[研发部 C语言编码风格指南](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760)

[背景](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E8%83%8C%E6%99%AF)

本指南的目标是通过详细描述编写C语言代码的注意事项，消除语言复杂性，使代码库易于管理。

代码风格，也称为可读性，就是我们称之为管理C语言编码规范的约定。

请注意，本指南不是C语言教程：我们假设读者熟悉该语言。

[C语言版本](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#c%E8%AF%AD%E8%A8%80%E7%89%88%E6%9C%AC)

代码使用C89标准，亦称C90标准。勿使用C11特性。勿使用GNU扩展。

从使用频率角度，C89标准影响最大的为：

1. 使用/\* \*/块注释而非//行注释；
2. **for**循环的循环变量必须定义在循环体之前。

/\* C89只支持块注释。而且循环表达式中不能定义变量，必须提前定义 \*/

**int** i;

**for** (i = 0; i < n; ++i)

{

}

[头文件](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%A4%B4%E6%96%87%E4%BB%B6)

通常，每个.c文件都应该有一个关联的.h文件。有一些常见的例外，例如单元测试和.c仅包含**main**()函数的小文件 。

正确使用头文件会对代码的可读性，大小和性能产生巨大影响。

以下规则将指导您使用头文件的各种麻烦。

[**#define 保护**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#define-%E4%BF%9D%E6%8A%A4)

所有头文件都应该有#**define**保护，以防止多次包含。符号名称的格式应为

\_<**PROJECT**>\_<**PATH**>\_<**FILE**>\_H\_

为了保证唯一性，它们应该基于项目源树中的完整路径。例如，foo/src/bar/baz.h项目中的文件foo应具有以下保护：

#**ifndef** \_FOO\_BAR\_BAZ\_H\_

#**define** \_FOO\_BAR\_BAZ\_H\_

...

#**endif** /\* \_FOO\_BAR\_BAZ\_H\_ \*/

[**头文件依赖**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%A4%B4%E6%96%87%E4%BB%B6%E4%BE%9D%E8%B5%96)

头文件中不使用前置声明，直接#**include**所需的其他头文件。

**注意：** 只包含且必须包含必要的头文件，避免包含不需要的头文件。

**例外：** SDK接口的**Type**\*句柄类型，可使用头文件中前置声明的方式，隐藏实现细节。

[**-inl.h文件**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#-inlh%E6%96%87%E4%BB%B6)

复杂的内联函数定义，应放在后缀名为**-inl**.h的头文件中。

和其他头文件一样，也需要#**define**保护。

[**包含头文件的名词及次序**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%8C%85%E5%90%AB%E5%A4%B4%E6%96%87%E4%BB%B6%E7%9A%84%E5%90%8D%E8%AF%8D%E5%8F%8A%E6%AC%A1%E5%BA%8F)

包含次序标准化可增强可读性，避免依赖。次序为：C库、其他库的.h、项目内的.h。

即： C系统文件 其他库头文件 本项目内头文件

项目内头文件按照项目源码的目录树结构排序，避免使用UNIX路径.和..

**总结：**

(1) 避免多重包含

(2) 前置声明是为了降低编译依赖，防止修改一个头文件引发多米诺效应

(3) 内联函数的合理使用可提高代码的执行效率

(4) **-inl**.h可提高代码可读性

(5) 包含文件使用.和..虽然方便，却易混乱，使用完整项目路径更清晰、美观。

[函数](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%87%BD%E6%95%B0)

[**编写短小函数**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E7%BC%96%E5%86%99%E7%9F%AD%E5%B0%8F%E5%87%BD%E6%95%B0)

函数长度尽量不超过40行，倾向选择短小、凝练的函数。

函数尽量短小、简单，便于他人阅读和修改代码。处理代码时如发现长函数，考虑分割为若干短小、易于管理的若干函数。

[**函数参数顺序**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%87%BD%E6%95%B0%E5%8F%82%E6%95%B0%E9%A1%BA%E5%BA%8F)

定义参数时，参数顺序为：输入参数在前，输出参数在后。

输入参数为传值或常量，输出参数为非常数指针。

本条应尽量遵守，函数中的输入输出两用参数作为中间参数。

/\* input: arg1, arg2, arg3

\* input/output: arg4

\* output: arg5, arg6

\*/

**int** **InputAndOutputFunction**(**int** arg1, **float** arg2, **const** struct Type3 \*arg3, struct Type4 \*arg4, struct Type5 \*arg5, struct Type5 \*\*arg6);

[**参数传递类型**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%8F%82%E6%95%B0%E4%BC%A0%E9%80%92%E7%B1%BB%E5%9E%8B)

基本数据类型和枚举类型形参允许使用值传递。

对于**struct**类型，统一使用指针传递：只读输入形参用**const** **Type**\*；输出形参使用**Type**\*。若需要在参数中返回指针类型，则使用**Type**\*\*。

[**返回函数中malloc的指针**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E8%BF%94%E5%9B%9E%E5%87%BD%E6%95%B0%E4%B8%ADmalloc%E7%9A%84%E6%8C%87%E9%92%88)

遵循谁分配谁释放的原则：如果接口函数中分配了指针空间并返回给上层调用者，则接口函数应提供配套的指针释放函数。

/\*\*

\* 接口函数

\*/

/\* 获取信息结构体指针 \*/

struct Information \***GetInformation**(...)

{

**return** (struct Information \*)malloc(...);

}

/\* 释放信息结构体指针 \*/

**void** **FreeInformation**(struct Information \*pInformation)

{

free(pInformation);

}

/\*\*

\* 上层调用

\*/

**void** **ProcessInformation**()

{

**struct** **Information** \***pInformation** = **GetInformation**(...);

/\* do something with pInformation

...\*/

FreeInformation(pInformation);

}

[**禁止使用缺省函数参数（测试框架例外）**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E7%A6%81%E6%AD%A2%E4%BD%BF%E7%94%A8%E7%BC%BA%E7%9C%81%E5%87%BD%E6%95%B0%E5%8F%82%E6%95%B0%E6%B5%8B%E8%AF%95%E6%A1%86%E6%9E%B6%E4%BE%8B%E5%A4%96)

[**局部变量**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%B1%80%E9%83%A8%E5%8F%98%E9%87%8F)

局部变量应统一在函数开头位置声明。

[其他C语言特性](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%85%B6%E4%BB%96c%E8%AF%AD%E8%A8%80%E7%89%B9%E6%80%A7)

[**禁止使用变长数组**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E7%A6%81%E6%AD%A2%E4%BD%BF%E7%94%A8%E5%8F%98%E9%95%BF%E6%95%B0%E7%BB%84)

[**前置自增和自减**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%89%8D%E7%BD%AE%E8%87%AA%E5%A2%9E%E5%92%8C%E8%87%AA%E5%87%8F)

迭代器和其他模板对象要求使用前置自增、自减运算符。其他场景优先使用前置自增、自减。

[**const的使用**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#const%E7%9A%84%E4%BD%BF%E7%94%A8)

任何可以使用的场景都要使用**const**，且**const**在前。

[**使用确定大小的整型**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E4%BD%BF%E7%94%A8%E7%A1%AE%E5%AE%9A%E5%A4%A7%E5%B0%8F%E7%9A%84%E6%95%B4%E5%9E%8B)

使用确定大小的整型，除表示数组位数外的场景禁止使用**unsigned**无符号型。

[**可移植性**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%8F%AF%E7%A7%BB%E6%A4%8D%E6%80%A7)

格式化输出及结构对齐时，注意32位和64位系统差异

[**0 和 NULL**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#0-%E5%92%8C-null)

整数用0，实数用0.0，指针用NULL，字符（串）用'\0'

[**sizeof**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#sizeof)

用 sizeof(**variableName**) 替代 sizeof(**TypeName**)

[命名约定](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%91%BD%E5%90%8D%E7%BA%A6%E5%AE%9A)

一致性命名规则，包括类型、变量、函数、常量、宏等等。

[**通用命名规则**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E9%80%9A%E7%94%A8%E5%91%BD%E5%90%8D%E8%A7%84%E5%88%99)

函数、变量、文件命名应具有描述性，不要过分缩写。

类型和变量应该是名词，函数名用"命令性质"的动词。

命名方法：

尽可能给出描述性名词，提高代码可读性，目标是让别人快速理解。

禁止使用模糊的缩写或随意的字符。

类型和变量名一般为名词，如numErrors。禁止使用泛称，尽量用完整的单词描述清楚作用和意图，不要怕字多。例如**struct** **ProcData**就不是好的命名。

函数名通常是指令性的，如OpenFile()、SetNumErrors()，访问函数需要描述的更细致，要与其访问的变量相吻合。

缩写：

除非是放在项目外也非常明了的缩写，如IOT、DNS、IP等

除linux命令常用缩写（如 ls、pwd、usr等）外，不要用省略字母的缩写。如count不要写成cnt

[**文件命名**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E6%96%87%E4%BB%B6%E5%91%BD%E5%90%8D)

文件名要全部小写，可以包含下划线（\_）。已有命名约定要求的项目遵循项目要求。

不要使用已经存在于系统中的文件名，如Linux系统不要使用/usr/include下的文件名，如db.h。

[**类型命名**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E7%B1%BB%E5%9E%8B%E5%91%BD%E5%90%8D)

类型命名每个单词以大写字母开头，不包含下划线，如：MyExcitingClass、MyExcitingEnum。

所有类型命名————结构体、类型定义（typedef）、枚举，使用相同约定，例如：

/\* structs \*/

**struct** **UrlTable** {...};

**struct** **UrlTableTester** {...};

**struct** **UrlTableProperties** {...};

/\* typedefs \*/

**typedef** LongSourceType PropertiesMap {...}

/\* enums \*/

**enum** UrlTableErrors{...}

[**变量命名**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%8F%98%E9%87%8F%E5%91%BD%E5%90%8D)

变量名使用驼峰命名法，第一个单词以小写开头，其他单词首字母大小，单词之间不使用分隔符，类的私有成员以下划线结尾，如

myExcitingLocalVariable、myExcitingLocalVariable\_

[**全局变量命名**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%85%A8%E5%B1%80%E5%8F%98%E9%87%8F%E5%91%BD%E5%90%8D)

不建议使用全局变量。如果一定要用，以g\_为前缀以便与局部变量区分。

[**静态变量命名**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E9%9D%99%E6%80%81%E5%8F%98%E9%87%8F%E5%91%BD%E5%90%8D)

不建议使用静态变量。如果一定要用，以s\_为前缀以便与非静态变量区分。静态全局变量仅需添加s\_前缀。

[**常量命名**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%B8%B8%E9%87%8F%E5%91%BD%E5%90%8D)

常量命名全部大写，单词间以下划线分隔，如：

**const** **int** DAYS\_IN\_A\_WEEK = 7;

[**函数命名**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%87%BD%E6%95%B0%E5%91%BD%E5%90%8D)

普通函数，函数名大写字母开头，每个单词首字母大写，没有下划线：

AddTableEntry();

[**枚举命名**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E6%9E%9A%E4%B8%BE%E5%91%BD%E5%90%8D)

枚举值全部大写，单词之间用下划线分隔

MY\_EXCITING\_ENUM\_VALUE。

枚举名称属于类型，因此大小写混合。

**enum** UrlTableErrors {

OK = 0,

ERROR\_OUT\_OF\_MEMORY = 1,

ERROR\_MALFORMED\_INPUT = 2,

};

[**宏命名**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%AE%8F%E5%91%BD%E5%90%8D)

尽量使用枚举、常量、内联替代宏。

如果要使用宏，宏的名称全部大写，单词之间用下划线分隔

MY\_MACRO\_THAT\_SCARES\_SMALL\_CHILDREN\_AND\_ADULTS\_ALIKE

[**命名规则例外**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%91%BD%E5%90%8D%E8%A7%84%E5%88%99%E4%BE%8B%E5%A4%96)

参考现有或相近命名约定

[代码注释](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E4%BB%A3%E7%A0%81%E6%B3%A8%E9%87%8A)

[**注释风格**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E6%B3%A8%E9%87%8A%E9%A3%8E%E6%A0%BC)

说明代码功能的关键注释统一使用/\* \*/块注释。

面向外部的头文件中，统一使用Doxygen注释风格。

[**文件注释**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E6%96%87%E4%BB%B6%E6%B3%A8%E9%87%8A)

文件注释需说明当前文件在项目中的核心作用。并使用Doxygen注释风格：

/\*\*

\* @file Sample.h

\* @author Author Name (username@ghostcloud.cn)

\* @brief Doxygen 注释风格示例

\* @version 0.1

\* @date 2021-03-11

\*

\* @copyright Copyright (c) 2021

\*

\*/

#**ifndef** \_GHOSTCLOUD\_PENGZE\_SAMPLE\_H\_

#**define** \_GHOSTCLOUD\_PENGZE\_SAMPLE\_H\_

#**endif** /\* !\_GHOSTCLOUD\_PENGZE\_SAMPLE\_H\_ \*/

[**结构体注释**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E7%BB%93%E6%9E%84%E4%BD%93%E6%B3%A8%E9%87%8A)

结构体的头文件中，对数据成员做说明。

[**函数注释**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%87%BD%E6%95%B0%E6%B3%A8%E9%87%8A)

函数申明处注释的内容：

(1) 函数作用：当前函数需要达成的目的

(2) 函数参数：说明是输入还是输出，及参数含义

(3) 返回值：含义和类型

面向外部的接口函数注释使用Doxygen风格：

/\*\*

\* @brief 创建一个示例对象

\*

\* @details 详细说明（可选）

\*

\* @note 注意事项说明（可选）

\*

\* @param arg0 参数0，参数0说明

\* @param arg1 参数1，参数1说明

\* @param arg2 参数2，参数2说明

\*

\* @return int ErrorCode

\* @retval 0 成功

\* @retval -1 失败

\*

\*/

**int** **CreateSample**(**int** arg0, **int** arg1, **int** arg2);

[**变量注释**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%8F%98%E9%87%8F%E6%B3%A8%E9%87%8A)

变量在上一行使用 /\* \*/注释

[**实现注释**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%AE%9E%E7%8E%B0%E6%B3%A8%E9%87%8A)

复杂或重要的逻辑实现，需要说明实现的意图、目的、主要思想。

**注意：** 注释是对逻辑的解释，而非代码的翻译。要假定代码阅读者比你更熟悉C++语言本身，但他不明白为什么要这样实现。

[**标点、拼写和语法**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E6%A0%87%E7%82%B9%E6%8B%BC%E5%86%99%E5%92%8C%E8%AF%AD%E6%B3%95)

注释要易读

[**TODO**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#todo)

临时、短期的解决方案，需要使用TODO注释。

TODO大写，需要备注作者名，备查找：

/\* TODO(Luffy): 需要优化逻辑 \*/

[代码格式](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E4%BB%A3%E7%A0%81%E6%A0%BC%E5%BC%8F)

1个Tab = 4个空格

[**行长度**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E8%A1%8C%E9%95%BF%E5%BA%A6)

每一行字符数不超过80

这一条存在例外情况：

(1) 如果一行注释包含了超过80字符的命令或URL，出于复制粘贴方便可以超过80字符；

(2) 包含长路径的可以超过80列，尽量避免

(3) 头文件保护可以不遵从该原则

[**非ASCII字符**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E9%9D%9Eascii%E5%AD%97%E7%AC%A6)

尽量不使用ASCII字符，使用时必须使用UTF-8格式

[**空格 or 制表符Tab**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E7%A9%BA%E6%A0%BC-or-%E5%88%B6%E8%A1%A8%E7%AC%A6tab)

只使用Tab，约定Tab长度为4个空格

[**函数声明与定义**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%87%BD%E6%95%B0%E5%A3%B0%E6%98%8E%E4%B8%8E%E5%AE%9A%E4%B9%89)

返回类型和函数名在同一行，参数也在同一行。

如果一行文本较多写不下所有参数，则参数换行Tab对齐：

ReturnType **ReallyReallyReallyLongFunctionName**(

Type par\_name1,

Type par\_name2,

Type par\_name3)

{

/\* 2 space indent \*/

DoSomething();

...

}

**注意：** (1) 返回值和函数在同一行

(2) 左圆括号(总是和函数名在同一行

(3) 函数名和左圆括号之间没有空格

(4) 圆括号与参数没有空格

(5) 函数的左花括号另起一行，单独置于一行

(6) 函数的右花括号在函数最后，单独置于一行

(7) 右圆括号和左花括号间换行

(8) 函数声明和实现处的所有形参名称必须保持一致

(9) 所有形参尽可能对齐

(10) 默认缩进为1个Tab，即4个空格

(11) 独立封装的参数保持4个空格缩进

const函数，关键字const和最后一个参数置于同一行

ReturnType LongClassName::ReallyReallyReallyLongFunctionName(

Type par\_name1,

Type par\_name2,

Type par\_name3) **const**

{

DoSomething();

...

}

用不到的参数需在函数的注释中说明

[**函数调用**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%87%BD%E6%95%B0%E8%B0%83%E7%94%A8)

尽量放在同一行，否则将括号内的参数换行对齐

[**条件语句**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E6%9D%A1%E4%BB%B6%E8%AF%AD%E5%8F%A5)

条件语句的左花括号放在条件语句的下一行行首。

注意：条件内哪怕只有一行语句，也要写全花括号，方便后续维护。

有些条件语句写在一行可以增强可读性，仅当语句简单且没有else子句时可以使用：

**if** (x == kFoo) **return** **new** Foo();

**if** (x == kBar) **return** **new** Bar();

注意：如果有else子句，则禁止不带花括号。

[**循环和开关选择语句（loop and switch）**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%BE%AA%E7%8E%AF%E5%92%8C%E5%BC%80%E5%85%B3%E9%80%89%E6%8B%A9%E8%AF%AD%E5%8F%A5loop-and-switch)

switch 语句尽量使用大括号分块

switch 语句必须要有default存在，如正常不会执行则以异常日志或控制台信息的方式处理。

空循环使用{}或者continue，禁止循环只用一个分号

[**指针表达式**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E6%8C%87%E9%92%88%E8%A1%A8%E8%BE%BE%E5%BC%8F)

句号.、箭头->前后不要有空格，指针\*、地址操作符&后不要有空格

value = \*test;

value = &test;

value = c.test;

value = c->test;

**char** \*cinput;

**const** **char** \*strName;

[**布尔表达式**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%B8%83%E5%B0%94%E8%A1%A8%E8%BE%BE%E5%BC%8F)

如果布尔表达式超过标准行宽（80字符），则断行且要求逻辑操作符置于行尾。

**if** (thisOneThing > thisOtherThing &&

aThirdThing == aFourthThing &&

yetAnother & lastOne) {

...

}

有多个逻辑操作符时，考虑使用圆括号()提示运算优先级

[**返回值**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E8%BF%94%E5%9B%9E%E5%80%BC)

函数返回（即：**return** 表达式）时不要使用圆括号。

[**初始化操作**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%88%9D%E5%A7%8B%E5%8C%96%E6%93%8D%E4%BD%9C)

使用等号=完成

[**预处理**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E9%A2%84%E5%A4%84%E7%90%86)

不要缩进，从行首开始。及时在缩进代码块中，也要从行首开始。

[**类格式**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E7%B1%BB%E6%A0%BC%E5%BC%8F)

类格式参考本规范声明次序章节。

[**初始化列表**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%88%9D%E5%A7%8B%E5%8C%96%E5%88%97%E8%A1%A8)

初始化列表置于同一行，或换行Tab补齐

[**命名空间格式化**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%91%BD%E5%90%8D%E7%A9%BA%E9%97%B4%E6%A0%BC%E5%BC%8F%E5%8C%96)

命名空间内容不缩进。

[**水平空白**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E6%B0%B4%E5%B9%B3%E7%A9%BA%E7%99%BD)

行尾不要增加无意义的空白。当行尾有注释需要Tab补齐时可以适当留空。

[**垂直空白**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E5%9E%82%E7%9B%B4%E7%A9%BA%E7%99%BD)

垂直空白尽量少。函数之间、模块之间留一样空白

[**规则之例外**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#%E8%A7%84%E5%88%99%E4%B9%8B%E4%BE%8B%E5%A4%96)

前文提到的编码习惯基本是强制性的，但优秀的规则都允许例外。

[**Windows代码**](http://newben.ghostcloud.cn:30038/pages/viewpage.action?pageId=25461760#windows%E4%BB%A3%E7%A0%81)

在Windows平台开发，除系统API不能更改外，自主开发内容需遵循本规则。