# 

# **Python开发规范**

目录

[分号 3](#_Toc12849)

[行长度 3](#_Toc29144)

[括号 3](#_Toc25170)

[空格 4](#_Toc25733)

[注释 6](#_Toc7044)

[类 7](#_Toc1663)

[字符串 7](#_Toc12103)

[文件和sockets 8](#_Toc27435)

[导入格式 8](#_Toc17163)

[语句 9](#_Toc5378)

[命名 10](#_Toc6154)

[项目结构 11](#_Toc27426)

[MySQL数据库 11](#_Toc16441)

[开发工具 13](#_Toc12753)

# 分号

1.不要在行尾加分号, 也不要用分号将两条命令放在同一行.

# 行长度

1.每行不超过80个字符，例外:1、长的导入模块2、语句注释里的URL

2.不要使用反斜杠连接行

3.Python会将[圆括号,中括号和花括号中的行隐式的连接起来](http://docs.python.org/2/reference/lexical_analysis.html" \l "implicit-line-joining)，你可以利用这个特点。如果需要，你可以在表达式外围增加一对额外的圆括号

Yes: foo\_bar(self, width, height, color='black', design=None, x='foo',

emphasis=None, highlight=0)

if (width == 0 and height == 0 and

color == 'red' and emphasis == 'strong'):

4.在注释中，如果必要，将长的URL放在一行上。

Yes: # See details at

# http://www.example.com/us/developer/documentation/api/content/v2.0/csv\_file\_name\_extension\_full\_specification.html

No: # See details at

# http://www.example.com/us/developer/documentation/api/content/\

# v2.0/csv\_file\_name\_extension\_full\_specification.html

# 括号

1.宁缺毋滥的使用括号

2.除非是用于实现行连接, 否则不要在返回语句或条件语句中使用括号. 不过在元组两边使用括号是可以的.

Yes: if foo:

bar()

while x:

x = bar()

if x and y:

bar()

if not x:

bar()

return foo

for (x, y) in dict.items(): ...

No: if (x):

bar()

if not(x):

bar()

return (foo)

# 空格

1.按照标准的排版规范来使用标点两边的空格

2.括号内不要有空格

Yes: spam(ham[1], {eggs: 2}, [])

No: spam( ham[ 1 ], { eggs: 2 }, [ ] )

3.不要在逗号, 分号, 冒号前面加空格, 但应该在它们后面加(除了在行尾)

Yes: if x == 4:

print x, y

x, y = y, x

No: if x == 4 :

print x , y

x , y = y , x

4.参数列表, 索引或切片的左括号前不应加空格

Yes: spam(1)

no: spam (1)

Yes: dict['key'] = list[index]

No: dict ['key'] = list [index]

5.在二元操作符两边都加上一个空格, 比如赋值(=), 比较(==, <, >, !=, <>, <=, >=, in, not in, is, is not), 布尔(and, or, not)

Yes: x == 1

No: x<1

6.当'='用于指示关键字参数或默认参数值时, 不要在其两侧使用空格.

Yes: def complex(real, imag=0.0): return magic(r=real, i=imag)

No: def complex(real, imag = 0.0): return magic(r = real, i = imag)

7.不要用空格来垂直对齐多行间的标记, 因为这会成为维护的负担(适用于:, #, =等):

Yes:

foo = 1000 # comment

long\_name = 2 # comment that should not be aligned

dictionary = {

"foo": 1,

"long\_name": 2,

}

No:

foo = 1000 # comment

long\_name = 2 # comment that should not be aligned

dictionary = {

"foo" : 1,

"long\_name": 2,

}

# 注释

1.模块注释推荐写法：

# -\*- coding: utf-8 -\*-  
# Description: 模块描述  
# Created: 作者 2020/06/22

2.方法注释推荐写法：

"""  
方法描述  
:param 参数  
:return:返回值  
"""

3.行注释推荐写法

# 注释

x+1

或

x+1 # 注释

# 类

1.如果一个类不继承自其它类, 就显式的从object继承. 嵌套类也一样.

class SampleClass(object):

pass

class OuterClass(object):

class InnerClass(object):

pass

class ChildClass(ParentClass):

"""Explicitly inherits from another class already."""

# 实体类要继承BaseEntity类

class Coproduct(BaseEntity):  
 """  
 副产品实体类  
 """  
  
 def \_\_init\_\_(self, coproduct=None):

# 字符串

1.避免在循环中用+和+=操作符来累加字符串，你可以将每个子串加入列表, 然后在循环结束后用 .join 连接列表.

Yes: items = ['<table>']

for last\_name, first\_name in employee\_list:

items.append('<tr><td>%s, %s</td></tr>' % (last\_name, first\_name))

items.append('</table>')

employee\_table = ''.join(items)

No: employee\_table = '<table>'

for last\_name, first\_name in employee\_list:

employee\_table += '<tr><td>%s, %s</td></tr>' % (last\_name, first\_name)

employee\_table += '</table>'

2.在同一个文件中, 保持使用字符串引号的一致性. 使用单引号'或者双引号"之一

Yes:

Python('Why are you hiding your eyes?')

Gollum("I'm scared of lint errors.")

Narrator('"Good!" thought a happy Python reviewer.')

No:

Python("Why are you hiding your eyes?")

Gollum('The lint. It burns. It burns us.')

Gollum("Always the great lint. Watching. Watching.")

# 文件和sockets

1. 在文件和sockets结束时, 显式的关闭它
2. 如果需要异常处理时，要将关闭操作放到finally代码块

# 导入格式

1.每个导入应该独占一行

Yes: import os

import sys

No: import os, sys

2.导入总应该放在文件顶部, 位于模块注释和文档字符串之后, 模块全局变量和常量之前. 导入应该按照从最通用到最不通用的顺序分组:

1、标准库导入

2、第三方库导入

3、应用程序指定导入

# 语句

1.通常每个语句应该独占一行

2.如果是if语句, 只有在没有else时才能放在一行.

3.try和except不能放在同一行.

Yes:

if foo: bar(foo)

No:

if foo: bar(foo)

else: baz(foo)

try: bar(foo)

except ValueError: baz(foo)

try:

bar(foo)

except ValueError: baz(foo)

# 命名

1.应该避免的名称

1、单字符名称, 除了计数器和迭代器.

2、包/模块名中的连字符(-)

3、双下划线开头并结尾的名称(Python保留, 例如\_\_init\_\_)

2.用单下划线(\_)开头表示模块变量或函数是protected的(使用from module import \*时不会包含).

3.用双下划线(\_\_)开头的实例变量或方法表示类内私有.

4.对类名使用大写字母开头的单词(如CapWords, 即Pascal风格), 但是模块名应该用小写加下划线的方式(如lower\_with\_under.py).

5.函数名一律小写，如有多个单词，用下划线隔开

6.变量名尽量小写, 如有多个单词，用下划线隔开

7.常量使用以下划线分隔的大写命名

# 项目结构

1.项目工程推荐目录结构如下：

- app 应用包  
 - analysis 离线分析包

- analysis 分析包  
 - models 训练模型包  
 - utils 工具类包  
 - main 业务程序包  
 - routes 路由包  
 - entity 业务实体包  
 - dao 数据库访问对象包  
 - service 业务逻辑包  
 - task 任务包  
 - rule 算法规则包  
- config.py 项目配置信息  
- model\_config.py 模型配置信息  
- manage.py 项目入口，启动文件  
- tests 测试包

2.entity包下存在基类BaseEntity，此包下所有类需要继承BaseEntity类

3.dao包下存在基类BaseDao，此包下所有类需要继承BaseDao类

# MySQL数据库

1. 建表规范：

1、表名、字段名必须使用小写字母或数字，禁止出现数字开头，禁止两个 下划线中间只出现数字

YES：

aliyun\_admin，rdc\_config，level3\_name

NO：

AliyunAdmin，rdcConfig，level\_3\_name

2、禁用保留字，如desc、range、match、delayed等，请参考MySQL官方保留字

3、数值类型推荐使用decimal，时间类型推荐使用datetime，如果存储的字符串长度几乎相等，推荐使用char定长字符串类型

4、必备三字段：rowid, create\_time, update\_time，其中rowid为自增主键

2.索引规范：

1、代码先行，索引后上。功能开发完毕，把涉及到该表相关sql都要拿出来分析之后再建立索引

2、联合索引尽量覆盖条件

3、不要在小基数字段上建立索引

4、长字符串推荐采用前缀索引

5、尽量少用左模糊或者全模糊，索引文件具有B-Tree最左前缀特性，左边值未确定无法使用索引

6、利用覆盖索引来进行查询操作，避免回表

7、建立组合索引时，区分度最高的在最左边

8、ADB FOR MYSQL不支持唯一索引

3.SQL语句

1、代码中禁止使用select \*

2、尽量少使用in操作，使用时需评估in后面的集合数量，控制在1000个之内。

3、in和exsits选择原则：小表驱动大表

4、模型禁止访问平台业务数据库或数仓ods源表，超过三张表禁止join，需要join的字段数据类型保持绝对一致，被关联字段需要有索引。涉及多个表的join，数仓做成可复用的宽表

5、使用数据库连接池，代码中禁止使用重新创建连接进行sql操作

# 开发工具

1.开发环境：Python3.7

2.开发IDE：Pycharm

3.数据库客户端：Navicat

4.Restful接口测试工具：Postman

5.压力测试工具：Jmeter

6.部署工具：Jenkins