



第五讲 数组与字符串

— 字符串 (字符数组)

潘建瑜@MATH.ECNU



字符串 (字符数组)

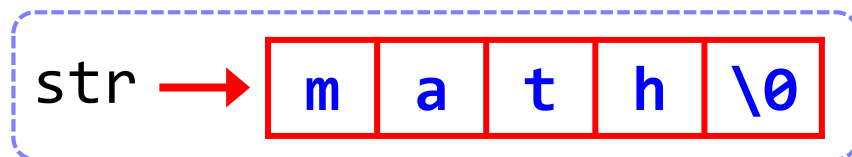
- 字符串的表示
- 字符串输入输出
- 字符串操作 --- 相关函数
- 字符串操作函数

字符串的声明与赋值

□ 字符串的表示：一维字符数组

```
char str[5]={'m','a','t','h','\0'}; // OK, 只能用于初始化  
char str[5]="math"; // OK, 只能用于初始化  
char str[]="math"; // OK, 只能用于初始化
```

- ▶ 字符串以 “\0” 为结束标志
- ▶ 使用双引号时, 会自动在最后添加结束标志



□ 字符串赋值：逐个赋值，循环实现

```
char str[5];  
str = "Math"; // ERROR: 一维数组, 不能直接赋值!
```

字符串的输出: cout

- 法一: 单个元素单个元素输出 → 循环 (数组方式)

```
char str[20]="C++ and Matlab";  
for(int i=0; i<20; i++)  
    if (str[i]!='\0')  
        cout << str[i]; // 逐个输出  
    else  
        break;
```

- 法二: 整体输出 cout

```
char str[20]="C++ and Matlab";  
cout << str << endl; // 整体输出
```

ex05_str_cout.cpp

注: 输出字符中不含 "\0"

字符串的输入: cin

- ❑ 整体输入 `cin`: 输入单个字符串时, 中间不能有空格
- ❑ 如果需要输入多个字符串, 以空格隔开

```
char str[5];  
cin >> str;
```

ex05_str_cin.cpp

```
char str1[5], str2[5], str3[5];  
cin >> str1 >> str2 >> str3;
```

输入数据: `How are you?` 内存中变量状态如下: `str1:`

<code>H</code>	<code>o</code>	<code>w</code>	<code>\0</code>	
<code>a</code>	<code>r</code>	<code>e</code>	<code>\0</code>	
<code>y</code>	<code>o</code>	<code>u</code>	<code>?</code>	<code>\0</code>

```
char str[13];  
cin >> str; // 输入数据: How are you? 结果会怎样
```

getline 和 getchar

□ 整行输入

```
cin.getline(str, N, 结束符);
```

- ▶ 连续读入多个字符（可以有空格），直到读满 $N-1$ 个为止，或遇到指定的结束符（不存储结束符）
- ▶ 如果省略结束符，则默认为 `'\n'`（换行），即读取整行

```
char str[13];  
cin.getline(str, 13);
```

[ex05_str_getline.cpp](#)

□ 单个字符的输入

```
getchar();
```

```
char ch;  
ch=getchar();
```

字符串操作相关函数

(需包含头文件 `cstring` 和 `cstdlib`)

函数	描述	用法
<code>strlen</code>	求字符串长度	<code>strlen(str)</code>
<code>strcat</code>	字符串连接	<code>strcat(dest, src)</code>
<code>strcpy</code>	字符串复制	<code>strcpy(dest, src)</code>
<code>strcmp</code>	字符串比较	<code>strcmp(str1, str2)</code>
<code>atoi</code>	将字符串转换为整数	<code>atoi(str)</code>
<code>atol</code>	将字符串转换为long	<code>atol(str)</code>
<code>atof</code>	将字符串转换为double	<code>atof(str)</code>
<code>itoa</code>	将整数转换为字符串	<code>itoa(int, str, radix)</code>

(更多函数可参见 <http://www.cppreference.com>)

字符串操作相关函数

`strlen(str)` // 返回字符串 str1 的长度 (不含结束符)

`strcat(str1, str2)` // 将 str2 的全部内容添加到 str1 中, str2 的内容保留

`strncat(str1, str2, n)` // 将 str2 的内容添加到 str1 中, 至多添加 n 个字符

`strcpy(str1, str2)` // 将 str2 复制到 str1 中, str1 的长度应该不小于 str2 的长度

`strncpy(str1, str2, n)`

// 将 str2 前 n 个字符复制到 str1 中, 若 n 大于 str2 的长度, 则复制全部内容

字符串

`strcmp(str1, str2)`

// 按字典顺序比较大小: $str1 > str2$ 返回正数, $str1 < str2$ 返回负数, 相等则返回 0

`strncmp(str1, str2, n)` // 按字典顺序比较前 n 个字符的大小

`atoi(str), atol(str), atof(str)`

// 分别将 str 转化为整型、长整型和双精度型数据, str 是由数字组成的字符串

```
int x; double y;  
x=atoi("66"); // x=66  
y=atof("14.5"); // y=14.5
```

`ex05_str_atoi.cpp`

`itoa(int, str, radix)` // 按指定的进制 radix 将一个整数 int 转化为字符串 str

字符操作相关函数

(需包含头文件 `cctype`)

函数	描述	用法
<code>isdigit</code>	是否为数字	<code>isdigit('3')</code>
<code>isalpha</code>	是否为字母	<code>isalpha('a')</code>
<code>isalnum</code>	是否为字母或数字	<code>isalnum('c')</code>
<code>islower</code>	是否为小写	<code>islower('b')</code>
<code>isupper</code>	是否为大写	<code>isupper('B')</code>
<code>isspace</code>	是否为空格	<code>isspace(' ')</code>
<code>tolower</code>	将大写转换为小写	<code>tolower('A')</code>
<code>toupper</code>	将小写转换为大写	<code>toupper('a')</code>

注意

以上函数只针对字符，不是字符串！

(更多参见 <http://www.cppreference.com>)

谢谢



Programming