

代码规范

代码规范

代码布局

缩进

制表符 or 空格

行的最大长度

二元运算符之前的换行

空行

源文件编码

包含文件的名称及次序

注释

行内注释

命名规范

最重要的原则

命名风格

应避免的名字

类名

Function Names 函数名

Constants 常量

本规范参考并修改了[PEP-0008](#)

代码布局

缩进

每一级缩进使用4个空格。

续行应该与其包裹元素对齐，要么使用圆括号、方括号和花括号内的隐式行连接来垂直对齐，要么使用挂行缩进对齐。当使用挂行缩进时，应该考虑到第一行不应该有参数，以及使用缩进以区分自己是续行。

推荐：

```
# 与左括号对齐
foo = long_function_name(var_one, var_two,
                          var_three, var_four);

# 用更多的缩进来与其他行区分
int long_function_name(
    var_one, var_two, var_three,
    var_four){
    cout << var_one;
}

# 挂行缩进应该再换一行
foo = long_function_name(
    var_one, var_two,
    var_three, var_four)
```

不推荐：

```
# 没有使用垂直对齐时，禁止把参数放在第一行
foo = long_function_name(var_one, var_two,
                          var_three, var_four)

# 当缩进没有与其他行区分时，要增加缩进
def long_function_name(
    var_one, var_two, var_three,
    var_four):
    cout << var_one;
```

四空格的规则对于续行是可选的。

在多行结构中的大括号/中括号/小括号的右括号与多行结构的第一行第一个字符对齐，就像这样：

```
result = some_function_that_takes_arguments(
    'a', 'b', 'c',
    'd', 'e', 'f',
);
```

制表符 or 空格

统一转成空格。

行的最大长度

所有行限制的最大字符数为79。

没有结构化限制的大块文本（文档字符或者注释），每行的最大字符数限制在72。

较长的代码行选择在小括号，中括号以及大括号中的隐式续行方式。通过小括号内表达式的换行方式将长串折成多行。这种方式应该优先使用,同时，反斜杠有时依然很有用。

二元运算符之前的换行

```
income = (gross_wages
          + taxable_interest
          + (dividends - qualified_dividends)
          - ira_deduction
          - student_loan_interest);
```

空行

顶层函数和类的定义，前后用两个空行隔开。

类里的方法定义用一个空行隔开。

在函数中使用空行来区分逻辑段（谨慎使用）。

源文件编码

总是以UTF-8格式编码

包含文件的名称及次序

包含文件应该按照以下顺序分组：

- C 库
- C++库
- 相关第三方库
- 本地应用/库

你应该在每一组导入之间加入空行。

推荐使用绝对路径导入，如果导入系统没有正确的配置，使用绝对路径会更加可读并且性能更好

注释

与代码相矛盾的注释比没有注释还糟，当代码更改时，优先更新对应的注释！

注释应该是完整的句子。如果一个注释是一个短语或句子，它的第一个单词应该大写，除非它是以小写字母开头的标识符(永远不要改变标识符的大小写！)。

行内注释

行内注释是与代码语句同行的注释。有节制地使用行内注释。

命名规范

最重要的原则

那些暴露给用户的API接口的命名，应该遵循反映使用场景而不是实现的原则。

命名风格

有许多不同的命名风格。以下是常见的命名方式：

- b 单个小写字母

- B 单个大写字母
- lowercase 小写字母
- lower_case_with_underscores 使用下划线分隔的小写字母
- UPPERCASE 大写字母
- UPPER_CASE_WITH_UNDERSCORES 使用下划线分隔的大写字母
- CapitalizedWords (或者叫 CapWords , 或者叫CamelCase 驼峰命名法 —— 这么命名是因为字母看上去有起伏的外观) 。有时候也被称为StudyCaps。

注意：当在首字母大写的风格中用到缩写时，所有缩写的字母用大写，因此，HTTPServerError 比 HttpServerError 好。

- mixedCase 不同于首字母大写，第一个单词的首字母小写
- Capitalized_Words_With_Underscores

应避免的名字

永远不要使用字母 'l' (小写的L) ， 'O' (大写的O) ，或者 'I' (大写的I) 作为单字符变量名。

在有些字体里，这些字符无法和数字0和1区分，如果想用 'l' ，用 'L' 代替。

类名

类名一般使用首字母大写的约定。

Function Names 函数名

函数名采用 `mixedCase` 这样的大小写混合

Constants 常量

常量通过下划线分隔的全大写字母命名。例如：MAX_OVERFLOW 和 TOTAL。

