



Programación

C #

TECNICATURA SUPERIOR EN PROGRAMACIÓN
LABORATORIO DE COMPUTACIÓN II



UTN
Facultad Regional
San Francisco

Clase Char



- Permite representar caracteres Unicode.
- Se usa cuando se necesita albergar únicamente **un carácter individual**.
- En caso de necesitarse varios, se puede utilizar una matriz de elementos Char. Por ejemplo:

```
char arreglo_char[];  
arreglo_char[0] = 'X'; // Caracter literal  
arreglo_char[1] = '\x0058'; // Hexadecimal  
arreglo_char[2] = '\u0058'; // Unicode
```
- El valor predeterminado de Char es el carácter con el *punto de código* 0. El punto de código es un entero que representa un determinado carácter.

Clase Char

DEC	HEX	OCT	CHAR	DEC	HEX	OCT	CH	DEC	HEX	OCT	CH	DEC	HEX	OCT	CH
0	0	000	NUL	32	20	040		64	40	100	@	96	60	140	`
1	1	001	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	2	002	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	3	003	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	4	004	EOT	36	24	044	\$	68	44	104	D	100	64	144	d
5	5	005	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	6	006	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	7	007	BEL	39	27	047	'	71	47	107	G	103	67	147	g
8	8	010	BS	40	28	050	(72	48	110	H	104	68	150	h
9	9	011	TAB	41	29	051)	73	49	111	I	105	69	151	i
10	A	012	LF	42	2A	052	*	74	4A	112	J	106	6A	152	j
11	B	013	VT	43	2B	053	+	75	4B	113	K	107	6B	153	k
12	C	014	FF	44	2C	054	,	76	4C	114	L	108	6C	154	l
13	D	015	CR	45	2D	055	-	77	4D	115	M	109	6D	155	m
14	E	016	SO	46	2E	056	.	78	4E	116	N	110	6E	156	n
15	F	017	SI	47	2F	057	/	79	4F	117	O	111	6F	157	o
16	10	020	DLE	48	30	060	0	80	50	120	80	112	70	160	p
17	11	021	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM)	57	39	071	9	89	59	131	Y	121	79	171	y
26	1A	032	SUB	58	3A	072	:	90	5A	132	Z	122	7A	172	z
27	1B	033	ESC	59	3B	073	;	91	5B	133	[123	7B	173	{
28	1C	034	FS	60	3C	074	<	92	5C	134	\	124	7C	174	
29	1D	035	GS	61	3D	075	=	93	5D	135]	125	7D	175	}
30	1E	036	RS	62	3E	076	>	94	5E	136	^	126	7E	176	~
31	1F	037	US	63	3F	077	?	95	5F	137	_	127	7F	177	DEL

Carácter

Punto de
código

FUNCIONALIDAD

- La clase Char proporciona métodos para:
 - Comparar objetos Char,
 - Convertir el valor del objeto Char actual en un objeto de otro tipo,
 - Determinar la categoría UNICODE de un objeto Char.

Clase Char

funcionalidad

COMPARAR OBJETOS

CompareTo

Compara una instancia con otra y devuelve una indicación de sus **valores relativos**.
Devuelve un entero.

```
char chA = 'A', chN = 'N';  
int resultado = chA.CompareTo('4');  
Console.WriteLine(resultado);  
// Salida: "13"  
Console.WriteLine(chA.CompareTo(chN));  
// Output: "-13"
```

Equals

Compara una instancia con otra, indicando si son iguales o no.
Devuelve un booleano.

```
char chA = 'A', chN = 'N', chA2 = 'A';  
string str = "cadena";  
Console.WriteLine(chA.Equals('A')); //True  
Console.WriteLine(chA.Equals(chA)); //True  
Console.WriteLine(chA.Equals(chN)); //False  
Console.WriteLine(chA.Equals(str)); //False
```

Ejercicio 1:

Escribir un programa que solicite al usuario que ingrese caracteres. Cada carácter ingresado debe ser comparado con el anterior e indicar si se trata del mismo, es mayor o menor. El primer carácter ingresado no se compara con ningún otro en especial. Utilice ambas funciones en el mismo ejercicio. ¿Qué diferencias encuentra? ¿Cuándo utilizaría una y cuándo otra?

CONVERSIÓN ENTRE TIPOS

- **GetNumericValue**

- Convierte un **carácter Unicode numérico** en un número de tipo double. Por ejemplo:

```
char ch1 = '1'; string str = "cadena 4";  
double numero = Char.GetNumericValue(ch1);  
Console.WriteLine(numero); // Salida: "1"  
Console.WriteLine(Char.GetNumericValue(str, 7)); // Salida: "4"
```

- **ToString**

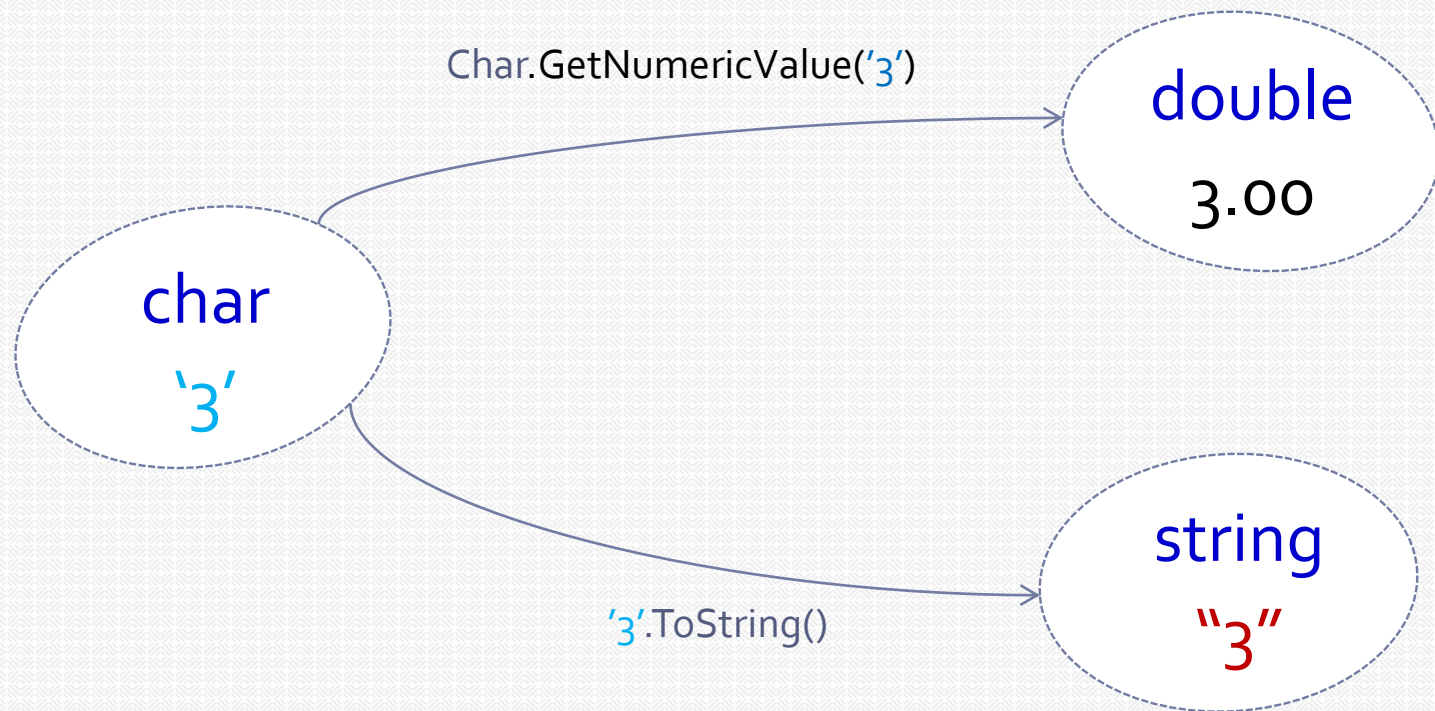
- Convierte el valor del caracter en una cadena string equivalente. Por ejemplo:

```
string cadena = 'x'.ToString();  
Console.WriteLine(cadena); // Salida: "x"  
Console.WriteLine(Char.ToString('b')); // Salida: "b"
```

Clase Char

funcionalidad

CONVERSIÓN ENTRE TIPOS



CONVERSIÓN ENTRE TIPOS

- **Parse**

- Convierte una cadena en el caracter equivalente. Por ejemplo:

```
string cadena = "s";  
char caracter = Char.Parse(cadena);  
Console.WriteLine(caracter); // Salida: 'S'
```

- **TryParse**

- ✦ Ídem Parse, pero devuelve un código indicando si la conversión fue exitosa o no. Por ejemplo:

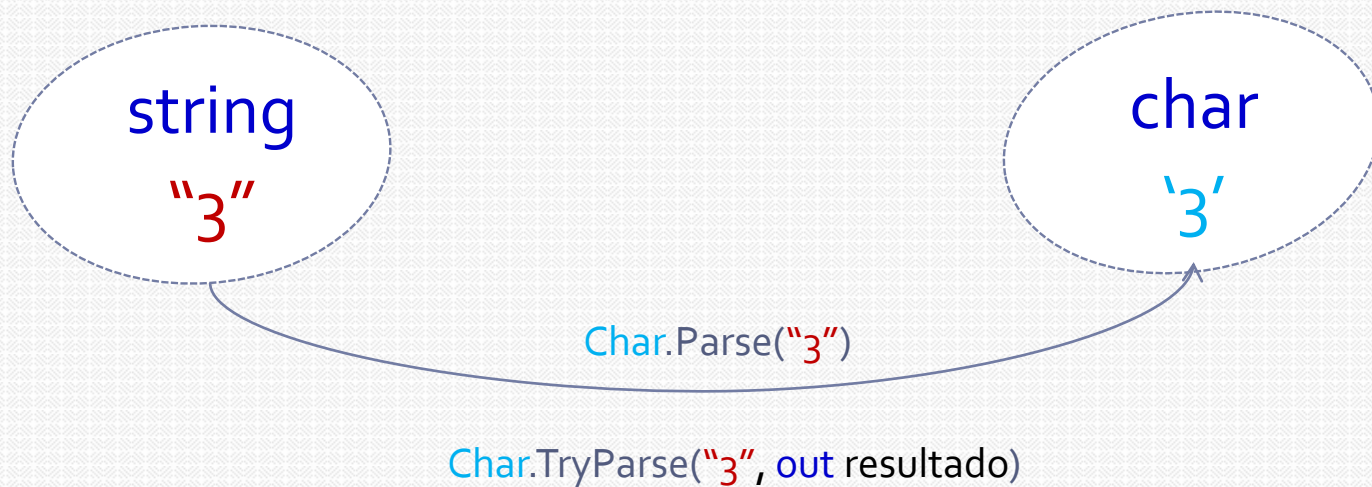
```
char chObtenido; bool resultado;  
resultado = char.TryParse("J", out chObtenido);  
Console.WriteLine(resultado + " " + chObtenido); // Salida: "true J"
```

¿Qué diferencias hay entre Parse y TryParse?

Clase Char

funcionalidad

CONVERSIÓN ENTRE TIPOS



Clase Char

funcionalidad

DETERMINAR CATEGORÍA UNICODE



- **GetUnicodeCategory**

- Indica a qué categoría pertenece un **carácter**. Las posibles categorías están definidas en la enumeración `UnicodeCategory`. Por ejemplo:

```
UnicodeCategory categoria = Char.GetUnicodeCategory('a');  
Console.WriteLine(categoria); // Salida: "LowercaseLetter"  
Console.WriteLine(Char.GetUnicodeCategory('1')); // Salida : "DecimalDigitNumber"
```

Nota: para acceder a la enumeración `UnicodeCategory`, deberá referenciar el espacio de nombres **System.Globalization**

- **Ejercicio 2:**

Escribir un programa que le solicite al usuario ingresar un carácter, lo evalúe y muestre la categoría a la que pertenece. Preguntar al usuario si desea continuar, a lo que deberá contestar 'S' o 'N' para ello.

DETERMINAR CATEGORÍA UNICODE

- Para conocer si un caracter pertenece a una categoría particular, se utilizan métodos específicos, como por ejemplo:
 - ✦ **IsDigit**: para indica si se clasifica como un dígito decimal. Por ejemplo:
 - `Console.WriteLine(Char.IsDigit('1')); // Salida: "True"`
 - ✦ **IsLetter**: para indicar si se clasifica como carácter alfabético. Por ejemplo:
 - `Console.WriteLine(Char.IsLetter(',')); // Salida: "False"`
 - ✦ **IsPunctuation**: para indicar si se clasifica como un signo de puntuación. Por ejemplo:
 - `Console.WriteLine(Char.IsPunctuation('.')); // Salida: "True"`

Además: `IsControl`, `IsHighSurrogate`, `IsLetterOrDigit`, `IsLower`, `IsLowSurrogate`, `IsNumber`, `IsSeparator`, `IsSurrogate`, `IsSurrogatePair`, `IsSymbol`, `IsUpper` y `IsWhiteSpace` para.

○ Ejercicio 3:

Modificar el programa anterior para que: si el usuario ingresa un carácter numérico, se le sume el doble del número y se muestre el resultado; si es una letra, se la muestre "Se ingresó la letra: X" y sino, muestre la categoría a la que pertenece. Continuar si el usuario ingresa 'S' al preguntarle.