**代 码 规 范**

1. **Python代码规范**
2. 包名、模块名、变量名通常由小写字母和下划线组成
3. 类名首字母用大写,其他字母采用小写
4. 函数名通常采用小写,并用下划线或单词首字母大写来增加名称的可读性,导入的函数以模块名作为前缀，如def has\_key(ch):
5. Python下划线的使用
   1. \_单下划线开头：弱“内部使用”标识，如：”from M import \*”，将不导入所有以下划线开头的对象，包括包、模块、成员
   2. 单下划线结尾\_：只是为了避免与python关键字的命名冲突
   3. \_\_双下划线开头：模块内的成员，表示私有成员，外部无法直接调用，例如类中的私有变量名称前带两个下划线 self.\_\_private\_name
   4. \_\_双下划线开头双下划线结尾\_\_：指那些包含在用户无法控制的命名空间中的“魔术”对象或属性，如类成员的\_\_name\_\_ 、\_\_doc\_\_、\_\_init\_\_、\_\_import\_\_、\_\_file\_\_、等。推荐永远不要将这样的命名方式应用于自己的变量或函数。
6. 全局常量的命名：全局常量通常定义在模块级别，以全大写命名，例如MAX\_OVERFLOW,TOTAL
7. 缩写：命名应当尽量使用全拼写的单词，缩写的情况有如下两种
   1. 常用的缩写，如xml、id等
   2. 命名中含有长单词，对某个单词进行缩写。这时应使用约定成俗的缩写方式，范例如下表

|  |  |
| --- | --- |
| function | fun |
| text | txt |
| object | obj |

1. Python命名规范总结成表格，如下

|  |  |
| --- | --- |
| **模块名** | 小写字母，单词之间用\_分割  ad\_stats.py |
| **包名** | 和模块名一样 |
| **类名** | 单词首字母大写（django中建立的model名遵循此例）  AdStats ConfigUtil |
| **全局变量名** | 大写字母，单词之间用\_分割  NUMBER COLOR\_WRITE |
| **普通变量** | 小写字母，单词之间用\_分割  this\_is\_a\_var |
| **实例变量** | 以\_开头，其他和普通变量一样  \_price \_instance\_var |
| **私有实例变量** | 以\_\_开头（2个下划线），其他和普通变量一样  \_\_private\_var |
| **专有变量** | \_\_开头，\_\_结尾，一般为python的自有变量，不要以这种方式命名  \_\_doc\_\_ \_\_class\_\_ |
| **普通函数** | 和普通变量一样：  get\_name() count\_number() ad\_stat() |
| **私有函数（外部访问会报错）** | 以\_\_开头（2个下划线），其他和普通函数一样  \_\_get\_name() |
| **单下划线结尾\_** | 只是为了避免与python关键字的命名冲突，比如str\_ |
| **\_单下划线开头** | 弱“内部使用”标识，如：”from M import \*”，将不导入所有以下划线开头的对象，包括包、模块、成员 |

1. **数据命名规范（django的model中和mysql中同时遵守）**
2. 表名、列名全部小写，单词间以下划线隔开 user\_name
3. 时间型的字段以’\_date’或’\_time’结尾
4. 布尔型字段以’is\_’作为开头
5. 外键字段使用’ 外键表名\_id’
6. 每个表都设置’create\_time’,’modify\_time’保存记录创建时间和上一次修改时间，不做任何业务使用
7. **CRUD相关命名规范**（创建、读取、更新和删除）

|  |  |
| --- | --- |
| **数据库和model的数据属性名称** | Read  Create  Update  Delete |
| **函数命名中** | Add  Del  Get  Set  示例：get\_message() |

1. **Url命名规范**

遵循restful的url规则

url命名示例如下表：(在django的urls.py中遵循此)

|  |  |
| --- | --- |
| **实例** | **含义** |
| /article/2007/8/a001 | /article/{year}/{month}/{id}  2007年8月id为a001的article的页面 |
| /business/1/message/33 | /business/{business\_id}/message/{messge\_id}  Id=1的business中id=33的message的页面 |

结合http协议的四种method完成CRUD：（服务器端和html中同时遵循此规则）

|  |  |
| --- | --- |
| 读取 | GET /article/2007/8/a001 |
| 创建 | POST /article/2007/8/a001 |
| 更新 | PUT /article/2007/8/a001 |
| 删除 | DELETE /article/2007/8/a001 |

**此规范一旦通过要求在以后的编码和对以前代码重构的工作中遵守，已有代码如有不遵守此规范之处，考虑代价地酌情修正**