<https://blog.csdn.net/jiede1/article/details/70809352>

**1**、C语言有**atoi、atol、atof**等库函数，可分别把ASCII编码的字符串转化为int、long、float类型的数字。需要注意的是，这个几个函数是C语言提供的扩展功能，并不是标准的函数，必须引入头文件# include <stdlib. h># include <stdlib. h>；若需要移植性，请用sscanf函数。

例如：int num=atoi(“12345”);//字符串”12345”转换为数字12345，并存入num变量中

**2、sscanf函数。**   
sscanf函数是C语言中从一个字符串中读进与指定格式相符的数据的函数。sscanf与scanf类似，都是用于输入的，只是后者以屏幕(stdin)为输入源，前者以固定字符串为输入源。使用sscanf函数可以实现字符串到任意数据类型的转换。

例如：char s[]=”12345”;   
int n;   
sscanf(s,”%d”,&n);//把字符串s转换为整形数据并存入变量n中

**3、字符串数字之间的转换**

(1)string --> char \*

string str("OK");

char \* p = str.c\_str();

(2)char \* -->string

char \*p = "OK";

string str(p);

(3)char \* -->CString

char \*p ="OK";

CString m\_Str(p);

//或者

CString m\_Str;

m\_Str.Format("%s",p);

(4)CString --> char \*

CString str("OK");

char \* p = str.GetBuffer(0);

...

str.ReleaseBuffer();

(5)string --> CString

CString.Format("%s", string.c\_str());

(6)CString --> string

string s(CString.GetBuffer(0));

GetBuffer()后一定要ReleaseBuffer()，否则就没有释放缓冲区所占的空间，CString对象不能动态增长了。

(7)double/float->CString

double data;

CString.Format("%.2f",data); //保留2位小数

(8)CString->double

CString s="123.12";

double d=atof(s);

(9)string->double

//c\_str函数的返回值是const char\*的，atof不能直接赋值给char\*，所以就需要我们进行相应的操作转化，下面就是这一转化过程

double d=atof(s.c\_str());

**4、数字转字符串：使用sprintf()函数**

char str[10];

int a=1234321;

sprintf(str,"%d",a);

char str[10];

double a=123.321;

sprintf(str,"%.3lf",a);

char str[10];

int a=175;

sprintf(str,"%x",a);//10进制转换成16进制

char *itoa(int value, char* string, int radix);

同样也可以将数字转字符串，不过itoa()这个函数是平台相关的（不是标准里的），故在这里不推荐使用这个函数。

**4、使用stringstream类，必须使用**#include<sstream>#include<sstream>

用ostringstream对象写一个字符串，类似于sprintf()

ostringstream s1;

int i = 22;

s1 << "Hello " << i << endl;

string s2 = s1.str();

cout << s2;

用istringstream对象读一个字符串，类似于sscanf()

istringstream stream1;

string string1 = "25";

stream1.str(string1);

int i;

stream1 >> i;

cout << i << endl; // displays 25

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/jiede1/article/details/70809352