

# Строковые литералы.

---

Дисциплина: «Конструирование программ и языки программирования»

Преподаватель: Гайшун Алеся Александровна ©

# 1 Типы строк в C++.

---

**Строка** – это группа символов, обрабатываемая как единый модуль.

Строка может включать **буквы, цифры** и специальные символы: **+, -, \*, /, \$** и др.  
Строки в C++ заключаются в двойные кавычки.

```
char str[]="Hello, World!";
```

## Типы строк в C++

массив  
переменных  
типа char

специальный  
класс string

```
char name[50];  
cin>>name;  
cout<<"Hello " <<name<<endl;  
  
cin.getline(name, 50);
```

```
#include <string>  
  
...  
  
string name;  
cin>>name;  
cout<<"My name is " <<name<<endl;
```

## Функции работы со строками:

**size** `strlen(char s)` - вычисляет длину строки `s` в байтах

```
int len=strlen(a);
```

`char` **strcat**(`char dest, char src`) - присоединяет строку `src` в конец строки `dest`, полученная строка возвращается в качестве результата

```
strcat(a,b);
```

`char` **strcpy**(`char s, char m`) - копирует строку `m` в строку `s`

```
strcpy(b, a);
```

**strcmp**(char s1, char s2) - сравнивает две строки с учетом различия прописных и строчных букв

```
int i=strcmp(a,b);
```

double **atof**(char s) - преобразует строку в вещественное число, в случае неудачного преобразования возвращается 0

```
double i=atof(a);
```

long **atol**(const char s) - преобразует строку в длинное целое число, в случае неудачного преобразования возвращается 0

```
long int i=atol(a);
```

char **strupr**(char s) - преобразует символы строки, на которую указывает s, в символы верхнего регистра, после чего возвращает ее

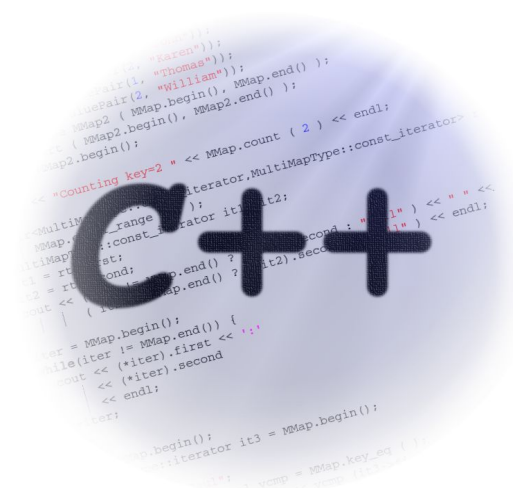
```
n=strupr(c);
```

```
string s;  
s="Hello, world!"
```

Над строками типа **string** определены следующие операции:

- присваивания, например `s1=s2`;
- объединения строк (`s1+=s2` или `s1=s1+s2`);
- сравнения строк на основе лексикографического порядка:

`s1=s2`, `s1!=s2`, `s1<s2`, `s1>s2`, `s1<=s2`, `s1>=s2`;



## Функции обработки строк типа

**string:**

- `s.append(str)` - добавляет в конец строки строку `str`.

`s.append(name) == s.append("Denis");`

- `s.assign(str)` - присваивает строке `s` значение строки `str`.

`s=str;`

- `int i=s.begin()` - записывает в `i` индекс первого элемента строки

- `int i=s.end()` - записывает в `i` индекс последнего элемента строки

- `s.clear()` - очищает строку.

- `s.compare(str)` - сравнивает строку `s` со строкой `str` и возвращает 0 в

случае совпадения

- **s.copy** (куда, сколько, начиная с какого) - копирует из строки `s` в куда (там может быть как строка типа `string`, так и строка типа `char`).
- **bool b=s.empty()** - если строка пуста, возвращает `true`, иначе `false`
- **s.erase** (откуда, сколько) - удаляет `n` элементов с заданной позиции
- **s.find** (str,позиция) - ищет строку `str` начиная с заданной позиции
- **s.insert** (позиция,str, начиная с beg, count) - вставляет в строку `s` начиная с заданной позиции часть строки `str` начиная с позиции `beg` и вставляя `count`

СИМВОЛОВ



- `int len=s.length()` - записывает в `len` длину строки
- `s.push_back(symbol)` - добавляет в конец строки символ
- `s.replace(index, n, str)` - берет `n` первых символов из `str` и заменяет символы строки `s` на них, начиная с позиции `index`
- `str=s.substr(n,m)` - возвращает `m` символов начиная с позиции `n`
- `s.swap(str)` - меняет содержимое `s` и `str` местами.
- `s.size()` - возвращает число элементов в строке.

```
#include "stdafx.h"
#include <iostream>
#include <string>
using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    string a,b;
    cout<<"Enter string 1"<<endl;
    getline(cin,a);
    cout<<"Enter string 2"<<endl;
    getline(cin, b);
    cout<<a<<endl;
    cout<<b<<endl;
    system("pause");
    return 0;
}
```

```
Enter string 1
Ivanov Semen
Enter string 2
Semenov Ivan
Ivanov Semen
Semenov Ivan
Для продолжения нажмите любую клавишу . . . _
```

## 2 Чтение внутренних пробелов

---

```
void main()  
{  
    char s []="Ya pamyatnik, sebe vozdvig nerykotvornui";  
    cout<<s<<endl;  
}
```



Признаком конца строки является нулевой символ **\0** или **NULL**.

Для считывания строк, содержащих пробелы, используется метод – **cin.get()** .

```
int main ( )
{
    const int MAX = 80;
    char str [ MAX];

    cout << "\nVvedite stroky: ";
    cin.get ( str, MAX );
    cout << "Vu vveli: " << str << endl;

    return 0;
}
```

## 3 Считывание нескольких строк

---

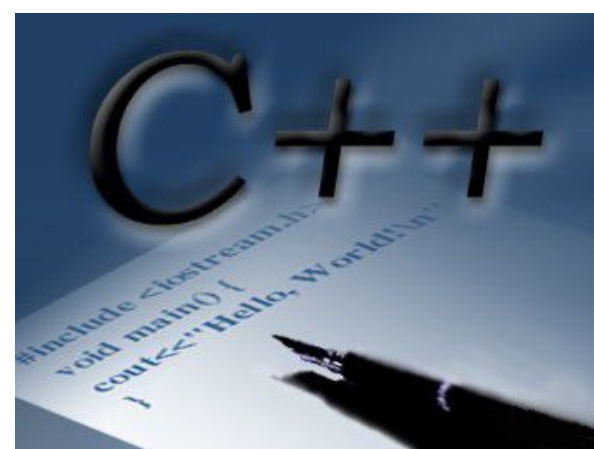
```
#include <iostream>
using namespace std;
const int MAX = 2000;
char str [ MAX ];
int main ( )
{
    cout << "\nВведите строку:\n";
    cin.get ( str, MAX, '$' );
    cout << "Вы ввели:\n" << str << endl;
    return 0;
}
```

```
int main ( )
{
    char str1 [ ] = "Умом Россию не понять.";

    const int MAX = 80;
    char str2 [ MAX ];

    for ( int j = 0; j < strlen ( str1 ); j++ )
        str2 [ j ] = str1 [ j ];
    str2 [ j ] = '\0';
    cout << str2 << endl;
    return 0;
}
```





**Спасибо за  
внимание!**

Дисциплина: «Конструирование программ и языки программирования»

Тема: «Строковые литералы»

Преподаватель: Гайшун Алеся Александровна ©