

## 1.原始字面量

定义方式为：R “xxx(原始字符串)xxx”

与python类似

在 R"(D:\hello\world\test.text)" 使用了原始字面量 R () 中的内容就是描述路径的原始字符串，无需做任何处理; 输出是 D:\hello\world\test.text

## 2.nullptr

在 C++ 程序开发中，为了提高程序的健壮性，一般会在定义指针的同时完成初始化操作，或者在指针的指向尚未明确的情况下，都会给指针初始化为 NULL，避免产生野指针（没有明确指向的指针，操作也这种指针极可能导致程序发生异常）。C++98/03 标准中，将一个指针初始化为空指针的方式有 2 种：

```
char *ptr = 0;
char *ptr = NULL;
```

在底层源码中 NULL 这个宏是这样定义的:

```
#ifndef NULL
#ifdef __cplusplus
#define NULL 0
#else
#define NULL ((void *)0)
#endif
#endif
```

也就是说如果源码是 C++ 程序 NULL 就是 0，如果是 C 程序 NULL 表示 (void\*)0。

那么为什么要这样做呢？是由于 C++ 中，void \* 类型无法隐式转换为其他类型的指针，此时使用 0 代替 ((void \*)0)，用于解决空指针的问题。

这个 0 (0x0000 0000) 表示的就是虚拟地址空间中的 0 地址，这块地址是只读的。

## 3.constexpr修饰常量表达式

C++ 程序从编写完毕到执行分为四个阶段：预处理、编译、汇编和链接 4 个阶段，得到可执行程序之后就可以运行了。需要额外强调的是，常量表达式和非常量表达式的计算时机不同，非常量表达式只能在程序运行阶段计算出结果，但是常量表达式的计算往往发生在程序的编译阶段，这可以极大提高程序的执行效率，因为表达式只需要在编译阶段计算一次，节省了每次程序运行时都需要计算一次的时间。

语法: 变量直接修饰即可, 不可用于修饰类, 但在实例化对象时可以修饰;

可以修饰函数->

`constexpr` 并不能修改任意函数的返回值，当这些函数成为常量表达式函数，必须要满足以下几个条件：

1. 函数必须要有返回值，并且 `return` 返回的表达式必须是常量表达式。
2. 函数在使用之前，必须有对应的定义语句。
3. 整个函数的函数体中，不能出现非常量表达式之外的语句（`using` 指令、`typedef` 语句以及 `static_assert` 断言、`return` 语句除外）。