

## C++ 引用

引用变量是一个别名，也就是说，它是某个已存在变量的另一个名字。一旦把引用初始化为某个变量，就可以使用该引用名称或变量名称来指向变量。

## C++ 引用 vs 指针

引用很容易与指针混淆，它们之间有三个主要的不同：

- 不存在空引用。引用必须连接到一块合法的内存。
- 一旦引用被初始化为一个对象，就不能被指向到另一个对象。指针可以在任何时候指向到另一个对象。
- 引用必须在创建时被初始化。指针可以在任何时间被初始化。

## C++ 中创建引用

试想变量名称是变量附属在内存位置中的标签，您可以把引用当成是变量附属在内存位置中的第二个标签。因此，您可以通过原始变量名称或引用来访问变量的内容。例如：

```
int i = 17;
```

我们可以为 i 声明引用变量，如下所示：

```
int& r = i;  
double& s = d;
```

在这些声明中，& 读作**引用**。因此，第一个声明可以读作 "r 是一个初始化为 i 的整型引用"，第二个声明可以读作 "s 是一个初始化为 d 的 double 型引用"。下面的实例使用了 int 和 double 引用：

### 实例

```
#include <iostream>  
using namespace std;  
int main ()  
{  
    // 声明简单的变量  
    int i;  
    double d;  
    // 声明引用变量  
    int& r = i;  
    double& s = d;  
    i = 5;  
    cout << "Value of i : " << i << endl;  
    cout << "Value of i reference : " << r << endl;  
    d = 11.7;  
    cout << "Value of d : " << d << endl;  
}
```

```
cout << "Value of d reference : " << s << endl;
return 0;
}
```

当上面的代码被编译和执行时，它会产生下列结果：

```
Value of i : 5
Value of i reference : 5
Value of d : 11.7
Value of d reference : 11.7
```

引用通常用于函数参数列表和函数返回值。下面列出了 C++ 程序员必须清楚的两个与 C++ 引用相关的重要概念：

概念	描述
<a href="#">把引用作为参数</a>	C++ 支持把引用作为参数传给函数，这比传一般的参数更安全。
<a href="#">把引用作为返回值</a>	可以从 C++ 函数中返回引用，就像返回其他数据类型一样。

3 篇笔记

写笔记