简体中文 English Spanish 日本語

# Qt 编码风格

这是我们编写Qt代码时所使用的编码惯例的一个概述。数据是通过挖掘Qt源码、论坛、邮件列表以及与开发者的协作而收集起来的。

Table of Content



## 缩进

采用4个空格

空格，不要用TAB!

## 变量声明

每行一个变量

尽可能避免短的变量名(比如“a”，“rbarr”，“nughdeget”)

单字符的变量只在临时变量或循环的计数中使用

等到真正需要使用时再定义变量

```
// Wrong
int a, b;
char *c, *d;

// Correct
int height;
int width;
char *nameOfThis;
char *nameOfThat;
```

以小写字符开头，后续单词以大写开头

避免使用缩写

```
// Wrong
short Cntr;
char ITEM_DELIM = '\t';

// Correct
short counter;
char itemDelimiter = '\t';
```

类名总是以大写开头。公有类以Q开头(QRgb)，公有函数通常以q开头(qRgb)。

## 空白

利用空行将语句恰当地分组

总是使用一个空行(不要空多行)

总是在每个关键字和大括号前使用一个空格

```
// Wrong
if(foo){
}

// Correct
if (foo) {
}
```

对指针和引用，在类型和*、&之间加一个空格，但在*、&与变量之间不加空格

```
char *x;
const QString &myString;
const char * const y = "hello";
```

二元操作符前后加空白

类型转换后不加空白

尽量避免C风格的类型转换

```
// Wrong
char* blockOfMemory = (char* ) malloc(data.size());

// Correct
char *blockOfMemory = reinterpret_cast<char *>(malloc(data.size()));
```

## 大括号

基本原则：左大括号和语句保持在同一行：

```
// Wrong
if (codec)
{
}

// Correct
if (codec) {
}
```

例外：函数定义和类定义中，左大括号总是单独占一行：

```
static void foo(int g)
{
    qDebug("foo: %i", g);
}

class Moo
{
};
```

控制语句的body中只有一行时不使用大括号

```
// Wrong
if (address.isEmpty()) {
    return false;
}

for (int i = 0; i < 10; ++i) {
    qDebug("%i", i);
}

// Correct
if (address.isEmpty())
    return false;

for (int i = 0; i < 10; ++i)
```

```
            qDebug("%i", i);
```

例外1：如果父语句跨多行，则使用大括号

```
        // Correct
        if (address.isEmpty() || !isValid()
            || !codec) {
            return false;
        }
```

例外2：在if-else结构中，有一处跨多行，则使用大括号

```
        // Wrong
        if (address.isEmpty())
            return false;
        else {
            qDebug("%s", qPrintable(address));
            ++it;
        }

        // Correct
        if (address.isEmpty()) {
            return false;
        } else {
            qDebug("%s", qPrintable(address));
            ++it;
        }

        // Wrong
        if (a)
            if (b)
                ...
            else
                ...

        // Correct
        if (a) {
            if (b)
                ...
            else
                ...
        }
```

如果控制语句的body为空，则使用大括号

```
        // Wrong
        while (a);

        // Correct
        while (a) {}
```

## 圆括号

使用圆括号将表达式分组

```
        // Wrong
        if (a && b || c)

        // Correct
        if ((a && b) || c)

        // Wrong
        a + b & c

        // Correct
        (a + b) & c
```

## Switch 语句

case 和 switch 位于同一列

每一个case必须有一个break(或renturn)语句，或者用注释说明无需break

```
        switch (myEnum) {
        case Value1:
            doSomething();
            break;
        case Value2:
            doSomethingElse();
            // fall through
        default:
            defaultHandling();
            break;
        }
```

## 断行

保持每行短于100 个字符，需要时进行断行

逗号放一行的结束，操作符放到一行的开头。如果你的编辑器太窄，一个放在行尾的操作符不容易被看到。

```
        // Correct
        if (longExpression
            + otherLongExpression
            + otherOtherLongExpression) {
        }

        // Wrong
        if (longExpression +
            otherLongExpression +
            otherOtherLongExpression) {
        }
```

## 继承与关键字 `virtual`

重新实现一个虚函数时，头文件中 不 放置 virtual 关键字。

## 通用例外

如果它使你的代码看起来不好，你可以打破任何一个规则 。

注：译文一开始发表在 CSDN博客 [blog.csdn.net] ，有问题欢迎讨论

Categories:

Developing_Qt
    Guidelines

SimplifiedChinese
    Developing_Qt