Python（计算机程序设计语言）编辑

Python是一种跨平台的计算机程序设计语言。 是一个高层次的结合了解释性、编译性、互动性和面向对象的脚本语言。最初被设计用于编写自动化脚本(shell)，随着版本的不断更新和语言新功能的添加，越多被用于独立的、大型项目的开发。

中文名

蟒蛇

外文名

Python

别 名

胶水语言

经典教材

Head First Python; Automate the Boring Stuff with Python

发行时间

1991年

作 者

Guido van Rossum

最新版本

稳定：3x:3.8.3 2x:2.7.18；测试：3.9.0a6 2.x:2.7.18rc1[1]

荣 誉

2017年度编程语言

最近更新时间

2020年04月28日

缩 写

py

语言类型

高级语言

目录

1Python简介及应用领域

2下载Python

3发展历程

4风格

5与MATLAB的对比

6设计定位

7执行

8基本语法

▪控制语句

▪表达式

▪函数

▪对象的方法

▪类型

▪数学运算

9帮助

▪1. 列出模块中的函数

▪2.查看完整的python内置函数清单

▪3. 查看某个函数的文档帮助信息

10CGI

▪服务器

▪程序

▪环境变量

11特点

▪优点

▪缺点

12应用

13工具功能

14标准库

15开发环境

16解释器

17著名应用

18库导入

▪内置库

▪外部库

19学习网站

Python简介及应用领域

编辑

Python是一种解释型脚本语言，可以应用于以下领域：[2]

●

Web 和 Internet开发

●

科学计算和统计

●

人工智能

●

桌面界面开发

●

软件开发

●

后端开发

●

网络爬虫

下载Python

编辑

在您开始之前，你的计算机需要Python，但您可能不需要下载它。首先检查(在同级目录下在命令行窗口输入python)有没有安装Python。如果你看到了一个Python解释器的响应，那么就能在它的显示窗口中得到一个版本号。通常的版本都可以做到Python的向前兼容。

如果您需要安装， 您不妨下载最近稳定的版本。 就是那个以没有被标记作为alpha或Beta发行的最高的版本。目前最稳定的版本是Python3.0以上[3]

如果你使用的操作系统是Windows：当前最稳定的Windows版本下载是"Python 3.8.3 for Windows"

如果你使用的是Mac，MacOS 10.2 (Jaguar), 10.3 (Panther) and 10.4 (Tiger)已经集成安装了Python，但是你大概需要安装最近通用的构架(build)。

对于Red Hat，安装python2和python2-devel包。

对于Debian，安装python2.5和python2.5-dev包。

发展历程

编辑

自从20世纪90年代初Python语言诞生至今，它已被逐渐广泛应用于系统管理任务的处理和Web编程。

Python的创始人为荷兰人吉多·范罗苏姆[4] （Guido van Rossum）。1989年圣诞节期间，在阿姆斯特丹，Guido为了打发圣诞节的无趣，决心开发一个新的脚本解释程序，作为ABC 语言的一种继承。之所以选中Python（大蟒蛇的意思）作为该编程语言的名字，是取自英国20世纪70年代首播的电视喜剧《蒙提.派森的飞行马戏团》（Monty Python's Flying Circus）。

ABC是由Guido参加设计的一种教学语言。就Guido本人看来，ABC 这种语言非常优美和强大，是专门为非专业程序员设计的。但是ABC语言并没有成功，究其原因，Guido 认为是其非开放造成的。Guido 决心在Python 中避免这一错误。同时，他还想实现在ABC 中闪现过但未曾实现的东西。

就这样，Python在Guido手中诞生了。可以说，Python是从ABC发展起来，主要受到了Modula-3（另一种相当优美且强大的语言，为小型团体所设计的）的影响。并且结合了Unix shell和C的习惯。

Python[5] 已经成为最受欢迎的程序设计语言之一。自从2004年以后，python的使用率呈线性增长。Python 2于2000年10月16日发布，稳定版本是Python 2.7。Python 3于2008年12月3日发布，不完全兼容Python 2。[4] 2011年1月，它被TIOBE编程语言排行榜评为2010年度语言。[6]

标识

由于Python语言的简洁性、易读性以及可扩展性，在国外用Python做科学计算的研究机构日益增多，一些知名大学已经采用Python来教授程序设计课程。例如卡耐基梅隆大学的编程基础、麻省理工学院的计算机科学及编程导论就使用Python语言讲授。众多开源的科学计算软件包都提供了Python的调用接口，例如著名的计算机视觉库OpenCV、三维可视化库VTK、医学图像处理库ITK。而Python专用的科学计算扩展库就更多了，例如如下3个十分经典的科学计算扩展库：NumPy、SciPy和matplotlib，它们分别为Python提供了快速数组处理、数值运算以及绘图功能。因此Python语言及其众多的扩展库所构成的开发环境十分适合工程技术、科研人员处理实验数据、制作图表，甚至开发科学计算应用程序。2018年3月，该语言作者在邮件列表上宣布Python 2.7将于2020年1月1日终止支持。用户如果想要在这个日期之后继续得到与Python 2.7有关的支持，则需要付费给商业供应商。[7]

风格

编辑

Python在设计上坚持了清晰划一的风格，这使得Python成为一门易读、易维护，并且被大量用户所欢迎的、用途广泛的语言。

设计者开发时总的指导思想是，对于一个特定的问题，只要有一种最好的方法来解决就好了。这在由Tim Peters写的Python格言（称为The Zen of Python）里面表述为：There should be one-- and preferably only one --obvious way to do it. 这正好和Perl语言（另一种功能类似的高级动态语言）的中心思想TMTOWTDI（There's More Than One Way To Do It）完全相反。

Python的作者有意的设计限制性很强的语法，使得不好的编程习惯（例如if语句的下一行不向右缩进）都不能通过编译。其中很重要的一项就是Python的缩进规则。

一个和其他大多数语言（如C）的区别就是，一个模块的界限，完全是由每行的首字符在这一行的位置来决定的（而C语言是用一对花括号{}来明确的定出模块的边界的，与字符的位置毫无关系）。这一点曾经引起过争议。因为自从C这类的语言诞生后，语言的语法含义与字符的排列方式分离开来，曾经被认为是一种程序语言的进步。不过不可否认的是，通过强制程序员们缩进（包括if，for和函数定义等所有需要使用模块的地方），Python确实使得程序更加清晰和美观。

与MATLAB的对比

编辑

说起科学计算，首先会被提到的可能是MATLAB。然而除了MATLAB的一些专业性很强的工具箱还无法被替代之外，MATLAB的大部分常用功能都可以在Python世界中找到相应的扩展库。和MATLAB相比，用Python做科学计算有如下优点：

● 首先，MATLAB是一款商用软件，并且价格不菲。而Python完全免费，众多开源的科学计算库都提供了Python的调用接口。用户可以在任何计算机上免费安装Python及其绝大多数扩展库。

● 其次，与MATLAB相比，Python是一门更易学、更严谨的程序设计语言。它能让用户编写出更易读、易维护的代码。

● 最后，MATLAB主要专注于工程和科学计算。然而即使在计算领域，也经常会遇到文件管理、界面设计、网络通信等各种需求。而Python有着丰富的扩展库，可以轻易完成各种高级任务，开发者可以用Python实现完整应用程序所需的各种功能。

设计定位

编辑

Python的设计哲学是“优雅”、“明确”、“简单”。因此，Perl语言中“总是有多种方法来做同一件事”的理念在Python开发者中通常是难以忍受的。Python开发者的哲学是“用一种方法，最好是只有一种方法来做一件事”。在设计Python语言时，如果面临多种选择，Python开发者一般会拒绝花俏的语法，而选择明确的没有或者很少有歧义的语法。由于这种设计观念的差异，Python源代码通常被认为比Perl具备更好的可读性，并且能够支撑大规模的软件开发。这些准则被称为Python格言。在Python解释器内运行import this可以获得完整的列表。

Python开发人员尽量避开不成熟或者不重要的优化。一些针对非重要部位的加快运行速度的补丁通常不会被合并到Python内。所以很多人认为Python很慢。不过，根据二八定律，大多数程序对速度要求不高。在某些对运行速度要求很高的情况，Python设计师倾向于使用JIT技术，或者用使用C/C++语言改写这部分程序。可用的JIT技术是PyPy。

Python是完全面向对象的语言。函数、模块、数字、字符串都是对象。并且完全支持继承、重载、派生、多继承，有益于增强源代码的复用性。Python支持重载运算符和动态类型。相对于Lisp这种传统的函数式编程语言，Python对函数式设计只提供了有限的支持。有两个标准库(functools, itertools)提供了Haskell和Standard ML中久经考验的函数式程序设计工具。

虽然Python可能被粗略地分类为“脚本语言”（script language），但实际上一些大规模软件开发计划例如Zope、Mnet及BitTorrent，Google也广泛地使用它。Python的支持者较喜欢称它为一种高级动态编程语言，原因是“脚本语言”泛指仅作简单程序设计任务的语言，如shellscript、VBScript等只能处理简单任务的编程语言，并不能与Python相提并论。

Python本身被设计为可扩充的。并非所有的特性和功能都集成到语言核心。Python提供了丰富的API和工具，以便程序员能够轻松地使用C语言、C++、Cython来编写扩充模块。Python编译器本身也可以被集成到其它需要脚本语言的程序内。因此，很多人还把Python作为一种“胶水语言”（glue language）使用。使用Python将其他语言编写的程序进行集成和封装。在Google内部的很多项目，例如Google Engine使用C++编写性能要求极高的部分，然后用Python或Java/Go调用相应的模块。《Python技术手册》的作者马特利（Alex Martelli）说：“这很难讲，不过，2004 年，Python 已在Google 内部使用，Google 召募许多 Python 高手，但在这之前就已决定使用Python，他们的目的是 Python where we can, C++ where we must，在操控硬件的场合使用C++，在快速开发时候使用 Python。”

执行

编辑

Python在执行时，首先会将.py文件中的源代码编译成Python的byte code（字节码），然后再由Python Virtual Machine（Python虚拟机）来执行这些编译好的byte code。这种机制的基本思想跟Java，.NET是一致的。然而，Python Virtual Machine与Java或.NET的Virtual Machine不同的是，Python的Virtual Machine是一种更高级的Virtual Machine。这里的高级并不是通常意义上的高级，不是说Python的Virtual Machine比Java或.NET的功能更强大，而是说和Java 或.NET相比，Python的Virtual Machine距离真实机器的距离更远。或者可以这么说，Python的Virtual Machine是一种抽象层次更高的Virtual Machine。基于C的Python编译出的字节码文件，通常是.pyc格式。除此之外，Python还可以以交互模式运行，比如主流操作系统Unix/Linux、Mac、Windows都可以直接在命令模式下直接运行Python交互环境。直接下达操作指令即可实现交互操作。

基本语法

编辑

Python的设计目标之一是让代码具备高度的可阅读性。它设计时尽量使用其它语言经常使用的标点符号和英文单字，让代码看起来整洁美观。它不像其他的静态语言如C、Pascal那样需要重复书写声明语句，也不像它们的语法那样经常有特殊情况和意外。

Python开发者有意让违反了缩进规则的程序不能通过编译，以此来强制程序员养成良好的编程习惯。并且Python语言利用缩进表示语句块的开始和退出（Off-side规则），而非使用花括号或者某种关键字。增加缩进表示语句块的开始，而减少缩进则表示语句块的退出。缩进成为了语法的一部分。例如if语句：python3

1

2

3

4

5 age = int(input("请输入你的年龄: "))

if age < 21:

print("你不能买酒。")

print("不过你能买口香糖。")

print("这句话在if语句块的外面。")

根据PEP的规定，必须使用4个空格来表示每级缩进（不清楚4个空格的规定如何，在实际编写中可以自定义空格数，但是要满足每级缩进间空格数相等）。使用Tab字符和其它数目的空格虽然都可以编译通过，但不符合编码规范。支持Tab字符和其它数目的空格仅仅是为兼容很旧的的Python程序和某些有问题的编辑程序。

控制语句

if语句，当条件成立时运行语句块。经常与else, elif(相当于else if) 配合使用。

for语句，遍历列表、字符串、字典、集合等迭代器，依次处理迭代器中的每个元素。

while语句，当条件为真时，循环运行语句块。

try语句，与except,finally配合使用处理在程序运行中出现的异常情况。

class语句，用于定义类型。

def语句，用于定义函数和类型的方法。

pass语句，表示此行为空，不运行任何操作。

assert语句，用于程序调试阶段时测试运行条件是否满足。

with语句，Python2.6以后定义的语法，在一个场景中运行语句块。比如，运行语句块前加密，然后在语句块运行退出后解密。

yield语句，在迭代器函数内使用，用于返回一个元素。自从Python 2.5版本以后。这个语句变成一个运算符。

raise语句，制造一个错误。

import语句，导入一个模块或包。

from … import语句，从包导入模块或从模块导入某个对象。

import … as语句，将导入的对象赋值给一个变量。

in语句，判断一个对象是否在一个字符串/列表/元组里。

表达式

Python的表达式写法与C/C++类似。只是在某些写法有所差别。

主要的算术运算符与C/C++类似。+, -, \*, /, //, \*\*, ~, %分别表示加法或者取正、减法或者取负、乘法、除法、整除、乘方、取补、取余。>>, <<表示右移和左移。&, |, ^表示二进制的AND, OR, XOR运算。>, <, ==, !=, <=, >=用于比较两个表达式的值，分别表示大于、小于、等于、不等于、小于等于、大于等于。在这些运算符里面，~, |, ^, &, <<, >>必须应用于整数。

Python使用and, or, not表示逻辑运算。

is, is not用于比较两个变量是否是同一个对象。in, not in用于判断一个对象是否属于另外一个对象。

Python支持“列表推导式”（list comprehension），比如计算0-9的平方和:

1

2 >>> sum(x \* x for x in range(10))

285

Python使用lambda表示匿名函数。匿名函数体只能是表达式。比如：

1

2

3 >>> add=lambda x, y : x + y

>>> add(3,2)

5

Python使用y if cond else x表示条件表达式。意思是当cond为真时，表达式的值为y，否则表达式的值为x。相当于C++和Java里的cond?y:x。

Python区分列表(list)和元组(tuple)两种类型。list的写法是[1,2,3]，而tuple的写法是(1,2,3)。可以改变list中的元素，而不能改变tuple。在某些情况下，tuple的括号可以省略。tuple对于赋值语句有特殊的处理。因此，可以同时赋值给多个变量，比如：

1 >>> x, y=1,2 # 同时给x,y赋值，最终结果：x=1, y=2

特别地，可以使用以下这种形式来交换两个变量的值：

1 >>> x, y=y, x #最终结果：y=1, x=2

Python使用'(单引号)和"(双引号)来表示字符串。与Perl、Unix Shell语言或者Ruby、Groovy等语言不一样，两种符号作用相同。一般地，如果字符串中出现了双引号，就使用单引号来表示字符串;反之则使用双引号。如果都没有出现，就依个人喜好选择。出现在字符串中的\(反斜杠)被解释为特殊字符，比如\n表示换行符。表达式前加r指示Python不解释字符串中出现的\。这种写法通常用于编写正则表达式或者Windows文件路径。

Python支持列表切割(list slices)，可以取得完整列表的一部分。支持切割操作的类型有str, bytes, list, tuple等。它的语法是...[left:right]或者...[left:right:stride]。假定nums变量的值是[1, 3, 5, 7, 8, 13, 20]，那么下面几个语句为真：

nums[2:5] == [5, 7, 8] 从下标为2的元素切割到下标为5的元素，但不包含下标为5的元素。

nums[1:] == [3, 5, 7, 8, 13, 20] 切割到最后一个元素。

nums[:-3] == [1, 3, 5, 7] 从最开始的元素一直切割到倒数第3个元素。

nums[:] == [1, 3, 5, 7, 8, 13, 20] 返回所有元素。改变新的列表不会影响到nums。

nums[1:5:2] == [3, 7] 从下标为1的元素切割到下标为5的元素，且步长为2。

函数

Python的函数支持递归、默认参数值、可变参数，但不支持函数重载。为了增强代码的可读性，可以在函数后书写“文档字符串”(Documentation Strings，或者简称docstrings)，用于解释函数的作用、参数的类型与意义、返回值类型与取值范围等。可以使用内置函数help()打印出函数的使用帮助。比如：

>>> def randint(a, b):

... "Return random integer in range [a, b], including both end points."...

>>> help(randint)

Help on function randint in module \_\_main\_\_:

randint(a, b)

Return random integer inrange[a, b], including both end points.

对象的方法

对象的方法是指绑定到对象的函数。调用对象方法的语法是instance.method(arguments)。它等价于调用Class.method(instance, arguments)。当定义对象方法时，必须显式地定义第一个参数，一般该参数名都使用self，用于访问对象的内部数据。这里的self相当于C++, Java里面的this变量，但是我们还可以使用任何其它合法的参数名，比如this 和 mine 等，self与C++,Java里面的this不完全一样，它可以被看作是一个习惯性的用法，我们传入任何其它的合法名称都行，比如：

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16 class Fish:

def eat(self,food):

if food is not None:

self.hungry=False

class User:

def \_\_init\_\_(myself,name):

myself.name=name

#构造Fish的实例：

f=Fish()

#以下两种调用形式是等价的：

Fish.eat(f,"earthworm")

f.eat("earthworm")

u=User('username')

print(u.name)

Python认识一些以“\_\_”开始并以“\_\_”结束的特殊方法名，它们用于实现运算符重载和实现多种特殊功能。

类型

Python采用动态类型系统。在编译的时候，Python不会检查对象是否拥有被调用的方法或者属性，而是直至运行时，才做出检查。所以操作对象时可能会抛出异常。不过，虽然Python采用动态类型系统，它同时也是强类型的。Python禁止没有明确定义的操作，比如数字加字符串。

与其它面向对象语言一样，Python允许程序员定义类型。构造一个对象只需要像函数一样调用类型即可，比如，对于前面定义的Fish类型，使用Fish()。类型本身也是特殊类型type的对象(type类型本身也是type对象)，这种特殊的设计允许对类型进行反射编程。

Python内置丰富的数据类型。与Java、C++相比，这些数据类型有效地减少代码的长度。下面这个列表简要地描述了Python内置数据类型(适用于Python 3.x)：

类型 描述 例子 备注

str（string/字符串） 一个由字符组成的不可更改的有序串行。 'Wikipedia'

"Wikipedia"

"""Spanning

multiple

lines""" 在Python 3.x里，字符串由Unicode字符组成

bytes（字节） 一个由字节组成的不可更改的有序串行。 b'Some ASCII'

b"Some ASCII" 在Python 2.x里，bytes为str的一种

list（列表） 可以包含多种类型的可改变的有序串行 [4.0, 'string', True]

tuple（元组） 可以包含多种类型的不可改变的有序串行 (4.0, 'string', True)

set, frozenset 与数学中集合的概念类似。无序的、每个元素唯一。 {4.0, 'string', True}

frozenset([4.0, 'string', True])

dict（字典） 一个可改变的由键值对组成的无序串行。 {'key1': 1.0, 3: False}

int（整数） 精度不限的整数 42

float（浮点数） 浮点数。精度与系统相关。 3.1415927

complex 复数 3+2.7j

bool 逻辑值。只有两个值：真、假 True

False

builtin\_function\_or\_method 自带的函数，不可更改也不可增加 print

input

type（类型） 显示某个值的类型，用type(x)获得 type(1) ->

type(‘1’) ->

range 按顺序排列的数 range(10)

......list(range(10))->[0,1,2,3,4,5,6,7,8,9]

range(1,10)

......list(range(1,10))->[1,2,3,4,5,6,7,8,9]

range(1,10,2)

......list(range(1,10,2))->[1,3,5,7,9] 在Python 2.x中，range为builtin\_function\_or\_method，获得的数为list

除了各种数据类型，Python语言还用类型来表示函数、模块、类型本身、对象的方法、编译后的Python代码、运行时信息等等。因此，Python具备很强的动态性。

数学运算

Python使用与C、Java类似的运算符，支持整数与浮点数的数学运算。同时还支持复数运算与无穷位数（实际受限于计算机的能力）的整数运算。除了求绝对值函数abs()外，大多数数学函数处于math和cmath模块内。前者用于实数运算，而后者用于复数运算。使用时需要先导入它们，比如：

>>> import math

>>> print(math.sin(math.pi/2))

1.0

fractions模块用于支持分数运算；decimal模块用于支持高精度的浮点数运算。

Python定义求余运行a % b的值处于开区间[0, b)内，如果b是负数，开区间变为(b, 0]。这是一个很常见的定义方式。不过其实它依赖于整除的定义。为了让方程式：b \* (a // b) + a % b = a恒真，整除运行需要向负无穷小方向取值。比如7 // 3的结果是2，而(-7) // 3的结果却是-3。这个算法与其它很多编程语言不一样，需要注意，它们的整除运算会向0的方向取值。

Python允许像数学的常用写法那样连着写两个比较运行符。比如a < b < c与a < b and b < c等价。C++的结果与Python不一样，首先它会先计算a < b，根据两者的大小获得0或者1两个值之一，然后再与c进行比较。

帮助

编辑

1. 列出模块中的函数

用import导入模块后，可使用函数dir(m)列出模块的所有函数，import是导入模块的命令，m是模块名。

例子：

1

2

3 >>>import math

>>>dir(math)

['\_\_doc\_\_', '\_\_loader\_\_', '\_\_name\_\_', '\_\_package\_\_', '\_\_spec\_\_', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan', 'pi', 'pow', 'radians', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']

这个例子列出math模块的一些函数，以双下划线( \_ \_ )开头的名称用于较复杂的python编程。

2.查看完整的python内置函数清单

查看完整的python内置函数清单，可在提示符后输入 >>>dir(\_ \_builtins\_ \_)。

例子：

1

2

3 >>> dir(\_\_builtins\_\_)

['ArithmeticError', 'AssertionError', 'AttributeError', 'BaseException', 'BlockingIOError', 'BrokenPipeError', 'BufferError', 'BytesWarning', 'ChildProcessError', 'ConnectionAbortedError', 'ConnectionError', 'ConnectionRefusedError', 'ConnectionResetError', 'DeprecationWarning', 'EOFError', 'Ellipsis', 'EnvironmentError', 'Exception', 'False', 'FileExistsError', 'FileNotFoundError', 'FloatingPointError', 'FutureWarning', 'GeneratorExit', 'IOError', 'ImportError', 'ImportWarning', 'IndentationError', 'IndexError', 'InterruptedError', 'IsADirectoryError', 'KeyError', 'KeyboardInterrupt', 'LookupError', 'MemoryError', 'ModuleNotFoundError', 'NameError', 'None', 'NotADirectoryError', 'NotImplemented', 'NotImplementedError', 'OSError', 'OverflowError', 'PendingDeprecationWarning', 'PermissionError', 'ProcessLookupError', 'RecursionError', 'ReferenceError', 'ResourceWarning', 'RuntimeError', 'RuntimeWarning', 'StopAsyncIteration', 'StopIteration', 'SyntaxError', 'SyntaxWarning', 'SystemError', 'SystemExit', 'TabError', 'TimeoutError', 'True', 'TypeError', 'UnboundLocalError', 'UnicodeDecodeError', 'UnicodeEncodeError', 'UnicodeError', 'UnicodeTranslateError', 'UnicodeWarning', 'UserWarning', 'ValueError', 'Warning', 'WindowsError', 'ZeroDivisionError', '\_', '\_\_build\_class\_\_', '\_\_debug\_\_', '\_\_doc\_\_', '\_\_import\_\_', '\_\_loader\_\_', '\_\_name\_\_', '\_\_package\_\_', '\_\_spec\_\_', 'abs', 'all', 'any', 'ascii', 'bin', 'bool', 'bytearray', 'bytes', 'callable', 'chr', 'classmethod', 'compile', 'complex', 'copyright', 'credits', 'delattr', 'dict', 'dir', 'divmod', 'enumerate', 'eval', 'exec', 'exit', 'filter', 'float', 'format', 'frozenset', 'getattr', 'globals', 'hasattr', 'hash', 'help', 'hex', 'id', 'input', 'int', 'isinstance', 'issubclass', 'iter', 'len', 'license', 'list', 'locals', 'map', 'max', 'memoryview', 'min', 'next', 'object', 'oct', 'open', 'ord', 'pow', 'print', 'property', 'quit', 'range', 'repr', 'reversed', 'round', 'set', 'setattr', 'slice', 'sorted', 'staticmethod', 'str', 'sum', 'super', 'tuple', 'type', 'vars', 'zip']

3. 查看某个函数的文档帮助信息

可以用函数help(函数)来查看某个函数的文档帮助信息。

例子：

1

2

3

4

5

6

7

8

9 >>>help(sum)

Help on built-in function sum in module builtins:

sum(iterable, start=0, /)

Return the sum of a 'start' value (default: 0) plus an iterable of numbers When the iterable is empty, return the start value.

This function is intended specifically for use with numeric values and may reject non-numeric types.

可以直接在提示符下输入help()，然后输入某个模块或函数名得到详细的帮助信息。

CGI

编辑

CGI 目前由NCSA维护，NCSA定义CGI如下：[5]

CGI(Common Gateway Interface)，通用网关接口，它是一段程序，运行在服务器上如：HTTP服务器，提供同客户端HTML页面的接口。

CGI程序可以是Python脚本、Perl脚本、Shell脚本、C或者C++程序等。

服务器

在你进行CGI编程前，确保您的Web服务器支持CGI及已经配置了CGI的处理程序。

所有的HTTP服务器执行CGI程序都保存在一个预先配置的目录。这个目录被称为CGI目录，并按照惯例，它被命名为/var/www/cgi-bin目录。

CGI文件的扩展名为.cgi，python也可以使用.py扩展名。

默认情况下，Linux服务器配置运行的cgi-bin目录中为/var/www。

如果想指定的其他运行CGI脚本的目录，可以修改httpd.conf配置文件，如下所示：

1

2

3

4

5

6

7

8

9 <Directory"/var/www/cgi-bin">

Allow Override None

Options ExecCGI

Order allow,deny

Allow from all

</Directory>

<Directory"/var/www/cgi-bin">

Options All

</Directory>

程序

使用Python创建第一个CGI程序，文件名为hello.py，文件位于/var/www/cgi-bin目录中，内容如下，修改文件的权限为755：[5]

1

2

3

4

5

6

7

8

9

10 #!/usr/bin/env python

print("Content-type:text/html\r\n\r\n")

print("<html>")

print("<head>")

print("")

print("</head>")

print("<body>")

print("<h2>Hello World! This is my first CGI program</h2>")

print("</body>")

print("</html>")

以上程序在浏览器访问显示结果如下：

1 Hello World! This is my first CGI program

这个的hello.py脚本是一个简单的Python脚本，脚本第一的输出内容"Content-type:text/html\r\n\r\n"发送到浏览器并告知浏览器显示的内容类型为"text/html"。

环境变量

所有的CGI程序都接收以下的环境变量，这些变量在CGI程序中发挥了重要的作用：[5]

变量名 描述

CONTENT\_TYPE 这个环境变量的值指示所传递来的信息的MIME类型。目前，环境变量CONTENT\_TYPE一般都是：application/x-www-form-urlencoded,他表示数据来自于HTML表单。

CONTENT\_LENGTH 如果服务器与CGI程序信息的传递方式是POST，这个环境变量即使从标准输入STDIN中可以读到的有效数据的字节数。这个环境变量在读取所输入的数据时必须使用。

HTTP\_COOKIE 客户机内的 COOKIE 内容。

HTTP\_USER\_AGENT 提供包含了版本数或其他专有数据的客户浏览器信息。

PATH\_INFO 这个环境变量的值表示紧接在CGI程序名之后的其他路径信息。它常常作为CGI程序的参数出现。

QUERY\_STRING 如果服务器与CGI程序信息的传递方式是GET，这个环境变量的值即使所传递的信息。这个信息经跟在CGI程序名的后面，两者中间用一个问号'?'分隔。

REMOTE\_ADDR 这个环境变量的值是发送请求的客户机的IP地址，例如上面的192.168.1.67。这个值总是存在的。而且它是Web客户机需要提供给Web服务器的唯一标识，可以在CGI程序中用它来区分不同的Web客户机。

REMOTE\_HOST 这个环境变量的值包含发送CGI请求的客户机的主机名。如果不支持你想查询，则无需定义此环境变量。

REQUEST\_METHOD 提供脚本被调用的方法。对于使用 HTTP/1.0 协议的脚本，仅 GET 和 POST 有意义。

SCRIPT\_FILENAME CGI脚本的完整路径

SCRIPT\_NAME CGI脚本的的名称

SERVER\_NAME 这是你的 WEB 服务器的主机名、别名或IP地址。

SERVER\_SOFTWARE 这个环境变量的值包含了调用CGI程序的HTTP服务器的名称和版本号。例如，上面的值为Apache/2.2.14(Unix)

以下是一个简单的CGI脚本输出CGI的环境变量：

1

2

3

4

5

6 #!/usr/bin/python

import os

print"Content-type:text/html\r\n\r\n"

print"Environment"

for param in os.environ.keys():

print"<b>%20s</b>:%s<\br>" %(param,os.environ[param])

特点

编辑

优点

简单：Python是一种代表简单主义思想的语言。阅读一个良好的Python程序就感觉像是在读英语一样。它使你能够专注于解决问题而不是去搞明白语言本身。

易学：Python极其容易上手，因为Python有极其简单的说明文档[8] 。

速度快：Python 的底层是用 C 语言写的，很多标准库和第三方库也都是用 C 写的，运行速度非常快。[5]

免费、开源：Python是FLOSS（自由/开放源码软件）之一。使用者可以自由地发布这个软件的拷贝、阅读它的源代码、对它做改动、把它的一部分用于新的自由软件中。FLOSS是基于一个团体分享知识的概念。

高层语言：用Python语言编写程序的时候无需考虑诸如如何管理你的程序使用的内存一类的底层细节。

可移植性：由于它的开源本质，Python已经被移植在许多平台上（经过改动使它能够工作在不同平台上）。这些平台包括Linux、Windows、FreeBSD、Macintosh、Solaris、OS/2、Amiga、AROS、AS/400、BeOS、OS/390、z/OS、Palm OS、QNX、VMS、Psion、Acom RISC OS、VxWorks、PlayStation、Sharp Zaurus、Windows CE、PocketPC、Symbian以及Google基于linux开发的android平台。

解释性：一个用编译性语言比如C或C++写的程序可以从源文件（即C或C++语言）转换到一个你的计算机使用的语言（二进制代码，即0和1）。这个过程通过编译器和不同的标记、选项完成。

运行程序的时候，连接/转载器软件把你的程序从硬盘复制到内存中并且运行。而Python语言写的程序不需要编译成二进制代码。你可以直接从源代码运行 程序。

在计算机内部，Python解释器把源代码转换成称为字节码的中间形式，然后再把它翻译成计算机使用的机器语言并运行。这使得使用Python更加简单。也使得Python程序更加易于移植。

面向对象：Python既支持面向过程的编程也支持面向对象的编程。在“面向过程”的语言中，程序是由过程或仅仅是可重用代码的函数构建起来的。在“面向对象”的语言中，程序是由数据和功能组合而成的对象构建起来的。

可扩展性：如果需要一段关键代码运行得更快或者希望某些算法不公开，可以部分程序用C或C++编写，然后在Python程序中使用它们。

可嵌入性：可以把Python嵌入C/C++程序，从而向程序用户提供脚本功能。

丰富的库：Python标准库确实很庞大。它可以帮助处理各种工作，包括正则表达式、文档生成、单元测试、线程、数据库、网页浏览器、CGI、FTP、电子邮件、XML、XML-RPC、HTML、WAV文件、密码系统、GUI（图形用户界面）、Tk和其他与系统有关的操作。这被称作Python的“功能齐全”理念。除了标准库以外，还有许多其他高质量的库，如wxPython、Twisted和Python图像库等等。

规范的代码：Python采用强制缩进的方式使得代码具有较好可读性。而Python语言写的程序不需要编译成二进制代码。

缺点

单行语句和命令行输出问题：很多时候不能将程序连写成一行，如import sys;for i in sys.path:print i。而perl和awk就无此限制，可以较为方便的在shell下完成简单程序，不需要如Python一样，必须将程序写入一个.py文件。

独特的语法

这也许不应该被称为局限，但是它用缩进来区分语句关系的方式还是给很多初学者带来了困惑。即便是很有经验的Python程序员，也可能陷入陷阱当中。

运行速度慢：这里是指与C和C++相比。

应用

编辑

系统编程：提供API（Application Programming Interface应用程序编程接口），能方便进行系统维护和管理，Linux下标志性语言之一，是很多系统管理员理想的编程工具。

图形处理：有PIL、Tkinter等图形库支持，能方便进行图形处理。

数学处理：NumPy扩展提供大量与许多标准数学库的接口。

文本处理：python提供的re模块能支持正则表达式，还提供SGML，XML分析模块，许多程序员利用python进行XML程序的开发。

数据库编程：程序员可通过遵循Python DB-API（数据库应用程序编程接口）规范的模块与Microsoft SQL Server，Oracle，Sybase，DB2，MySQL、SQLite等数据库通信。python自带有一个Gadfly模块，提供了一个完整的SQL环境。

网络编程：提供丰富的模块支持sockets编程，能方便快速地开发分布式应用程序。很多大规模软件开发计划例如Zope，Mnet 及BitTorrent. Google都在广泛地使用它。

Web编程：应用的开发语言，支持最新的XML技术。

多媒体应用：Python的PyOpenGL模块封装了“OpenGL应用程序编程接口”，能进行二维和三维图像处理。PyGame模块可用于编写游戏软件。

pymo引擎：PYMO全称为python memories off，是一款运行于Symbian S60V3,Symbian3,S60V5, Symbian3, Android系统上的AVG游戏引擎。因其基于python2.0平台开发，并且适用于创建秋之回忆（memories off）风格的AVG游戏，故命名为PYMO。

黑客编程：python有一个hack的库,内置了你熟悉的或不熟悉的函数，但是缺少成就感。

用Python写简单爬虫

首先，要通过urllib2这个Module获得对应的HTML源码。(PS:在python3.3之后urllib2已经不能再用，代之以urllib)

1

2

3

4 import urllib2 #调用urllib2

url='http://www.baidu.com/s?wd=cloga' #把等号右边的网址赋值给url

html=urllib2.urlopen(url).read() #html随意取名 等号后面的动作是打开源代码页面，并阅读

print html #打印

通过上面这三句就可以将URL的源码存在content变量中，其类型为字符型。

接下来是要从这堆HTML源码中提取我们需要的内容。用Chrome查看一下对应的内容的代码（也可以用Firefox的Firebug）。

可以看到url的信息存储在span标签中，要获取其中的信息可以用正则表达式。

工具功能

编辑

Tkinter

Python默认的图形界面接口。Tkinter是一个和Tk接口的Python模块，Tkinter库提供了对Tk API的接口，它属于Tcl/Tk的GUI工具组。

PyGTK

用于python GUI程序开发的GTK+库。GTK就是用来实现GIMP和Gnome的库。

PyQt

用于python的Qt开发库。QT就是实现了KDE环境的那个库，由一系列的模块组成，有qt, qtcanvas, qtgl, qtnetwork, qtsql, qttable, qtui and qtxml，包含有300个类和超过5750个的函数和方法。PyQt还支持一个叫qtext的模块，它包含一个QScintilla库。该库是Scintillar编辑器类的Qt接口。

wxPython

GUI编程框架，熟悉MFC的人会非常喜欢，简直是同一架构（对于初学者或者对设计要求不高的用户来说，使用Boa Constructor可以方便迅速的进行wxPython的开发）属于外置库，要先下载。

PIL

python提供强大的图形处理的能力，并提供广泛的图形文件格式支持，该库能进行图形格式的转换、打印和显示。还能进行一些图形效果的处理，如图形的放大、缩小和旋转等。是Python用户进行图象处理的强有力工具。

Psyco

一个Python代码加速度器，可使Python代码的执行速度提高到与编译语言一样的水平。

xmpppy

Jabber服务器采用开发的XMPP协议，Google Talk也是采用XMPP协议的IM系统。在Python中有一个xmpppy模块支持该协议。也就是说，我们可以通过该模块与Jabber服务器通信，是不是很Cool。

PyMedia

用于多媒体操作的python模块。它提供了丰富而简单的接口用于多媒体处理(wav, mp3, ogg, avi, divx, dvd, cdda etc)。可在Windows和Linux平台下使用。

Pmw

Python megawidgets，Python超级GUI组件集，一个在python中利用Tkinter模块构建的高级GUI组件，每个Pmw都合并了一个或多个Tkinter组件，以实现更有用和更复杂的功能。

PyXML

用Python解析和处理XML文档的工具包，包中的4DOM是完全相容于W3C DOM规范的。它包含以下内容：

xmlproc: 一个符合规范的XML解析器。Expat: 一个快速的，非验证的XML解析器。还有其他和他同级别的还有 PyHtml PySGML。

PyGame

用于多媒体开发和游戏软件开发的模块。可以直接使用pip install pygame下载

PyOpenGL

模块封装了“OpenGL应用程序编程接口”，通过该模块python程序员可在程序中集成2D和3D的图形。

NumPy、NumArray、SAGE

NumArray是Python的一个扩展库，主要用于处理任意维数的固定类型数组，简单说就是一个矩阵库。它的底层代码使用C来编写，所以速度的优势很明显。SAGE是基于NumPy和其他几个工具所整合成的数学软件包，目标是取代Magma, Maple, Mathematica和Matlab 这类工具。

MySQLdb

用于连接MySQL数据库。还有用于zope的ZMySQLDA模块，通过它就可在zope中连接mysql数据库。

Sqlite3

用于连接sqlite数据库。

Python-ldap

提供一组面向对象的API，可方便地在python中访问ldap目录服务，它基于OpenLDAP2.x。

smtplib

发送电子邮件。

ftplib

定义了FTP类和一些方法，用以进行客户端的ftp编程。如果想了解ftp协议的详细内容，请参考RFC959。

PyOpenCL

OpenCL的Python接口，通过该模块可以使用GPU实现并行计算。

标准库

编辑

Python拥有一个强大的标准库。Python语言的核心只包含数字、字符串、列表、字典、文件等常见类型和函数，而由Python标准库提供了系统管理、网络通信、文本处理、数据库接口、图形系统、XML处理等额外的功能。Python标准库命名接口清晰、文档良好，很容易学习和使用。

Python社区提供了大量的第三方模块，使用方式与标准库类似。它们的功能无所不包，覆盖科学计算、Web开发、数据库接口、图形系统多个领域，并且大多成熟而稳定。第三方模块可以使用Python或者C语言编写。SWIG,SIP常用于将C语言编写的程序库转化为Python模块。Boost C++ Libraries包含了一组库，Boost.Python，使得以 Python 或 C++ 编写的程序能互相调用。借助于拥有基于标准库的大量工具、能够使用低级语言如C和可以作为其他库接口的C++，Python已成为一种强大的应用于其他语言与工具之间的胶水语言。

Python标准库的主要功能有：

文本处理，包含文本格式化、正则表达式匹配、文本差异计算与合并、Unicode支持，二进制数据处理等功能

文件处理，包含文件操作、创建临时文件、文件压缩与归档、操作配置文件等功能

操作系统功能，包含线程与进程支持、IO复用、日期与时间处理、调用系统函数、写日记(logging)等功能

网络通信，包含网络套接字，SSL加密通信、异步网络通信等功能

网络协议，支持HTTP，FTP，SMTP，POP，IMAP，NNTP，XMLRPC等多种网络协议，并提供了编写网络服务器的框架

W3C格式支持，包含HTML，SGML，XML的处理

其它功能，包括国际化支持、数学运算、HASH、Tkinter等

开发环境

编辑

●IDLE：Python内置IDE (随python安装包提供)

●PyCharm[9] ：详见百度百科PyCharm，由著名的JetBrains公司开发，带有一整套可以帮助用户在使用Python语言开发时提高其效率的工 具，比如调试、语法高亮、Project管理、代码跳转、智能提示、自动完成、单元测试、版本控制。此外，该IDE提供了一些高级功能，以用于支持Django框架下的专业Web开发。

●Komodo和Komodo Edit：后者是前者的免费精简版

●Spyder：安装Anaconda自带的高级IDE

●PythonWin：ActivePython或pywin32均提供该IDE，仅适用于Windows

●SPE（Stani's Python Editor）：功能较多的自由软件，基于wxPython

●Ulipad：功能较全的自由软件，基于wxPython；作者是中国Python高手limodou

●WingIDE：可能是功能最全的IDE，但不是自由软件(教育用户和开源用户可以申请免费key)

●Eric：基于PyQt的自由软件，功能强大。全名是：The Eric Python IDE

●DrPython

●PyScripter：使用Delphi开发的轻量级的开源Python IDE， 支持Python2.6和3.0。

●PyPE：一个开源的跨平台的PythonIDE。

●bpython： 类Unix操作系统下使用curses库开发的轻量级的Python解释器。语法提示功能。

●eclipse + pydev插件：方便调试程序

●emacs：自带python支持，自动补全、refactor等功能需要插件支持

●Vim: 最新7.3版编译时可以加入python支持，提供python代码自动提示支持

●Visual Studio 2003 + VisualPython：仅适用Windows，已停止维护，功能较差

●SlickEdit

●Visual Studio 2010 + Python Tools for Visual Studio

●TextMate

●Netbeans IDE

●Sublime

●ipython

另外，诸如Notepad++、EditPlus、UltraEdit等通用的程序员文本编辑器软件也能对Python代码编辑提供一定的支持，比如代码自动着色、注释快捷键等，但是否够得上集成开发环境的水平，尚有待评估。

解释器

编辑

Python是一门跨平台的脚本语言，Python规定了一个Python语法规则，实现了Python语法的解释程序就成为了Python的解释器。

CPython（ClassicPython，也就是原始的Python实现，需要区别于其他实现的时候才以CPython称呼；或解作C语言实现的Python）。这是最常用的Python版本。

Jython（原名JPython；Java语言实现的Python，现已正式发布）。Jython可以直接调用Java的各种函数库。

PyPy（使用Python语言写的Python）

IronPython（面向.NET和ECMA CLI的Python实现）。IronPython能够直接调用.net平台的各种函数库。可以将Python程序编译成.net程序。

ZhPy（周蟒）（支持使用繁/简中文语句编写程序的Python语言）

著名应用

编辑

Digwebs-Web应用框架

Pylons-Web应用框架

Zope- 应用服务器

Plone- 内容管理系统

Django- 鼓励快速开发的Web应用框架

Uliweb- 国人开发的轻量级Web框架

TurboGears- 另一个Web应用快速开发框架

Twisted--Python的网络应用程序框架

Python Wikipedia Robot Framework- MediaWiki的机器人程序

MoinMoinWiki- Python写成的Wiki程序

flask- Python 微Web框架

tornado- 非阻塞式服务器

Webpy- Python 微Web框架

Bottle- Python 微Web框架

EVE- 网络游戏EVE大量使用Python进行开发

Reddit - 社交分享网站

Dropbox - 文件分享服务

TurboGears - 另一个Web应用快速开发框架

Fabric - 用于管理成百上千台Linux主机的程序库

Trac - 使用Python编写的BUG管理系统

Mailman - 使用Python编写的邮件列表软件

Mezzanine - 基于Django编写的内容管理系统

Blender - 以C与Python开发的开源3D绘图软件

库导入

编辑

内置库

可以直接使用 import语句导入。

外部库

需要先下载，再在CMD命令窗口在pip.exede同级目录下输入 pip install 库名。

外部库储存在include文件夹中。

学习网站

编辑

Python官方文档英文版[10]

Python官方文档中文版[11]

python 网络教育-百度传课[12]

Python官网[2]

百度百科内容由网友共同编辑，如您发现自己的词条内容不准确或不完善，欢迎使用本人词条编辑服务（免费）参与修正。立即前往>>

词条图册

更多图册

概述图册(3)

参考资料

1.  Python Documentation by Version ．Python Documentation by Version | Python.org[引用日期2020-03-02]

2.  官网首页 ．python官网[引用日期2018-01-05]

3.  黄海涛．Python3破冰人工智能从入门到实战．北京：人民邮电出版社，2019

4.  李金．自学Python：编程基础、科学计算及数据分析．北京：机械工业出版社，2018

5.  Python CGI 编程 ．自强学堂[引用日期2014-01-22]

6.  python流行指数 ．python吧[引用日期2013-02-20]

7.  不再有更新：Python 2.7支持将于2020年1月1日终止 ．it之家[引用日期2018-03-14]

8.  python tutorial中文 ．python中文文档[引用日期2013-02-20]

9.  PyCharm产品官网 ．JetBrains的PyCharm产品官网[引用日期2013-09-10]

10.  Python3 官方文档（英文） ．Python3 官方文档[引用日期2015-01-14]