

[Перейти на главную страничку сайта \(список статей, файлы для скачивания\)](#)

[ФОРУМ](#) (здесь можно обсудить эту статью, а также любые проблемы программирования на различных макроязыках и в скриптовых средах)

[Описание объектов Windows Script Host, их свойств и методов](#)

Объект RegExp

Содержание:

[1. Создание объекта](#)

[2. Свойства](#)

[2.1. Global](#)

[2.2. IgnoreCase](#)

[2.3. Pattern](#)

[2.4. Multiline](#)

[3. Методы](#)

[3.1. Replace](#)

[3.2. Test](#)

[3.3. Execute](#)

[4. Метасимволы](#)

1. Создание объекта

Пример:

```
Set objRegExp = CreateObject("VBScript.RegExp")
```

Возможности регулярных выражений:

- Сравнить текст с заданным шаблоном. Например, введенный пользователем телефонный номер должен соответствовать шаблону "8(***)***_***_***", где * - любая цифра, но не буква.
- Заменить или удалить из введенной пользователем строки текст, соответствующий заданному шаблону.
- Извлечь из введенной пользователем строки текст, соответствующий заданному шаблону.

Регулярное выражение - это образец текста, который состоит из обычных символов и/или специальных метасимволов.

2. Свойства

2.1. Global

Синтаксис:

Global

Возвращаемое значение: число (булево). False - проверять до первого соответствия, True - проверять по всему тексту. По умолчанию - False.

Замечание: чтение и запись.

Пример:

```
Set objRegExp = CreateObject("VBScript.RegExp")
objRegExp.Pattern = "abc"
TempStr = "abc 123 abc"
MsgBox TempStr 'исходная строка
Result = objRegExp.Replace(TempStr, "def")
MsgBox Result
objRegExp.Global = True
Result = objRegExp.Replace(TempStr, "def")
MsgBox Result
```

2.2. IgnoreCase

Синтаксис:

IgnoreCase

Возвращаемое значение: число (булево). False - учитывать регистр символов, True - игнорировать регистр символов. По умолчанию - False.

Замечание: чтение и запись.

```
Set objRegExp = CreateObject("VBScript.RegExp")
objRegExp.Pattern = "АБВ"
Str = "абв 123"
MsgBox Str 'исходная строка
Res = objRegExp.Replace(Str, "где")
MsgBox Res
objRegExp.IgnoreCase = True
Res = objRegExp.Replace(Str, "где")
MsgBox Res
```

2.3. Pattern

Синтаксис:

Pattern

Возвращаемое значение: строка, используемая как шаблон.

Замечание: чтение и запись.

Пример:

```
Set objRegExp = CreateObject("VBScript.RegExp")
objRegExp.Pattern = "абв"
Str = "абв 123"
MsgBox Str 'исходная строка
Res = objRegExp.Replace(Str, "где")
MsgBox Res
```

2.4. Multiline

Синтаксис:

Multiline

Возвращаемое значение: число (булево). False - однострочный объект, True - многострочный. По умолчанию - False.

Замечание: чтение и запись. Влияет на работу метасимволов ^ и \$.

Пример:

```
Set objRegExp = CreateObject("VBScript.RegExp")
objRegExp.Global = True
objRegExp.Pattern = "^абв"
Str = "абв 123 абв" & vbCrLf & "абв"
MsgBox Str 'исходная строка
Res = objRegExp.Replace(Str, "где")
MsgBox Res
objRegExp.Multiline = True
Res = objRegExp.Replace(Str, "где")
MsgBox Res
```

3. Методы

3.1. Replace

Синтаксис:

Replace(<strSource>,<strReplace>)

Назначение: замена соответствующих шаблону вхождений в строке-оригинале на указанную подстроку. Возвращает (возможно) изменённую строку.

Параметры:

- <strSource> - строка-оригинал (где заменять).
- <strReplace> - подстрока для замены (на что заменять).

Пример:

```
Set objRegExp = CreateObject("VBScript.RegExp")
objRegExp.Pattern = "абв"
Str = "абв 123"
MsgBox Str 'исходная строка
Res = objRegExp.Replace(Str, "где")
MsgBox Res
```

3.2. Test

Синтаксис:

Test(<strSource>)

Назначение: проверка соответствия шаблону. Возвращаемое значение - булево (число).

Параметры:

- <strSource> - строка для проверки.

Пример:

```
Set objRegExp = CreateObject("VBScript.RegExp")
objRegExp.Pattern = "абв+"
Str = "абвбвбвбвбв"
MsgBox objRegExp.Test(Str)
Str = "авббвбвбвбв"
MsgBox objRegExp.Test(Str)
```

3.3. Execute

Синтаксис:

Execute(<strSource>)

Назначение: поиск соответствующих шаблону вхождений в строке-оригинале. Возвращает коллекцию найденных подстрок в виде агрегатного объекта.

Параметры:

- <strSource> - строка для поиска.

Пример:

```
Set objRegExp = CreateObject("VBScript.RegExp")
objRegExp.Pattern = "абв+"
objRegExp.Global = True
Str = "абв 123 абв абвв 456"
Set objMatches = objRegExp.Execute(Str)
For i=0 To objMatches.Count-1
    Set objMatch = objMatches.Item(i)
    'найденное значение (подстрока)
    'индекс первого символа найденной подстроки в строке-оригинале
    'длина найденной подстроки
    MsgBox objMatch.Value & ", " & "FirstIndex=" & objMatch.FirstIndex & ", " & "Length=" & objMatch.Length
Next
```

4. Метасимволы

\	<p>Показывает, что следующий символ является спецсимволом. Последовательность "\\" соответствует "\", а последовательность \"(\" соответствует \"(\".</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Pattern = "[A-Z]:\\" Str = "C:\" Res = objRegExp.Test(Str) MsgBox "строка " & Str & " соответствует шаблону " & objRegExp.Pattern & " = " & Res</pre>
^	<p>Определяет начало входной строки. Если установлено свойство Multiline, определяет также позицию сразу после "\n" (новая строка) и "\r" (возврат каретки).</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True objRegExp.Pattern = "^абв" Str = "абв 123 абв" & vbCrLf & "абв" MsgBox Str 'исходная строка Res = objRegExp.Replace(Str, "где") MsgBox Res objRegExp.Multiline = True Res = objRegExp.Replace(Str, "где") MsgBox Res</pre>
\$	<p>Определяет конец входной строки. Если установлено свойство Multiline, определяет также позицию непосредственно перед "\n" (новая строка) и "\r" (возврат каретки).</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True objRegExp.Pattern = "абв\$" Str = "абв 123 абв" & vbCrLf & "абв" MsgBox Str 'исходная строка Res = objRegExp.Replace(Str, "где") MsgBox Res objRegExp.Multiline = True Res = objRegExp.Replace(Str, "где") MsgBox Res</pre>
*	<p>Определяет ни одного или несколько символов, стоящих перед ним. Эквивалентно {0,}.</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Pattern = "длинноше*" Str = "длинношеее" Res = objRegExp.Test(Str) MsgBox "строка " & Str & " соответствует шаблону " & objRegExp.Pattern & " = " & Res</pre>
+	<p>Определяет один или несколько символов, стоящих перед ним. Эквивалентно {1,}.</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Pattern = "длинноше+" Str = "длинношеее" Res = objRegExp.Test(Str) MsgBox "строка " & Str & " соответствует шаблону " & objRegExp.Pattern & " = " & Res</pre>
?	<p>Определяет ни одного или один символ, стоящий перед ним. Эквивалентно {0,1}. Если этот метасимвол идёт непосредственно за конструкциями (*, +, ?, {n}, {n,}, {n,m}), это приводит к некоторому изменению алгоритма поиска по заданному шаблону, что проиллюстрировано примером ниже.</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "000111000" objRegExp.Pattern = "0+" Set objMatches = objRegExp.Execute(Str) MsgBox "в строке " & Str & " найдено " & objMatches.Count & _ " вхождений по шаблону " & objRegExp.Pattern & "" objRegExp.Pattern = "0+?" Set objMatches = objRegExp.Execute(Str) MsgBox "в строке " & Str & " найдено " & objMatches.Count & _ " вхождений по шаблону " & objRegExp.Pattern & ""</pre>

{n}	<p>Определяет точное количество символов, стоящих перед ним. "n" - неотрицательное целое число.</p> <pre> Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "000111001110000" objRegExp.Pattern = "0{3}" Set objMatches = objRegExp.Execute(Str) MsgBox "в строке " & Str & " найдено " & objMatches.Count & _ " вхождений по шаблону " & objRegExp.Pattern & " " </pre>
{n,}	<p>Определяет минимальное количество символов, стоящих перед ним. "n" - неотрицательное целое число. "{0,}" эквивалентно "*", а "{1,}" эквивалентно "+".</p> <pre> Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "000111001110000" objRegExp.Pattern = "0{3,}" Set objMatches = objRegExp.Execute(Str) MsgBox "в строке " & Str & " найдено " & objMatches.Count & _ " вхождений по шаблону " & objRegExp.Pattern & " " </pre>
{n,m}	<p>Определяет количество символов, стоящих перед ним (от - до). "n" и "m" - неотрицательные целые числа, причём n <= m. "{0,1}" эквивалентно "?".</p> <pre> Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "000111001110000" objRegExp.Pattern = "0{2,3}" Set objMatches = objRegExp.Execute(Str) MsgBox "в строке " & Str & " найдено " & objMatches.Count & _ " вхождений по шаблону " & objRegExp.Pattern & " " </pre>
.	<p>Определяет любой символ, кроме "\n" (новая строка). Чтобы определить любой символ, включая "\n", следует использовать "[\s\S]".</p> <pre> Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "00" & vbCrLf & "00" objRegExp.Pattern = "." Set objMatches = objRegExp.Execute(Str) MsgBox "в строке " & Str & " найдено " & objMatches.Count & _ " вхождений по шаблону " & objRegExp.Pattern & " " objRegExp.Pattern = "[\s\S]" Set objMatches = objRegExp.Execute(Str) MsgBox "в строке " & Str & " найдено " & objMatches.Count & _ " вхождений по шаблону " & objRegExp.Pattern & " " </pre>
(шаблон)	<p>Фиксирует подбор по шаблону в коллекции SubMatches. Чтобы определить символы "(" и ")", следует использовать "\(и \)".</p> <pre> Set objRegExp = CreateObject("VBScript.RegExp") Str = "One = 111" & vbCrLf & "Two = 222" & vbCrLf & "Three = 333" 'прочитаем параметр "Two" objRegExp.Pattern = "Two = (.+\\r)" Set objMatches = objRegExp.Execute(Str) Set objMatch = objMatches.Item(0) MsgBox Trim(objMatch.Value) Set objSubmatches = objMatch.Submatches For i=0 To objSubmatches.Count-1 MsgBox Trim(objSubmatches.Item(i)) Next </pre>

(?:шаблон)	<p>Полезно для комбинирования частей шаблона с помощью символа " " ("или"). Например, "госпо(?:дин жа)" короче, чем "господин госпожа". НЕ фиксирует подбор в коллекции SubMatches.</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "господин госпожа господа" objRegExp.Pattern = "госпо(?:дин жа)" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count</pre>
(?=шаблон)	<p>"Windows(?:=95 98)" подберёт подстроку "Windows" в строке "Windows95", но не в строке "WindowsNT". Дальнейший подбор начинается немедленно, а не после символов, входящих в скобки. НЕ фиксирует подбор в коллекции SubMatches.</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "WindowsWindowsWindows" objRegExp.Pattern = "Windows(?:=Win)" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count For i=0 To objMatches.Count-1 MsgBox objMatches.Item(i).Value Next</pre>
(?!шаблон)	<p>"Windows(?:!95 98)" подберёт подстроку "Windows" в строке "WindowsNT", но не в строке "Windows95". В остальном - полностью аналогично "(?=шаблон)".</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "WindowsWindowsWindows" objRegExp.Pattern = "Windows(?:!Win)" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count For i=0 To objMatches.Count-1 MsgBox objMatches.Item(i).Value Next</pre>
x y	<p>Условие "или". "вз перевод" подберёт "вз" и "перевод". "(вз пере)вод" подберёт "взвод" и "перевод".</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "Windows Lindows Linux" objRegExp.Pattern = "(Wi Li)ndows" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count</pre>
[abc]	<p>Набор символов. Находит любой из указанных символов.</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "Windows Lindows Linux" objRegExp.Pattern = "[WL]indows" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count</pre>
[^abc]	<p>Исключающий набор символов. Находит любой из неуказанных символов.</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "Windows Lindows Linux" objRegExp.Pattern = "in[^uv]" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count</pre>
[a-z]	<p>Диапазон символов. Находит любой символ из диапазона.</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "Windows Lindows Linux" objRegExp.Pattern = "[L-W]indows"</pre>

	<pre>Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count</pre>
[^a-z]	<p>Исключающий диапазон символов. Находит любой символ, не входящий в диапазон.</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "Windows Lindows Linux" objRegExp.Pattern = "[^A-K]indows" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count</pre>
\b \B	<p>\b - Граница слова, т.е. позиция между концом слова и пробелом. \B - НЕ граница слова.</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "Windows Windows Windows98" objRegExp.Pattern = "ows\b" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count objRegExp.Pattern = "ows\B" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count</pre>
\d \D	<p>\d - цифра. Эквивалентно [0-9]. \D - НЕ цифра. Эквивалентно [^0-9].</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "WindowsNT Windows2000 Windows98" objRegExp.Pattern = "Windows\d" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count objRegExp.Pattern = "Windows\D" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count</pre>
\w \W	<p>\w - любой символ слова, включая символ подчёркивания. Эквивалентно [A-Za-z0-9_]. \W - обратно "\w". Эквивалентно [^A-Za-z0-9_].</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "Windows® Windows2000 Windows98" objRegExp.Pattern = "Windows\w" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count objRegExp.Pattern = "Windows\W" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count</pre>
\s \S	<p>\s - пробельный символ. Эквивалентно [\f\n\r\t\v]. \S - непробельный символ. Эквивалентно [^\f\n\r\t\v].</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "Windows Windows Windows98" objRegExp.Pattern = "Windows\s" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count objRegExp.Pattern = "Windows\S" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count</pre>
\cx	<p>Управляющий символ. Параметр "x" лежит в диапазоне A-Z или a-z. Например, "\cM" определяет Control-M или символ возврата каретки, "\cJ" - новая строка, "\cL" - новая страница, "\cI" - символ табуляции, "\cK" - символ вертикальной табуляции.</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = vbCrLf & vbCrLf</pre>

	<pre>objRegExp.Pattern = "\cM" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count</pre>
\f	Символ новой страницы. Эквивалент "\x0c" и "\cL".
\n	<p>Символ новой строки. Эквивалент "\x0a" и "\cJ".</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = vbCrLf objRegExp.Pattern = "\r\n" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count</pre>
\r	<p>Символ возврата каретки. Эквивалент "\x0d" и "\cM".</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = vbCrLf objRegExp.Pattern = "\r\n" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count</pre>
\t	<p>Символ табуляции. Эквивалент "\x09" и "\cI".</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = vbTab objRegExp.Pattern = "\t" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count</pre>
\v	<p>Символ вертикальной табуляции. Эквивалент "\x0b" и "\cK".</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = vbVerticalTab objRegExp.Pattern = "\v" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count</pre>
\xn	<p>Шестнадцатеричный ASCII-код символа. Должен содержать ровно два символа.</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = vbTab objRegExp.Pattern = "\x09" Set objMatches = objRegExp.Execute(Str) MsgBox objMatches.Count</pre>
\num	<p>Ссылка на предыдущие зафиксированные (найденные) подстроки-соответствия шаблону (SubMatches). "num" - положительное целое число, номер соответствия в коллекции SubMatches; отсчёт с единицы. Например, "(.)\1" определяет два идентичных символа, следующих друг за другом.</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True Str = "0:0:1-2:3:3-2:2:1-4:5:5-7:7:8" ' найдём группы, в которых две последние цифры совпадают objRegExp.Pattern = "(\d)(:)(\d)(:)\3" Set objMatches = objRegExp.Execute(Str) For i=0 To objMatches.Count-1 Set objMatch = objMatches.Item(i) MsgBox objMatch.Value Next</pre>
\un	<p>Определяет Unicode-символ по его шестнадцатеричному коду. "n" - четырёхзначный шестнадцатеричный код.</p> <pre>Set objRegExp = CreateObject("VBScript.RegExp") objRegExp.Global = True</pre>


```
Str = "© Microsoft © Sun"  
objRegExp.Pattern = "\u00A9 Sun"  
Set objMatches = objRegExp.Execute(Str)  
For i=0 To objMatches.Count-1  
    Set objMatch = objMatches.Item(i)  
    MsgBox objMatch.Value  
Next
```

Составление - [Людоговский Александр](#)

[Перейти на главную страничку сайта \(список статей, файлы для скачивания\)](#)

© 2007 <http://www.script-coding.com> При любом использовании материалов сайта обязательна **ссылка** на него как на источник информации, а также сохранение **целостности** и **авторства** материалов.