

# DEVBEAN'S WORLD

Colorful Technologies...

🏠 主页 / C++ / C++11 新特性：用户定义字面量

## C++11 新特性：用户定义字面量

👤 豆子 | 📅 2012年5月20日 | 📁 C++ | 💬 3条评论

参考文章：[https://blogs.oracle.com/pcarlini/entry/c\\_1x\\_tidbits\\_user\\_defined](https://blogs.oracle.com/pcarlini/entry/c_1x_tidbits_user_defined)

今天说的是一个小小的改进，但是许多人都会喜欢的。

来看一下下面的程序：

C++

```
1 #include <iostream>
2
3 long double operator"" _mm(long double x) { return x / 1000; }
4 long double operator"" _m(long double x) { return x; }
5 long double operator"" _km(long double x) { return x * 1000; }
6
7 int main()
8 {
9     std::cout << 1.0_mm << '\n';
10    std::cout << 1.0_m << '\n';
11    std::cout << 1.0_km << '\n';
12 }
```

其输出是：

```
1 0.001
2 1
3 1000
```

这种新语法其实很容易理解： `#include` 之后的三行代码定义了一个用户自定义的新的类型的操作符，称为字面量操作符 *literal operator*。在这个例子中，这个运算符能够转换相应的长度单位，例如，1 mm = 10<sup>-3</sup> m，1 km = 10<sup>3</sup> m，而 1 m = 1 m。因此，我们的操作符就可以自动计算每个长度单位是多少米。

搜索

搜索站点

订阅到邮箱

📧 订阅到邮箱

如果觉得还不错...

使用支付宝钱包扫描二维码进行捐助



在开始试验这个特性之前，我们应该了解这种字面量操作符的参数数量和类型的限制。根据 C++ 11 标准，只有下面这些签名是合法的：

```
1 char const*
2 unsigned long long
3 long double
4 char const*, std::size_t
5 wchar_t const*, std::size_t
6 char16_t const*, std::size_t
7 char32_t const*, std::size_t
```

最后四个对于字符串相当有用，因为第二个参数会自动推断为字符串的长度。例如：

```
1 std::size_t operator"" _len(char const*, std::size_t l)
2 {
3     return l;
4 }
5
6 int main()
7 {
8     std::cout << "ABCDEFGH"_len << '\n';
9 }
```

其输出是 8。

上面列出的第一个签名不要同字符串相混淆，应该被称为原始字面量 *raw literal* 操作符。例如：

```
1 char const* operator"" _r(char const* s)
2 {
3     return s;
4 }
5
6 int main()
7 {
8     std::cout << 12_r << '\n';
9 }
```

其输出值是 12。

字面量的返回值并没有被严格限定。我们完全可以提供相容类型的返回值。例如：

```
1 std::string operator"" _rs(char const* s)
2 {
3     return 'x' + std::string(s) + 'y';
4 }
5
6 int main()
7 {
8     std::cout << 5_rs << '\n';
9 }
```



### Paypal



### 标签

2013   2014   C++   C++11   canvas  
Flex   git   HTML5   Java   Objective-C  
OrbitsWriter   QML   Qt   Qt5  
QtCreator   QtQuick   sqlite   STL   style  
XML-RPC   学习   学习之路   插件   照片  
生活   目录   组件   进阶   随笔

### 分类目录

分类目录

选择分类目录 ▼

### 相关文章

C++11 新特性：非 static 数据成员初始化器 五月 11, 2012