



# СИМВОЛЫ

Лекция 8



# СИМВОЛЫ

Отдельные символы могут храниться в переменных типа **char**.

```
char c = '#';
```

```
char c = Convert.ToChar(Console.ReadLine());
```



# Представление символов в компьютере

	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	0	1	2	3	4	5
6	7	8	9	:	;	<	=	>	?	@	A	B	C	D	E	F	G	H	I	J	K
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_	`	a
b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
x	y	z	{		}	~	€	,	f	„	...	†	‡	^	‰	Š	‹	Œ	Ž	'	'
“	”	•	–	—	~	™	š	›	œ	ž	ÿ		i	ç	£	¤	¥	¦	§	¨	©
ª	«	¬	-	®	-	°	±	²	³	'	μ	¶	·	¸	¹	º	»	¼	½	¾	¿
À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ	Ò	Ó	Ô	Õ
Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë
ì	í	î	ï	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ		



# Представление символов в компьютере

Стандартная часть таблицы ASCII											
№ п/п	символ	двоичный код	№ п/п	символ	двоичный код	№ п/п	символ	двоичный код	№ п/п	символ	двоичный код
32	пробел	00100000	56	8	00111000	80	P	01010000	104	h	01101000
33	!	00100001	57	9	00111001	81	Q	01010001	105	i	01101001
34	"	00100010	58	:	00111010	82	R	01010010	106	j	01101010
35	#	00100011	59	;	00111011	83	S	01010011	107	k	01101011
36	\$	00100100	60	<	00111100	84	T	01010100	108	l	01101100
37	%	00100101	61		00111101	85	U	01010101	109	m	01101101
38	&	00100110	62	>	00111110	86	V	01010110	110	n	01101110
39	'	00100111	63	?	00111111	87	W	01010111	111	o	01101111
40	(	00101000	64	@	01000000	88	X	01011000	112	p	01110000
41	)	00101001	65	A	01000001	89	Y	01011001	113	q	01110001
42	*	00101010	66	B	01000010	90	Z	01011010	114	r	01110010
43	+	00101011	67	C	01000011	91	[	01011011	115	s	01110011
44	,	00101100	68	D	01000100	92	\	01011100	116	t	01110100
45	-	00101101	69	E	01000101	93	]	01011101	117	u	01110101
46	.	00101110	70	F	01000110	94	^	01011110	118	v	01110110
47	/	00101111	71	G	01000111	95	_	01011111	119	w	01110111
48	0	00110000	72	H	01001000	96	'	01100000	120	x	01111000
49	1	00110001	73	I	01001001	97	a	01100001	121	y	01111001
50	2	00110010	74	J	01001010	98	b	01100010	122	z	01111010
51	3	00110011	75	K	01001011	99	c	01100011	123	{	01111011
52	4	00110100	76	L	01001100	100	d	01100100	124		01111100
53	5	00110101	77	M	01001101	101	e	01100101	125	}	01111101
54	6	00110110	78	N	01001110	102	f	01100110	126	~	01111110
55	7	00110111	79	O	01001111	103	g	01100111	127		01111111



# Пример

Нахождение **кода** символа

```
char c = '#';  
int code = (int)c;  
Console.WriteLine(code); // 35
```

По **коду** символа, восстановление **символа**

```
int code = 35;  
char c = (char)code;  
Console.WriteLine(c); // '#'  
  
Console.WriteLine((char)48); // '0'
```



# Сравнение символов

```
char first = 'B';  
char second = 'v';  
Console.WriteLine(first < second); // true
```

```
char c = Convert.ToChar(Console.ReadLine());  
if (c >= 'A' && c <= 'Z') //если введенный символ лежит в диапазоне от 'A' до 'Z' включительно  
{  
    Console.WriteLine("Заглавная буква английского алфавита");  
}
```



# Арифметические операции

```
char c = 'a';  
c = (char)(c + 3); // 'd'
```

```
char b = '0';  
c = (char)(c + b);  
c = (char)(c - b);  
c = (char)(c * b);  
c = (char)(c / b);  
c = (char)(c % b);
```



# Выведем все заглавные буквы латинского алфавита

```
for (char w = 'A'; w <= 'Z'; w++)  
{  
    Console.WriteLine(w);  
}
```





**Символы можно использовать в любых местах, где допускается использование целого числа**

*Например:*

```
char c = 'R';  
int x = c - 'A' + 1;
```



# Задачи

1. Определите является ли данный символ цифрой или нет.
2. По заданной строчной букве латинского алфавита, выведите все строчные буквы латинского алфавита, начиная от начала до заданной буквы.
3. По заданной строчной букве латинского алфавита, выведите все строчные буквы латинского алфавита, начиная от заданной буквы до конца алфавита.
4. Измените регистр символа, если он был латинской буквой: сделайте его заглавным, если он был строчной буквой и наоборот.