

集训课-12：字符型与字符数组

课程目标

01

字符型

02

强制类型转换

03

cctype库

04

字符数组

05

总结回顾



Part 1

字符型

字符型

字符型 char

存储单个字符的数据类型

字符型

字符型定义

数据类型 变量名;

例: `char a;`

字符型

字符型常量用单引号括起来

'a'

'A'

'O'

'0'

'#'

'+'

字符型

字符型赋值

数据类型 变量名 = 初始值;

例如: `char a = ' A' ;`

字符型

下面哪些数据是字符类型呢？你来选一选吧！

'a'

“a”

123

“aBC”

'*'

‘0’

字符型

字符型数据是用ASCII码存储的

ASCII码范围：0~127

字符型

ASCII 码表完整版							
ASCII 值	控制字符	ASCII 值	控制字符	ASCII 值	控制字符	ASCII 值	控制字符
0	NUT	32	(space)	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	,	71	G	103	g
8	BS	40	(	72	H	104	h
9	HT	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DCI	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	X	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	TB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS	60	<	92	/	124	
29	GS	61	=	93	]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	—	127	DEL

字符型

ASCII码	字符
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K
76	L
77	M

ASCII码	字符
78	N
79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
88	X
89	Y
90	Z

字符型

ASCII码	字符
97	a
98	b
99	c
100	d
101	e
102	f
103	g
104	h
105	i
106	j
107	k
108	l
109	m

ASCII码	字符
110	n
111	o
112	p
113	q
114	r
115	s
116	t
117	u
118	v
119	w
120	x
121	y
122	z

字符型

‘A’ : 65

‘a’ : 97

‘B’ : 65 + 1

‘b’ : 97 + 1

‘C’ : 65 + 2

‘c’ : 97 + 2

.....

.....

字符型

已知 'A' 的ASCII码值是65，那么 'D' 的ASCII码值是多少呢？

- 67

- 68

- 69

字符型

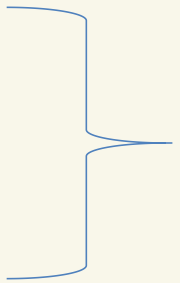
ASCII码	字符
65	A
66	B
67	C
68	D
.....	
87	W
88	X
89	Y
90	Z

ASCII码	字符
97	a
98	b
99	c
100	d
.....	
119	w
120	x
121	y
122	z

字符型

同一字母的大小写ASCII码

‘A’ : 65
‘a’ : 97



相差32

‘Z’ : 90
‘z’ : 122



相差32

字符型

已知 'D' 的ASCII码值是68, 那么 'd' 的ASCII码值是多少呢?

A.100

B.69

C.36

字符型

大写字母 = 小写字母 - 32

```
char lower = 'b';  
char upper = lower - 32;  
cout << upper;
```

字符型

小写字母 = 大写字母 + 32

```
char upper = 'B';  
char lower = upper + 32;  
cout << lower;
```

字符型

ASCII 码	字符
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9

常用的ASCII码：

‘0’ : 48

字符0~9的ASCII码是连续的

字符型

数字字符 = 数字 + 48

```
int num = 2;  
char ch = num + 48;  
cout << ch;
```

字符型

现有数字5，如何将它转化为字符'5'呢？

- `char ch = 5;`
- `char ch = 5 - 48;`
- `char ch = 5 + 48;`



Part 2

强制类型转换

强制类型转换

被转化的类型(要转化的变量)

```
char ch = 'a';  
cout << int(ch);  
  
int ascii = 97;  
cout << char(ascii);
```

强制类型转换

被转化的类型(要转化的变量)

```
int a = 10, b = 3;  
cout << double(a) / b;
```

```
double c = 10, d = 3; cout << int(c)  
% int(d);
```



Part 3

cctype库

cctype库

<cctype> 是一个与字符处理有关的头文件，它包含了一系列用于检测和转换单个字符的函数。

```
#include <cctype>
```

cctype库

函数名称	函数功能
isupper()	如果参数是大写字母，返回值为true
islower()	如果参数是小写字母，返回值为true
isdigit()	如果参数是数字字符0~9，返回值为true
tolower()	如果参数是大写字符，则返回其小写，否则返回该参数
toupper()	如果参数是小写字符,则返回其大写,否则返回该参数

cctype库

使用islower函数进行字符检测，下面哪个选项的返回值是非零值？

A

islower('A')

B

islower('2')

C

islower('a')

D

islower('@')

cctype库

使用isupper函数进行字符检测，下面哪个选项的返回值是非零值？

A

islower('A')

B

islower('2')

C

islower('a')

D

islower('@')



Part 4

字符数组

字符数组

字符数组定义

★ `char 数组名[元素个数];`

例

```
char a[5];
```

字符数组

字符数组初始化：

依次初始化所有元素

```
char a[5] = {'a', 'b', 'c', 'd', 'e'};
```

字符数组

数组元素获取

- 数组下标从0开始
- 最后一个下标为数组长度-1
- 下标必须为自然数，不可以为负数和浮点数
- 注意越界问题

字符数组



字符数组元素输出

格式: `cout << 数组名[下标];`

```
char a[5] = {'a', 'b', 'c', 'd', 'e'};
```

```
cout << a[0];
```

字符数组

字符数组遍历输入与遍历输出

```
char a[5];  
for (int i = 0; i < 5; i++)  
{  
    cin >> a[i];  
}  
for (int i = 0; i < 5; i++)  
{  
    cout << a[i];  
}
```

字符数组

多个字符串可以使用二维字符数组进行存储。

```
char 数组名[字符串个数][字符串最大长度+1];
```

```
char a[5][100];
```

字符数组

二维字符数组可以借助字符串的形式进行初始化。

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char a[5][100] = { "hello",
                      "nihao",
                      "xqwcode",
                      "xixi",
                      "nicai"
                    };

    return 0;
}
```

字符数组

字符输出：① 通过数组名[行下标]获取对应的字符串

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char a[5][100]={ "hello","nihao","xxxcode",
    "xixi","nicai"};
    cout << a[0] << endl;
    cout << a[1] << endl;
    cout << a[2] << endl;
    cout << a[3] << endl;
    cout << a[4] << endl;
    return 0;
}
```

字符数组

字符输出：②循环输出

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char a[5][100]={ "hello","nihao","xxxcode",
    "xixi","nicai"};
    for (int i = 0; i < 5; i++)
    {
        cout << a[i] << " " ;
    }
    return 0;
}
```



Part 5

总结回顾

总结回顾

一、字符型

表示：用单引号括起来的单个字符

类型名：char

ASCII码：字符型中用于存储字符的编码

常用的ASCII码：'0' → 48, 'A' → 65, 'a' → 97

总结回顾

char与int的转化:

```
char a = 'a';    cout << int(a);
```

int与char的转化:

```
int a = 97;      cout << char(a);
```

double与int的转化:

```
double a = 1.2;  cout << int(a);
```

int与double的转化:

```
int a = 1.2;     cout << double(a);
```

总结回顾

大小写转化：

小写字母 = 大写字母 + 32

大写字母 = 小写字母 - 32

数字字符和数字的转化：

数字字符 = 数字 + 48 数字 = 数字字符 -

48

总结回顾

<cctype> 是一个与字符处理有关的头文件，它包含了一系列用于检测和转换单个字符的函数。

函数名称	函数功能
isupper()	如果参数是大写字母，返回值为true
islower()	如果参数是小写字母，返回值为true
isdigit()	如果参数是数字字符0~9，返回值为true
tolower()	如果参数是大写字符，则返回其小写，否则返回该参数
toupper()	如果参数是小写字符,则返回其大写,否则返回该参数

总结回顾

二维字符数组可以借助字符串的形式进行初始化。

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char a[5][100] = { "hello",
                      "nihao",
                      "xqwcode",
                      "xixi",
                      "nicai"
                    };

    return 0;
}
```

A horizontal yellow banner with blue triangular ends, centered on a solid blue background. The text "Class is over" is written in blue within the banner.

Class is over