

# c/c++中sizeof()、strlen()、length()、size() 详解和区别

- **c/c++中获取字符串长度。有以下函数：  
size()、sizeof()、strlen()、str.length()**
- **c/c++ strlen(str)和str.length()和str.size()都可以求字符串长度。**
- **其中str.length()和str.size()是用于求string类对象的成员函数**
- **strlen(str)是用于求字符数组的长度，其参数是char\*。**

# 一、数组或字符串的长度:sizeof()、strlen()

1、sizeof():返回所占总空间的字节数

(1)、对于整型字符型数组

(2)、对于整型或字符型指针

2、strlen():返回字符数组或字符串所占的字节数

(1)、针对字符数组

(2)、针对字符指针

- **sizeof(...)**是运算符，其值在编译时即计算好了，参数可以是数组、指针、类型、对象、函数等。它的功能是：获得保证能容纳实现所建立的最大对象的字节大小。由于在编译时计算，因此**sizeof**不能用来返回动态分配的内存空间的大小。

- **strlen(...)**是函数，要在运行时才能计算。参数必须是字符型指针（char\*）。当数组名作为参数传入时，实际上数组就退化成指针了。它的功能是：返回字符串的长度。该字符串可能是自己定义的，也可能是内存中随机的，该函数实际完成的功能是从代表该字符串的第一个地址开始遍历，直到遇到结束符'\0'。返回的长度大小不包括'\0'。
- **strlen(char\*)**函数求的是字符串的实际长度，它求得方法是从开始到遇到第一个'\0'，如果你只定义没有给它赋初值，这个结果是不定的，它会从aa首地址一直找下去，直到遇到'\0'停止。

# 举例:

- 1) `char* ss = "0123456789";`  
`sizeof(ss)`为4, `ss`是指向字符串常量的字符指针, `sizeof` 获得的是指针所占的空间,则为4  
`sizeof(*ss)`为1, `*ss`是第一个`char`字符, 则为1
- 2) `char ss[] = "0123456789";`  
`sizeof(ss)`为11, `ss`是数组, 计算到`'\0'`位置, 因此是(10+1)  
`sizeof(*ss)`为1, `*ss`是第一个字符
- 3) `char ss[100] = "0123456789";`  
`sizeof(ss)`为100, `ss`表示在内存中预分配的大小, `100*1`  
`strlen(ss)`为10, 它的内部实现用一个循环计算字符串的长度,直到`'\0'`为止。
- 4) `int ss[100] = "0123456789";`  
`sizeof(ss)`为400, `ss`表示在内存中预分配的大小, `100*4`  
`strlen(ss)`错误, `strlen`参数只能是`char*`, 且必须是以`'\0'`结尾
- 5) `char[] a={'a','b','c'};`  
`sizeof(a)`的值应该为3。  
`char[] b={"abc"};`  
`sizeof(b)`的值应该是4。  
若`string str={'a','b','c','\0','X'};`  
那么`sizeof(str)`为5, `strlen(str)`为3。

# sizeof()、strlen()两者区别:

1) sizeof操作符的结果类型是size\_t, 它在头文件中typedef为unsigned int类型。

该类型保证能容纳实现所建立的最大对象的字节大小。

2) sizeof是运算符, strlen是函数。

3) sizeof可以用类型做参数, strlen只能用char\*做参数, 且必须是以'\0'结尾的。

sizeof还可以用函数做参数, 比如:

```
short f();
```

```
printf("%d\n", sizeof(f()));
```

输出的结果是sizeof(short), 即2。

4) 数组做sizeof的参数不退化, 传递给strlen就退化为指针了。

5) 大部分编译程序 在编译的时候就把sizeof计算过了 是类型或是变量的长度。这就是sizeof(x)可以用来定义数组维数的原因

```
char str[20]="0123456789";
```

```
int a=strlen(str); //a=10;
```

```
int b=sizeof(str); //而b=20;
```

6) strlen的结果要在运行的时候才能计算出来，用来计算字符串的长度，不是类型占内存的大小。

7) sizeof后如果是类型必须加括弧，如果是变量名可以不加括弧。这是因为sizeof是个操作符不是个函数。

8) 当使用了一个结构类型或变量时， sizeof 返回实际的大小， 当使用一静态地空间数组， sizeof 返回全部数组的尺寸。 sizeof 操作符不能返回被动态分配的数组或外部的数组的尺寸



## 二、string中length()和size()

c++中的size()和length()没有区别

如：

```
string str="0123456789";
```

```
cout <<"str.length()=" <<str.length() <<endl;//结果为10
```

```
cout <<"str.size()=" <<str.size() <<endl;//结果为10
```

为了兼容，这两个函数一样。length()是因为沿用C语言的习惯而保留下来的，string类最初只有length()，引入STL之后，为了兼容又加入了size()，它是作为STL容器的属性存在的，便于符合STL的接口规则，以便用于STL的算法。string类的size()/length()方法返回的是字节数，不管是否有汉字。