简明 python 编码规范

(v1.0)

编者：trollbird@gmail.com

目标

以简明可读性为前提，简巧的约定，文章般的风格，希望以此能共建朴实稳健的系统，绚丽的产品。

文档风格

1. 编码
   - 统一编码格式 utf-8
   - 文件头部引入： #coding=utf-8
2. 注释
   - 组成：语句注释(#开头)和文档注释(模块、类、方法下的字符串)
   - 原则：适当语句注释，不存在技术难点坚持不注释，存在难点必须注释，文档注释尽量有（除非代码结构非常简单）
3. 缩进
   - 意义：python 依赖缩进表示逻辑
   - 以 4 空格的 tab 进行缩进（tab 用空格代替）
4. 空格
   - 组成：除去**缩进**外（即非前导空格），其它可增强阅读性的空格
   - 建议：
     - 二元算术、逻辑运算符前加： `a = b + c` no a=b+c
     - 一元前缀运算符不加空格： `if !flg: pass` no if ! flg: pass
     - “:”语句尾端不加空格：如分支、循环、函数定义等，非语句尾加空格，如 dict d= {'key': 'value'}
     - 括号（含指大中小）前后不加空格：`do_some(arg1, arg2)` no do_some (arg1, arg2)
     - 逗号后加空格
5. 空行
   - 类、函数定义间加空行（类 4 行，函数 2 行）

- import 不同模块间加空行（1 行）
- 逻辑段落加空行(1 行)

6. 断行
   - 长度：为了一目了然，坚持不超过 78 个字符，若超过，则采用以下方式来解决
   - 方式：
     - 逻辑:
       - this_is_very_long_var_name.attr = this_is_very_long_var_name.attr * 2
       - > tmp = this_is_very_long_var_name
       - > tmp.attr = tmp.attr * 2
     - 语句：
       - if case1 and case2 and case3 or case4 … :
       - > if case1 and \
       - >   case2 and \
       - >   case3 or \
       - > … :

模块变量规范

1. 常量：

   字母全部大写，由下划线连接各单词，如：

   WHITE = '0xFFFFFF'

   COLOR_WHITE = '0xFFFFFF'

2. 变量：

   变量全部小写，由下划线连接各个单词，如：

   color = 'ffffff'

   color_white = 'ffffff'

   私有属性，前面加 1 个 _（下划线）

   python 动态特性，所以：

   部分全局变量**均不用**前缀（如 m、g） 不加类型信息（如：iValue, dict_obj）

3. 函数：（和变量一致）

4. 类：

首字母大写，用驼峰风格，如：

```
class ThisIsAClass(object):
        pass
```

5. 模块和包：

全部小写字母组成，用下划线连接

其他和变量差不多

6. 缩写：

function > fnc

text > txt

object > obj

count > cnt

number > num 等

7. 系统保留命名：

__xxx___

如：

```
class Base(object):
        def __init__(self):
                pass
```

语句规范

1. import 引入模块
   - 顺序： 内置 > 第三方 > 自己的
   - 一条语句 import 一个模块
   - 引入模块中多个对象，超过一行用**断行**法
   - 不使用模糊导入 from xxx import * 会造成导入同名变量的风险
2. 赋值和注释

- 不需要做无谓的对齐，浪费精力且不便于后期维护。如：

```
a =          1            # 注释
bb =         2            # 注释
c =          33333333333 # 注释
```

- 如写文章流畅直接表示即可

```
a = 1 # 注释
bb = 2 # 注释
```

3. 分支和循环,不要写成一行，如：

```
if !flg: pass
for i in range(10): print I
```

应该写得更 pythonic 点：

```
if !flg:
   pass

for I in range(10):
   print i
```

项目协作建议

1. 文件夹常规表示：

core 表示核心

util 表示工具

ext 表示扩展

bus 表示业务层

orm 表示数据层

servs 表示服务层

tmp 表示临时实验

tests 表示测试

public 表示公开的静态文件

docs 表示生成的文档

2. 功能模块定义：

servs 包下的类命名

ServXxxx

bus 包下的类命名

BusXxxxx

Orm 包下的类命名

Xxxx

Test 包下的类命名

TestXxxxx

3. 字段约定

对于orm数据层的字段约定如下：

主键：

默认：id

避免冲突：

auto id: aid

guid id: gid

时间：

date: create_on, modify_on

datetime: create_at, modify_at

second: wait_seconds (全写，简明)

物件：

编号： creator_id, modifier_id

属形： creator_xxx, bird_xxxx