

LibreOffice

Guías de Referencia LibreOffice

LibreOffice Basic n.º 5

Biblioteca de Ejecución

v. 1.14 – 26/04/2019

Principiante

Redactado con LibreOffice v. 5.3.3 – Plataforma : Todas

Opciones de ejecución

Especificaciones, **por módulo**, antes de cualquier código ejecutable.

Option Explicit	Impone la declaración explícita de variables.
Option Compatible	Para que LibO Basic se comporte como VBA.
Option VBASupport 1	Activa la compatibilidad o el apoyo a VBA.
Option Base 1	Indexar los arrays empezando por 1 en lugar de 0.
Option ClassModule	Utilizado para crear clases (+ Option Compatible).

Constantes Basic

True	Verdadero (Boolean)	Empty	Variable no inicializada.
False	Falso (Boolean)	Null	Variable sin contenido útil
Pi	3.14159265358979 (Double)	Nothing	(objetos) elimina la asignación anterior.

Funciones

☞ **Sintaxis general de las funciones** Resultado = NombreFunción(argumentos)

Funciones, Cadenas de texto (o caracteres) (tipo String)

Asc()	Devuelve el valor Ascii (del 1 ^{er} carácter) de una cadena de texto. ☛ Asc("Azerty") → 65 *Vea Chr(), Tabla Ascii.
Chr()	Devuelve el carácter al que pertenece el código Ascii expresado. ☛ Chr(65) → "A" *Vea Asc(), Tabla Ascii.
ConvertFromURL()	Convierte un nombre de archivo del formato URL al formato del S.O. Format URL: protocolo://host/ruta/a1/archivo.ext ☛ Windows: file:///c:/rep/archivo.ods ☛ Linux: file:///home/user/rep/archivo.ods
ConvertToURL()	Convierte un nombre de archivo del formato del S.O. al formato URL *Vea ConvertFromURL()
Format()	Covierte un número en texto con el formato de una máscara. ☛ 14/4/2021, Format(Now(), "yyyy") → "2021" *Vea Función Format – Máscaras de formato.
InStr()	Busca en una cadena de texto la posición de otra cadena y devuelve el número que ocupa. *Si no la encuentra, devuelve 0. ☛ InStr("LibreOffice", "Office") → 6
Join()	Une los elementos de un array para convertirlos en una cadena Array1 = Array("C:", "Rep", "SsRep", "MiArchivo.ods") ☛ Join(Array1, "\") → "C:\Rep\SsRep\MiArchivo.ods" *Vea Split()
LCase()	Convierte una cadena a minúsculas ☛ LCase("LibreOffice") → "libreoffice" *Vea UCase()
Left()	Left(cadena, N) Extrae N caracteres de la cadena desde la izq. ☛ Left("LibreOffice", 5) → "Libre" *Vea Mid(), Right()
Len()	Devuelve el número de caracteres de una cadena. ☛ Len("LibreOffice") → 11
LTrim()	Elimina los espacios a la izquierda de una cadena (iniciales). Vea RTrim(), Trim()
Mid()	Mid(cadena, P, N). Extrae N caracteres. Del interior de una cadena a partir de la posición P. ☛ Mid("14/7/2017", 4, 1) → "7" *Vea Left(), Right()
Right()	Right(cadena, N). Extrae N caracteres de una cadena desde la derecha. *Vea Left(), Mid()
RTrim()	Elimina los espacios a la derecha de una cadena (finales) Vea LTrim(), Trim()
Space()	Devuelve una cadena constituida por N espacios. ☛ Space(3) → " " *Vea String()
Split()	Convierte una cadena en un array tipo texto separando los elementos por un carácter especificado (se elimina en el resultado) MiCadena = "C:\Rep\SsRep\MiArchivo.ods" ☛ Split(MiCadena, "\") → un array de 4 elementos: "C:", "Rep", "SsRep", "Miarchivo.ods" *Vea Join()
Str()	Convierte una expresión numérica en cadena de texto. ☛ Str(-65) → "-65" ☞ Un espacio precede al texto. El separador decimal tiene la clave. *Vea CStr(), Val()
StrComp()	Compara dos cadenas y devuelve un valor entero que indica el resultado de la comparación.
String()	Crema una cadena de N veces un carácter. ☛ String(4, "Y") → "YYYY" *Vea Space()
Trim()	Elimina los espacios a izquierda y derecha de una cadena. *Vea LTrim(), RTrim()
UCase()	Convierte una cadena en Mayúsculas. ☛ UCase("LibreOffice") → "LIBREOFFICE" *Vea LCase()
Val()	Convierte una cadena en un valor numérico (0 si no se puede conv.) ☛ Val("12,34") → 12,34 *Vea Str(), Val()

Funciones Numéricas

Abs()	Devuelve el valor absoluto de un número.
Exp()	Exponencial. (Eleva a una potencia).
Fix()	Devuelve la parte entera de un número (sin redondearlo).
Hex()	Devuelve el valor hexadecimal de un número decimal.
Int()	Devuelve la parte entera de un número (redondeado a la baja).
Log()	Devuelve el logaritmo de un número.
Oct()	Devuelve el valor octal de un número decimal.

Randomize()	Inicializa un generador de números aleatorios *para la función Rnd().
Rnd()	Devuelve un número aleatorio entre 0 y 1. *Vea Randomize()
Sgn()	Devuelve el signo de un número.
Sqr()	Calcula la raíz cuadrada de un número.

Funciones Trigonométricas

Ángulos en radianes. (use la formula) radianes = (grados * Pi)/180
 Atn() Arco tangente. Cos() Coseno. Tan() Tangente.

Funciones fecha / hora

Formato de fecha «UNO»

L'API LibreOffice utiliza de manera habitual las fechas «Uno»
 com.sun.star.util.DateTime (.Date o bien .Time), que se estructura así:
 IsUTC True si la zona hor. es UTC. Hours Horas (0-23).
 Year Nº del año Minutes Minutos (0-59).
 Month Nº del mes (0 si está vacía). Seconds Segundos (0-59).
 Day Nº del día (0 si está vacía). NanoSeconds Nanosegundos.

☞ **Date → Uno Date:** utilice las funciones de conversión CDateXxx siguientes.

Funciones fecha / hora

CDateFromISO()	Convierte una cadena fecha en formato ISO al formato Date *formato iso = (AAAAMMJJ). Date = en función de conf. local ☛ CDateFromISO("20170714") → 14/07/2017.																				
CDateFromUnoDate()	Convierte una estructura UNO com.sun.star.util.Date en un valor de tipo Date.																				
CDateFromUno	Convierte una estructura UNO																				
DateTime()	com.sun.star.util.DateTime en un valor de tipo Date.																				
CDateFromUnoTime()	Convierte una estructura UNO com.sun.star.util.Time en un valor de tipo Date.																				
CDateToISO()	Convierte la fecha en formato Date al formato ISO ☛ 14/7/2017, CDateToISO(Now()) → "20170714"																				
CDateToUnoDate()	Convierte la fecha a la forma estructurada UNO com.sun.star.util.Date.																				
CDateToUnoDateTime()	Convierte fecha y hora a la forma estructurada UNO com.sun.star.util.DateTime.																				
CDateToUnoTime()	Convierte hora a la forma estructurada UNO com.sun.star.util.Time.																				
Date()	Devuelve la fecha actual (tipo Date). *Vea Now(), Time()																				
DateAdd()	Devuelve una nueva fecha calculada por una fecha de inicio, una máscara de tiempo y un criterio de ajuste (±). ☛ 14/7/2017, DateAdd("m", -1, Now()) → 14/6/2017 Máscaras de unidades de fecha: <table border="0" style="margin-left: 20px;"> <tr> <td>yyyy</td> <td>Año</td> <td>ww</td> <td>Semana del año</td> </tr> <tr> <td>q</td> <td>Trimestre</td> <td>d</td> <td>Día</td> </tr> <tr> <td>m</td> <td>Mes</td> <td>h</td> <td>Hora</td> </tr> <tr> <td>y</td> <td>Día del año</td> <td>n</td> <td>Minuto</td> </tr> <tr> <td>w</td> <td>Día de la semana</td> <td>s</td> <td>Segundo</td> </tr> </table>	yyyy	Año	ww	Semana del año	q	Trimestre	d	Día	m	Mes	h	Hora	y	Día del año	n	Minuto	w	Día de la semana	s	Segundo
yyyy	Año	ww	Semana del año																		
q	Trimestre	d	Día																		
m	Mes	h	Hora																		
y	Día del año	n	Minuto																		
w	Día de la semana	s	Segundo																		
DateDiff()	Calcula la diferencia entre dos fechas, expresada en una unidad a elegir *Vea tabla en DateAdd(). ☛ DateDiff("m", "14/8/2017", "14/7/2017") → 1																				
DatePart()	Devuelve la parte especificada de la fecha (unidad elegida) *Vea tabla en DateAdd(). ☛ DatePart("q", "14/7/2017") → 3																				
DateSerial()	Devuelve un valor fecha para una fecha calculada por tres elementos separados por coma (año, mes, día). ☛ DateSerial(2017, 7, 14) → 14/7/2017																				
DateValue()	Convierte una cadena de texto a un formato tipo Date ☛ DateValue("14/7/2017") → 14/07/2017 (tipo Date)																				
Day()	Devuelve el número del día del mes. ☛ Day("14/7/2017") → 14																				
Hour()	Devuelve la hora ☛ Hour(Now()) → 12 (si es mediodía)																				
Minute()	Devuelve los minutos ☛ Minute(Now()) → 0 (si es una hora en punto)																				
Month()	Devuelve el número del mes. ☛ Month("14/7/2017") → 7																				
Now()	Devuelve la fecha y hora actual (en formato Date). *Vea Date(), Time()																				
Second()	Devuelve los segundos de un valor de tipo Date. ☛ Second(Now()) → 0 (si es la hora en punto)																				
Time()	Devuelve la hora actual en tipo Date. *Vea Date(), Now()																				
Timer()	Devuelve un Double que indica el número de segundos desde la medianoche. ☞ Asigne Timer() a una variable antes de usarlo																				
TimeSerial()	Devuelve un valor dato calculado desde tres elementos separados por coma (hora, minuto y segundo) ☛ TimeSerial(12,25,14) → 12:25:14 (tipo Date)																				
TimeValue()	Convierte una cadena de texto válida en un valor tipo Date. ☛ TimeValue("12:25:14") → 12:25:14 (tipo Date) (instrucción) hace una pausa de un tiempo especificado en milisegundos. - wait 1000 → pausa de 1 segundo.																				
Wait	Devuelve el número del día de la semana (1 = domingo). ☛ Weekday("14/7/2017") → 6 (viernes)																				
WeekDay()	Devuelve el número del año ☛ Year("14/7/2017") → 2017																				

Funciones, Colores

Los colores se almacenan en variables Long.
 Red(), Green(), Blue() Extraen los valores de los componentes de un color
 RGB() Devuelve el número (Long) corresp. a un color expresado en componentes
 RGB(128,0,0) → 8388608 (marrón)

Funciones, Arrays

Array() Crea un array a partir de valores discretos.
 MiArray = Array("Uno", 2, Now())

DimArray() Equivalente a Array() ➔ MiArray = DimArray("Uno", 2, Now())
 Use esta opción sólo si está activa la declaración implícita de variables.
 Erase (Instrucción) Borra el contenido del array. Si es dinámico lo libera de memoria. ➔ Erase MiArray

LBound() Límite inferior. UBound() Límite superior.

Funciones de consulta de tipo

Estas funciones permiten obtener información de las variables

De cualquier variable

TypeName() Devuelve una cadena de texto detallando la variable indicada.
 VarType() Devuelve un valor numérico relativo a la variable indicada.
 IsUnoStruct() Devuelve True si la variable indicada es una estructura UNO.

Las dos primeras funciones devuelven los valores indicados en la tabla:

Tipo	Nombre de l tipo	Tipo	Nombre de l tipo	Tipo	Nombre de l tipo
0	Empty	5	Double	11	Boolean
1	Null	6	Currency	12	Variant
2	Integer	7	Date	17	Byte
3	Long	8	String	37	Decimal
4	Single	9	Object		

De tipo variants

Devuelvenn True según el tipo real detectado.

Función	Verifica el tipo	Función	Verifica el tipo
IsArray()	Array	IsNull()	Null (sin datos).
IsDate()	Date.	IsNumeric()	Valor numérico.
IsEmpty()	Variable no inicializada	IsObject()	Objeto OLE.
IsError()	Valor de error.	IsUNOStruct()	True si es estructura UNO.

De tipo objeto

(Vea Estructuras y objetos UNO)

Funciones de conversión de tipos (typecast)

Estas funciones convierten un valor de un tipo (compatible) a otro. El nombre de la función refleja el tipo destino.

Legibilidad del código: Utilice con preferencia un typecast explícito a uno implícito.

CBool()	a Boolean	Cdbl()	a Double	CSng()	a Single
CByte()	a Byte	CDec()	a Decimal	CStr()	a String
CCur()	a Currency	CInt()	a Integer	CVar()	a Variant
CDate()	a Date	CLng()	a Long	CVerr()	a Variant (Error)

Estructuras y objetos UNO

CreateUnoService(Nomb)	Crea un servicio UNO. ⚠ Nomb sensible a mayúsculas
IsUNOStruct()	True si es una estructura UNO.
(struct.)Dbg_Properties	Devuelve el nombre de la estructura UNO (String).
HasUnoInterfaces()	True si el objeto UNO promueve Interfaces.
(obj.)SupportsService()	True si obj promueve el servicio aumentado (String).
EqualUnoObjects(o1, o2)	True si dos var. se refieren a la misma instancia del objeto.

Funciones de información de errores

ErL	Nº de línea de error	Error	Mensaje asociado al error.
Err	Código del error		

Otras Funciones

GetGUIType()	Devuelve un valor que indica el sistema operativo usado
	1 Windows 4 OSX ou Linux
	3 MacOS
GetSolarVersion()	Devuelve la versión de LibreOffice.
IsMissing()	Comprueba si se ha omitido un parámetro opcional (sub o func.)

Llamadas a comandos del sistema

Syntaxis del comando: Shell(Comand, Estilo, Param, Sinc), opciones:

Comando	El comando que se desea ejecutar (String).
Estilo	Se refiere a la ventana donde ejecutar el comando (Integer). Opciones:
0	El programa recibe el foco y su ventana está oculta.
1	El programa recibe el foco y se ejecuta en una ventana estándar.
2	El programa recibe el foco y se ejecuta en una ventana minimizada.
3	El programa recibe el foco y se ejecuta en una ventana maximizada.
4	El programa se inicia en una ventana sin foco.
6	El programa se inicia minimizado, el foco se queda en la ventana actual.
10	El programa se inicia en modo pantalla completa.
Param	Parámetros de ejecución pasados como argumento al programa (String).
Sinc	Señal de seguimiento de la ejecución
	True Esperar al final de la ejecución del comando.
	False No esperar al final de la ejecución del comando.

Función Format – Máscaras de formato

La función Format() convierte un núm. en cadena, (formato según máscara de formato).

Una **máscara de formato** es una cadena que puede ser mostrada en tres secciones separadas por punto y coma ➔ val>0;val<0;val=0. *Una sección sola = todos los números.

Los números se formatean según el idioma configurado en Herramientas ► Opciones ► Configuración de idiomas ► Idiomas.

Números

0	Cifra obligatoria en la posición (0 si no se indica)	%	El resultado se muestra en formato de porcentaje.
#	Cifra opcional	E- E+	Formato científico.
.	Separador decimal	e- e+	
+ - espacio	Caracteres literales se muestran igual en el resultado.	\	Caracter de escape: el car. que sigue se muestra en el resultado.

Fechas

D o DD	Nº del día (1 o 2 car.)	Q o QQ	Nº trimestre (1 o 2 car.)
M o MM	Nº del mes (1 o 2 car.)	W o WW	Nº semana (1 o 2 car.)
MMM	Nombre del mes.	h o hh	Hora (1 o 2 car.)
YY o YYYY	Nº del año (2 o 4 car.)	m o mm	Minutos (1 o 2 car.)
NNN	Nombre del día.	s o ss	Segundos (1 o 2 car.)

Soporte VBA

La compatibilidad con VBA no es completa.

Opciones de entorno

Herramientas ► Opciones ► Cargar / guardar ► Propiedades de VBA

Cargar el código Basic Carga y guarda el código VBA de un documento MSOffice en un módulo especial LibreOffice Basic.

Código ejecutable El código VBA se cargará preparado para su ejecución.

Guardar el código Basic original El código VBA en el documento se guarda aparte durante la carga del documento en LibreOffice.

Opciones de ejecución

La compatibilidad con VBA necesita que se indique Option VBASupport 1 y Option Compatible en la cabecera del módulo.

Funciones VBA

AscW	FV()	IRR()	Round()
ChrW	Input()	Me()	RTL()
DDB()	InStrRev()	MIRR()	StrReverse()
FormatDateTime()	IPmt()	NPer()	WeekDayName()

Encuentre más detalles en la ayuda en línea

Instrucciones VBA

```
Enum Enum MiEnumeracion
    WINDOWS = 1 ' Windows
    OS2PM = 2 ' OS/2 Presentation Manager
    MACINTOSH = 3 ' Macintosh
    MOTIF = 4 ' Motif Window Manager / Unix-like
    OPENLOOK = 5 ' Open Look / Unix-like
End Enum
```

A los valores enumerados se les asigna el tipo Long.

Los nombres de las enumeraciones y sus valores tienen que ser **únicos** dentro de una biblioteca.

Tabla Ascii

Déc	Hex	Val									
0	0	NUL	32	20	SPC	64	40	@	96	60	`
1	1	SOH	33	21	!	65	41	A	97	61	a
2	2	STX	34	22	"	66	42	B	98	62	b
3	3	ETX	35	23	#	67	43	C	99	63	c
4	4	EOT	36	24	\$	68	44	D	100	64	d
5	5	ENQ	37	25	%	69	45	E	101	65	e
6	6	ACK	38	26	&	70	46	F	102	66	f
7	7	BEL	39	27	'	71	47	G	103	67	g
8	8	BS	40	28	(72	48	H	104	68	h
9	9	HT	41	29)	73	49	I	105	69	i
10	0A	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	
29	1D	GS	61	3D	=	93	5D]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL

Creditos

Autor : Jean-François Nifenecker – jean-francois.nifenecker@laposte.net

Somos como enanos sentados sobre los hombros de gigantes. Si vemos más cosas y más lejanas que ellos, no es por la perspicacia de nuestra visión, ni por nuestra grandeza, sino porque son ellos los que nos elevan. (Bernard de Chartres [atribuido])

Historial

Versión	Fecha	Comentarios
1.01	1/10/17	Primera versión.
1.14	26/04/19	Correcciones y Ajustes
	07/04/2021	Traducción al español : B. Antonio Fernández

El documento original se puede obtener en la [Wiki francesa de publicaciones de L.O.](#)

Licencia

Esta guía de referencia está bajo licencia

Creative Commons BY-SA v3 (fr).

Información de la licencia : [en español](#)

<https://creativecommons.org/licenses/by-sa/3.0/fr/>

