Лекция 4.

Функции VBA (Часть 2)

РХТУ им. Д.И. Менделеева
Каф. ИКТ
Курс создал: ст. преп. А.М. Васецкий

2013 г

Функции проверки типов

IsArray(x)	Является ли переменная массивом
IsDate(x)	Является ли переменная датой
IsEmpty(x)	Была ли переменная описана инструкцией Dim
IsError(x)	Является ли переменная кодом ошибки
IsNull(x)	Является ли переменная пустым значением (Null)
IsNumeric(x)	Является ли переменная числовым значением
IsObject(x)	Является ли переменная объектом

Функции преобразования типов

Функция	Тип, в который преобразуется выражение
CBool (Выражение)	Boolean
CByte (Выражение)	Byte
CCur (Выражение)	Currency
CDate (Выражение)	Date
CDbl (Выражение)	Double
CDec (Выражение)	Decimal
CInt (Выражение)	Integer
CLng (Выражение)	Long
CSng (Выражение)	Single
CVar (Выражение)	Variant
CStr (Выражение)	String

Функции преобразования форматов. Преобразование строки в число и обратно

Val(строка)	Возвращает числа, содержащиеся в строке, как числовое значение соответствующего типа. Распознаёт также 8-ричные и 16-ричные представления чисел.
Str(число)	Возвращает значение типа Variant (String), являющееся строковым значением числа

Примеры:

Val("2457") ☞ 2457
Val(" 2 45 7") ☞ 2457
Val("24 and 57") ☞ 24
Val("24,57") ☞ 24

Str(459) ☞ " 459".
Str(-459.65) ☞ "-459.65".
Str(459.001) ☞ " 459.001".

Прим. **В качестве разделителя распознаётся только десятичная точка «.»**. В иных случаях использовать операторы **CDbl** и **CStr**

Зарезервированные константы для работы со строками

Код	Константа	Действие
0	vbNullChar	Chr(0)
	vbNullString	Пустая строка
8	vbBack	Backspace
9	vbTab	Tab
10	vbLf	Новая строка
13	vbCr	Возврат каретки
	VbCrLf	VbCr+VbLf
	vbNewLine	VbCr+VbLf в зависимости от платформы

Функции обработки строк

Asc	Возвращает ASCII-код начальной буквы строки. Синтаксис: **Asc(Строка)**
Chr	Преобразует ASCII-код в строку. Синтаксис: **Chr(Код)**
Lcase	Преобразует строку к нижнему регистру. Синтаксис: **Lcase(Строка)**
Ucase	Преобразует строку к верхнему регистру. Синтаксис: **Ucase(Строка)**
Space	Возвращает строку, состоящую из указанного числа пробелов. Синтаксис: **Space (Число)**
String	Возвращает строку, состоящую из указанного числа повторений одного и того же символа. Синтаксис: **String (number, character)** Аргументы: number – число повторений символа character – повторяемый символ
Tab	Размножает символы табуляции. Если никакое количество не указано, просто вставляет символ табуляции. Tab(Число)

Функции обработки строк (продолжение)

Mid	Возвращает подстроку строки, содержащую указанное число символов. Синтаксис: Mid [string, start [, length]) Аргументы: string – строковое выражение, из которого извлекается подстрока start – позиция символа в строке string, с которого начинается нужная подстрока length – число возвращаемых символов подстроки.
Left	Возвращает подстроку, состоящую из заданного числа первых символов исходной строки. Синтаксис: Left (string, length) Аргументы: length – число символов; string – исходная строка
Right	Возвращает строку, состоящую из заданного числа последних символов исходной строки. Синтаксис: Right (string, length) Аргументы: length – число символов, string – исходная строка

Функции обработки строк (продолжение)

Len	Возвращает число символов строки. Синтаксис: **Len (Строка)**
LTrim	Возвращает копию строки без пробелов в начале. Синтаксис: **LTrim (Строка)**
RTrim	Возвращает копию строки без пробелов в конце. Синтаксис: **RTrim (Строка)**
Trim	Возвращает копию строки без пробелов в начале и в конце Синтаксис: **Trim (Строка)**
Replace	Возвращает строку, в которой заданная подстрока заменена на другую Синтаксис: **Replace(expression, find, replace[, start[, count[, compare]]])** Аргументы: expression – исходная строка; find – подстрока, поиск которой ведётся; replace – подстрока, на которую заменяется найденная. start – номер символа в исходной строке с которого начинается поиск; count – количество замен. По умолчанию заменяются все вхождения искомой подстроки. compare – режим сравнения (бинарный, текстовой и пр.).

Функции обработки строк (продолжение)

StrComp

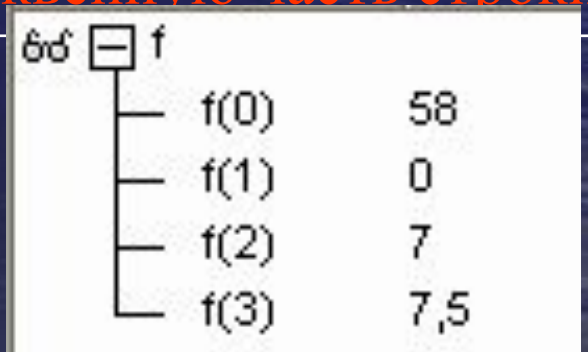
Возвращает результат сравнения двух строк.
Синтаксис: StrComp(string1, string2 [, compare])
Аргументы:
- string1 и string2 – два любых строковых выражения
- compare – указывает способ сравнения строк.

Допустимые значения: 0 (двоичное сравнение), 1 (посимвольное сравнение без учета регистра)
Возвращаемые значения:
- string1 меньше, чем string2, то -1
- string1 равняется string2, то 0
- string1 больше, чем string2, то 1

Val

Преобразует строку в соответствующий численный тип.
Синтаксис: Val(Строка)
Строка – Конвертируемая строка
При конвертации игнорирует буквенную часть строки!

f(0) = Val("58AB")
f(1) = Val("AB")
f(2) = Val("7,5")
f(3) = Val("7.5")



Функции обработки строк (продолжение)

InStr

InStrRev

Возвращает позицию первого вхождения одной строки внутри другой строки (**InStr** - с начала, **InStrRev** - с конца).

Синтаксис:

InStr ([start,] string1, string2 [, compare])

InstrRev(string1, string2[, start[, compare]])

Аргументы:

- start – числовое выражение, задающее позицию, с которой начинается каждый поиск. Если этот аргумент опущен, поиск начинается с первого символа строки
- string1 – строковое выражение, в котором выполняется поиск
- string2 – искомое строковое выражение
- compare – указывает способ сравнения строк. Допустимые значения: 0 (для двоичного сравнения), 1 (посимвольное сравнение без учета регистра)

Функции обработки строк (продолжение)

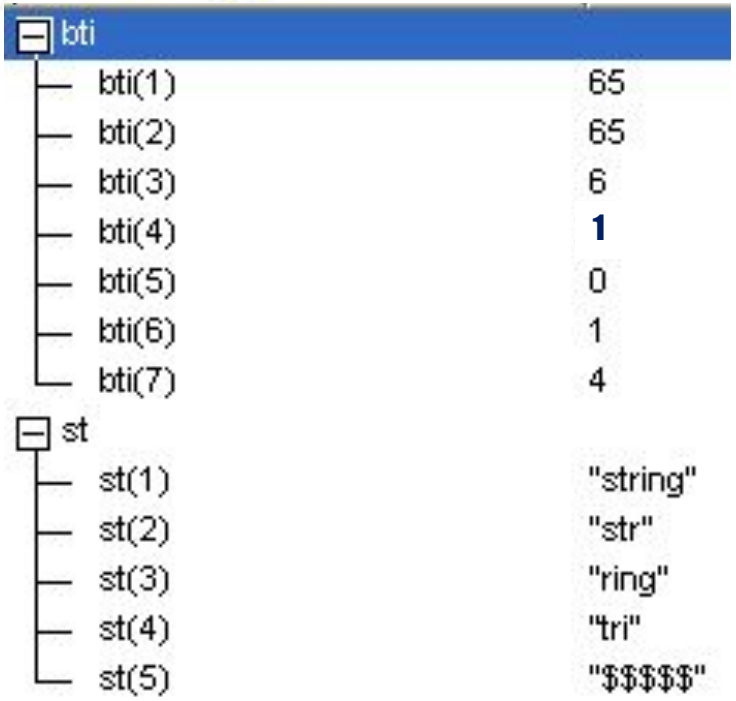
StrConv	Преобразует строку в соответствии с заданным форматом. Синтаксис: StrConv(Строка, Конверсия) Строка – Конвертируемая строка Конверсия – Сумма констант типа Integer, определяющие тип конверсии. Константы: • vbUpperCase – (1) – Конвертирует строку в верхний регистр • vbLowerCase – (2) – конвертирует строку в нижний регистр • vbProperCase – (3) – конвертирует первую буквы каждого слова в заглавную. • vbUnicode – (64) – Конвертирует строку в символы Юникода • vbFromUnicode – (128) – Конвертирует строку из Юникода
	Функции **Join** и **Split** рассмотрены ниже

Примеры

```
Sub string1()
Dim str As String, st(1 To 10) As String
Dim bti(1 To 7) As Byte, i As Byte
str = "string"
bti(1) = Asc("A")
bti(2) = Asc("Apple")
bti(3) = Len(str)

st(1) = LCase("STRING")
st(2) = Left(str, 3)
st(3) = Mid(str, 3)
st(4) = Mid(str, 2, 3)
st(5) = String(5, "$")

bti(4) = StrComp(str, st(1), vbBinaryCompare)
bti(5) = StrComp(str, st(1), vbTextCompare)
bti(6) = StrComp(str, st(2), vbTextCompare)
bti(7) = InStr(2, str, "i", vbBinaryCompare)
End Sub
```



Использование символов, которые нельзя ввести с клавиатуры

Иногда необходимо включить в строку какой-либо символ, для которого нет соответствующей клавиши на клавиатуре, а также какой-либо символ, который уже имеет особое значение для VBA, такой как символ кавычек ("). Чтобы включить в строку символы, которые невозможно ввести с клавиатуры, или которые имеют особое значение для VBA, используется VBA-функцию Chr.

Chr имеет следующий синтаксис: Chr(n)

Где n – число от 0 до 255 (см. следующий слайд)

Обозначения:

□ , € - не поддерживается MS Windows

** - специальный символ. Не отображается

Таблица кодов символов, используемая в Excel-2003

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

128	€	160	[space]	192	À	224	à
129	€	161	¡	193	Á	225	á
130	€	162	¢	194	Â	226	â
131	€	163	£	195	Ã	227	ã
132	€	164	¤	196	Ä	228	ä
133	€	165	¥	197	Å	229	å
134	€	166	¦	198	Æ	230	æ
135	€	167	§	199	Ç	231	ç
136	€	168	¨	200	È	232	è
137	€	169	©	201	É	233	é
138	€	170	ª	202	Ê	234	ê
139	€	171	«	203	Ë	235	ë
140	€	172	¬	204	Ì	236	ì
141	€	173		205	Í	237	í
142	€	174	®	206	Î	238	î
143	€	175	¯	207	Ï	239	ï
144	€	176	°	208	Ð	240	ð
145	€	177	±	209	Ñ	241	ñ
146	€	178	²	210	Ò	242	ò
147	€	179	³	211	Ó	243	ó
148	€	180	´	212	Ô	244	ô
149	€	181	µ	213	Õ	245	õ
150	€	182	¶	214	Ö	246	ö
151	€	183	·	215	×	247	÷
152	€	184	¸	216	Ø	248	ø
153	€	185	¹	217	Ù	249	ù
154	€	186	º	218	Ú	250	ú
155	€	187	»	219	Û	251	û
156	€	188	¼	220	Ü	252	ü
157	€	189	½	221	Ý	253	ý
158	€	190	¾	222	Þ	254	þ
159	€	191	¿	223	ß	255	ÿ

0=	1=□	2=□	3=□	4=□	5=□	6=□	7=□	8=□	9=
10=									
	11=□	12=□	13=	14=□	15=□	16=□	17=□	18=□	19=□
20=□	21=□	22=□	23=□	24=□	25=□	26=□	27=□	28=□	29=□
30=□	31=□	32=	33=!	34="	35=#	36=$	37=%	38=&	39='
40=(	41=)	42=*	43=+	44=,	45=-	46=.	47=/	48=0	49=1
50=2	51=3	52=4	53=5	54=6	55=7	56=8	57=9	58=:	59=;
60=<	61==	62=>	63=?	64=@	65=A	66=B	67=C	68=D	69=E
70=F	71=G	72=H	73=I	74=J	75=K	76=L	77=M	78=N	79=O
80=P	81=Q	82=R	83=S	84=T	85=U	86=V	87=W	88=X	89=Y
90=Z	91=[	92=\	93=]	94=^	95=_	96=`	97=a	98=b	99=c
100=d	101=e	102=f	103=g	104=h	105=i	106=j	107=k	108=l	109=m
110=n	111=o	112=p	113=q	114=r	115=s	116=t	117=u	118=v	119=w
120=x	121=y	122=z	123={	124=|	125=}	126=~	127=□	128=Ъ	129=Ѓ
130=‚	131=ѓ	132=„	133=…	134=†	135=‡	136=€	137=‰	138=Љ	139=‹
140=Њ	141=Ќ	142=Ћ	143=Џ	144=ђ	145=‘	146=’	147=“	148=”	149=•
150=–	151=—	152=□	153=™	154=љ	155=›	156=њ	157=ќ	158=ћ	159=џ
160=	161=Ў	162=ў	163=J	164=¤	165=Ґ	166=¦	167=§	168=Ё	169=©
170=Є	171=«	172=¬	173=-	174=®	175=Ї	176=°	177=±	178=I	179=i
180=ґ	181=µ	182=¶	183=·	184=ё	185=№	186=є	187=»	188=j	189=S
190=s	191=ї	192=A	193=Б	194=B	195=Г	196=Д	197=E	198=Ж	199=З
200=И	201=Й	202=K	203=Л	204=M	205=H	206=O	207=П	208=P	209=C
210=T	211=У	212=Ф	213=X	214=Ц	215=Ч	216=Ш	217=Щ	218=Ъ	219=Ы

Таблица, полученная программно

	0	1	2	3	4	5	6	7	8	9
0	0=	1=□	2=□	3=□	4=□	5=□	6=□	7=□	8=□	9=□
1	10=□	11=□	12=□	13=□	14=□	15=□	16=□	17=□	18=□	19=□
2	20=□	21=□	22=□	23=□	24=□	25=□	26=□	27=□	28=□	29=□
3	30=□	31=□	32=	33=!	34="	35=#	36=$	37=%	38=&	39='
4	40=(	41=)	42=*	43=+	44=,	45=-	46=.	47=/	48=0	49=1
5	50=2	51=3	52=4	53=5	54=6	55=7	56=8	57=9	58=:	59=;
6	60=<	61==	62=>	63=?	64=@	65=A	66=B	67=C	68=D	69=E
7	70=F	71=G	72=H	73=I	74=J	75=K	76=L	77=M	78=N	79=O
8	80=P	81=Q	82=R	83=S	84=T	85=U	86=V	87=W	88=X	89=Y
9	90=Z	91=[	92=\	93=]	94=^	95=_	96=`	97=a	98=b	99=c
10	100=d	101=e	102=f	103=g	104=h	105=i	106=j	107=k	108=l	109=m
11	110=n	111=o	112=p	113=q	114=r	115=s	116=t	117=u	118=v	119=w
12	120=x	121=y	122=z	123={	124=|	125=}	126=~	127=□	128=Ђ	129=Ѓ
13	130=‚	131=ѓ	132=„	133=…	134=†	135=‡	136=€	137=‰	138=Љ	139=‹
14	140=Њ	141=Ќ	142=Ћ	143=Џ	144=ђ	145=‘	146=’	147=“	148=”	149=•
15	150=–	151=—	152=□	153=™	154=љ	155=›	156=њ	157=ќ	158=ћ	159=џ
16	160=	161=Ў	162=ў	163=Ј	164=¤	165=Ґ	166=¦	167=§	168=Ё	169=©
17	170=Є	171=«	172=¬	173=-	174=®	175=Ї	176=°	177=±	178=І	179=і
18	180=ґ	181=µ	182=¶	183=·	184=ё	185=№	186=є	187=»	188=ј	189=Ѕ
19	190=ѕ	191=ї	192=А	193=Б	194=В	195=Г	196=Д	197=Е	198=Ж	199=З
20	200=И	201=Й	202=К	203=Л	204=М	205=Н	206=О	207=П	208=Р	209=С
21	210=Т	211=У	212=Ф	213=Х	214=Ц	215=Ч	216=Ш	217=Щ	218=Ъ	219=Ы
22	220=Ь	221=Э	222=Ю	223=Я	224=а	225=б	226=в	227=г	228=д	229=е
23	230=ж	231=з	232=и	233=й	234=к	235=л	236=м	237=н	238=о	239=п
24	240=р	241=с	242=т	243=у	244=ф	245=х	246=ц	247=ч	248=ш	249=щ
25	250=ъ	251=ы	252=ь	253=э	254=ю	255=я				

Функции, возвращающие строки

Некоторые функции имеют по две версии, одна из которых возвращает тип данных variant а другая – тип данных string. Первая версия является более удобной, так как при этом для значений типа variant преобразование типов данных выполняется автоматически. Вторая версия, возвращающая тип string, использует меньше памяти и может быть полезна в следующих случаях:

- Для экономии памяти, если в программе имеется очень много переменных
- При выполнении прямой записи данных в файлы с произвольным доступом

Chr$	CurDir$	Date$	Dir$
Error$	Format$	Input$	InputB$
LCase$	Left$	LTrim$	Mid$
Right$	Rtrim$	Space$	Str$
String$	Time$	Trim$	Ucase$

Прочие функции

DoEvents - важная функция. Она позволяет на время отвлечься от выполнения какой-то операции VBA и передать управление операционной системе, чтобы обработать накопившиеся в операционной системе события (например, нажатия клавиш пользователем). После этого продолжение операции VBA продолжается. Если запущена очень долгая операция (поиск на дисках, обработка большого объема данных и т.п.) и требуется дать пользователю возможность быстро прервать эту операцию, можно выполнять эту команду, например, каждый раз после обработки определенной «порции» данных.

Пример 1 «Антизамерзание»

```
Dim i, OpenForms
For i = 1 To 150000                'Начало цикла.
  If i Mod 1000 = 0 Then           ' 1000 циклов пройдено.
    OpenForms = DoEvents ' отдаём управление ОС.
  End If
Next i
```

Пример 2 «Таймер»

```
Sub SleepVB(Seconds) 'ожидание Seconds секунд
Dim Start
Start = Timer ' текущее время в секундах
Do While Timer < Start + Seconds
   DoEvents 'обеспечивает параллельное
выполнение других процессов
Loop
End Sub
```

Функция RGB

RGB - позволяет вернуть цветовой код, который можно использовать для присвоения цвета в коде, приняв три значения для цветов:

- красного (Red),
- зеленого (Green)
- синего (Blue).

Значение для каждого из основных цветов могут варьироваться от 0 до 255. Пример: зелёный цвет RGB(0,255,0).

Функция Shell

Shell - позволяет запустить из VBA внешний программный файл и вернуть информацию о его Program ID в операционной системе. Обычно используется опытными разработчиками при применении ими в программах возможностей Windows API. С практической точки зрения эту функцию можно использовать для запуска любых внешних программ из приложения

Синтаксис:

Shell(Путь[,Вид_окна])

В случае успеха функция возвращает ID запущенной программы.

- Путь – Путь к файлу
- Вид_окна – опциональный, отвечает за вид окна запущенной программы.

Значения:

- vbHide – скрытое окно
- vbNormalFocus – окно в фокусе
- vbMinimizedFocus – свёрнутое окно в фокусе
- vbMaximizedFocus – развёрнутое окно в фокусе
- vbNormalNoFocus – окно не в фокусе
- vbMinimizedNoFocus – свёрнутое не в фокусе

Пример:

```
Dim RetVal
RetVal = Shell("C:\WINDOWS\CALC.EXE", vbNormalFocus).
```

Функции для работы с массивами

Array - позволяет автоматически создать массив нужного размера и типа и сразу загрузить в него переданные значения.

Пример:

```
Dim MyWeek, MyDay 'переменные должны быть типа Variant
MyWeek = Array("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")
MyDay = MyWeek(2) ' в MyDay  "Wed".
```

Примечание: индексы массива начинаются с 0

Границы массива

UBound(arrayname[, dimension]) - возвращает информацию о верхней границе массива

LBound(arrayname[, dimension]) - возвращает информацию о нижней границе массива

- arrayname – имя массива
- dimension – индекс размерности

Пример:

```
Dim A(1 To 100, 0 To 3, -3 To 4)

B=UBound(A, 1) ‘ B=100
B=UBound(A, 2) ‘ B=3
B=LBound(A, 3) ‘ B=-3
B=UBound(A, 3) ‘ B=4
```

Функция Join

Join(sourcearray[, delimiter]) - возможность слить множество строк из массива строк в одну строковую переменную. В качестве разделителя по умолчанию используется пробел.

- sourcearray – имя массива
- delimiter – разделитель. Если он "" то строки склеиваются без разделителя.

Пример:

```
Dim avArr
avArr = Array("Первый элемент", "Второй элемент", "3", 4, "Последний")
MsgBox Join(avArr, "-") 'разделитель "-"
```

Функция Split

Split(expression[, delimiter[, limit[, compare]]]) - функция, разбивающая строку на массив строк. В качестве разделителя по умолчанию используется пробел, можно указать свой разделитель.

- expression – исходное строковое выражение, содержащее разделители
- delimiter – разделитель
- limit – количество возвращаемых подстрок
- compare – режим сравнения

Пример:

```
Dim avArr
avArr=Split("Первый-Второй-3-4-Последний", "-")
    'показываем 3-й по порядку элемент
    MsgBox avArr(2)
```

Формат

Чтобы представить числовое значение как дату, время, денежное значение или в специальном формате, следует использовать функцию **Format**.

Синтаксис: **Format (Выражение [, Формат [, ПервыйДеньНедели [, ПерваяНеделяГода]]])**

- Выражение – любое допустимое выражение
- Формат – любое допустимое именованное или определяемое пользователем выражение формата. Примером именованного формата является Fixed – формат действительного числа с двумя значащими цифрами после десятичной точки
- ПервыйДеньНедели – постоянная, определяющая первый день недели
- ПерваяНеделяГода – постоянная, определяющая первую неделю года

Константы

vbUseSystem	0	Использовать NLS API настройки
VbSunday	1	Воскресенье (по умолчанию)
vbMonday	2	Понедельник
vbTuesday	3	Вторник

vbWednesday	4	Среда
vbThursday	5	Четверг
vbFriday	6	Пятница
vbSaturday	7	Суббота

vbUseSystem	0	Использовать NLS API настройки
vbFirstJan1	1	Начинать с недели, где 1 января (по умолчанию)
vbFirstFourDays	2	Начинать с недели, где есть по крайней мере 4 дня года
vbFirstFullWeek	3	Начинать с первой полной недели года

Системные форматы даты и времени

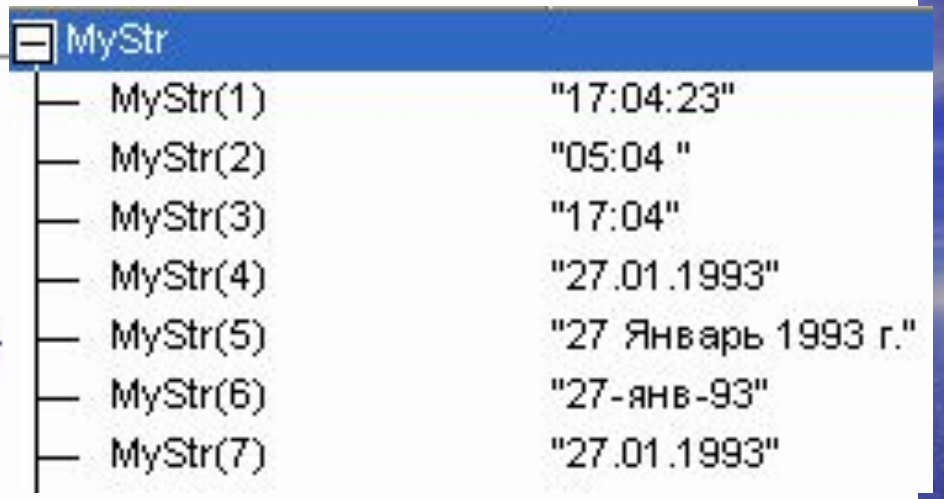
General Date	Показывает дату и/или время. Например: 4/3/93 05:34 PM Если нет дробной части, Показывает только дату, например: 4/3/93. Если нет целой части – только время. Например: 05:34 PM.
Long Date	Показывает время в системном формате Long date (длинный формат даты)
Medium Date	Показывает дату в формате Medium Date (средний формат даты) в зависимости от языковой версии приложения
Short Date	Показывает дату в кратком формате Short Date
Long Time	Показывает время в Long Time формате (часы, минуты, секунды)
Medium Time	Показывает время в 12-часовом формате с использованием AM/PM обозначений
Short Time	Показывает время в 24-часовом формате

Примеры формата и времени

```
Option Explicit

Sub dat()
Dim MyTime, MyDate
Dim i As Byte
Dim MyStr(1 To 10) As String
MyTime = #5:04:23 PM#
MyDate = #1/27/1993#


MyStr(1) = Format(MyTime, "Long Time")
MyStr(2) = Format(MyTime, "Medium Time")
MyStr(3) = Format(MyTime, "Short Time")
MyStr(4) = Format(MyDate, "General Date")
MyStr(5) = Format(MyDate, "Long Date")
MyStr(6) = Format(MyDate, "Medium Date")
MyStr(7) = Format(MyDate, "Short Date")
```



Пользовательские форматы даты и времени

:	Разделитель часов, минут и секунд в категории форматов Время (Time)
/	Разделитель дня, месяца и года в категории форматов Дата (Date)
c	Дата показывается как ddddd и время как ttttt. Показывает только дату если нет дробной части и только время, если нет целой части
d	Показывает день (1-31)
dd	Показывает день (01-31)
ddd	Показывает день как аббревиатуру (Пн-Вс)
dddd	Показывает день в полном виде (Понедельник - Воскресенье)
ddddd	Показывает дату в полном виде (день, месяц, год) в соответствии с системными настройками короткого представления
dddddd	Показывает дату в полном виде (день, месяц, год) в соответствии с системными настройками полного представления.

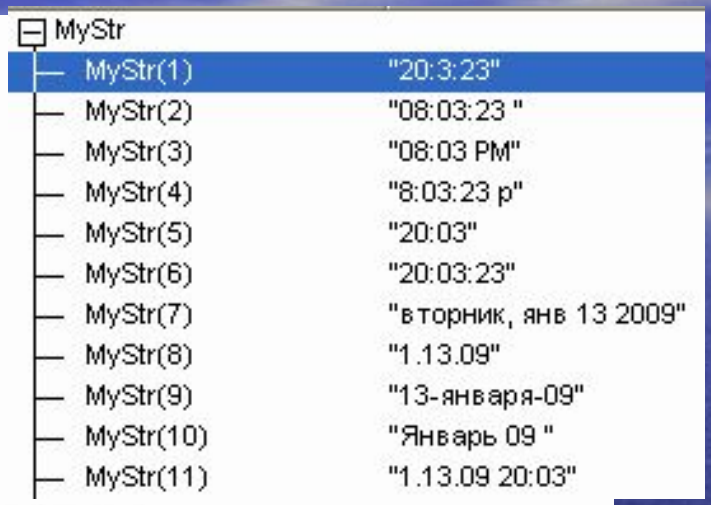
aaaa	То же, что и dddd, только это локализованная версия строки
w	Показывает порядковый номер дня недели (1- Вс, 7 – Сб)
ww	Номер недели (1-54)
m	Номер месяца (1-12)
mm	Номер месяца (01-12)
mmm	Месяц как аббревиатура (Янв-Дек)
mmmm	Полное название месяца (Январь-Декабрь)
oooo	То же, что и mmmm, только локализованная версия
q	Номер квартала 1-4
y	Номер дня 1-366
yy	Номер года 00-99
yyyy	Номер года 100-9999

Примеры пользовательских форматов даты и времени

```
Sub DatFormat()
Dim MyTime, MyDate
Dim i As Byte
Dim MyStr(1 To 20) As String
MyTime = #8:03:23 PM#
MyDate = #1/13/2009#

MyStr(1) = Format(MyTime, "h:m:s")
MyStr(2) = Format(MyTime, "hh:mm:ss AMPM")
MyStr(3) = Format(MyTime, "hh:mm AM/PM")
MyStr(4) = Format(MyTime, "h:mm:ss a/p")
MyStr(5) = Format(MyTime, "h:mm")
MyStr(6) = Format(MyTime, "h:mm:ss")

MyStr(7) = Format(MyDate, "dddd, mmm d yyyy")
MyStr(8) = Format(MyDate, "m/d/yy")
MyStr(9) = Format(MyDate, "d-mmmm-yy")
MyStr(10) = Format(MyDate, "mmmm yy ")
MyStr(11) = Format(MyDate & " " & MyTime, "m/d/yy h:mm")
```

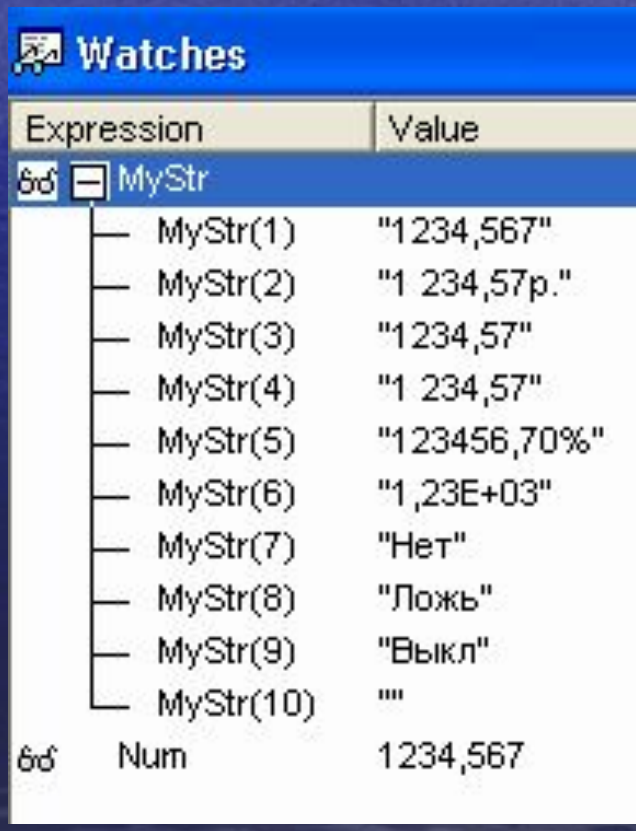


Общие числовые форматы

General Number	Общий числовой формат без разделителей тысяч
Currency	Показывает число с разделителем тысяч. Два разряда справа от десятичного сепаратора. Вывод основывается на базе локальных языковых настроек
Fixed	Показывает по крайней мере 1 разряд слева и 2 разряда справа от десятичной точки
Standard	Показывает число с разделителем тысяч. Присутствует по крайней мере один разряд слева и 2 разряда справа от десятичной точки
Percent	Показывает число, умноженное на 100 со знаком % справа. Всегда показывает 2 разряда справа от разделителя
Scientific	Использует стандартную научную запись
Yes/No	**0** – Нет; **не 0** – Да
True/False	**0** – False (Ложь); **не 0** – True (Истина)
On/Off	**0** – Off (Выкл); **не 0** – On (Вкл)

Примеры числовых форматов

```
Option Explicit
Sub NumFormat()
Dim i As Byte, Num As Double
Dim MyStr(1 To 10) As String
Num = 1234.567
i = 0
MyStr(1) = Format(Num, "General Number")
MyStr(2) = Format(Num, "Currency")
MyStr(3) = Format(Num, "Fixed")
MyStr(4) = Format(Num, "Standard")
MyStr(5) = Format(Num, "Percent")
MyStr(6) = Format(Num, "Scientific")
MyStr(7) = Format(i, "Yes/No")
MyStr(8) = Format(i, "True/False")
MyStr(9) = Format(i, "On/Off")
```



Пользовательские форматы

0	Резервирует позицию цифрового разряда. Отображает цифру или нуль. Если у числа, представленного аргументом, есть какая-нибудь цифра в той позиции разряда, где в строке формата находится 0, функция отображает эту цифру аргумента, если нет – в этой позиции отображается нуль
#	Резервирует позицию цифрового разряда. Отображает цифру или ничего не отображает. Если у числа, представленного аргументом, есть какая-нибудь цифра в той позиции разряда, где в строке формата находится #, функция отображает эту цифру аргумента, если нет – в исходной позиции не отображается ничего. Действие данного символа аналогично действию 0, за исключением того, что лидирующие нули не отображаются
.	Резервирует позицию десятичного разделителя. Указание точки в строке формата определяет, сколько разрядов необходимо отображать слева и справа от десятичной точки

Пользовательские форматы (продолжение)

%	Резервирует процентное отображение числа
,	Разделитель разряда сотен от тысяч
E+, E-, e+, e-	Разделитель мантиссы и порядка в экспоненциальном формате
-+$()	Литералы. Чтобы отобразить любой другой символ, надо заключить его в кавычки или поставить перед ним знак «\»
\	Показывает следующий за обратным слэшем символ. Чтобы вывести сам обратный слэш надо использовать его дважды \\
["ABC"]	Выводит текст, заключённый в кавычки. Сами кавычки можно отобразить если использовать оператор Chr(34)
>	Преобразует символы строки в малые прописные
<	Преобразует символы строки в большие прописные

Примеры

Если формат не поддерживается, то возвращается числовое значение: **MyStr = Format(23)** ☞ "23"

Пользовательские форматы:

MyStr = Format(5459.4, "##,##0.00") ☞ **"5,459.40"**

MyStr = Format(334.9, "###0.00") ☞ **"334.90".**

MyStr = Format(5, "0.00%") ☞ **"500.00%".**

MyStr = Format("HELLO", "<") ☞ **"hello".**

MyStr = Format("This is it", ">") ☞ **"THIS IS IT".**

Примеры

Format	+5	-5	0,5	Null
("")	5	-5	0.5	
0	5	-5	1	
0.00	5.00	-5.00	0.50	
#,##0	5	-5	1	
#,##0.00;;;Nil	5.00	-5.00	0.50	Nil
$#,##0; ($#,##0)	$5	($5)	$1	
$#,##0.00; ($#,##0.00)	$5.00	($5.00)	$0.50	
0%	500%	-500%	50%	
0.00%	500.00%	-500.00%	50.00%	
0.00E+00	5.00E+00	-5.00E+00	5.00E-01	
0.00E-00	5.00E00	-5.00E00	5.00E-01	

СПАСИБО ЗА ВНИМАНИЕ!