[1.语言综述 4](#_Toc514010824)

[1.1. 语言历史 4](#_Toc514010825)

[1.2. 设计哲学——Python之禅 4](#_Toc514010826)

[1.3. 语言特点 5](#_Toc514010827)

[1.4. 应用前景 6](#_Toc514010828)

[1.5. 如何获得帮助 8](#_Toc514010829)

[2. 开发环境搭建 8](#_Toc514010830)

[2.1. 搭建Python环境 8](#_Toc514010831)

[2.2. PyCharm开发工具 9](#_Toc514010832)

[2.3. 文本编辑工具 10](#_Toc514010833)

[3. 编码规范 10](#_Toc514010834)

[3.1. 命名规范 10](#_Toc514010835)

[3.2. 注释规范 11](#_Toc514010836)

[3.3. 导入规范 13](#_Toc514010837)

[3.4. 代码排版 14](#_Toc514010838)

[3.4.1. 空行 15](#_Toc514010839)

[3.4.2. 空格 16](#_Toc514010840)

[3.4.3. 缩进 18](#_Toc514010841)

[3.4.4. 断行 19](#_Toc514010842)

[4. Python语法基础 20](#_Toc514010843)

[4.1. 标识符和关键字 20](#_Toc514010844)

[4.2. 常量和变量 20](#_Toc514010845)

[4.3. 注释 20](#_Toc514010846)

[4.4. 语句 20](#_Toc514010847)

[4.5. 模块 20](#_Toc514010848)

[4.6. 包 20](#_Toc514010849)

[5. 数据类型 20](#_Toc514010850)

[5.1. 数字类型 20](#_Toc514010851)

[5.2. 数字类型相互转换 20](#_Toc514010852)

[5.3. 字符串类型 20](#_Toc514010853)

[6. 运算符 20](#_Toc514010854)

[6.1. 算数运算符 20](#_Toc514010855)

[6.2. 关系运算符 20](#_Toc514010856)

[6.3. 逻辑运算符 21](#_Toc514010857)

[6.4. 位运算符 21](#_Toc514010858)

[6.5. 赋值运算符 21](#_Toc514010859)

[6.6. 其他运算符 21](#_Toc514010860)

[6.7. 运算符优先级 21](#_Toc514010861)

[7. 控制语句 21](#_Toc514010862)

[7.1. 分支语句 21](#_Toc514010863)

[7.2. 循环语句 21](#_Toc514010864)

[7.3. 跳转语句 21](#_Toc514010865)

[7.4. 使用范围 21](#_Toc514010866)

# 语言综述

## 语言历史

## 设计哲学——Python之禅

优美胜于丑陋*——指编码规范*

明了胜于晦涩*——表达的意思要明了*

简洁胜于复杂*——算法或编码规范*

复杂胜于凌乱*——算法或编码规范*

扁平胜于嵌套*——使用多层嵌套并不好*

宽松胜于紧凑*——行与行之间的距离*

可读性很重要*——也指编码规范*

1：即便是特例，也不可违背以上规则

2：不要捕获所有错误，除非你确定需要这样做

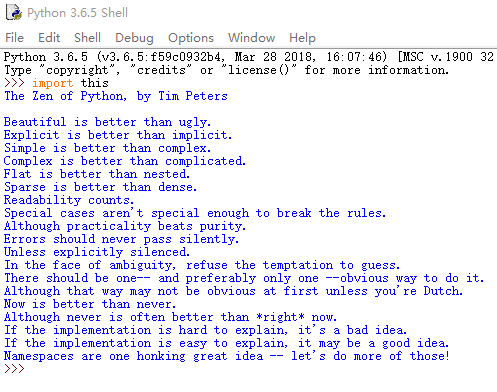
3：如果存在多种可能，不要猜测

4：通常只有唯一一种是最佳的解决方案（最重要的）

5：虽然这并不容易，因为你不是Python之父（对前面两句话的总结）

6：做比不做要好，但不假思索就动手还不如不做

7：如果你的方案很难懂，那肯定不是一个好方案，反之亦然



## 语言特点

### 简单易学

Python设计目标之一就是能够方便学习，使用简单。

使你能够专注于解决问题而不是过多关注语言本身。

### 面向对象

Python支持面向对象的编程。

与其他主要的语言相比入C++和Java相比，Python以一种非常强大而又简单的方式实现面向对象编程。

### 解释性

Python是解释执行的，即Python语言不需要被编译成二进制代码，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为中间字节码形式，然后再把它解释为计算机使用的机器语言执行。

### 免费开源

Python是免费开放源代码软件之一。

简单地说，你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。

### 可移植性

Python解释器已经被移植在许多平台上，Python程序无需修改就可以在多个平台上运行。

### 胶水语言

Python被称为胶水语言。

所谓胶水语言是用来连接其他语言编写的软件组建或模块。

Python能够称为胶水语言是因为标准版本Python使用C编译的，称为CPython。所以Python可以调用C语言，借助于C接口Python几乎可以驱动所有已知的软件。

### 丰富的库

Python标准库（官方提供的）种类繁多，它可以帮助处理各种工作，这些库不需要安装直接可以使用。

除了标准库意外，还有许多其他高质量的库可以使用。

### 规范的代码

Python使用强制缩进的方式使得代码具有极佳的可读性。

### 支持函数式编程

虽然Python并不是一种单纯的函数式编程，但是也提供了函数式编程的支持，如函数类型、Lambda表达式、高阶函数和匿名函数等。

### 动态类型

Python是动态类型语言，它不会检查数据类型，在变量声明是不需要指定数据类型。

## 应用前景

### 桌面应用开发

Python语言可以开发传统的桌面应用程序。

Tkinter、PyQt、PySide、wxPython和PyGTK等Python库可以快速开发桌面应用程序。

### Web应用开发

Python也经常被用于Web开发。

很多网站是基于Python Web开发的，如豆瓣、知乎和Dropbox等。

很多熟悉的Python Web框架，如Django、Flask、Tornado、Bottle和web2py等Web框架。可以帮助开发人员快速开发Web应用。

### 自动化运维

Python可以编写服务器运维自动化脚本。

很多服务器采用Linux和UNIX系统，以前很多运维人员编写系统管理Shell脚本实现运维工作。而现在使用Python编写系统管理，在可读性、性能、代码可重用、可扩展性等几方面优于普通Shell脚本。

### 科学计算

Python语言也可广泛地应用科学计算，NumPy、SciPy和Pandas是优秀的数值计算和科学计算库。

### 数据可视化

Python语言也可将复杂的数据通过图表展示出来，便于数据分析。

Matplotlib库是优秀的可视化库。

### 网络爬虫

Python语言很早就用来编写网络爬虫。

谷歌等搜索引擎公司大量地使用Python语言编写网络爬虫。

从技术层面上讲Python语言有很多这方面的工具，urllib、Selenium和BeautifulSoup等。

还可以网络爬虫框架scrapy。

### 人工智能

人工智能是现在非常火的一个方向。

Python广泛应用于深度学习、机器学习和自然语言处理等方向。

由于Python语言的动态特点，很多人工智能框架是采用Python语言实现的。

### 大数据

大数据分析中涉及到的分布式计算、数据可视化、数据库操作等，Python中都有成熟库可以完成这些工作。

Hadoop和Spark都可以直接使用Python编写计算逻辑。

### 游戏开发

Python可以直接调用Open GL势线3D绘制，这是高性能游戏引擎的技术基础。

所有很多Python语言势线的游戏引擎入Pygame、Pyglet和Cocos2d等。

## 如何获得帮助

对于一个初学者必须要熟悉如下几个Python相关网址：

Python标准库：https://docs.python.org/3/library/index.html

Python HOWTO：https://docs.python.org/3/howto/index.html

Python教程：https://docs.python.org/3/tutorial/index.html

PEP规范：https://www.python.org/dev/peps/

# 开发环境搭建

## 搭建Python环境

### Python环境产品有多个

1：CPython

CPython是Python官方提供，一般情况下提到的Python就是指CPython，CPython是基于C的语言编写的，它实现的Python解释器能够将源代码编译位字节码（Bytecode）,类似于Java语言，然后由虚拟机执行，这样当再次执行相同源代码文件时，如果源代码文件没有修改过，那么它会直接解释执行字节码文件，这样会提高程序的运行速度。

2：PyPy

基于Python实现的Python环境。速度比CPython，但兼容性不如CPython。官网[www.pypy.org](http://www.pypy.org)。

3：Jython

基于Java实现的Python环境。可以将Python代码编译为Java字节码，可以在Java虚拟机下运行。官网[www.jython.org](http://www.jython.org)。

4：IronPython

基于.NET平台实现的Python环境，可以.NET Framework链接库。官网[www.jython.org](http://www.jython.org)。

### Python环境下载

<https://www.python.org/downloads>

## PyCharm开发工具

### 下载和安装

<https://www.jetbrains.com/pycharm/download/>

### 设置Python解释器

Python解释器虚拟环境：

* Virtuallenv Environment
* Conda Environment

Conda是一个开源的软件包管理系统和环境管理系统。安装Conda一般通过安装Anaconda实现的，Anaconda是一个Python语言的免费增值发行版，用于进行大规模数据处理、预测分析和科学计算，致力于简化包的管理和部署。

* System Interpreter

## 文本编辑工具

### Sublime Text

近年来发展和壮大的文本编辑工具，所有的设置没有图形界面，在JSON格式的文件中进行的，初学者入门比较难，官网[www.sublimetext.com](http://www.sublimetext.com)。

### UltraEdit

历史悠久强大的文本编辑工具，可支持文本列模式等很多有用的功能，官网[www.ultraedit.com](http://www.ultraedit.com)。

### Notepad++

Notepad++本意是Windows平台Notepad（记事本）的升级，但其功能非常强大，能够很好地支持中文等多种语言，内置支持多达27种语言的语法高亮度显示。更重要的是它是免费的。官网[www.notepad-plus-plus.org](http://www.notepad-plus-plus.org)。

### EditPlus

历史悠久强大的付费文本编辑工具，小巧、轻便、灵活，官网[www.editplus.com](http://www.editplus.com)。

# 编码规范

Python编码规范借鉴了Python官方的PEP8编码规范和谷歌Python编码规范。

参考地址<https://www.python.org/dev/peps/pep-0008>。

参考地址<https://google.github.io/styleguide/pyguide.html>。

## 命名规范

### 包名

全部小写字母，中间可以由点分隔开，不推荐使用下划线。

作为命名空间，包名应该具有唯一性，推荐采用公司或组织域名的倒置，如com.apple.quicktime.v2。

### 模块名

全部小写字母，如果是多个单词构成，可以用下划线隔开，如dummy\_threading。

### 类名

采用大驼峰法命名法，如SplitViewController。

### 异常名

异常属于类，命名同类命名，应该使用Error作为后缀。如FileNotFoundError。

### 变量名

全部小写字母，如果是多个单词构成，可以用下划线隔开。如果变量应用于模块或函数内部，则变量名可以单下划线开头：如果变量类内部私有使用变量名可以双下划线开头。不要命名双下划线开头和结尾的变量，这是Python保留的。另外，避免使用小写L、大写O和大写I作为变量名。

### 函数名和方法名

命名同变量命名。如balance\_account、\_push\_cm\_exit。

### 常量名

全部大写字母，如果是由多个单词构成，可以用下划线隔开，如YEAR和WEEK\_OF\_MONTH。

## 注释规范

### 文件注释

文件注释就是在每一个文件开头添加注释，采用多行注释。

文件注释通常包括如下信息：版权信息、文件名、所在模块、作者信息、历史版本信息、文件内容和作用等。

例：

#

# 版权所有2015某某科技有限公司

# 许可信息查看LICENSE.txt文件

# 描述：

# 实现某某某功能

# 历史版本：

# 2015-7-22：创建 谁谁谁

# 2015-8-20：添加socket库

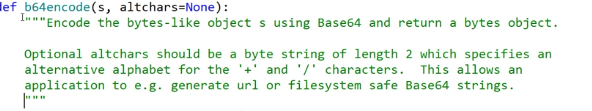
# 2015-8-22：添加math库

#

### 文档注释

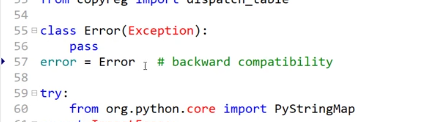
文档注释就是文档字符串，注释内容能够生成API帮助文档，可以使用Python官方提供的pydoc工具从Python源代码文件中提取这些信息，也可以生成HTML文件。所有公有的模块、函数、类和方法都应该进行文档注释。

文档注释推荐使用一对三重双引号（”””）包裹起来，注意不推荐使用三重单引号（’’’）。

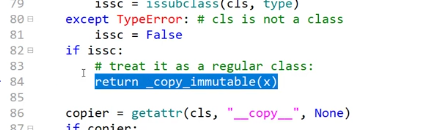


### 代码注释

一般变量声明或常量声明会在尾部注释：

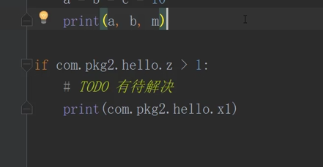


如果代码很长，则需要在这行代码的上部进行注释，或者注释内容比较多的：



### 使用TODO注释

也被称为地标注释，在完成程序代码时有些地方还不能确定，或者没有完成：

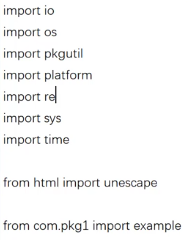


## 导入规范

导入语句总是放在文件顶部，位于模块注释和文档注释之后，模块全局变量和常量之前。每一个导入语句只能导入一个模块，示例代码如下：

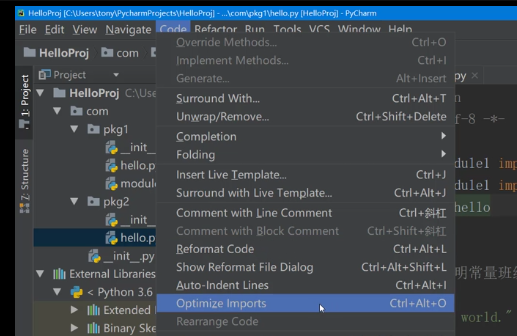


导入语句应该按照从通用到特殊的顺序分组，顺序是：标准库—第三方库—自己模块，每一组之间有一个空行，而且组中模块是按照英文字典顺序排序的。



可以利用PyCharm工具进行整理：

叫做分析导入：



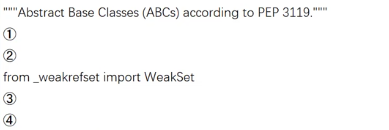
## 代码排版

代码排版包括空行、空格、断行和缩进等内容。

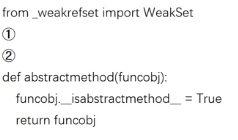
### 空行

Import语句块前后保留两个空行，示例代码如下：

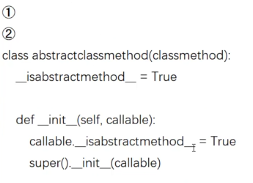
其中①②处和③④处是两个空行：



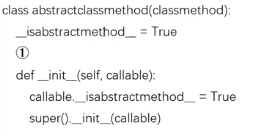
函数声明之前后保留两个空行，示例代码如下，其中①②处是两个空行。



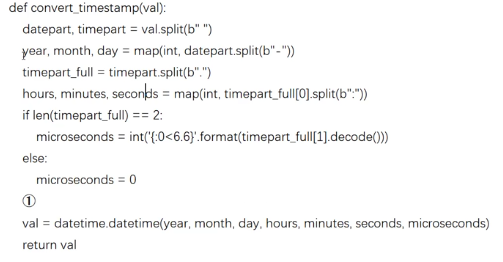
类声明之前后保留两个空行，示例代码如下，其中①②处是两个空行。



方法（类中所定义的函数）声明之前保留一个空行，示例代码如下，其中①处是一个空行。



两个逻辑代码块之间应该保留一个空行，示例代码如下，其中①处是一个空行。



### 空格

使用空格的规范：

1：赋值符号“=”前后各有一个空格。



2：所有的二元运算符都应该使用空格与操作数分开。



3：一元运算符：算法运算符取反“-”和算数运算符取反“~”。



4：括号内不要有空格，Python中括号包括：小括号“（）”、中括号“【】”和大括号“{}”。

推荐：



不推荐：



5：不要在逗号、分号、冒号前面有空格，而是要在他们后面有一个空格，除非该符号已经是行尾了。

推荐：



不推荐：



6：参数列表，索引或切片的左括号前不应有空格。

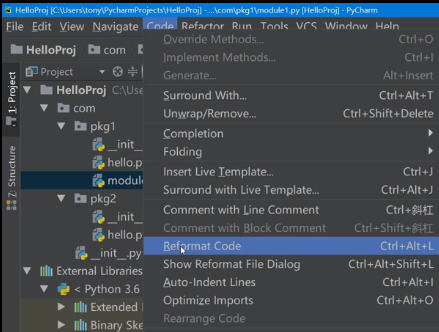
推荐：



不推荐：



注：PyCharm工具可以帮助进行整理：

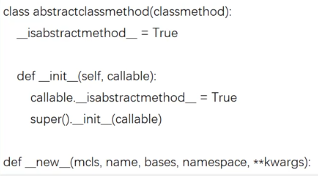


### 缩进

4个空格常被作为缩进排版的一个级别。

虽然在开发时程序员可以使用制表符进行缩进，而默认情况下一个制表符等于8个空格，但是不同的IDE工具中一个制表符与空格对应个数会有不同，所以不要使用制表符缩进。

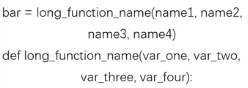
代码块的内容相对于首行缩进一个级别（4个空格），示例如下：



### 断行

一行代码中最多79个字符。对于文档注释和多行注释时一行最多72个字符，但是如果注释中包含URL地址可以不受这个限制。否则，如果超过则需断行，可以依据下面的一般规范断开：

1：在逗号后面断开



2：在运算符前面断开



3：尽量不要使用续行符（\），当有括号（包括：大括号、中括号和小括号）则在括号中断开，这样可以不使用续行符。

# Python语法基础

## 标识符和关键字

### 标识符

Python语言中标识符的命名规则如下：

1：区分大小写

Myname与myname是两个不同的标识符。

2：首字符

可以是下划线（\_）或字母，但不能是数字。

3：除首字符外其他字符

可以是下划线（\_）或字母和数字。

4：关键字不能作为标识符。

5：不要使用Python内置函数作为自己的标识符。

例如：

身高、identifier、userName、User\_Name、\_sys\_val等合法的标识符，注意中文“身高”命名的变量是合法的；

而2mail、room#、$Name和class为非法的标识符，注意#和$不能构成标识符。

### 关键字

Python语言中由33个关键字。

只有三个，即False、None和True首字母大写，其他的全部小写。

|  |  |  |  |
| --- | --- | --- | --- |
| False | def | if | raise |
| None | del | import | return |
| True | elif | in | try |
| and | else | is | while |
| as | except | lambda | with |
| assert | finally | nonlocal | yield |
| break | for | not |  |
| class | from | or |  |
| continue | global | pass |  |

## 常量和变量

### 变量

Python是动态类型语言。

### 常量

不能定义真正意义的常量。

## 注释

### 代码注释

#

### 编码注释

第一种情况：

# coding = utf-8注释

第二种情况：指定Python解释器在一个什么路径

# ！/usr/bin/python

# -\*- coding:utf-8 -\*-

注意：在程序代码中，对容易引起误解的代码进行注释是必要的，但应避免对已清晰表达信息的代码进行注释。需要注意的是，频繁的注释有时反映了代码的低质量。当觉得被迫要加注释的时候，不妨考虑一下重写代码使其更清晰。

## 语句

1：一行代码表示一条语句，语句结束可以加分号，也可以省略分号。

2：多个变量赋相同的数值，可以这样表示：

a = b = c =10

3：if、for和while有代码块的语句中，代码块不是通过大括号来界定的，而是通过缩进，缩进在一个级别的代码是在相同的代码块中。

4：一个缩进级别一般是一个制表符（Tab）或4个空格，考虑到不同的编辑器制表符显示的宽度不同，大部分编程语言规范推荐使用4个空格。

## 模块

一个模块就是一个文件，模块是保存代码的最小单位，模块中可以声明变量、常量、函数、属性和类等Python程序元素。

例如：

* import <模块名>
* from <模块名> import <代码元素>

## 包

包（package）实际上就是一个命名空间。其实就是为了防止命名冲突。

# 数据类型

在Python中所有的数据类型都是类，每一个变量都是类的“实例”。

没有基本数据类型的概念，所以整数、浮点和字符串也都是类。

Python有6种标准数据类型：数字、字符串、列表、元组、集合和字典。

## 数字类型

整数类型、浮点类型、复数类型和布尔类型。

### 整数类型

Python整数类型位int，整数类型的范围可以很大，可以表示很大的整数，这只受所在计算机硬件的限制。

二进制数、八进制数和十六进制整数表示方式如下：

* 二进制数：

以0b或0B为前缀，注意0是阿拉伯数字，不要误认为是英文字母o或O。

* 八进制数：

以0o或0O为前缀，第一个字符是阿拉伯数字0，第二个字符是英文字母o或O。

* 十六进制数：

以0x或0X为前缀，注意0是阿拉伯数字。

### 浮点类型

Python浮点类型为float，Python只支持双精度浮点类型，而且是与本机相关。

### 复数类型

Python中复数类型为complex，例如1+2j表示的是实部1，虚部2的复数。

### 布尔类型

Python中布尔类型为bool，bool是int的字类，它只有两个值：True和False。

注意：任何类型数据都可以通过bool()函数转换为布尔值，那些被认为“没有的”、“空的”值会转换为False，反之转换为True。如None(空对象)、False、0、0.0、0j（复数）、””（空字符串）、[]（空列表）、（）（空元组）和{}（空字典）这些值会转换为False，否则是True。

## 数字类型相互转换

数字类型主要包含整数、浮点和布尔互相转换。

### 隐式类型转换

多个数字类型数据之间可以进行数学计算，由于参与进行计算的数字类型可能不同，此时会发生隐式类型转换。

|  |  |  |
| --- | --- | --- |
| 操作数1类型 | 操作数2类型 | 反转后的类型 |
| 布尔 | 整数 | 整数 |
| 布尔、整数 | 浮点 | 浮点 |

### 显式类型转换

int（）、float（）和bool（）函数

## 字符串类型

### 字符串表示方式

Python中字符串表示方式三种：

1：普通字符串

采用单引号（’）或双引号（”）包裹起来的字符串。

2：原始字符串（raw string）

在普通字符串前加r，字符串中的特殊字符不需要转义，按照字符串的本来“面目”呈现。

3：长字符串

字符串中包含了换行缩进等排版字符，可以使用三重单引号（’’’）或三重双引号（”””）包裹起来，这就是长字符串。

|  |  |  |
| --- | --- | --- |
| **字符表示** | **Unicode编码** | **说明** |
| \t | \u0009 | 水平制表符 |
| \n | \u000a | 换行 |
| \r | \u000d | 回车 |
| \” | \u0022 | 双引号 |
| \’ | \u0027 | 单引号 |
| \\ | \u005c | 反斜线 |

### 字符串格式化

Format()方法，以及占位符{}

格式控制符是位于占位符索引或占位符名字的后面，之间用冒号分割，例如{1：d}表示索引为1的占位符格式参数是十进制整数。

字符串格式化控制符

|  |  |
| --- | --- |
| 控制符 | 说 明 |
| s | 字符串格式化 |
| d | 十进制整数 |
| f、F | 十进制浮点数 |
| g、G | 十进制整数或浮点数 |
| e、E | 科学计算法表示浮点数 |
| o | 八进制整数，符号是小英文字母o |
| x、X | 十六进制整数，x是小写表示，X是大写表示 |

### 字符串查找

# 运算符

## 算数运算符

## 关系运算符

## 逻辑运算符

## 位运算符

## 赋值运算符

## 其他运算符

## 运算符优先级

# 控制语句

## 分支语句

## 循环语句

## 跳转语句

## 使用范围