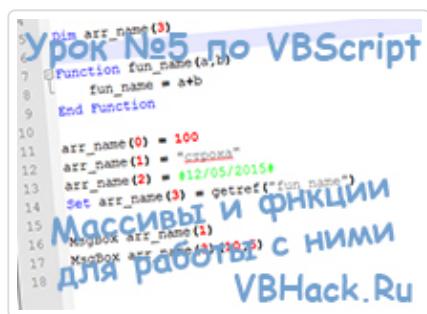


## ◀ (http://vbhack.ru) Урок VBScript №5: Массивы и функции для работы с НИМИ

📅 24 июля, 2015    📁 Уроки VBScript (http://vbhack.ru/category/uroki-vbscript)    🔍

Array (http://vbhack.ru/tag/array)	End Function (http://vbhack.ru/tag/end-function)	Erase (http://vbhack.ru/tag/erase)
Function (http://vbhack.ru/tag/function)	IsArray (http://vbhack.ru/tag/isarray)	LBound (http://vbhack.ru/tag/lbound)
Preserve (http://vbhack.ru/tag/preserve)	ReDim (http://vbhack.ru/tag/redim)	UBound (http://vbhack.ru/tag/ubound)
VBScript (http://vbhack.ru/tag/vbscript)	массивы (http://vbhack.ru/tag/massivy)	



В этой статье я расскажу про **массивы языка VBScript** и про функции для работы с ними. Массивы бывают одномерными и многомерными. Они будут очень полезны нам в будущем, и если с ними сразу разобраться, то никаких проблем в следующих уроках у Вас не возникнет.

Основы основ:

- Название массива vbs имеет произвольный вид.
- После его названия, в круглых скобках, пишется его длина.
- Если в скобках ничего не прописано, то массив динамический и его длину можно изменять.
- Массив начинается с нуля.

Давайте сразу перейдём к самому простому примеру:

</>



```
'Урок VBScript №5:
'Массивы и фнкции для работы с ними
'file_1.vbs

dim arr_name(2)           'Массив из трёх элементов
dim matrix_name(1,3)      'матрица 1 (строки) на 4 (столбцов)
dim dyn_name()            'динамический массив

arr_name(0) = 100
arr_name(1) = "строка"
arr_name(2) = #12/05/2015#

matrix_name(0,0) = 1
matrix_name(0,1) = 10
matrix_name(0,2) = 100
matrix_name(0,3) = 1000

matrix_name(1,0) = 2
matrix_name(1,1) = 20
```

После того как объявили массивы и их длину надо каждому элементу массива присвоить значение, что и показано в примере. Хочу так же отметить, что массивы хранят не только числа, но и строки, а так же даты. Про работу с датой и временем (<http://vbhack.ru/uroki-vbscript/urok-vbscript-n7-funkcii-dlya-raboty-s-datoy-i-v.html>) я расскажу в следующих уроках.

Далее мы при помощи функции **MsgBox** получаем значение трёх элементов массивов. Каждый элемент показан с новой строки. В этом нам помогает **константа «vbCrLf»**, о которой мы уже знаем из первого урока.

**Пример динамического массива vbs** надо рассматривать отдельно:

```
</>
'Урок VBScript №5:
'Массивы и фнкции для работы с ними
'file_2.vbs

dim dyn_name()            'динамический массив

i = 0                     'начальное значение переменной i и массива dyn_name
Do                         'начало цикла
    Redim Preserve dyn_name(i) 'перезаписываем размерность массива с сохранением имеющийся инфо
    dyn_name(i) = i          'присваиваем значение каждому элементу массива
    i = i + 1                'увеличиваем значение переменной i на единицу
loop Until (i>100)          'выполнять пока i не станет больше 100 т.е. 101 элемент массива

MsgBox dyn_name(55)         'получаем значение 55'ого элемента массива
```

В данном примере я использовал **цикл Do ... Loop** (<http://vbhack.ru/uroki-vbscript/urok-vbscript-n4-cikly-while-wend-i-do-loop.html>) , который мы проходили в прошлом уроке.

Если мы используем **динамический массив VBScript**, то для назначения его размера используется ключевое слово **ReDim**.

Так же мы можем изменить это значение сколько угодно раз. Для того, что бы сохранить уже имеющиеся данные используется ключевое слово **Preserve (Redim Preserve)**, которое мы использовали в данном примере.

Если понадобится полностью отчистить содержимое массива, то используется ключевое слово **Erase (Erase dyn\_name)**. Этот способ работает для всех массивов.

## Функции: Array, IsArray, UBound и LBound.

- **Array(значения)** — Функция Array позволяет записать значение всех элементов массива в одну строку в круглых скобках через запятую.
- **IsArray(имя переменной)** — проверяет, является ли переменная массивом. Если да, то вернёт True в противном случае False
- **UBound(имя массива,измерение)** — позволяет определить верхнюю границу массива (номер элемента). Параметр «Измерение» не обязателен. О применяется с многомерными массивами.
- **LBound(имя массива,измерение)** — то же самое, что и UBound, но определяет нижнюю границу массива VBScript.

Нужно отметить, что функция Array не может быть применена с матрицей.

</>



```
'Урок VBScript №5:
'Массивы и функции для работы с ними
'file_3.vbs

dim arr_name
dim matrix_name(1,3)      'матрица 1 (строки) на 4 (столбцов)

arr_name = Array(100,"строка",#12/05/2015#) 'задаём значения для элементов массива
if IsArray(arr_name) then 'проверяем является ли переменная arr_name массивом
    MsgBox arr_name(1) 'выводим первый элемент массива
    a = UBound(arr_name) 'присваиваем переменной a верхнюю границу (Номер последнего элемен
    b = LBound(arr_name) 'присваиваем переменной b нижнюю границу (Номер первого элемента)
    MsgBox "Значение последнего элемента массива: " & arr_name(a)
    MsgBox "Значение первого элемента массива: " & arr_name(b)
else
    MsgBox "Переменная не является массивом" 'если переменная arr_name не является массивом
end if
```

На примере видно, что нам уже не надо указывать длину для одномерного массива. Мы просто превращаем **переменную arr\_name** в массив, используя функцию **Array**, указав круглых скобках значение элементов (*от 0 до 2*).

На 11-ой и 12-ой строке показан пример работы **функций UBound и LBound**. На примере 13-ой и 14-ой строки видно, что в массивах можно использовать переменные. Их так же можно использовать в момент присваивания значений элементов массива.

На 29-ой и 30-ой строке показана работа **функций UBound и LBound с массивом матрицей**. Если параметр «измерение» = 1, то мы получим номер строки, а если «измерение» = 2, то номер столбца. Если этот параметр не указывать, то мы увидим оба значения.

Надеюсь не возникло никакой путаницы из-за того, что я уместил все функции в одном примере. На всякий случай я оставил подробные комментарии.

Теперь можно усложнить задачу и выполнить **массив в массиве**. Язык VBScript нам это позволяет. Смотрим пример:

```
</>
'Урок VBScript №5:
'Массивы и функции для работы с ними
'file_4.vbs

dim arr_name

arr_name = Array(Array(10,"строка",#12/05/2015#,54),Array(11,3,15,44),Array(18,37,16,34)) 'задаём з
MsgBox arr_name(0)(2) & vbCrLf & arr_name(1)(2) & vbCrLf & arr_name(2)(2)
```

На примере видно, что внутри **функции Array** находятся ещё три функции Array со своими значениями.

Все эти три функции являются элементами **массива arr\_name** (*от 0 до 2*), со своими элементами (*от 0 до 3*).

В конце мы видим как мы получаем значение второго элемента из всех трёх массивов внутри массива.

Проще говоря: **arr\_name(Номер массива в массиве)(Номер элемента массива в массиве)**

С этим я думаю всё понятно.

## Функция Filter

Данная функция вернёт отфильтрованный массив по заданному критерию.

**Синтаксис:** **Filter(inputStrings, value[, include[, compare]])**

- **InputStrings** — Исходный одномерный массив состоящий из строк.
- **Value** — Строка, критерий отбора (фильтра).
- **Include** — Если указано True, то в результат фильтра попадут только строки содержащие критерий «Value». Если указано False, то в результат фильтра попадут строки не содержащие данного критерия.
- **Compare** — Сравнение: 0 — двоичный (с учётом регистра), 1 — текстовый (без учёта регистра).

```
</>
'Урок VBScript №4:
'Циклы While ... Wend и Do ... Loop
'file_5.vbs

dim i, arr_name(5), Arr_Filter, List

Arr_name(0) = "Понедельник"
Arr_name(1) = "Вторник"
Arr_name(2) = "Среда"
Arr_name(3) = "Четверг"
Arr_name(4) = "Пятница"
Arr_name(5) = "Суббота"

Arr_Filter = Filter(Arr_name, "ик", false, 1)

i = 0

Do Until i = UBound(Arr_Filter) + 1
List = List & Arr_Filter(i) & vbCrLf
```

В данном примере нам вернутся названия дней недели, в которых отсутствует «ИК».

## Функции Join и Split

Данные функции относятся к строковым функциям vbs (<http://vbhack.ru/uroki-vbscript/urok-vbscript-n9-strokovye-funkcii-rabota-s-t.html>), но так как они работают с массивами, я решил их описать и в этой статье.

**Функция Join** — Вернёт строку состоящую из элементов массива разделённых указанным разделителем.

**Синтаксис:** **Join(List[, Delim])**

- **List** — Разбираемый массив.
- **Delim** — Разделитель. По умолчанию пробел.

**Функция Split** — Вернёт массив из строк, полученных в результате разбития строки при помощи указанного разделителя.

**Синтаксис:** **Split(Expr[, Delim[, Count[, Compare]]])**

- **Expr** — Строка, которую разбиваем.
- **Delim** — Разделитель. По умолчанию пробел.
- **Count** — Число подстрок (создаваемых элементов массива). По умолчанию все (-1).
- **Compare** — Сравнение: 0 — двоичный (с учётом регистра), 1 — текстовый (без учёта регистра).

</>



```
'Урок VBScript №4:
'Циклы While ... Wend и Do ... Loop
'file_6.vbs

dim arr_name(5), Arr_Join, Arr_Split, i, List

Arr_name(0) = "Понедельник"
Arr_name(1) = "Вторник"
Arr_name(2) = "Среда"
Arr_name(3) = "Четверг"
Arr_name(4) = "Пятница"
Arr_name(5) = "Суббота"

Arr_Join = Join(Arr_name, ",")'массив в виде строки
MsgBox Arr_Join

Arr_Split = Split(Arr_Join, ",", -1, 0) 'создаём массив из строки

i = 0
```

## Функция (Function ... End Function) в качестве значения элемента массива VBScript.

Теперь посмотрим, как установить **функцию (Function ... End Function)** (<http://vbhack.ru/uroki-vbscript/urok-vbscript-n3-funkcii-function-end-function-i-procedura-sub-end-sub.html>) в качестве значения. Это уже немного потруднее. Давайте посмотрим на пример:

</>



```

'Урок VBScript №5:
'Массивы и фнкции для работы с ними
'file_7.vbs

Dim arr_name(3)

Function fun_name(a,b)
    fun_name = a+b
End Function

arr_name(0) = 100
arr_name(1) = "строка"
arr_name(2) = #12/05/2015#
Set arr_name(3) = getref("fun_name")

MsgBox arr_name(1)
MsgBox arr_name(3)(10,5)

```

В примере четвёртый элемент массива vbs выглядит иначе. Для того, что бы мы могли присвоить функцию (Function ... End Function) вместо значения элемента массива, мы *создали ссылку* ключевым словом **«Set»** а уже сама ссылка получена при помощи **функции getref(имя функции)**.

Далее при выводе элемента массива, в дополнительных круглых скобках, мы прописали параметры функции.

Добавить мне больше нечего. Думаю, на этом тема массивов VBScript закрыта.

3 comments



**Владимир** 3 года ago

reply (<http://vbhack.ru/uroki-vbscript/urok-vbscript-n5-massivy-i-fnkcii-dlya-raboty-s.html?replytoocom=13172#respond>)

В разделе «Функция Filter» допущена ошибка:

```

Arr_Filter = Filter(Arr_name, "ик", false, 1)
i = 0
Do Until i = UBound(Arr_Filter) + 1
List = List & Arr_Filter(i) & vbCrLf
i = i+1
Loop
MsgBox List

```

"Do Until i = UBound(Arr\_Filter) + 1" - здесь единица как бы лишняя, в противном случае

Вот немного переделал ваше решение в виде функции, с подсчетом кол-ва элементов (файл

```

Set DestFolder = objFSO.GetFolder("C:\video\ups\")
Dim testFiles()
ReDim Preserve testFiles(DestFolder.Files.Count)
i = 0
For Each bFile In DestFolder.Files
    i = i + 1
    testFiles(i) = bFile.Path
Next

GetDimensions(testFiles)

Function GetDimensions(testFiles)
    z=0
    Do Until z = UBound(testFiles)
        List = List & testFiles(z) & vbCrLf
        z = z+1
    Loop
    MsgBox z
End function

```

o



**admin** 3 года ago

<http://vbhack.ru/uroki-vbscript/urok-vbscript-n5-massivy-i-fnkcii-dlya-raboty-s.html?replytoocom=13915#respond>

Ошибки как раз нет. Попробуйте убрать единицу



**V** 7 месяцев ago

[reply \(http://vbhack.ru/uroki-vbscript/urok-vbscript-n5-massivy-i-fnkcii-dlya-raboty-s.html?replytoocom=24602#respond\)](http://vbhack.ru/uroki-vbscript/urok-vbscript-n5-massivy-i-fnkcii-dlya-raboty-s.html?replytoocom=24602#respond)

В первом примере перепутаны строка и столбец для функций **UBound** и **LBound**, должно быть вот так

~~~

a = UBound(matrix\_name,2) 'присваиваем переменной a номер строки последнего элемента массива матрицы

b = LBound(matrix\_name,1) 'присваиваем переменной b номер столбца начального элемента массива матрицы

~~~



Reply

Preview

Help



Markdown enabled

Name

E-Mail

URL

Код безопасности \*



Введите символы отображаемые выше:

Post comment

↑ Back to top

© VBHack

Theme by nehalist.io (<http://nehalist.io>)