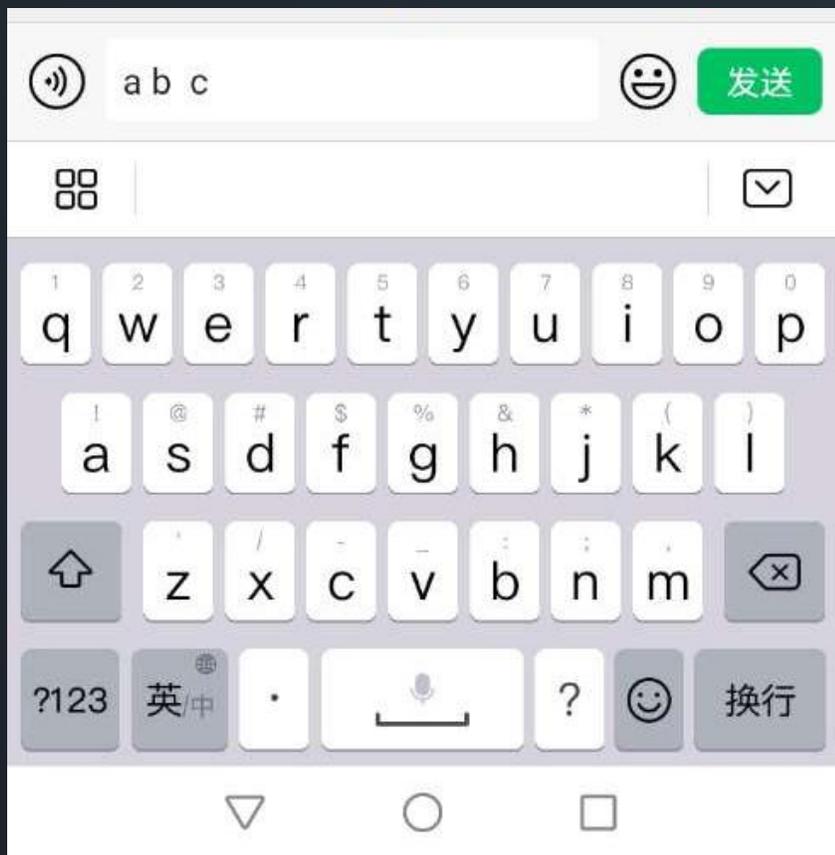


# 字符型-数据类型



华清远见 | 创客学院 小美老师

# 计算机如何处理字符



- 计算机存储数据是0、1二进制数据
- 计算机存储字符数据，也需要把字符数据按照特定的编码格式转换成二进制数据进行存储。

# ASCII码

- **字符数据**在机器内也被变换成**二进制**编码的形式。
- 国际上普遍采用的一种编码是**美国国家信息交换标准代码**，  
简称为**ASCII码**。

# man ASCII

Oct	Dec	Hex	Char	Oct	Dec	Hex	Char
000	0	00	NUL '\0' (null character)	100	64	40	@
001	1	01	SOH (start of heading)	101	65	41	A
002	2	02	STX (start of text)	102	66	42	B
003	3	03	ETX (end of text)	103	67	43	C
004	4	04	EOT (end of transmission)	104	68	44	D
005	5	05	ENQ (enquiry)	105	69	45	E
006	6	06	ACK (acknowledge)	106	70	46	F
007	7	07	BEL '\a' (bell)	107	71	47	G
010	8	08	BS '\b' (backspace)	110	72	48	H
011	9	09	HT '\t' (horizontal tab)	111	73	49	I
012	10	0A	LF '\n' (new line)	112	74	4A	J
013	11	0B	VT '\v' (vertical tab)	113	75	4B	K
014	12	0C	FF '\f' (form feed)	114	76	4C	L
015	13	0D	CR '\r' (carriage ret)	115	77	4D	M
016	14	0E	S0 (shift out)	116	78	4E	N
017	15	0F	SI (shift in)	117	79	4F	O
020	16	10	DLE (data link escape)	120	80	50	P
021	17	11	DC1 (device control 1)	121	81	51	Q
022	18	12	DC2 (device control 2)	122	82	52	R
023	19	13	DC3 (device control 3)	123	83	53	S
024	20	14	DC4 (device control 4)	124	84	54	T
025	21	15	NAK (negative ack.)	125	85	55	U
026	22	16	SYN (synchronous idle)	126	86	56	V
027	23	17	ETB (end of trans. blk)	127	87	57	W
030	24	18	CAN (cancel)	130	88	58	X
031	25	19	EM (end of medium)	131	89	59	Y
032	26	1A	SUB (substitute)	132	90	5A	Z
033	27	1B	ESC (escape)	133	91	5B	[

# ASCII

- ✓ 第一部分由 00H 到 1FH 共 32 个，一般用来通讯或作为控制之用，有些字符可显示于屏幕，有些则无法显示在屏幕上。
- ✓ 第二部分是由 20H 到 7FH 共 96 个，这 96 个字符是用来表示阿拉伯数字、英文字母大小写和底线、括号等符号，都可以显示在屏幕上。
- ✓ 第三部分由 80H 到 0FFH 共 128 个字符，一般称为『扩充字符』，这 128 个扩充字符是由 IBM 制定的，并非标准的 ASCII 码。这些字符是用来表示框线、音标和其它欧洲非英语系的字母。

# 字符型数据

数据类型	字节	取值范围
char	1	-128~127
unsigned char	1	0~255

# 字符型变量

- 定义格式:

```
char c1 = 'A';
```

```
char c2 = 'a';
```

```
char c3 = 66;
```

注意字符常量要用单引号括起来, 也可以用字符对应的ASCII码值进行赋值;

printf对应的格式符是%c;

# 代码验证

```
#include <stdio.h>

int main() {
    char ch1 = ' ';
    char ch2 = 'A';
    char ch3 = 'B';
    char ch4 = 'a';
    char ch5 = 'b';
    char ch6 = '0';
    char ch7 = '1';

    printf("%c %d\n", ch1, ch1);
    printf("%c %d\n", ch2, ch2);
    printf("%c %d\n", ch3, ch3);
    printf("%c %d\n", ch4, ch4);
    printf("%c %d\n", ch5, ch5);
    printf("%c %d\n", ch6, ch6);
    printf("%c %d\n", ch7, ch7);

    return 0;
}
```

\$ ./a.out

32

A-65

B-66

a-97

b-98

0-48

1-49

'0' ---48 (差48)

'1' ---49

.....

'A' ---65

'B' ---66

.....

'a' ---97 (差32)

'b' ---98

# 大小写字母转换

- 大写字母转小写字母 +32
- 小写字母转大写字母 -32

```
#include <stdio.h>
```

```
int main() {
```

```
    char ch1 = 'E';
```

```
    char ch2 = 'w';
```

```
    printf("%c %c\n", ch1 + 32, ch2 - 32);
```

```
    return 0;
```

```
}
```

```
$ gcc ASCII_3.c -Wall
```

```
$ ./a.out
```

```
e W
```

# 数字0和字符 '0' 的转换

```
#include <stdio.h>
```

```
int main() {
```

```
    int a = 5, b = 8;
```

```
    printf("%c %c\n", a + 48, b + 48);
```

```
    return 0;
```

```
}
```

```
$ gcc ASCII_4.c
```

```
$ ./a.out
```

```
5 8
```

扫一扫，获取更多信息



THANK YOU