

声明：本课件及视频版权归小武老师所有，禁止任何组织及个人分发、抄袭、售卖等，违者将追究其法律责任！

《C++ 语言基础-L1》

Day7 字符串与字符数组

主讲人：小武老师

字符

字符本质上还是用计算机二进制来存储，字符与二进制数存在一个一一对应的关系，比如 ASCII 码



数据类型



数据类型	占用空间	取值范围
char	1 字节, 8 位	-128 ~ 127
int	4 字节, 32 位	$-2^{31} \sim 2^{31} - 1$, 大约能够表示绝对值不超过 2.1×10^9 的整数
unsigned int	4 字节, 32 位	$0 \sim 2^{32} - 1$, 大约能够表示不超过 4.2×10^9 的非负整数
long long	8 字节, 64 位	$-2^{63} \sim 2^{63} - 1$, 大约能够表示绝对值不超过 9.2×10^{18} 的整数
unsigned long long	8 字节, 64 位	$0 \sim 2^{64} - 1$, 大约能够表示不超过 1.8×10^{19} 的非负整数
float	4 字节, 32 位	大约指数绝对值不超过 37, 6 位有效数字
double	8 字节, 64 位	大约指数绝对值不超过 307, 15 位有效数字



字符类型



char 是一种存储字符的变量

```
#include<iostream>
using namespace std;
int main() {
    cout << "hello world";
    return 0;
}
```

```
#include<iostream>
using namespace std;
int main() {
    char c = 'A';
    cout << c;
    return 0;
}
```

思考：如何在计算机当中存储一个字符？



211 —> “su bing tian”

400 —> “zhang san”

001 —> “xiao wu lao shi”



ASCII码



ASCII ((American Standard Code for Information Interchange): 美国信息交换标准代码)

数字	字符	数字	字符	数字	字符	数字	字符	数字	对应字符	数字	字符
32	[空格]	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	E	117	u
38	&	54	6	70	F	86	V	102	F	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(56	8	72	H	88	X	104	h	120	x
41)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o		

ASCII，美国信息交换标准代码，发明于1963年，ASCII码是7位代码，足够存储128个不同的值。扩展范围后，足够表示大小写字母、数字和标点符号等。核心思想：用数字代表字符



英文字母



大写字母、小写字母和数字在 ASCII 表中都是按照顺序依次排列的

```
ans1 = 'C' - 'A' + 1;
```

'C'-'A'+1 是计算字母 'C' 和字母 'A' 中间的差距。ans1 等价于 $67 - 65 + 1$ 。67 和 'C' 等价，65 和 'A' 等价

```
ans2 = 'A' + 1; // 表示 'B' 的 ASCII
```

```
// ans 2 = 65 + 1;
```



英文字母



```
#include<iostream>
using namespace std;
int main() {
    int ans1;
    char ans2;
    ans1 = 'C' - 'A' + 1;
    ans2 = 'A' + 4 - 1;
    cout << ans1 << endl;
    cout << ans2 << endl;
    return 0;
}
```

使用 cout 输出 char 类型的变量，
会输出这个变量存储的数字对应的
的字符。



大写字符转小写



例子 大写字符转小写字符

输入一个大写字符，输出对应的小写字符

输入

A

输出

a

```
#include<iostream>
using namespace std;
int main() {
    char c, ans;
    cin >> c;
    ans = c - 'A' + 'a';
    cout << ans;
    return 0;
}
```

字符串

字符串可以用字符数组和string 表示

字符数组实质上 and 整数数组没什么区别，只是数组中的每一个元素都是一个字符（实际上，都是存成对应 ASCII 的数字）。



字符串



字符串：由若干个有效字符组成的序列有效字符包括字母，数字，专用字符，转义字符，如：

"zhang san"

"123456 "

"a45-7"

"m k\n"

C风格-字符数组

```
#include<cstdio>
using namespace std;
char s[101];
int main(){
    scanf("%s",s);
    printf("%s",s);
    return 0;
}
```

C++风格——string类型

```
#include<iostream>
#include<string>
using namespace std;
string s;
int main(){
    cin >> s;
    cout << s;
    return 0;
}
```

字符数组

字符数组实质上 and 整数数组没什么区别，只是数组中的每一个元素都是一个字符（实际上，都是存成对应 ASCII 的数字）。



字符数组



字符数组初始化

字符数组实质上 and 整数数组没什么区别，只是数组中的每一个元素都是一个字符（实际上，都是存成对应 ASCII 的数字）。

```
char c[8] = {"hello"};  
char c[8] = "hello";  
char c[8] = {'h', 'e', 'l', 'l', 'o', '\0'};
```

```
char c[5] = "hello";
```

h	e	l	l	o	\0
---	---	---	---	---	----

错误的初始化

注意：字符数组以 “\0” 作为结束标志。



字符数组



因为有“\0”，所以可以在初始化时一次性输入

```
#include<cstdio>
using namespace std;
char s[101];
int main(){
    scanf("%s",s);
    printf("%s",s);
    return 0;
}
```

```
#include<iostream>
using namespace std;
char s[101];
int main(){
    cin >> s;
    cout << s;
    return 0;
}
```



字符数组相关函数



注意使用到了 `cstring` 头文件，其中一些的用法如下：

<code>strlen(s)</code>	获得字符数组的 长度 ，数到 ' \0'
<code>strcpy(a,b)</code>	将 b 字符数组的数据 复制 到 a
<code>strcmp(a,b)</code>	比较两个字符串， $a > b$ 则返回 1
<code>char a[100]; strcpy(a, "hello");</code>	给字符数组 赋值常量

声明：本课件及视频版权归小武老师所有，禁止任何组织及个人分发、抄袭、售卖等，违者将追究其法律责任！

编程实践 Online Judge

P0019 P0075 P0237



string 类

而c++中的string类（面向对象）也可以用来处理字符串，string类处理起字符串来会方便很多，完全可以代替C风格字符数组。



string 类型



使用 C 语言风格的字符数组有诸多不便，于是再在 C++ 中提供了一些更好的工具，将很多有用的功能进行了封装，开箱即用。

```
#include<iostream>
#include<string>
using namespace std;
string s;
int main(){
    cin >> s;
    int len = s.length(); //s.size();
    cout << len;
    return 0;
}
```



string 类型



使用 string 头文件来操作字符串，其中一些的用法如下

功能	说明
string s;	定义一个名字为 s 的字符串变量
s+=str 或 s.append(str)	在字符串 s 后面拼接字符串 str
s<str	比较字符串 s 的是否在字符串 str 的字典序之前
s.size() 或 s.length()	得到字符串 s 的长度
s.substr(pos,len)	截取字符串 s，从第 pos 个位置开始 len 个字符，并返回这个字符串
s.insert(pos, str)	字符串 s 的第 pos 个字符之前，插入字符串 str，并返回这个字符串
s.find(str, [pos])	字符串 s 中从第 pos 个字符开始寻找 str，并返回位置，如果找不到返回 string::npos。pos 可以省略，默认值是 0



string 类型



例子 用string 实现字符串大小写转换



string 类型



思考：如何输入一个带空格的字符串？

```
#include<iostream>
#include<string>
using namespace std;
string s;
int main(){
    getline(cin,s);
    int len = s.length(); //s.size();
    cout << len;
    return 0;
}
```

getline(cin,s);



string 类型



思考：如何输入一个带空格的字符串？

```
#include<cstdio>
using namespace std;
int main(){
    char s1[20],s2[20],i;
    i = 0;
    while((s1[i] = getchar()) != '\n') i++;
    s1[i] = '\0';
    i = 0;
    while(s1[i] != '\0' ){
        putchar(s1[i]);
        i++;
    }
    return 0;
}
```

getchar();

声明：本课件及视频版权归小武老师所有，禁止任何组织及个人分发、抄袭、售卖等，违者将追究其法律责任！

编程实践 Online Judge

P0053 P0238

可达信奥—小武老师—keda.ac

可达信奥—小武老师—keda.ac



字符串总结



不带空格字符串

```
char s[101];    strlen(s);
```

```
string s;    s.length();
```

带空格字符串

```
getline(cin, s);
```

```
getchar() 逐个读入
```

课后习题与实验

Talk is cheap, show me the code !



声明：本课件及视频版权归小武老师所有，禁止任何组织及个人分发、抄袭、售卖等，违者将追究其法律责任！

下节课见啦！

可达信奥—小武老师—keda.ac

可达信奥—小武老师—keda.ac