**Python代码风格规范**

1. **分号**

代码行尾不要使用分号，也不要使用分号进行命令的相连。

1. **行长度**

行长度一般不要超过100字符，也尽可能的不要使用反斜杠来连接字符串，如果有必要，使用括号将其连接。

例外：

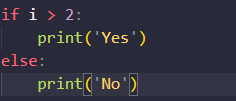
注释中的URL



导入模块的长度不必根据此有限制

1. **括号**

不要过度的使用括号，除非用于行连接



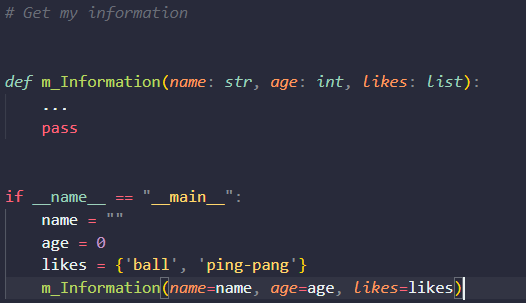
1. **缩进**

选择一种缩进方式进行缩进，严格来说是4个空格，不允许空格和tab混用

1. **空行**

严格按照：

方法之间存在一个空行，方法注释与方法相距两个空行，main与相邻上部函数有两个空行，且结尾有空行。



1. **空格**

严格使用规范使用空格：

括号内不要有空格



不要再标点前加空格，但是应该在行尾的标点后处加

二元操作符两边都加空格



1. **注释**

确保注释的风格要正确，错误的注释风格可能会造成不可不要的麻烦。

注释的使用时在函数定义之初必须执行，除非满足下面的条件：

①程序短小

②简单明了

③外部不可见

按照Google的代码风格规范，正确的注释应该是这样的（这里引用官网实例）：

**Args:**

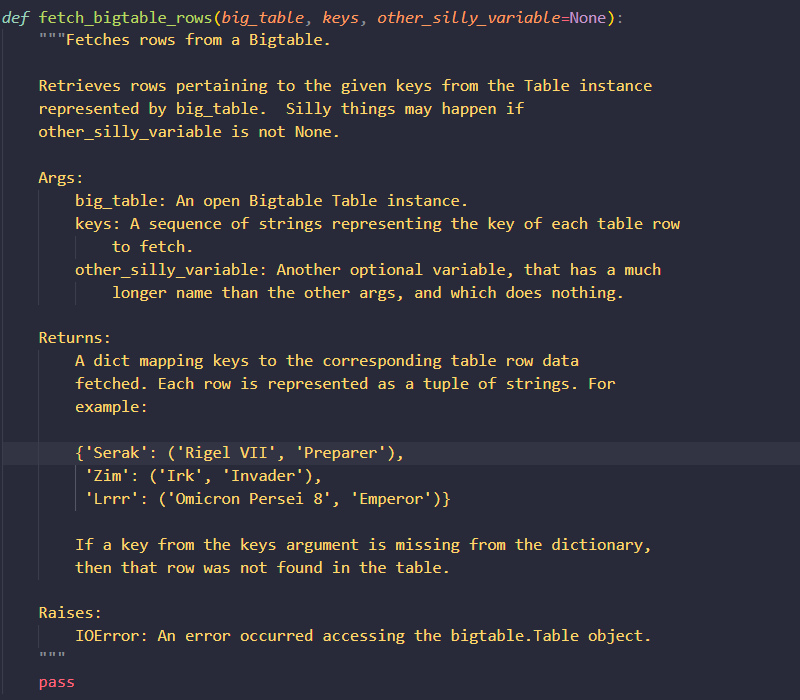
列出每个参数的名字, 并在名字后使用一个冒号和一个空格, 分隔对该参数的描述.如果描述太长超过了单行80字符,使用2或者4个空格的悬挂缩进(与文件其他部分保持一致). 描述应该包括所需的类型和含义. 如果一个函数接受\*foo(可变长度参数列表)或者\*\*bar (任意关键字参数), 应该详细列出\*foo和\*\*bar.

**Returns:** (或者 Yields: 用于生成器)

描述返回值的类型和语义. 如果函数返回None, 这一部分可以省略.

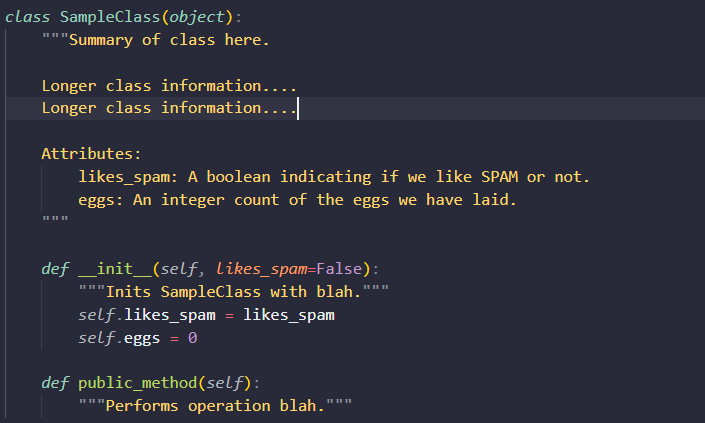
**Raises:**

列出与接口有关的所有异常.



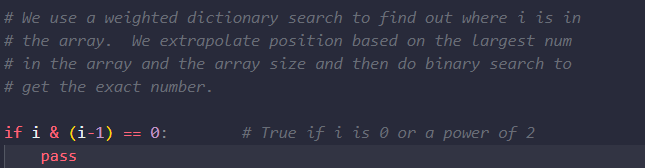
**类**

类应该在其定义下有一个用于描述该类的文档字符串. 如果你的类有公共属性(Attributes), 那么文档中应该有一个属性(Attributes)段. 并且应该遵守和函数参数相同的格式.



**块注释和行注释**

最需要写注释的是代码中那些技巧性的部分. 如果你在下次 代码审查 的时候必须解释一下, 那么你应该现在就给它写注释. 对于复杂的操作, 应该在其操作开始前写上若干行注释. 对于不是一目了然的代码, 应在其行尾添加注释.



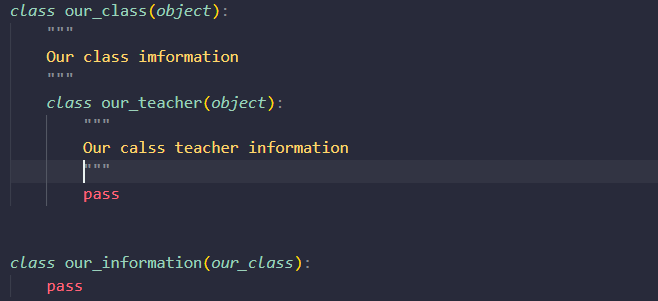
为了提高可读性, 注释应该至少离开代码2个空格.

另一方面, 绝不要描述代码. 假设阅读代码的人比你更懂Python, 他只是不知道你的代码要做什么.



1. **类**

对于类的继承，如果她不继承其他类，那么就显示的object继承



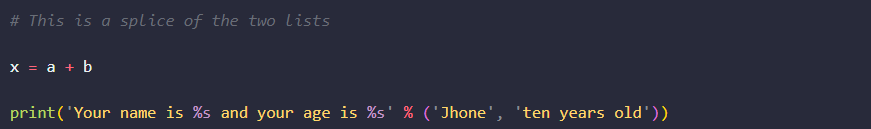
这样的好处在于能够保护自己的代码，使其免受一些不兼容性的影响

1. **字符串**

对于一般的字符串，要经常使用format进行



但也要视情况而定，在% 和 + 之间做选择



1. **导入**

每个导入应该在单独的一行，对于较长的导入要进行必要的简写



导入的位置应该位于文件的文档与函数模块介绍之后，位于所有方法之前，且应该进行字典排序式的导入，以便在查询模块时能够迅速的找到

1. **语句**

每条语句应该单独的占一行，对于if语句，当没有else的时候可以占一行



1. **命名**

自己的命名应该避免使用双下划线开头，因为这极有可能与 python本身的保留变量发生冲突

虽然python3以后支持汉字命名，但是在实际项目中任然不要使用汉字进行命名

1. **Main**

即使是一个打算被用作脚本的文件, 也应该是可导入的. 并且简单的导入不应该导致这个脚本的主功能(main functionality)被执行, 这是一种副作用. 主功能应该放在一个main()函数中.

要习惯 的使用 if \_\_name\_\_ == '\_\_main\_\_' ，因为这样在你的使用过程中会在执行主程序前总被检查，而在导入时不会被执行