目录

[第1章： Python简介 3](#_Toc518888985)

[1.1： Python定义及发行时间 3](#_Toc518888986)

[1.2： Python强制用空白符作为语句缩进。 3](#_Toc518888987)

[1.3： Python具有丰富强大的库，常被昵称为胶水语言。 3](#_Toc518888988)

[1.4： Python风格 4](#_Toc518888989)

[第2章： Python基础语法 5](#_Toc518888990)

[2.1： 编程 5](#_Toc518888991)

[2.1.1： 交互式编程 5](#_Toc518888992)

[2.1.2： 脚本式编程 5](#_Toc518888993)

[2.2： 注释 5](#_Toc518888994)

[2.3： 行和缩进 5](#_Toc518888995)

[2.4： Python标识符 6](#_Toc518888996)

[第3章： Python变量类型 7](#_Toc518888997)

[3.1： 标准数据类型 7](#_Toc518888998)

[3.1.1： Numbers（数字） 7](#_Toc518888999)

[3.1.2： String（字符串） 8](#_Toc518889000)

[3.1.3： List（列表） 8](#_Toc518889001)

[3.1.4： Tuple（元组） 8](#_Toc518889002)

[3.1.5： Dictionary（字典） 8](#_Toc518889003)

[3.2： 变量赋值 8](#_Toc518889004)

[3.3： 数据类型转换 8](#_Toc518889005)

# Python简介

## Python定义及发行时间

Python是一种面向对象的解释型计算机程序设计脚本语言，由荷兰人Guido van Rossum于1989年发明，并于1991年公开发行第一版。

## Python强制用空白符作为语句缩进。

Python语法简洁清晰，特色之一是强制用空白符（white space）作为语句缩进。这使得不好的变成习惯（例如 if语句的下一行不向右缩进）都不能通过编译。这点儿和其他大多数语言（如C）都有所不同，它使得一个模块的界限，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。

## Python具有丰富强大的库，常被昵称为胶水语言。

Python具有丰富强大的库。它常被昵称为胶水语言，能够把用其他语言制作的各种模快（尤其是 C/C++）很轻松地联结在一起。常见的一种应用情形是，使用Python快速生成程序的原型（有时甚至是程序的最终界面），然后对其中有特别要求的部分，用更适合的语言改写，比如3D游戏中的 图形渲染模块，性能要求特别高，就可以用 C/C++重写，而后 封装为 Python可以调用的扩展类库。需要注意的是：在使用扩展类库时可能需要考虑平台问题，某些可能不提供跨平台的实现。

## Python风格

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，是用途广泛的语言。

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（There is more than one way to do it）完全相反。（注：动态语言指在编译时变量的数据类型在运行时由系统自动确定，不需要在编写程序时进行类型声名）

因此，Perl语言中“总是有很多种方法来做同一件事儿”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儿”。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基础语法

## 编程

### 交互式编程

交互式编程不需要创建脚本文件，是通过Python解释器的交互模式进行编写代码。

### 脚本式编程

通过脚本参数调用解释器开始执行脚本，直到脚本执行完毕。当脚本执行完成后，解释器不再有效。所有Python文件以.py为扩展名。

## 注释

单行注释采用”#”开头，多行注释可以用三引号。

因为Python可以使用引号(‘)、双引号(“)三引号(‘’’或”””)来表示字符串，其中三引号可以由多行组成，是编写多行文本的快捷语法。常用于文档字符串，在文件的特定地方，则被当作注释。

## 行和缩进

学习Python与其他语言最大的区别就是，Python的代码块不使用大括号{}来控制类、函数以及其他逻辑判断。Python最具特色的就是用缩进来写模块。

所进的空白数量是可变的，但是所有代码块语句必须包含相同的缩进空白数量，这个必须严格执行。

若有的地方用TAB键缩进，有的用空格缩进，则会提示IndentationError: unindent does not match any outer indentation level因此，在Python的代码块中必须使用相同数目的行首缩进空格数。

建议在每个缩进层次使用单个制表符或两个空格或四个空格 ，切记不能混用。

另外函数之间或类的方法之间建议用空行分隔，表示一段新的代码的开始。类和函数入口之间也用一行空行分隔，以突出函数入口的开始。尽管空行与缩进不同，它并不是Python语法的一部分。书写时即使不插入空行，Python解释器运行也不会出错。但是空行的作用在于分隔两段不同功能或含义的代码，便于日后代码的维护或重构。

## Python标识符

在Python里，标识符由字母、数字、下划线组成。所有标识符可以包括英文、数字以及下划线(\_)，但不能以数字开头。Python中的标识符是区分大小写的。

以下划线开头的标识符是有特殊意义的。以 单下划线\_foo的代表不能直接访问的类属性，需通过类提供的接口进行访问，不能用from xxx import \*而导入；以双下划线开头的\_\_foo代表类的私有成员；以双下划线开头和结尾的\_\_foo\_\_代表Python里特殊方法专用的标识，如\_\_init\_\_()代表类的构造函数。

Python可以同一行显示多条语句，方法是用分号 ; 分开。

Python中的保留字不能用作常数或变数，或任何其他标识符名称。

所有Python的关键字只包含小写字母，以下是Python的保留字符

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| and | exec | not | assert | finally | or |
| break | for | pass | class | from | print |
| continue | global | raise | def | if | return |
| del | import | try | elif | in | while |
| else | is | with | except | lambda | yield |

# Python变量类型

## 标准数据类型

### Numbers（数字）

数字数据类型用于存储数值。

他们是不可改变的数据类型，这意味着改变数字数据类型会分配一个新的对象。

可以使用del语句删除一些对象的引用，当在同一行代码中删除多个对象的引用时，各引用之间用 , 隔开。

Python支持四种不同的数字类型：

* int（有符号类型）

如：10、100、-786、-0x260、0x69

* long（长整型）

如：51924361L、-0x19323L、0122L、0xDEFABCECBDAECBFBAEL

注：Python使用字母L（大小写均可）来显式生成长整型的对象，尽管长整型也可用小写的l，但还是建议使用大写的L，因为这样可以避免其与数字1混淆。

* float（浮点型）

如：0.0、15.20、-21.9、32.3e+18

* complex（复数）

如：3.14j、45j、9.322e-36j、5+6j、complex(5,6)

### String（字符串）

字符串或串(String)是由数字、字母、下划线组成的一串字符。

Python的字符串列表 有两种取值顺序：

* 从左到右索引默认从0开始，最大范围是字符串长度少1
* 从右到左索引默认从-1开始，最大范围是字符串开头

如果要实现从字符串中获取一段子字符串的话，可以使用变量[头下标:尾下标]，就可以截取相应的字符串，其中下标是从0开始算起，不写头下标表示从头开始取，不写尾下标表示取到尾。

当使用以冒号分隔的字符串，Python返回一个新的对象，结果包含了以这对偏移标识的连续的内容。

加号（+）是字符串连接运算符，星号（\*）是重复操作

### List（列表）

List（列表）是Python中使用最频繁的数据类型，列表可以完成大多数集合类的数据结构实现。它支持字符，数字，字符串甚至可以包含列表（即嵌套）。

列表用[ ]标识，列表中的值也可以用到变量[头下标:尾下标]，就可以截取相应的列表，从左到右索引默认0开始，从右到左索引默认-1开始，头下标为空表示从头开始取，尾下标为空表示取到尾。

加号+是列表连接运算符，星号\*是重复操作。

### Tuple（元组）

Tuple（元组）是一个类似于List（列表）的数据类型，元组用 ( ) 标识，内部元素用逗号隔开，但是元组不能二次赋值，相当于只读列表。（即元组不允许更新，而列表是允许更新的）

### Dictionary（字典）

字典（Dictionary）是除列表以外，Python中最灵活的内置数据结构类型，列表是有序的对象集合，字典是无序的对象集合。两者之间的区别在于：字典当中的元素是通过键来存取的，而不是通过偏移存取。字典用 { } 标识。字典由索引（Key）和它对应的值（Value）组成。

### Set（集合）

在数学中，我们知道集合中的元素由三个特征：1.确定性（集合中的元素必须是确定的）。2.互异性（集合中的元素互不相同）。3.无序性（集合中的元素没有先后之分）

当Set（集合）与List（列表）对比时，我们会发现List（列表）中的元素是可以重复的，而Set（集合）中的元素是不可重复的（即是互异的）。

当Set（集合）与Dict（字典）对比时，我们会发现Set（集合）与Dict（字典）类似，也是一组key（键）的集合，但不存储Value（值）。

## 变量赋值

Python允许同时为多个变量赋值，如：

a = b =c =1

表示创建一个整形对象，值为1，三个变量被分配到相同的内存空间上。

也可以为多个对象指定多个变量。如：

a, b, c = 1, 2, “Jerry”

表示两个整形对象1和2分别分配给变量a和b，字符串对象”John”分配给变量c。

## 数据类型转换

有时候，我们需要对数据内置的类型进行转换，数据类型的转换，只需要将数据类型作为函数名即可。

以下几个内置的函数可以执行数据类型之间的转换，这些函数返回一个新的对象，表示转换的值。

|  |  |
| --- | --- |
| 函数 | 描述 |
| int(x) | 将x转换为一个整数 |
| long(x) | 将x转换为一个长整数 |
| float(x) | 将x转换到一个浮点数 |
| str(x) | 将对象x转换为字符串 |
| repr(x) | 将对象x转换为表达式字符串 |
| eval(str) | eval函数就是实现list、tuple、dict与str之间的转化 |
| tuple(s) | 将序列s转换成一个元组 |
| list(s) | 将序列s转换成一个列表 |
| set(s) | 转换为可变集合 |
| frozenset(s) | 转换为不可变集合 |
| dict(d) | 创建一个字典，d必须是一个包含一维子元组（key,value）的二维元组。 |
| chr(x) | 将一个整数转换为一个字符 |
| unichr(x) | 将一个整数转换为Unicode字符 |
| ord(x) | 将一个字符转换为它的整数值 |
| hex(x) | 将一个整数转换为一个十六进制字符串 |
| oct(x) | 将一个整数转换为一个八进制字符串 |
| bin(x) | 将一个整数转换为一个二进制字符串 |