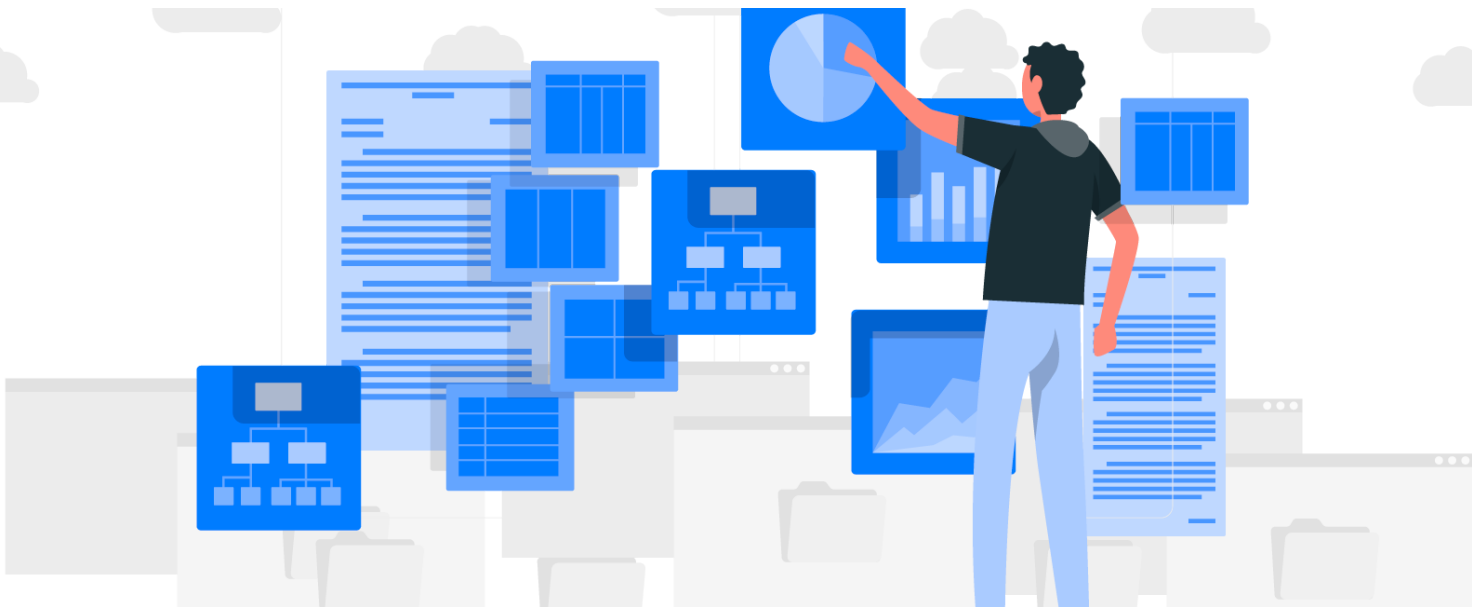


MATERIAL DE LECTURA

Temas:

Tipos de datos. Operadores. Expresiones. Calificadores.
Conversiones de tipos y otros operadores. Código Ascii.

TIPOS DE DATOS

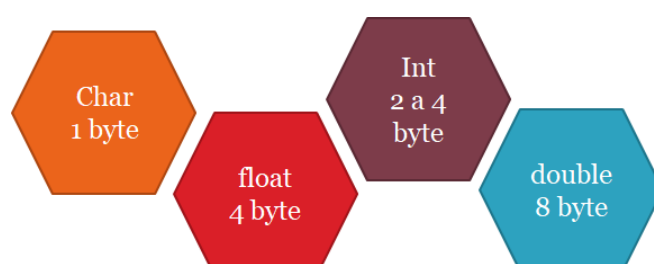


Podemos definir **Dato** como una expresión general que describe los objetos con los cuales opera una computadora. A nivel de hardware, están representados por "tiras de bits" y son manejados por instrucciones propias del procesador.

El uso de **tipo de datos** es muy importante en ciencias de la computación. Un tipo identifica aquellas propiedades comunes a un grupo de objetos que lo distinguen como una clase identificable.

Tipos de datos Simples

Los tipos de datos simples se caracterizan por que sus componentes no se pueden descomponer en tipos más simples. En otras palabras, representan valores atómicos. Por ejemplo: entero, carácter.



TIPO DE DATO ENTERO: TAMAÑO, CALIFICADORES

¿Qué valores puede asumir una variable int?

Los valores que puede tomar una variable entera están en el rango -32768 hasta 32767.

Y surge la pregunta ¿Por qué un entero tiene límites de rango tan extraños?.

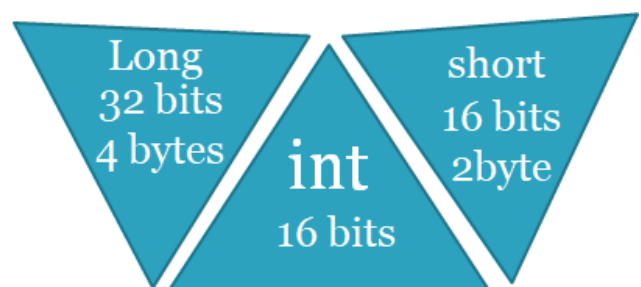
Esto tiene que ver con la forma en que la computadora almacena los números. Sabemos que los almacena en forma binaria en vez de nuestro familiar sistema decimal. Un entero tiene 16 dígitos binarios o bits ($2^{16} = 32768$), uno de los cuales se usa para el signo.

Calificadores

Existen algunos calificadores que se aplican a los tipos básicos. El uso de calificadores sirve para proporcionar diferentes longitudes de enteros donde sea práctico.

Cada compilador puede seleccionar libremente los tamaños apropiados para su propio hardware, sujeto sólo a restricción de que los short e int son por lo menos de 16 bits, los long son por lo menos de 32 bits.

Se debe cumplir : $\text{short} < \text{int} < \text{long}$



Otros calificadores

Los números unsigned son siempre positivos o cero, obedecen a la aritmética módulo 2^n , con n número de bits.

Los números signed pueden asumir valores menores a cero.

Tabla del tipo de dato int

int	16	-32768 a 32767
unsigned int	16	0 a 65535
signed int	16	-32768 a 32767
short int	16	-32768 a 32767
unsigned short int	16	0 a 65535
signed short int	16	-32768 a 32767
long int	32	-2147483648 a 2147483647
signed long int	32	-2147483648 a 2147483647
unsigned long int	32	0 a 4294967295
long	32	-2147483648 a 2147483647
unsigned long	32	0 a 4294967295

TIPO DE DATO FLOAT Y DOUBLE: TAMAÑO, CALIFICADORES

El tipo float es punto flotante de precisión normal (4Bytes = 32 Bits).

El tipo double es punto flotante de precisión extendida (8Bytes = 64 Bits)..

Calificador para los tipos de datos de de punto flotante

Tres tipos de punto flotante, este calificador permite trabajar con precisión extendida.

float < double < long double

Tabla del tipo de dato float y double

float	32	3.4E-38 a 3.4E+38
double	64	1.7E-308 a 1.7E+308
Long double	64 o 80 (según versión)	1.7E-308 a 1.7E+308 ó 3.4E-4932 a 1.1E+4932

¿Cómo saber los valores máximos y mínimos según el tipo de dato?

Los archivos de encabezado (ordenes para el pre-procesador) : <limits.h> y <float.h> contienen constantes simbólicas para los tipos de datos int, longint, float, double y longdouble.

```
INT_MAX  INT_MIN  LONG_MAX  LONG_MIN
FLT_MAX  FLT_MIN  DBL_MAX   DBL_MIN
```

OPERADORES

El lenguaje C es rico en operadores incorporados. Un operador es un símbolo que le indica al compilador que realice manipulaciones matemáticas o lógicas específicas. Existen cuatro clases de operadores:

- Aritméticos Binarios: + - * / %
- De Relación : < <= > >=
- De Igualdad : == (igualdad) != (distinto)
- Lógicos: && (y) || (o)

Asignación

Algoritmo	C	Descripción
←	=	Asignación

Aritméticos

Algoritmo	C	Descripción
+	+	Suma
-	-	Resta
.	*	Producto
<u>Div</u>	/	Cociente división entera
<u>Mod</u>	%	Resto división entera
/	/	División

Prioridades de Operadores

Prioridad	Operadores
Más Alta	()
	!
	* / %
	+ -
	< <= > >=
	== !=
	&&
Más Baja	= += -= *= /=

Conversiones y asignación para los tipos numéricos

En una expresión, cuando un operador tiene operandos de distintos tipos, ambos se convierten a un tipo común. En general las conversiones son de un tipo angosto a un tipo ancho sin pérdida de información.

`int < float < double < long double`

`Operando1 = operando2;`

Si son de distinto tipo, el `operando2` se convierte al tipo del `operando 1`.

Si `i` es de tipo `int`:

`i=34.567;`  `i=34;`

El valor del punto flotante se trunca si se asigna a una variable entera.

Si `j` es de tipo `float`:

`j=34 + 100;`  `i=134.000000;`

Los enteros se convierten a flotante.

Operadores de incremento y decremento

Pueden ser usados como prefijos y postfijos, incrementando antes o después de la utilización del valor. El incremento (++) o decremento (--) es 1.

Operadores de asignación y expresiones

$i = i + 2$ es equivalente a $i += 2$

El operador de asignación es: operador =

Los operadores aritméticos binarios se asocian con un operador de asignación:

$+=$ $-=$ $*=$ $/=$ $\%=$

Ejemplos:

$x *= y + 1; \rightarrow x = x * (y + 1);$

$i -= g; \rightarrow i = i - g;$

$j \% = (j - 2) \rightarrow j = j \% (j - 2)$

Código ASCII: código estándar para el intercambio de Información

ASCII (American Standard Code for Information Interchange Estándar)

Creado en 1963 por el Comité Estadounidense de Estándares este organismo cambio su nombre en 1969 por "Instituto Estadounidense de Estándares Nacionales" o "ANSI" como se lo conoce desde entonces.

¿Por qué hace falta la conversión?

Cada procesador de palabras codifica textos, números y los datos necesarios para el formato de márgenes, paginación, etc. En cambio, un texto en ASCII es simplemente un texto sin atributos de ninguna especie y se rige por una tabla universal.

La opción "Sólo texto" de los procesadores de palabras mas populares transforma el archivo en un texto ASCII.

El código ASCII establece un criterio de orden:


- cada signo
 - cada número
 - cada letra
 - cada carácter de control
- 
- Tiene un código numérico

Tabla ASCII

Caracteres ASCII de control

\	barra invertida (alt + 92)
@	arroba (alt + 64)
ñ	eñe minúscula (alt + 164)
'	comilla simple, apóstrofe (alt + 39)
#	signo numeral (alt + 35)
!	signo de admiración (alt + 33)
_	guión bajo, subrayado (alt + 95)
*	asterisco (alt + 42)
~	equivalencia, tilde (alt + 126)
·	guión medio (alt + 45)

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

Caracteres ASCII imprimibles

¿Cuál es el objetivo del uso de ASCII?

En concreto, el ASCII establece una correspondencia entre estas cadenas y los símbolos que representa (como letras y números). Esto hace que se facilite el almacenamiento y el procesamiento de información en una computadora y la comunicación entre diferentes aparatos digitales.

El uso de un código de caracteres en particular resulta extremadamente útil para evitar la pérdida de información como consecuencia de la incompatibilidad entre diferentes programas.

TIPO DE DATO CARÁCTER: TAMAÑO, CALIFICADORES

Las variables de tipo char almacenan 1 solo carácter. Declaración de una variable de tipo carácter:

```
char letra;
```

Declaración e inicialización: `char letra = 'a';`

Tamaño del tipo de dato carácter

2^n , n : número de bits.

8 bits = 1 byte

n = 1 byte

$2^8=256$

La tabla de
caracteres ASCII
básica tiene 256
caracteres.

ASCII, caracteres, valores enteros

Una constante de caracteres un entero, escrita entre ' ' (comillas simples).

```
char letra = 'a'; // 97 en ASCII
```

```
char digito = '0'; // 48 en ASCII
```

¿Operaciones permitidas ?

Las constantes de carácter pueden participar de operaciones aritméticas (por su “respaldo” ASCII), aunque se utilizan mas en comparaciones con otros caracteres. Algunos caracteres se representan como constantes de cadena: `\n`
Son 2 caracteres que representan uno solo (Secuencias de escape).