能工具），用于实现从 haskell 和 standard ml 借来的功能工具。[48]

该语言的核心哲学总结在文件 *python (pep 20)*，其中包括格言，如:[49]

- 美丽胜于丑陋
- 显式比隐式好
- 简单胜于复杂
- 复杂比复杂好
- 可读性计数

python 的设计并不是将其所有功能都内置到其核心中，而是具有高度可扩展性。这种紧凑的模块化使其作为向现有应用程序添加可编程接口的一种手段特别受欢迎。van rossum 认为，他的观点是一种小型的核心语言，拥有一个庞大的标准库和易于扩展的口译服务，这源于他对 abc 的不满，而 abc 支持相反的方法。[31]

在提供编码方法选择的同时，python 哲学拒绝了旺盛的语法（如 perl 语法），而倾向于更简单、更不杂乱的语法。正如亚历克斯·马尔泰利所说: "在 python 文化中，将一些东西

描述为 ' 聪明 '*并不*被认为是一种赞美"[50] python 的哲学拒绝 perl "有不止一种方法可以做到这一点" 的语言设计方法，而倾向于 "应该有一种——最好是只有一种——明显的方法来做到这一点"。[49]

python 的开发人员努力避免过早优化，并拒绝对 cpython 的非关键部分进行修补，这些补丁将以清晰为代价提供极快的速度。[51] 当速度很重要时，python 程序员可以将时间关键的函数移动到用 c 等语言编写的扩展模块上，或者使用实时编译器 pypy。cython 也可用，它将 python 脚本转换为 c，并将直接的 c 级 api 调用转换为 python 解释器。

python 开发人员的一个重要目标是保持使用乐趣。这反映在语言的名称中——对英国喜剧团体蒙蒂·毕顿[52]的赞扬[52]——以及偶尔对教程和参考材料的好玩的方法，例如引用垃圾邮件和鸡蛋的示例（来自著名的蒙蒂·毕顿素描)而不是标准的 foo 和 bar。[53] [25]

python 社区中常见的新词是*巨蟒*，它可以有与程序风格有关的广泛含义。说代码是巨蟒，是说它很好地使用 python 成语，它是自然的或显示流利的语言，它符合 python 的最低限度的哲学和对可读性的强调。相反，难以理解的代码或从另一种编程语言读出的粗抄本*被称为非巨生语*.

python 的使用者和崇拜者，特别是那些被认为知识渊博或经验丰富的人，通常被称为 *pythonisters*、 *pythonistas* 和 *Pythonists*。[25] [56]

语法和语义

主要文章: python 语法和语义

python 是一种易于阅读的语言。它的格式在视觉上很整洁，它经常使用英语关键字，其他语言使用标点符号。与许多其他语言不同，它不使用大括号来分隔块，并且语句后的分号是可选的。与 c 或 pascal 相比，它的语法异常和特殊情况较少。[57]

压 痕

主要文章: python 语法和语义§缩进

python 使用空白缩进（而不是大括号或关键字）来分隔块。缩进的增加在某些声明以后;缩进的减少表示当前块的结束。[58] 因此，程序的视觉结构准确地代表了程序的语义结构。[1] 此功能有时也称为越位规则.

语句和控制流

python 的陈述包括（除其他外):

- 赋值语句（标记 "="，等号）。这与传统的命令式编程语言的操作方式不同，这种基本机制（包括 python 变量版本的性质）说明了该语言的许多其他功能。c 中的赋值（x = 2）转换为 "类型化变量名 x 接收数值 2 的副本"。(右侧）值被复制到分配的存储位置，其中（左侧)变量名是符号地址。分配给变量的内存对于声明的类型来说足够大（可能相当大）。在 python 赋值的最简单情况下，使用相同的示例 x = 2 将 x 转换为 "(泛型）名称 x 接收对值为 2 的数字（int）类型的单独动态分配对象的引用。这称为*将名称绑定*到对象。由于名称的存储位置不*包含*指示的值，因此将其称为*变量*是不合适的。名称随后可能会在任何时候反弹到类型千差万别的对象，包括字符串、过程、包含数据和方法的复杂对象等。将一个共同值连续分配给多个名称，例如，x = 2;y = 2;z = 2 存储分配给（最多）三个名称和一个数字对象，所有三个名称都绑定到这些名称和对象。由于名称是通用引用持有者，因此将固定数据类型与其关联是不合理的。但是，在给定的时间，名称将绑定到*某个*对象，该**对象将**具有类型;因此，有动态键入.
- if 语句，它有条件地执行代码块，以及 else 和 elif (elif if 的收缩)。

- for 语句，它遍历可迭代对象，将每个元素捕获到局部变量中，以供附加块使用。
- while 语句，它执行一个代码块，只要它的条件为真。
- try 语句，它允许通过子句以外的子句捕获和处理在其附加代码块中 except 引发的异常;它还可确保无论块如何退出 finally 块中的清理代码都将始终运行。
- raise 声明，用于引发指定的异常或重新引发捕获的异常。
- class 语句，它执行一个代码块，并将其本地命名空间附加到一个类，以便在面向对象的编程中使用.
- def 语句，它定义函数或方法.
- with 来自 2006 年 9 月发布的 python 2.5 [59]，它在上下文管理器中包含一个代码块（例如，在运行代码块之前获取锁并释放锁之后，或打开一个文件，然后将其关闭），允许资源获取是初始化(raii) 类似的行为，并替换一个常见的尝试/最后的成语。[60]
- pass 语句，用作 nop。在语法上需要创建一个空的代码块。
- assert 语句，在调试过程中用于检查应应用的条件。
- yield 语句，它返回来自生成器函数的值。从 python 2.5 开始,yield 也是一个运算符。此窗体用于实现 coroutines.
- import 语句，用于导入其函数或变量可在当前程序中使用的模块。使用导入的方法有三种:import <module name> [as <alias>]from <module name> import * ;]: 导入 & lt; 模块名称 &

gt; & & & & 导入 *from <module name> import <definition 1> [as <alias 1>], <definition 2> [as <alias 2>], ...

- print 语句已更改为 python 3 中的 print()函数。[61]

python 不支持尾部呼叫优化或一级延续,根据 guido van rossum 的说法，它永远不会支持。[62][63] 然而，通过扩展 python 的生成器,2.5 中提供了对类似 coroutine 的功能的更好的支持。[64] 在 2.5 之前，发电器是懒惰迭代器;信息是从生成器中单向传递的。从 python 2.5 中，可以将信息传递回生成器函数，从 python 3.3，可以通过多个堆栈级别传递信息。[65]

表达 式

有些 python 表达式类似于 c 和 java 等语言，而有些则不是:

- 加法、减法和乘法是相同的，但除法的行为不同。python 中有两种类型的划分。它们是地板划分和整数划分。[66] python 还为指数化添加**运算符。
- 从 python 3.5 中，引入了新的@ infix 运算符。它是打算由库，如 numpy 矩阵乘法。[67][68]
- 在 python 中，= = 按值比较，而 java 则按值[69] 比较数字，按引用比较对象。[70] (equals()方法在对象上执行 java 中的值比较。pythonis 运算符可用于比较对象标识

(通过引用进行比较)。在 python 中，比较可能是链接的，例如 a <= b <= c.

- python 使用的词 andor 为它的布尔运算符，而 not 符号 & amp &&, ||!java 和 c 中使用。
- python 有一种称为*列表理解*的表达方式。python 2.4 扩展列表理解为一种更一般的表达式，称为*生成器表达式*.[47]
- 匿名函数是使用 lambda 表达式实现的；匿名函数是使用 lambda 表达式实现的。然而，这些都是有限的，因为身体只能是一个表达。
- python 中的条件表达式被编写为 x if c else y[71] (按操作数的顺序与许多其他语言共有的 c? x:y c ? x : y 不同)。
- python 对列表和元组进行了区分.列表是[1, 2, 3]的形式编写的，是可变的，不能用作字典的键（字典键在 python 中必须是不可变的)。元组编写为(1, 2, 3)，是不可变的，因此可以用作字典的键，前提是元组的所有元素都是不可变的。+运算符可用于连接两个元组，这不直接修改其内容，而是生成一个新的元组，其中包含两个提供的元组的元素。因此，给定变量 t 最初等于(1, 2, 3))，执行 t = t + (4, 5)首先计算 t + (4, 5))，产生(1, 2, 3, 4, 5)，然后被分配回 t 从而有效地 "修改 t 的内容 "，同时符合元

组对象的不可变性质。对于明确上下文中的元组，括号是可选的。[72]

- python 特征*序列解包*,其中多个表达式（每个表达式都计算到可以分配给的任何东西（变量、可写属性等），都以与形成元组文本的方式相同的方式关联，并作为一个整体,赋值语句中等号的左侧。该语句需要在等号的右侧有一个*可迭代*对象，该对象在迭代时生成与所提供的可写表达式相同数量的值，并将遍历该对象，将每个生成的值分配给在左边相应的表达式。[73]
- python 具有 "字符串格式" 运算符%。此函数类似于 c 中 printf 格式字符串，例如"spam=%s eggs=%d" % ("blah", 2)的计算结果为"spam=blah eggs=2"。在 python 3 和 2.6 + 中，str 类的 format() spamsthx{ "spam={0} eggs={1}".format("blah", 2)对此进行了补充。python 3.6 增加了 "f–字符串" blah = "blah"; eggs = 2; f'spam={blah} eggs={eggs}'[74]
- python 有各种字符串文本:
  - 由单引号或双引号分隔的字符串。与 unix shell、perl 和 perl 受影响的语言不同，单引号和双引号的功能相同。这两种字符串都使用反斜杠(\) 作为转义字符。字符串插值在 python 3.6 中作为 "格式化字符串文本" 可用。[74]

    - 三引号字符串，以三个单引号或双引号的系列开头和结尾。它们可以跨越多条线，像这里的文件在 shell, perl 和 ruby.
    - 原始字符串品种，通过在字符串文本之前加上 r 来表示. r 逃逸序列不被解释;因此，在文本反斜杠很常见的地方，例如正则表达式和 windows 样式的路径，原始字符串非常有用。比较 c#中的 "@报价".
- python 在列表上有数组索引和数组切片表达式，表示为 a[key]、 a[start:stop]或 a[start:stop:step]。索引从零开始，负索引相对于端。切片将元素从*起始*索引中提取到停止索引，但不包括停止索引。第三个切片参数（称为*"步长"* 或 *"步幅"*）允许跳过和反转元素。切片索引可以省略，例如 a[:]返回整个列表的副本。切片的每个元素都是浅层副本.

在 python 中，与公共 lisp、scheme 或 ruby 等语言相比，对表达式和语句之间的区别是严格强制的。这将导致重复某些功能。例如:

- 列表理解与–循环 for
- 条件表达式与 if 块

- eval()与 exec() ）内置函数（在 python 2 中，exec是一个语句);前者表示表达式，后者表示语句。

语句不能是表达式的一部分，因此列表和其他理解或 lambda 表达式(所有表达都是表达式）不能包含语句。这种情况的一个特殊情况是，赋值语句（a = 1 不能构成条件语句的条件表达式的一部分。这样做的优点是避免了一个经典的 c 错误，错误地将=赋值运算符==条件下:if (c = 1) { ... } {...} 在语法上是有效的（但可能是意外的）c 代码，但 if c = 1: ...在 python 中导致语法错误。

方法

对象上的方法是附加到对象的类的函数；语法 instance.method(argument)，对于正常方法和函数，语法糖的 class.方法（Class.method(instance, argument)。python self 方法具有一个显式 self 来访问实例数据，这与其他一些面向对象 this 的编程语言（例如，c++、 java、 objjec-uby）. [75]

打字

python 使用鸭子类型，并且具有类型化对象，但非类型化变量名。在编译时不检查类型约束;相反，对对象的操作可能会失败，这表示给定的对象不是合适的类型。尽管 python 是动态类型化的，但它是强类型的，禁止未定义的操作（例如，向字符串中添加一个数字），而不是默默尝试理解它们。

python 允许程序员使用类定义自己的类型，这些类最常用于面向对象的编程。类的新实例是通过调用类（例如，SpamClass()或 EggsClass()）构造的，这些类是元类 type 的实例（本身就是一个实例），允许元编程和反射.

在 3.0 版之前，python 有两种类:*旧式*类和*新型*类。[76] 这两种样式的语法是相同的，区别是类 object 是从继承的，直接还

是间接（所有新样式类继承从 object 并且是 type 的实例）。在 python 2.2 开始的 python 2 版本中，可以使用这两种类。python 3.0 中删除了旧式类。

长期计划是支持逐步键入[77]，从 python 3.5 中，语言的语法允许指定静态类型，但它们不会在默认实现 cpython 中进行检查。一个名为 *mypy* 的实验性可选静态类型检查器支持编译时类型检查。[78]

python 3 的内置类型摘要

类型	可变	描述	语法示例
bool	变	布尔值	True False
bytearray	可变	字节的顺序	bytearray(b'Some ASCII') bytearray(b"Some ASCII") bytearray([119, 105, 107, 105])
bytes	变	字节的顺序	b'Some ASCII' b"Some ASCII" bytes([119, 105, 107, 105])
complex	变	具有真实和虚构部分的复数	3+2.7j
dict	可变	关联数组键和值对的（或字典);可以包含混合类型（键和值），键必须是可哈希类型	{'key1': 1.0, 3: False}
ellipsis		要用作 numpy 数组中的索引的省略号占位符	...
float	变	浮点数，系统定义的精度	3.1415927
frozenset	变	无序设置，不包含重复项;可以包含混合类型，如果可共享	frozenset([4.0, 'string', True])
int	变	无限大小的整数[79]	42

list	可变	列表，可以包含混合类型	[4.0, 'string', True]
set	可变	无序设置，不包含重复项;可以包含混合类型，如果可共享	{4.0, 'string', True}
str	变	字符串：unicode 代码点的序列	'Wikipedia' "Wikipedia" """Spanning multiple lines"""
tuple	变	可以包含混合类型	(4.0, 'string', True)

数学

python 具有通常的 c 语言算术运算符+、– *、/ % /. -它还具有**指数化，例如 5**3 == 125 9**0.5 == 3.0，3.5 版中包括一个新的矩阵@运算符。[80]另外，它有一个一元运算符（~），根本上倒置它的一个论据的所有位。对于整数，这意味着 ~x=-x-1。[81]其他运算符包括按位移位运算符 x << y，将 x 移动到左侧 y 位置 x*(2**y)）相同，X x >> y 将 X 移位 x 到正确 y 的地方，就像 x//(2**y) 。[82]

随着时间的推移，分裂的行为发生了重大变化:[83][*为什么？*]

- python 2.1 和更早版本使用 c 除法行为。如果两个操作数都是整数,//运算符为整数除法，否则为浮点除法。整数除法舍入为 0 7/3 == 2 -7/3 == -2
- python 2.2 将整数除法改为负无穷大，例如 7/3 == 2 -7/3 == -3 介绍了地板事业部//操作人员。所以 7//3 == 2 -7//3 ==

-3，7.5//3 == 2.0 -7.5//3 == -3.0from __future__ import division 中添加会导致模块使用 python 3.0 规则进行除法（请参阅下一个）。

- python 3.0/始终是浮点划分。用 python 术语来说，前 3.0 /是*经典的除法*，版本 3.0 /是真正的*除法*，和//是*地板划分*.

舍入到负无穷大，虽然不同于大多数语言，但它增加了一致性。例如，这意味着方程(a + b)//b == a//b + 1 始终为真。这也意味着方程 b*(a//b) + a%b == a 对 a 的正值 a 负值都有效。但是，保持此公式的有效性意味着 a%b 的结果，如预期的那样，在半开放间隔 [0, *b*)，其中 b 是一个正整数，它必须位于区间（*b*, 0]当 b 为负数时。[84]

python 提供了一个 round 函数，用于将浮点舍入到最近的整数。对于打线，3 个之前的版本使用往返从零: round(0.5) ）为 1.0,round(-0.5) 为 –1.0 。[85] python 3 使用圆均匀: round(1.5) 是 2, round(2.5)）是 2。[86]

python 允许具有多个相等关系的布尔表达式的方式与数学中的一般用法一致。例如，表达 a < b < c 测试 a 是否 a 小于 b ，b 是否小于 c。[87] C 派生的语言不同地解释这个表达式: 在 C 中，表达式将首先计算 & a < bb，导致 0 或 1，然后将结果与 c 进行比较。[88]

python 对任意精度算法有广泛的内置支持。整数透明地从机器支持的最大固定精度（通常为 32 位或 64 位）（属于 python 类型 int）透明地切换到任意精度（在需要时属于 python 类型 long。后者的文本表示形式中有一个 "l" 后缀。[89]（在 python 3 中,int 和 long 类型之间的区别；这种行为现在完全包含在 int 类中。模块 decimal 中的 Decimal 类型类（自 2.4 版开始）为任意精度和多个舍入模式提供小数点浮点数。[90] 模块分数中的 Fraction 类型 fractions（因为版本 2.6）为有理数提供任意精度。[91]

由于 python 的大量数学库，以及进一步扩展本机功能的第三方库 numpy ，它经常被用作科学脚本语言，以帮助解决数字数据处理和数据处理等问题。操纵。[*需要引文*]

图书馆

python 的大型标准库通常被引用为其最大优势之一 [92]，它提供了适合许多任务的工具。对于面向 internet 的应用程序，支持许多标准格式和协议，如 mime 和 http 。它包括用于创建图形用户界面、连接到关系数据库、生成伪随机数、具有任意精度小数的算术的模块 [93] 操作正则表达式和单元测试.

标准库的某些部分包含在规范中（例如，web 服务器网关接口(wsgi）实现后，wsgiref 按照 pep 333[94]），但大多数模块

不是. wsgiref 它们由其代码、内部文档和测试套件（如果提供）指定。但是，由于大多数标准库都是跨平台的 python 代码，因此只有少数模块需要更改或重写变体实现。

截至 2018 年 3 月,python 包索引（ pypi）是第三方 python 软件的官方存储库，包含超过 130,000 个 [95] 包，具有广泛的功能，包括:

- 图形用户界面
- 网络框架
- 多媒体
- 数据库
- 网络
- 测试框架
- 自动化
- 网刮 [96]
- 文档
- 系统管理
- 科学计算
- 文字处理
- 图像处理

开发环境

参见:集成开发环境的比较§ python

大多数 python 实现（包括 cpython）都包括一个读–eval–打印循环(repl)，允许它们作为命令行解释器，用户可以为其按顺序输入语句并立即接收结果。

其他外壳，包括 idle 和 ipython，添加了更多的功能，如自动完成、会话状态保留和语法突出显示.

除了标准的桌面集成开发环境外，还有基于 web 浏览器的 ide;萨吉数学(旨在开发科学和与数学有关的 python 项目);pythonanywhere，基于浏览器的 ide 和宿主环境;和树冠 ide，一个强调科学计算的商业 python ide。[97]

实现

参见: python 软件列表§ python 实现

参考实施

cpython 是 python 的参考实现。它是用 c 编写的，符合 c89 标准，有几个选定的 c99 功能。[98] 它将 python 程序编译成一个中间字节码[99]，然后由其虚拟机执行。[100] cpython 与一个大型标准库一起分布，该库是用 c 和本机 python 的混合物编写的。它适用于许多平台，包括 windows 和大多数现

代类似 unix 的系统。平台可移植性是其最早的优先事项之一。[101]

其他实现

pypy 是 python 2.7 和 3.5 的快速、合规[102]解释器。与 cpython 相比，它的实时编译器带来了显著的速度改进。[103]

无堆栈 python 是实现微线程的 cpython 的一个重要分支；它不使用 c 内存堆栈，从而允许大规模并发程序。pyy 也有一个无堆叠的版本。[104]

micropython 和 circuitpython 是针对微控制器优化的 python 3 变体.

不支持的实现

其他实时 python 编译器已经开发，但现在不受支持：

- 谷歌在 2009 年开始了一个名为"解登燕子"的项目，目的是通过使用 llvm 来加快 python 解释器的 5 倍，并提高其多线程扩展到数千个内核的能力。[105]
- psyco 是一个实时专用的编译器，它与 cpython 集成，并在运行时将字节码转换为机器代码。发出的代码专门用于某些数据类型，并且比标准的 python 代码更快。

2005 年,诺基亚发布了一款名为 pys60 的 60 系列手机的 python 解释器。它包括 CPython 实现中的许多模块和一些与 symbian 操作系统集成的其他模块。该项目保持最新状态,可在 s60 平台的所有变体上运行,并且有几个第三方模块可用。诺基亚 n900 还支持 python 与 gtk 小部件库,使程序能够在目标设备上编写和运行。[106]

交叉编译器到其他语言

高级对象语言有几个编译器,可以是不受限制的 python、python 的受限子集,也可以是类似于 python 的源语言:

- jython 将编译为 java 字节代码,然后每个 java 虚拟机实现都可以执行该代码。这还允许使用 python 程序中的 java 类库函数。
- ironpython 遵循类似的方法,以便在. net 公共语言运行时运行 python 程序.
- rpython 语言可以编译到 c、java 字节码或通用中间语言,并用于构建 python 的 pyty 解释器。
- pyjs 将 python 编译为 javascript.
- cython 将 python 编译为 c 和 c++.
- pythran 将 python 编译为 c++.
- pyrex (2010 年的最新版本)和应 shed skin (2013 年的最新版本)分别编译为 c 和 c++。

- 谷歌的脾气暴躁编译 python 去.
- myhdl 将 python 编译为 vhdl.
- nuitka 将 python 编译为 c++。[107]

性能

在 euroscipi ' 13 中介绍了关于非数字（组合）工作负载的各种 python 实现的性能比较。[108]

发展

python 的开发主要是通过 *python 增强建议*(pep) 过程进行的，该过程是提出主要新功能、收集有关问题的社区输入和记录 python 设计决策的主要机制。[109]杰出的 pep 由 python 社区和 guido van rossum (python 的仁慈的生活独裁者）进行审查和评论。[109]

语言的增强与 cpython 引用实现的开发相对应。邮件列表 python-dev 是该语言发展的主要论坛。具体问题在 python.org 上的 roundup bug 跟踪器中进行了讨论。开发最初是在运行 mercurial 的自托管源代码存储库上进行的，直到 2017 年 1 月 python 搬到 github. [111]

cpython 的公共版本有三种类型，由版本号的一部分递增:

- 向后不兼容的版本，其中代码预计将中断，需要手动移植。版本号的第一部分将递增。这些版本很少出现，例如，3.0 版本是在 2.0 后发布的。
- 主要或 "功能" 版本，大约每 18 个月发布一次，基本上是兼容的，但引入了新的功能。版本号的第二部分将递增。每个主要版本在发布后的几年内都由错误修复程序支持。[112]
- 错误修复版本，没有引入新的功能，大约每 3 个月发生一次，并在自上次版本以来在上游修复了足够数量的错误时进行。这些版本中还修补了安全漏洞。版本号的第三部分也是最后一部分将递增。[113]

许多 alpha、beta 和发布候选项也会作为预览发布，并在最终发布前进行测试。虽然每个版本都有一个粗略的时间表，但如果代码还没有准备好，它们通常会被延迟。python 的开发团队通过在开发过程中运行大型单元测试套件和使用构建机器人连续集成系统来监视代码的状态。[114]

python 开发人员社区还贡献了超过 86，000 个[115] 软件模块（截至 2016 年 8 月 20 日）到 python 包索引（pypi），这是第三方 python 库的官方存储库。

关于 python 的主要学术会议是 pycon。还有专门的 python 辅导方案，如 Pyladies.

命名

python 的名字来自英国喜剧团体 monty python, python 创作者 guido van rossum 在开发语言时喜欢他。在 python 代码和区域性中，蒙蒂 python 引用频繁出现;[116] 例如，python 文献中经常使用的元策略变量是*垃圾邮件*和*鸡蛋*，而不是传统的 *foo* 和 *bar*。[11][117] 正式 python 文档还包含对 monty python 例程的各种引用。[118][119]

前缀 *py–用于*显示与 python 有关的东西。在 python 应用程序或库的名称中使用此前缀的示例包括 pygame, 这是 sdl 与 python 的绑定（通常用于创建游戏);pyqt 和 pygtk, 分别将 qt 和 gtk 与 python 绑定在一起;和 pypy，最初用 python 编写的 python 实现。

api 文档生成器

python api 文档生成器包括:

- 狮身人面像
- 埃皮多克
- 头文档
- 皮多克

使用

主要文章: python 软件列表

自 2003 年以来, python 一直在 tiobe 编程社区索引中名列前十位最受欢迎的编程语言之列, 截至 2018 年 1 月, 它是第四种最流行的语言（仅次于 java、 c 和 c++）。[120] 它在 2007 年和 2010 年被选择了年度编程语言。[121]

一项实证研究发现, 脚本语言（如 python）比传统语言（如 c 和 java）相比, 对于涉及字典中字符串操作和搜索的编程问题更有成效, 并确定内存消耗是通常 "比 java 好, 也不比 c 或 c++ 差多少"。[122]

使用 python 的大型组织包括维基百科、谷歌、[123] yahoo!、[124] cern、[125]nasa, [126] facebook,[127] amazon, instagam, spotify[128] 和一些较小的实体, 如 ilm[129] 和 ita。[130] 社交新闻网络网站 reddit 完全用 python 编写。

python 可以作为 web 应用程序的脚本语言, 例如, 通过 apache web 服务器的 mod _ wsgi 。[131] 使用 web 服务器网关接口, 已经发展出一个标准 api 来促进这些应用程序。网络框架, 如 django, pylons,金字塔,涡轮齿轮, 网络 2 比, 龙卷风,烧瓶, 瓶子和 zope 支持开发人员在 复杂应用的设计和维护。pyjs 和 ironpython 可用于开发基于 ajax 的应用程

序的客户端。sqlalchemy 可用作关系数据库的数据映射器。扭曲是一个编程计算机之间通信的框架,由 dropbox 使用(例如).

像 numpy、 s 学皮和 matplotlib 这样的库允许 python 在科学计算中有效使用,[132] [133] 有专门的库, 如 biopython 和所有的人提供域特定的功能。sage 解 2 成是一个数学软件, 在 python 中具有可编程的笔记本接口: 它的库涵盖了数学的许多方面, 包括代数、组合学、 数值数学、数论和微积分.

python 已成功地嵌入到许多软件产品中, 作为脚本语言, 包括有限元方法软件, 如 abaqus, 3ds 参数建模器, 如 freecad, 3ds 动画包, 如 3ds 最大, blender,影院 4d,光波, 胡迪尼, 玛雅, 莫多,摩托车制造商, 软体, 视觉效果作曲家 nuke,2d 成像节目象 gimp,[134] inkscape,涂鸦并且油漆商店临,[135] 并且音乐记数器节目象记分员和卡佩拉。gnu 调试器使用 python 作为漂亮的打印机来显示复杂的结构, 如 c++ 容器。esri 提倡 python 作为在 arcgis 中编写脚本的最佳选择。[136] 它也被用于几个电子游戏,[137][138] 并且被采纳了作为第一三种可用的编程语言在 google 应用程序引擎, 另外两个是 java 和 go。 [139]python 也用于算法交易和定量金融。[140] python 还可以通过使用包装在其他语言上运行的在线经纪公司的 api 中实现。[141]

python 通常用于人工智能项目中，借助 tensorflow、keras 和科学学习等库。[142] [143] [144] [145] 作为一种具有模块化体系结构、简单语法和丰富的文本处理工具的脚本语言，python 通常用于自然语言处理。[146]

许多操作系统都将 python 作为标准组件。它与大多数 linux 发行版、amigaos 4、freebsd、netbsd、openbsd 和 macos 一起发送，并且可以从命令行（终端）使用。许多 linux 发行版使用用 python:ubuntu 编写的安装程序，而红帽 linux 和 fedora 使用 anaconda 安装程序。gentoo linux 在其软件包管理系统中使用 python，portage.

python 在信息安全行业中被广泛使用，包括在开发开发中。[147] [148]

现在在糖实验室开发的每个孩子 xo 一台笔记本电脑的糖软件大多是用 python 编写的。[149] 树莓派单板计算机项目采用 python 作为其主要的用户编程语言。

libreoffice 包括 python，并打算用 python 替换 java。自 2013 年 2 月 7 日版本 4.0 以来，其 python 脚本提供程序是一个核心功能[150]。

受 python 影响的语言

python 的设计和理念影响了许多其他编程语言:

- boo 使用缩进、类似的语法和类似的对象模型。[151]
- 眼镜蛇使用缩进和类似的语法，它的 "鸣谢" 文档首先列出了 python 在影响它的语言中。[152] 然而，眼镜蛇直接支持按合同设计、单元测试和可选的静态类型.[153]
- coffescript 是一种跨编译到 javascript 的编程语言，它具有 Python-inspired 的语法。
- ecmascript 从 python 中借用迭代器和生成器。[154]
- go 是为 "使用 python 这样的动态语言的工作速度"[155] 而设计的，并共享相同的数组语法。
- groovy 的动机是希望将 python 设计理念引入 java。[156]
- 朱莉娅被设计为 "与真正的宏[..。并] 可作为 python [和] 的常规编程可用 "，速度应与 c 一样快"。[22] 电话对或从朱莉娅是可能的;与 py 卡尔. jl 和 python 包 pyulia 允许调用，在另一个方向，从 python。
- kotlin 是一种函数式编程语言，具有类似于 python 的交互式外壳。但是，kotlin 是强类型的，可以访问标准 java 库。[157]
- ruby 的创作者松本幸弘曾说过: "我想要一种比 perl 更强大、比 python 更面向对象的脚本语言。这就是为什么我决定设计自己的语言。[158]

- swift 是苹果开发的一种编程语言，它有一些 python-b 起的语法。[159]
- gdscript，用于创建视频游戏的动态类型化编程语言。它与 python 极为相似，有一些细微的差异。

python 的开发实践也被其他语言效仿。例如，tcl[160] 和 erlang 也采用了要求提供一个文档来描述语言更改的原理和相关问题的做法（在 python 中，pep 中）。[161]

python 在 2007 年和 2010 年获得 tiobe 年度编程语言奖。该奖项是以 tiobe 指数衡量的，颁发给一年来人气增长最快的语言。[162]

另请参见

引用

1. ^ ᵃ ᵇ *guttag, john v. (2016-08-12)。使用 python 进行计算和编程的简介：用于了解数据。麻省理工出版社。国际标准书号 970-0-262-52962-4.*
2. *deily, ned (2018 年 10 月 20 日)。"python 3.7.1 和 3.6.7 现在可用"。python 内幕.python 核心开发人员。2018 年 10 月 20 日检索.*

3. *benjamin peterson (2018 年 5 月 1 日)。"python 2.7.15 发布"。python 内幕.python 核心开发人员。检索 2018 年 5 月 1 日.*
4. *"pep 483----类型提示理论"。python. org.*
5. 文件扩展名. pyo 在 python 3.5 中被删除。见 pep 0488
6. *moore holth (2014 年 3 月 30 日)。"pep 0441--改进 python zip 应用程序支持"的影响。检索 2015 年 11 月 12 日.*
7. ^ ᵃ ᵇ *"为什么 python 一开始就被创造出来了？"一般 python 常见问题。python 软件基金会。2007 年 3 月 22 日检索.*
8. *kkuchling, andrew m. (2006 年 12 月 22 日)。"采访 guido van rossum (1998 年 7 月)"。2007 年 5 月 1 日从原件中存档。2012 年 3 月 12 日检索.*
9. ^ ᵃ ᵇ *"迭代工具–为高效循环创建迭代器的函数–python 3.7.1 文档"。docs.python.org.*
10. *van rossum, guido (1993 年)。"unixsc 程序员的 python 简介"。nlug najaarsconferentie 的论文集（荷兰 unix 用户组)。尽管 c 的设计远不理想，但它对 python 的影响是相当大的.* 行馈送字符在|last=在位置 4 (帮助)

11. ^ ᵃ ᵇ "课程"。python 教程.python 软件基金会。检索 2012 年 2 月 20 日。它是 c++ 和 modoula-3 中的类机制的混合体
12. 伦德，弗雷德里克"按对象调用"。埃伯博阿克的影响。检索 2017 年 11 月 21 日。将 "clu" 替换为 "python"，将 "记录" 替换为 "实例"，将 "过程" 替换为 "函数或方法"，您将获得对 python 对象模型的相当准确的描述.
13. 西米尼亚托，米歇尔。"python 2.3 方法解析顺序"。python 软件基金会。c3 方法本身与 python 无关，因为它是由在迪伦工作的人发明的，在一篇针对 lispers 的论文中进行了描述
14. 库奇林, a. m. "功能编程 howto"。python v2.7.2 文档。python 软件基金会。检索 2012 年 2 月 9 日.
15. 架构，尼尔;彼得斯，蒂姆;hetland, magnus lie (2001 年 5 月 18 日)。"pep 255—简单的发电机"。python 增强方案。python 软件基金会。检索 2012 年 2 月 9 日.
16. 史密斯，凯文 d.;jewett, jim j.;蒙塔纳罗，斯基普;baxter, anthony (2004 年 9 月 2 日)。"pep 318—功能和方法的装饰器"。python 增强方案。python 软件基金会。检索 2012 年 2 月 24 日.
17. "更多控制流程工具"。python 3 文档。python 软件基金会。检索 2015 年 7 月 24 日.

18. *"coffescript 借用 python 链接的比较".*

19. *"精灵语言——————————————————————的影响。检索 2015 年 12 月 28 日.*

20. *"perl 和 python 在 javascript 中的影响"。www.2ality.com.检索 2015 年 5 月 15 日.*

21. *劳施迈尔，阿克塞尔。第 3 章: javascript 的本质;影响"。奥赖利，说 javascript 的影响。检索 2015 年 5 月 15 日.*

22. ^ ᵃ ᵇ *"为什么我们创造了朱莉娅"朱莉娅网站 2012 年 2 月。检索 2014 年 6 月 5 日.*

23. *环队（2017 年 12 月 4 日）。"戒指和其他语言"。灵兰.网.*

24. *比尼, ola (2007 年)。rails web 2.0 项目实用的 jruby: 将 rails 上的 ruby 引入 java 平台。伯克利: apreses。第 3 页。国际标准书号 978-1-599059-88-8.*

25. *lattner, chris (2014 年 6 月 3 日)。"克里斯·拉特纳的主页"。克里斯·拉特纳检索 2014 年 6 月 3 日。swift 语言是语言专家、文档专家、编译器优化忍者团队和一个非常重要的内部基础思维小组不懈努力的产物，他们提供反馈以帮助完善和测试想法。当然，它也从该领域许多其他语言来之不易的经验中获益匪浅，这些语言*

借鉴了 objecjecal-c、rust"、haskell、ruby、python、c#、clu 等太多语言的创意.

26. 库尔曼，戴夫"python 书：开始 python、高级 python 和 python 练习"。2012 年 6 月 23 日从原件中存档。

27. "吉多范罗苏姆从毕顿的仁慈的生活独裁者的角色中走下来 linux 杂志 "。www.linuxjournal.com.

28. "python 老板 guido van rossum 在 30 年后下台 ".http://www.theinquirer.net，我的时间，我的网站中的外部链接|website=（帮助）

29. "关于 python".python 软件基金会。检索 2012 年 4 月 24 日.，第二部分 "python 的粉丝使用短语" 包含电池 "来描述标准库，它涵盖了从异步处理到 zip 文件的所有内容。

30. "历史和许可"的影响。检索 2016 年 12 月 5 日."所有 python 版本都是开源的"

31. ^ ᵃ ᵇ《文纳斯法案》（2003 年 1 月 13 日）。"巨蟒的制作 "。阿米玛开发者。阿米玛 2007 年 3 月 22 日检索.

32. van rossum, guido(2000 年 8 月 29 日）。"setl（是：温情关于范围文字）"。python-deve(邮寄名单)的影响。2011 年 3 月 13 日检索.

33. van rossum, guido (2009 年 1 月 20 日)。"python 的简短时间"。python 的历史。谷歌。2009 年 1 月 20 日检索.
34. fairchild, carlie (2018 年 7 月 12 日)。"guido van rossum 从 python 的仁慈的生活独裁者的角色中走了下来"。linux 杂志。2018 年 7 月 13 日检索.
35. 库奇林, a. m.;zadka, moshe (2000 年 10 月 16 日)。"python 2.0 中的新增功能"。python 软件基金会。检索 2012 年 2 月 11 日.
36. "python 3.0 版本"。python 软件基金会。检索 2009 年 7 月 8 日.
37. van rossum, guido (2006 年 4 月 5 日)。"pep 3000—python 3000"。python 增强方案。python 软件基金会。2009 年 6 月 27 日检索.
38. "自动 python 2 到 3 代码翻译–python 文档"的影响。检索 2018 年 2 月 11 日.
39. "pep 373--python 2.7 发布时间表"。python. org 的影响。检索 2017 年 1 月 9 日.
40. "pep 466--python 2.7. x 的网络安全增强功能"。python. org 的影响。检索 2017 年 1 月 9 日.
41. "谷歌开源博客: 脾气暴躁: 去运行 python!"2017 年 1 月 4 日。检索 2017 年 3 月 7 日.

42. 该隐刚有限公司"python Metaclasses: 谁？为什么？什么时候？"(pdf)。2009年12月10日从原件(pdf)存档。2009年6月27日检索.
43. "3.3。特殊方法名称"。python 语言参考。python 软件基金会。2009年6月27日检索.
44. "pydbc: python 的方法前置点、方法后置条件和类不变性"的影响。2011年9月24日检索.
45. "python 合同"的影响。2011年9月24日检索.
46. "分析数据库"的影响。2012年7月22日检索.
47. ^ ᵃ ᵇ hettinger, raymond (2002年1月30日)。"pep 289—生成器表达式"。python 增强方案。python 软件基金会。检索2012年2月19日.
48. "6.5 迭代工具—创建用于高效循环的迭代器的函数"。Docs.python.org，我的时间，我的检索2016年11月22日.
49. ^ ᵃ ᵇ peters, tim (2004年8月19日)。"pep 20— python 的禅宗"。python 增强方案。python 软件基金会。检索2008年11月24日.
50. martelli, alex;Ravenscroft, anna;ascher, david (2005年)。python 食谱，第2版。奥赖利媒体。第230页。国际标准书号978-0-596-00797-3.
51. "python 文化".

52. "一般 python 常见问题"。python v2.7.3 文档。Docs.python.org，我的时间，我的检索 2012 年 12 月 3 日.
53. "15 种方法 python 是网络上的强大力量".
54. "打印-数据漂亮的打印机-python 文档".
55. 古德杰，大卫。"像巨蟒一样的代码：习语巨蟒".
56. "如何像毕托塔那样思考".
57. "python 是初程序员的好语言吗？"一般 python 常见问题。python 软件基金会。检索 2007 年 3 月 21 日.
58. "关于 python 中的缩进的神话"。塞克尼蒂克斯。2011 年 4 月 19 日检索.
59. "python 2.5 发布"。python. org.
60. "亮点：python 2.5"。python. org.
61. sweigart，al (2010 年)。"附录 a：python 2 和 3 之间的区别"。使用 python 创建您自己的计算机游戏(2 版)。国际标准书号 978-0-9821060-1-3 的影响。2014 年 2 月 20 日检索.
62. van rossum，guido (2009 年 4 月 22 日)。"尾递归消除"。Neopythonic.blogspot.be，我的时间，我的检索 2012 年 12 月 3 日.

63. van rossum, guido (2006 年 2 月 9 日)。"语言设计不仅仅是解决难题"。阿米玛论坛。阿米玛检索 2007 年 3 月 21 日.
64. van rossum, guido;eby, phillip j. (2005 年 5 月 10 日)。"pep 342—通过增强型发电机的 Coroutines"。python 增强方案。python 软件基金会。检索 2012 年 2 月 19 日.
65. "pep 380"。python. org。检索 2012 年 12 月 3 日.
66. "分裂"。python. org.
67. "pep 0465--矩阵乘法专用内缀运算符"。python. org 的影响。检索 2016 年 1 月 1 日.
68. "python 3.5.1 发布和更改日志"。python. org 的影响。检索 2016 年 1 月 1 日.
69. "" 第 15 章。表达—15.21.1。数字平等操作员 == 和!="。甲骨文公司的影响。检索 2016 年 8 月 28 日.
70. "" 第 15 章。表达—15.21.3。参考平等经营者 == 和!="。甲骨文公司。检索 2016 年 8 月 28 日.
71. van rossum, guido;hettinger, raymond (2003 年 2 月 7 日)。"pep 308—条件表达式"。python 增强方案。python 软件基金会。2011 年 7 月 13 日检索.
72. "4。内置类型—python 3.6.3rc1 文档 ".python. org 的影响。检索 2017 年 10 月 1 日.

73. "5.3。元组和序列–python 3.7.1rc2 文档 ".python.org 的影响。检索 2018 年 10 月 17 日.
74. ^ ᵃ ᵇ "pep 498–––文字字符串插值"。python. org 的影响。检索 2017 年 3 月 8 日.
75. "为什么必须在方法定义和调用中显式使用 ' self '? " 设计和历史常见问题.python 软件基金会。检索 2012 年 2 月 19 日.
76. "python 语言参考，第 3.3 节。新风格和经典的类，发行 2.7.1 "。2011 年 1 月 12 日检索.
77. "python 的类型提示"。lwn. net. 2014 年 12 月 24 日。检索 2015 年 5 月 5 日.
78. "mypy–python 的可选静态键入 "的影响。检索 2017 年 1 月 28 日.
79. zadka, moshe;van rossum, guido (2001年3月11日)。"pep 237 —统一长整数和整数"。python 增强方案。python 软件基金会。2011 年 9 月 24 日检索.
80. "pep 465––矩阵乘法专用内缀运算符"。python. org.
81. "python 中的倾斜运算符–堆栈溢出"。斯塔克维流.com.
82. "比特威斯运营商–python wiki"。wiki.python.org.

83. zadka, moshe;van rossum, guido (2001年3月11日)。"pep 238 —更改部门运营商"。python 增强方案。python 软件基金会。检索 2013 年 10 月 23 日.

84. "为什么 python 的整数分区楼层"的影响。2010 年 8 月 25 日检索.

85. "轮", python 标准库，版本 2.7, §2: 内置功能，检索 2011 年 8 月 14 日

86. "轮", python 标准库，第 3.2 版，第 2 版: 内置函数, 2011 年 8 月 14 日检索

87. beazley, david m. (2009 年)。python 基本参考(第 4 版)。第 66 页。

88. kernighan, brian w.;ritchie, dennis m. (1988 年)。 c 编程语言(第 2 版)。第 206 页。

89. "内置类型"。docs.python.org.

90. 巴蒂斯塔，法孔多"pep 0327--十进制数据类型"。python. org 的影响。检索 2015 年 9 月 26 日.

91. "python 2.6 中的新增功能 –python v2.6.9 文档".docs.python.org 的影响。检索 2015 年 9 月 26 日.

92. piotrowski, przemyslaw (2006 年 7 月)。"为 python 服务器页面和 oracle 构建快速 web 开发环境"。甲骨文技术网络。甲骨文 2012 年 3 月 12 日检索.

93. 巴蒂斯塔，法孔多（2003 年 10 月 17 日）。"pep 327—十进制数据类型"。python 增强方案。python 软件基金会。检索 2008 年 11 月 24 日.

94. eby, phillip j. (2003 年 12 月 7 日)。"pep 33— python web 服务器网关接口 v1.0"。python 增强方案。python 软件基金会。检索 2012 年 2 月 19 日.

95. 德比尔，艾瑞克"模块计数"。模块计数的影响。检索 2017 年 9 月 20 日.

96. "20 + python 网络刮的例子（美丽的汤 & amp; 硒)–像极客"。2017 年 12 月 5 日.2018 年 3 月 12 日检索.

97. 想，天篷。"天篷"。www.enthought.com 的影响。2016 年 8 月 20 日检索.

98. van rossum, guido (2001 年 6 月 5 日)。"pep 7— c 代码的样式指南"。python 增强方案。python 软件基金会。检索 2008 年 11 月 24 日.

99. "cbython 字节代码"。Docs.python.org，我的时间，我的检索 2016 年 2 月 16 日.

100. "python 2.5 内部"(pdf)的影响。2011 年 4 月 19 日检索.

101. "与吉多·范·罗斯姆的访谈"。orreilly. com。检索 2008 年 11 月 24 日.

102. "pypy 兼容性"。ypy. org。检索 2012 年 12 月 3 日.

103. "cpython 和 Pypy 之间的速度比较"。Speed.pypy.org，我的时间，我的检索 2012 年 12 月 3 日.

104. "应用程序级无堆叠功能–pypy 2.0.2 文档".Doc.pypy.org，我的时间，我的检索 2013 年 7 月 17 日.

105. "优化 python 的计划"。谷歌项目托管。谷歌。2009 年 12 月 15 日。2011 年 9 月 24 日检索.

106. "诺基亚 n900 上的 python".随机几何.

107. "努伊特卡之家 nuitka home"。nuitka. net 的影响。检索 2017 年 8 月 18 日.

108. murri, riccardo (2013 年)。python 运行时间在非数字科学代码上的性能。欧洲科学巨蟒会议 (euroscipy) 。arxiv/1404. 6388 。bibcode:2014arxiv1404.6388m.

109. ^ ᵃ ᵇ 华沙，巴里;海尔顿，杰里米;goodger，david (2000 年 6 月 13 日)。"pep 1— pep 宗旨和准则"。python 增强方案。python 软件基金会。2011 年 4 月 19 日检索.

110. 卡农，布雷特。"guido，一些人，和一个邮件列表: python 是如何开发的"。python. org. python 软件基金会。2009 年 6 月 1 日原稿存档 。2009 年 6 月 27 日检索.

111. "python 开发人员指南".

112. norwitz, neal (2002 年 4 月 8 日)。"[python-dev] 发布计划（是稳定性和变化）"的影响。2009 年 6 月 27 日检索.

113. aahz;baxter, anthony (2001年3月15日)。"pep 6—错误修复版本"。python 增强方案。python 软件基金会。2009 年 6 月 27 日检索.

114. "python 生成机器人"。python 开发人员指南。python 软件基金会。2011 年 9 月 24 日检索.

115. 德比尔，艾瑞克"模块计数"。www.modulecounts.com 的影响。2016 年 8 月 20 日检索.

116. ^ ᵃ ᵇ "你的胃口"python 教程.python 软件基金会。检索 2012 年 2 月 20 日.

117. "在 python 中，我应该在 if 块中返回后使用其他方法吗？"堆栈溢出。堆栈交换。2011 年 2 月 17 日。检索 2011 年 5 月 6 日.

118. lutz, mark (2009 年)。学习 python: 强大的面向对象编程。奥赖利传媒公司第 17 页。国际标准书号 9781449379322.

119. 费莉，克里斯（2002）。python。桃坑出版社。十五. p。国际标准书号 9780201748840.

120. "tiobe 指数"。tiobe–软件质量公司。检索 2017 年 3 月 7 日.

121. tiobe 软件索引（2015 年）。"tiobe 编程社区索引 python"的影响。检索 2015 年 9 月 10 日.

122. prechelt, lutz (2000 年 3 月 14 日)。"c、c++、java、perl、python、rexx 和 tcl 的经验比较"(pdf)的影响。检索 2013 年 8 月 30 日.

123. "关于 python 的语录"。python 软件基金会。2012 年 1 月 8 日检索.

124. "使用 python 的组织"。python 软件基金会。2009 年 1 月 15 日检索.

125. "python: 编程的圣杯"。欧洲核子研究中心公报。欧洲核研究组织出版物（31/2006 年）。2006 年 7 月 31 日。检索 2012 年 2 月 11 日.

126. shafer, daniel g. (2003 年 1 月 17 日)。"python 简化航天飞机飞行任务设计"。python 软件基金会。检索 2008 年 11 月 24 日.

127. "龙卷风: facebook 的 python 实时 web 框架–开发人员的 facebook"。面向开发者的脸书的影响。检索 2018–06–19 .

128. "我们如何在 spotify 使用 python"。spotify 实验室的影响。检索 2018–07–25 .

129. fortenberry, tim (2003 年 1 月 17 日)。"工业光和魔法运行在 python 上"。python 软件基金会。检索 2012 年 2 月 11 日.

130. taft, darryl k. (2007 年 3 月 5 日)。"python 滑入系统"。e 周. com. ziff davis holdings。2011 年 9 月 24 日检索.

131. "网站 python 的使用情况统计和市场份额"。2012 年。检索 2012 年 12 月 18 日.

132. oliphant, travis (2007 年)。"科学计算的 python". 科学与工程中的计算.

133. 米尔曼, k. 杰罗德;Aivazis, michael (2011 年)。"科学家和工程师的 python"。科学与工程领域的计算。**13** (2): 9—12。

134. "windows gimp 的安装程序-常见问题"。2013 年 7 月 26 日。2013 年 7 月 17 日从原件中存档。检索 2013 年 7 月 26 日.

135. "jasc psp9 组件"。2008 年 3 月 19 日原稿存档.

136. "关于开始编写地理处理脚本"。桌面帮助 9.2。环境系统研究所。2006 年 11 月 17 日。检索 2012 年 2 月 11 日.

137. 中共波克贝里 (2010 年 8 月 24 日)。"无支架 python 2.7"。eve 社区发展博客。中共游戏。您可能知

道，eve 的核心是被称为 stackless python 的编程语言.

138. caudill, barry (2005 年 9 月 20 日)。《塑造希德·梅耶尔的文明四》。希德·梅耶尔的文明 iv 开发者博客。第一游戏。2010 年 8 月 11 日从原件中存档。我们创造了三个层次的工具...。下一个级别提供 python 和 xml 支持，让具有更多经验的调制解调器操作游戏世界和其中的一切.

139. "python 语言指南（v1.0)"。谷歌文档列表数据 api v1.0。谷歌。2010 年 8 月 11 日从原件中存档。

140. "python–算法交易系统的最佳编程语言"。2016 年 3 月 9 日。检索 2016 年 10 月 3 日.

141. "使用 python 与互动经纪商进行交易：ibp 教程"。2016 年 9 月 19 日。检索 2016 年 10 月 3 日.

142. 院长，杰夫;mona, ra 枯 at;等人（2015 年 11 月 9 日)。"张力流：异构系统上的大规模机器学习"(pdf)。entsorfloww. org. google research。检索 2015 年 11 月 10 日.

143. 皮亚泰茨基，格雷戈里"python 蚕食 r：2018 年用于分析、数据科学、机器学习的顶级软件：趋势和分析"。克丁根格克丁根格 2018 年 5 月 30 日检索.

144. "谁在用科学学习？ -- 科学学习 0.20.1 文档 ".scikit-reng. org.

145. 朱皮，诺姆"谷歌使用 tpu 自定义芯片为机器学习任务增压"。谷歌云平台博客的影响。检索 2016 年 5 月 19 日.

146. "自然语言工具包".

147. "豁免：知道你是安全的"。2009 年 2 月 16 日原稿存档.

148. "corelabs 网站".

149. "什么是糖？"糖实验室。检索 2012 年 2 月 11 日.

150. "4.0 新功能和修复程序"。libreoffice. org. 文件基金会。2013 年。检索 2013 年 2 月 25 日.

151. "python 用户的 Gotchas"。boo.codehaus.org. codehaus 基金会。2008 年 12 月 11 日从原件中存档。检索 2008 年 11 月 24 日.

152. 埃斯特布鲁克，查尔斯。"鸣谢"。眼镜蛇语言。2010 年 4 月 7 日检索.

153. 埃斯特布鲁克，查尔斯。"与 python 的比较"。眼镜蛇语言。2010 年 4 月 7 日检索.

154. "建议：迭代器和生成器 [es4 wiki]"。wiki.ecmascript.org. 2007年10月20日从原件中存档。检索 2008 年 11 月 24 日.

155. kincaid, jason (2009 年 11 月 10 日)。"谷歌的 go: 一种新的编程语言, python 满足 c++"。泰克·克伦奇。2010 年 1 月 29 日 .

156. strachan, james (2003 年 8 月 29 日)。"groovy— java 平台的一种新的动态语言的诞生".

157. "使用命令行编译器–kotlin 编程语言"。科特林的影响。2018 年 3 月 12 日检索.

158. "采访红宝石的创造者"。linuxdevcenter. com。检索 2012 年 12 月 3 日.

159. 拉特纳，克里斯(2014 年 6 月 3 日)。"克里斯·拉特纳的主页"。克里斯·拉特纳检索 2014 年 6 月 3 日。2010 年 7 月，我开始学习快速编程语言。我实施了很多基本的语言结构，只有少数人知道它的存在。其他一些（惊人的）人在 2011 年底开始认真地做出贡献，它在 2013 年 7 月成为 apple 开发人员工具组的主要重点 [.] 从 obecjec 色、rust 集、haskell、ruby、python、c#、clu 和太多其他人那里汲取想法.

160. 库普里雷斯，安德烈亚斯;研究员，donal k. (2000 年 9 月 14 日)。"提示 #3: tip 格式"。tcl 开发人员 xchanges。检索 2008 年 11 月 24 日.

161. gustafsson, per;niskanen, raimo (2007 年 1 月 29 日)。"eep 1: eep 宗旨和准则"。erlang.orgorg2011 年 4 月 19 日检索.

162. "2012 年 3 月 tiobe 编程社区指数"。tiobe 软件。2012 年 3 月。2012 年 3 月 25 日检索.

来源

- "人工智能的 python"。2012 年 7 月 19 日，Wiki.python.org。2012 年 11 月 1 日原稿 存档。检索 2012 年 12 月 3 日.
- 潘恩, jocelyn, ed. (2005 年 8 月)。"python 中的 ai"。ai 专家通讯。安齐!的影响。检索 2012 年 2 月 11 日.
- "pyaiml 0.8.5: python 包索引"。Pypi.python.org，我的时间，我的检索 2013 年 7 月 17 日.
- 罗素，斯图尔特 j.peter，norvig(2009 年)。人工智能：一种现代方法(第 3 版)。上马鞍河，新泽西州：普伦蒂斯大厅。第 1062 页。国际标准书号 978-0-13-60252-4 的影响。检索 2012 年 2 月 11 日.

进一步阅读

- downey, allen b. (2012 年 5 月)。思考 python：如何像计算机科学家一样思考(版本 1.6.6 ed.)。国际标准书号 978-0-521-72596-5.
- hamilton, naomi (2008 年 8 月 5 日)。"编程语言的 a-z: python".计算机世界。2008 年 12 月 29 日从原件中存档。2010 年 3 月 31 日检索.
- lutz, mark (2013 年)。学习 python(第 5 版)。奥赖利媒体。国际标准书号 978-0--59, 15806-4.
- 朝圣者，马克 (2004)。潜入 python。阿压特国际标准书号 978-1-599059-356-1.
- 朝圣者，马克 (2009 年)。潜入 python 3。阿压特国际标准书号 978-1-4320-2415-0. 原版 2011-10-17 版存档.
- summerfield, mark (2009 年)。python 3 中的编程(第 2 版)。阿迪森 - 韦斯利职业。国际标准书号 978-0-321-68056-3.

外部链接