

Некоторые допустимые значения аргумента **Buttons** (для MsgBox)

<i>Константа</i>	<i>Значение</i>	<i>Описание</i>
<code>vbOKOnly</code>	0	Кнопка "ОК".
<code>VbOKCancel</code>	1	Кнопки "ОК" и "Отмена" (Cancel).
<code>VbAbortRetryIgnore</code>	2	Кнопки "Прервать" (Abort), "Повторить" (Retry) и "Пропустить" (Ignore).
<code>VbYesNoCancel</code>	3	Кнопки "Да" (Yes), "Нет" (No) и "Отмена" (Cancel).
<code>VbYesNo</code>	4	Кнопки "Да" (Yes) и "Нет" (No).
<code>VbRetryCancel</code>	5	Кнопки "Повторить" (Retry) и "Отмена" (Cancel).
<code>VbCritical</code>	16	Значок "Критическое сообщение".
<code>VbQuestion</code>	32	Значок "Предупреждающий запрос".
<code>VbExclamation</code>	48	Значок "Предупреждение".

Возвращаемые значения (какая кнопка «щелкалась»)

Нажатая кнопка	Значение
ОК	1
Отмена (Cancel)	2
Прервать (Abort)	3
Повторить (Retry)	4
Пропустить (Ignore)	5
Да (Yes)	6
Нет (No)	7

Или используя системные переменные, например, vbYes vbIgnore

Типы переменных указывают, что может хранить переменная

Описать переменную можно с помощью следующей конструкции

Dim имя_переменной As тип_переменной

Описание типа переменной может быть помещено в любом месте программы, но обычно располагается вначале программы

Если переменная **не была описана**, она «воспринимается» программой, как **переменная типа Variant**.

Тип Variant – указывает на то, что переменная может хранить все, что в нее поместят, т.е. ее тип изменяется, в зависимости от последнего присвоения.

Числовые типы (для хранения чисел)

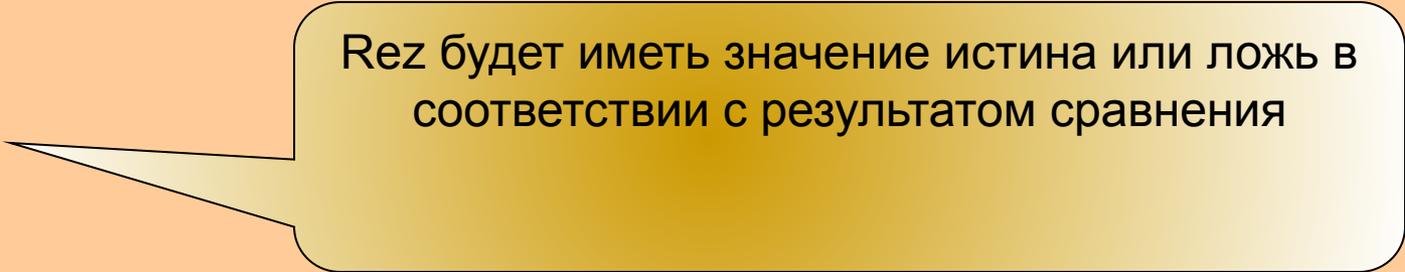
<i>Тип</i>	<i>Данные</i>
Integer (целое)	числа от $-3 \cdot 10^4$ до $3 \cdot 10^4$, занимает 2 байта памяти
Long (длинное целое)	числа от $-2 \cdot 10^9$ до $2 \cdot 10^9$, занимает 4 байта памяти
Single (с плавающей точкой обычной точности)	числа от $-3 \cdot 10^{38}$ до $-1 \cdot 10^{-45}$ для <u>отрицательных</u> значений и от $1 \cdot 10^{-45}$ до $3 \cdot 10^{38}$ для <u>положительных</u> значений, занимает 4 байта памяти
Double (с плавающей точкой двойной точности)	числа от $-1 \cdot 10^{308}$ до $-4 \cdot 10^{-324}$ для <u>отрицательных</u> значений и от $4 \cdot 10^{-324}$ до $1 \cdot 10^{308}$ для <u>положительных</u> значений, занимает 8 байтов памяти

Логический тип.

Boolean (логический). Переменные этого типа могут принимать значения True (истина) или False (ложь). Этим переменным могут присваиваться логические выражения, например

```
Dim Rez As Boolean
```

```
Rez = A > B
```



Rez будет иметь значение истина или ложь в соответствии с результатом сравнения

К переменным Boolean применяются логические операции **And** (И) **Or** (Или) **Not** (Не) **Xor** (исключающее Или).

Строковый тип.

String (строка переменной длины) 10 байт + длина строки, хранит данные от 0 до приблизительно 2 миллиардов символов.

По умолчанию строковая или текстовая переменная является массивом переменной длины, которая содержит символы

Dim R As String

Но можно описать **текстовую переменную фиксированной длины**, например, 25 символов

*Dim R As String*25*

В этом случае, если переменной *R* будет присвоена строка длиной более 25 символов – она будет усечена.

Дата и время.

Date переменные этого типа хранят значения начиная с 1 января 1900 г. 0:00:00 до 31 декабря 2078 г. 23:59:59.

Для присвоения даты и времени можно использовать одно из следующих выражений:

Xd = #31/3/64 9:50#

Xd = #19:35#

Xd = #Март 3, 1964 12:51#

Xd = #3 Март 1998#

Xd = #1 Апр#

Значение даты заключается в символы #

Функции даты и времени

Получить *текущее время и дату*

$n = Now$

Выделить *число часов* из даты

$h = Hour(n)$

Выделить *число минут* из даты

$m = Minute(n)$

Выделить *число дней* из даты

$d = Day(n)$

Выделить *год лет* из даты

$y = Year(n)$

Выделить *месяц* из даты

$mn = Month(n)$

Для переменных типа дата можно использовать **математические операции**:

Прибавить *один день*

$xd = xd + 1$

Прибавить *один час*

$xd = xd + 1 / 24$

Прибавить *одну минуту*

$xd = xd + 1 / 1440$

Прибавить *одну секунду*

$xd = xd + 1 / 86400$

Функции преобразования типов данных

Правило использования:

ИмяФункции(Выражение)

Обязательный аргумент Выражение является любым строковым выражением или числовым выражением

Возвращаемые типы

ИмяФункции	Тип	Выражение
CBool	Boolean	Любая строка или числовое выражение.
CByte	Byte	Число от 0 до 255.
CCur	Currency	Число от $-9,5808 \cdot 10^{14}$ до $9,5807 \cdot 10^{14}$.
CDate	Date	Любое допустимое выражение даты.
CDbl	Double	От $-1 \cdot 10^{308}$ до $-4 \cdot 10^{-324}$ для отрицательных чисел; от $4 \cdot 10^{-324}$ до $1 \cdot 10^{308}$ для положительных чисел.
CDec	Decimal	числа $\pm 7 \cdot 10^{28}$ (без дробной части) или числа с 28 знаками в дробной части. Минимальное ненулевое число $1 \cdot 10^{-28}$.
CInt	Integer	От -32 768 до 32 767 с округлением дробной части.
CLng	Long	От $-2 \cdot 10^9$ до $2 \cdot 10^9$ с округлением дробной части.
CSng	Single	От $-3 \cdot 10^{38}$ до $-1 \cdot 10^{-45}$ для отрицательных чисел; от $1 \cdot 10^{-45}$ до $3 \cdot 10^{38}$ для положительных чисел.
CVar	Variant	Диапазон значений Double для числовых значений. Диапазон значений String для нечисловых значений.
CStr	String	Возвращаемые значения функции CStr зависят от аргумента Выражение.

Обработка текстовых переменных (строк)

Конкатенация (склейка) строк

Конкатенация двух строк – это объединение их в одну строку. Конкатенация строк выполняется с помощью оператора «&». Для конкатенации используют и знак «+», но такой метод является устаревшим.

Пример

A = "Премия"

B = "за год"

C = A & " " & B

Переменная C получит значение
Премия за год

Функции

обработки текстовых строк

Функция **Len**

Возвращает число символов, из которых образована строка (Сколько символов в строке).

Правило записи: **$K = Len(\text{строка})$**

здесь **K** – целочисленная переменная, для хранения числа символов, которые возвращает функция;

строка – текстовое выражение, длину которого необходимо определить .

Пример

$A = \text{“Иван”}$

$KB = Len(A)$

Переменная KB станет
равной 4

Усечение строк

Функция *Right*

Возвращает значение содержащее указанное число последних символов строки. Из текста (строки) извлекает указанное число символов справа.

Правило записи: ***PS = Right(строка, длина)***

здесь *PS* – текстовая переменная, для хранения символов, которые возвращает функция;

строка Текстовое выражение, из которого извлекаются символы. Если *строка* имеет значение Null, возвращается значение Null.

длина Числовое выражение, указывающее число возвращаемых символов. Если 0, возвращается пустая строка (""). Если превышает число символов в строке string, возвращается вся строка.

Усечение строк

Функция *Left*

Возвращает значение, содержащее указанное число первых символов строки.
Аналогична функции *Right*.

Правило записи: ***FS = Left(строка, длина)***

Примеры

Пусть имеется переменная *MS = "Всем привет"*

A	<i>t</i>
B=	<i>ивет</i>
C	<i>Всем привет</i>

A	<i>B</i>
B	<i>Всем</i>
C	<i>Всем привет</i>

Функция *Mid*

Возвращает значение, содержащее **указанное число символов** строки. Из текста (строки) извлекает его часть (подстроку).

Правило записи: ***CS = Mid(строка, начало[, длина])***

Здесь **CS** – текстовая переменная, для хранения символов, которые возвращает функция (подстрока);

строка Строковое выражение, из которого извлекаются символы. Если **string** имеет значение Null, возвращается значение Null.

начало номер позиции символа в строке, с которого начинается нужная подстрока. Если **начало** больше числа символов в строке, функция *Mid* возвращает пустую строку ("").

длина **Необязательный**. Число возвращаемых символов. Если этот аргумент опущен или превышает число символов, расположенных справа от позиции **начало**, то возвращаются все символы от позиции **начало** до конца строки.

Инструкция *Mid*

Заменяет указанное число символов в переменной символами из другой строки.

Правило записи: ***Mid(пр, начало[, длина]) = стр***

здесь

пр Имя строковой переменной, подлежащей изменению.

начало Позиция символа в переменной, с которого начинается замена.

длина **Необязательный**. Число символов, подлежащих замене. Если этот аргумент опущен, используется вся строка.

стр Строковое выражение, предназначенное для замены части строки.

Примечание. Число замененных символов не может превышать число символов в переменной.

Примеры

$Ms = \text{“Заседание состоится в четверг”}$

Функция $Mid()$

$F = Mid(Ms, 11, 9)$

переменная **F** получит
значение
СОСТОИТСЯ

Инструкция $Mid()$

$Mid(Ms, 11, 11) = \text{“переносится на”}$

переменная **Ms** получит значение
Заседание переносится на четверг

Удаление пробелов в текстовой строке

Функции *Rtrim* *Ltrim* *Trim*

Возвращают значение, содержащее копию строки, из которой удалены пробелы, находившиеся в начале строки (LTrim), в конце строки (RTrim) или в начале и конце строки (Trim).

Правило записи: $CT = LTrim(\text{строка})$

$CT = RTrim(\text{строка})$

$CT = Trim(\text{строка})$

Обязательный аргумент *строка* представляет любое допустимое строковое выражение. Если строка имеет значение Null, возвращается значение Null.

CT копия строки без пробелов.

Изменение регистра

Функции *UCase* *LCase*

UCase «делает» все буквы большими

LCase «делает» все буквы маленькими

Примечание на цифры не влияет.

Правило записи: ***СТ = UCase(строка)***

СТ = LCase(строка)

Пример

СТ = ПРИВЕТ

СТ = пока