

Лабораторная работа

Использование языка VBScript

Использование языка VBScript позволяет упростить процесс создания скриптов при разработке сайтов.

Целью лабораторной работы является изучение некоторых полезных объектов языка VBScript, способов их использования и приобретение навыков в написании и отладке специальных программ при WEB-дизайне.

Конкретные задания на работу выделены в тексте *курсивом*.

1. Вводные замечания

VBScript является подмножеством языка Microsoft Visual Basic for Applications (сокращенно – VBA), поставляемого с Microsoft Office и Visual Basic.. Основные синтаксические принципы VBScript как языка программирования:

- VBScript нечувствителен к регистру;
- чтобы закомментировать код до конца строки, используется одинарная кавычка (') или команда REM;
- символьные значения должны заключаться в двойные кавычки;
- максимальная длина любого имени в VBScript (переменные, константы, процедуры) - 255 символов;
- начало нового оператора - перевод на новую строку (точка с запятой, как в C++, Java, JavaScript для этого не используется);
- ограничений на максимальную длину строки нет. Несколько операторов в одной строке разделяются двоеточиями (:).

Прежде чем начинать знакомиться с VBScript, сравним две одинаковые, простые программы на Java Script и VBScript. Суть программ: вводим текст в верхнее поле, нажимаем кнопку "Скопировать" и текст копируется в нижнее поле.

Java Script


```
<FORM NAME=forma>  
<INPUT TYPE="text" NAME="text1"><BR>  
<INPUT TYPE="text" NAME="text2"><BR>  
<INPUT TYPE="button" NAME="knopka"  
VALUE="Скопировать" OnClick=copyfun () >  
</FORM>
```

```
<SCRIPT LANGUAGE="JavaScript">  
function copyfun () {  
    var copytext;  
    copytext=document.forma.text1.value;  
    document.forma.text2.value=copytext;  
}  
</SCRIPT>
```

VB Script


```
<FORM NAME=forma>  
<INPUT TYPE="text" NAME="text1"><BR>  
<INPUT TYPE="text" NAME="text2"><BR>  
<INPUT TYPE="button" NAME="knopka"  
VALUE="Скопировать" OnClick=copyfun () >  
</FORM>
```

```
<SCRIPT LANGUAGE="VBScript">  
sub copyfun  
    dim copytext  
    copytext=document.forma.text1.value  
    document.forma.text2.value=copytext  
end sub  
</SCRIPT>
```

Как видно, различий практически нет. Java Script поддерживает только функции, VBScript и функции и процедуры (подпрограммы). Так же следует еще раз отметить, что VBScript в отличие от Java Script не чувствителен к регистру символов, т.е. для него что соруfun(), что СоPYfuN() – одно и то же.

Используя VBScript можно проверять правильность заполнения форм, автоматически генерировать HTML-страницы и многое другое. VBScript интегрирован с броузерами WWW и приспособлен к работе с управляющими элементами ActiveX и другими объектами, встраиваемыми в активные HTML-документы.

Скрипты - это текстовые файлы, поэтому в принципе можно обойтись любым текстовым редактором, вроде Notepad. Если запомнить текстовый файл со скриптом с расширением htm или html, то при его запуске он будет автоматически обрабатываться броузером.

Задание 1.

Заполните два файла текстами представленных выше программ и убедитесь в их идентичной работе.

2. Пример программы на VBScript

Поставим задачу, написать программу, которая бы случайным образом загадывала число от 1 до 100, и предлагала Вам его угадать. При неправильном предположении, программа должна выводить сообщение о том, больше загаданное число или меньше. Ведется подсчет попыток. В случае победы выводится поздравление.

Код типичной страницы с VBScript программой будет выглядеть примерно так:

```
<html>
<head>
<title>Угадай число</title>
</head>
<body bgcolor="white" text="black">
```

... HTML код формы взаимодействия с VBScript программой ...

```
<script language="VBscript">
```

... VBScript программа ...

```
</script>
</body>
</html>
```

Рассмотрим особенности реализации двух «вставок»: HTML кода формы взаимодействия с VBScript программой и самой программы.

HTML код формы взаимодействия с VB программой будет иметь вид:

```
<FORM NAME=ugadai>
<INPUT TYPE="button" NAME="begin" VALUE="Загадать число">
```

```

<BR><BR>
Ваш вариант:<BR>
<INPUT TYPE="text" NAME="chislo">
<BR>
<INPUT TYPE="button" NAME="but" VALUE="Угадал?">
</FORM>

```

1. Броузер интерпретирует представленный код в виде, представленном на рисунке

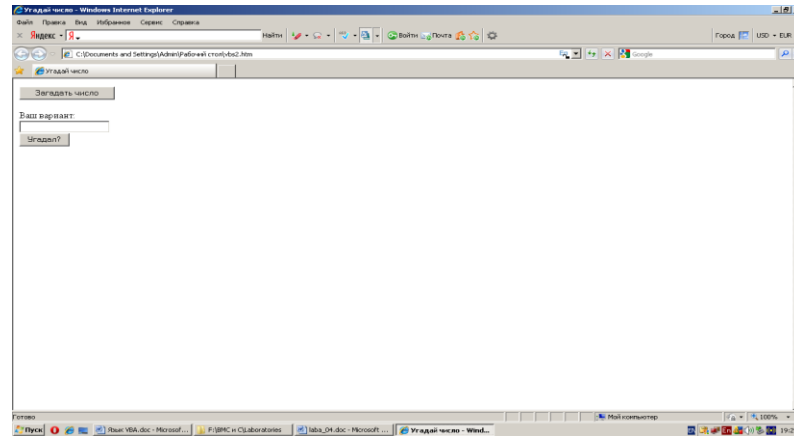


Рис.1. Интерпретация HTML кода формы взаимодействия с VBScript программой

VBScript программа представлена ниже:

```

<!--***** угадай число *****-->
<script language="vbscript"><!--
dim a          ' определяем переменные
dim p
dim v
sub begin_onclick ' процедура срабатывающая при нажатии (событие onclick)
                ' кнопки с именем begin
    randomize   ' включение генератора случайных чисел
    a=int(rnd(1)*100+1) ' присваивание переменной (a) случайного значения
    p = 1       ' обнуление счетчика
    alert "число загадано" ' вывод сообщения в отдельном окне
end sub

sub but_onclick ' процедура срабатывающая при нажатии (событие onclick)
                ' кнопки с именем but

    v = document.ugadai.chislo.value

' переменной (v) присваивается значение
' содержащееся в поле ввода формы
' путь: документ (просто эта html страница).форма с именем ugadai.
' объект- поле для ввода с именем chislo. свойство value

v=cint(v) ' поскольку поле для ввода содержит текстовую
          ' информацию, то переменную (v) надо преобразовать
          ' в целочисленный тип (функция cint)

if a > v then
    alert "загаданное число больше, попробуйте еще"
    p = p + 1
end if

if a < v then

```

```

    alert "загаданное число меньше, попробуйте еще"
    p = p + 1
end if

if a = v then
    document.write"<center>Победа за "&p&" ходов.</center>"
end if
        ' оператор document.write выводит содержащуюся
        ' в кавычках информацию в новое окно браузера
        ' как html код.

end sub
--></script>

```

Если внимательно изучить приведенный код, то думается многое станет ясным. Обсудим некоторые вопросы, которые могут возникнуть по ходу осмысления программы:

1. Что бы произошло, если бы мы в начале скрипта не объявили переменные? Тогда бы значение переменной "a" заданной в процедуре "sub begin_onclick" не было "видно" в других процедурах (например "sub but_onclick"). Это упущение приведет или к ошибке, или к тому, что значению "a" не будет ничего присвоено. Можно также воспользоваться оператором "public a" вместо "dim a" тогда значение "a" станет видимым во всех открытых на данный момент документах, это актуально при фреймовой структуре.

Во многих случаях переменные можно и не объявлять, если тело программы находится в одной процедуре.

2. Что такое эти процедуры? Процедура это общее название функций или подпрограмм. Имеет синтаксис:

```
sub name_событие ..... end sub.
```

name - это имя процедуры, например мы написали, что кнопка "загадать число" называется "chislo", значит подпрограмма sub chislo_onclick выполнится при нажатии на эту кнопку. Событие - то, что может "случиться" в окне браузера. Например "onclick" означает, что процедура выполнится, при нажатии на кнопку, предварительно заданную в форме (как в нашем случае). Существует несколько видов событий. Из самых распространенных можно отметить: window_onload - запускается при полной загрузке документа:

```

<Script language="VBscript"><!--
Sub window_onload
Alert "Добро пожаловать на мою домашнюю страницу!"
end sub
--></Script>

```

При загрузке документа появится окошко с данной надписью и кнопкой ОК

Событие onmouseover - возникает при наведении указателя мыши на гиперссылку.

[Наведите на эту ссылку указатель мыши](#)

Пример:

```

<a href="http://wanderfly.da.ru/" name="link">
Наведите на эту ссылку указатель мыши</a>

```

```
<script language="vbscript"><!--
sub link_onmouseover
alert "ссылка"
end sub
--></script>
```

Разумеется, данный пример Вы можете использовать только как шутку на ссылку "xxx", потому что нажать на нее невозможно:) Но если поменять alert "ссылка" на status="Моя домашняя страница", то это сообщение появится в строке статуса.

Задание 2.

Проверьте, как работают два последних скрипта.

3. Операторы и функции VBScript

Каков полный синтаксис операторов указывающих как "добраться" до данных в поле ввода?

x = top.имя_фрейма.document.имя_формы.имя_объекта_text_box.value

Таким образом, можно добраться до любых данных в окне браузера, даже если они расположены в другом фрейме.

Приведем основные операторы и функции VBscript:

| СИНТАКСИС | ПРИМЕР | ОПИСАНИЕ |
|-----------|--------|----------|
|-----------|--------|----------|

Операторы работы с массивами.

| | | |
|------------------------------|--------------------|-----------------------------|
| Dim имя_массива (индексы) | dim a(10),b(20,30) | Объявление массива |
| Erase имя_массива | erase a | Очистка содержимого массива |

Операторы присваивания.

| | | |
|-------------------------|----------------------|--|
| переменная = значение | a=137 a=b c="привет" | Присваивание |
| Set переменная = объект | set a=document.form1 | Присваивание переменной значения ссылки на объект. |

Комментарии.

| | | |
|-----------------|-----------|--|
| ' комментарии | ' текст | |
| Rem комментарии | rem текст | |

Константы Boolean.

| | | |
|--------------------|---------------------|------------------------------------|
| переменная = False | if a=false then ... | Значение типа boolean равное 0 |
| переменная = True | if a=true then ... | Значение типа boolean равное -1 |

Функции.

| | | |
|-----------------|-------------------------------|------------------------------------|
| Abs (число) | a = abs(-1.34) ... a=1.34 | Абсолютное значение числа |
| Asc (символ) | a = asc("s") ... a=115 | ANSI код символа |
| Chr (число) | a = chr(115) ... a="s" | Символ соответствующий коду |
| Fix (число) | a = fix(-1.5) ... a=-1 | Целое число, округление |
| Int (число) | a = int(-1.5) ... a=-2 | Целое число, округление |
| Hex (число) | a = hex(543) ... a="21F" | Шестнадцатеричное представление |
| Oct (число) | a = oct(543) ... a="1037" | Восьмеричное представление |
| Sgn (число) | a = sgn(-1.3) ... a=-1 | Знак числа |
| Date | a = date ... a= 06.02.2009 | Текущая дата |
| Day(выражение) | a = day(date) ... a= 6 | День |
| ' комментарии | ' текст | |
| Rem комментарии | rem текст | |

Константы Boolean.

| | | |
|--------------------|---------------------|------------------------------------|
| переменная = False | if a=false then ... | Значение типа boolean равное 0 |
| переменная = True | if a=true then ... | Значение типа boolean равное -1 |

Функции.

| | | |
|-------------|---------------------------|---------------------|
| Abs (число) | a = abs(-1.34) ... a=1.34 | Абсолютное значение |
|-------------|---------------------------|---------------------|

| | | |
|---|--|---|
| | | числа |
| Asc (символ) | $a = \text{asc}("s") \dots a=115$ | ANSI код символа |
| Chr (число) | $a = \text{chr}(115) \dots a="s"$ | Символ соответствующий коду |
| Fix (число) | $a = \text{fix}(-1.5) \dots a=-1$ | Целое число, округление |
| Int (число) | $a = \text{int}(-1.5) \dots a=-2$ | Целое число, округление |
| Hex (число) | $a = \text{hex}(543) \dots a="21F"$ | Шестнадцатеричное представление |
| Oct (число) | $a = \text{oct}(543) \dots a="1037"$ | Восьмеричное представление |
| Sgn (число) | $a = \text{sgn}(-1.3) \dots a=-1$ | Знак числа |
| Date | $a = \text{date} \dots a=$ <input type="text" value="06.02.2009"/> | Текущая дата |
| Day(выражение) | $a = \text{day}(\text{date}) \dots a=$ <input type="text" value="6"/> | День |
| Time | $a = \text{time} \dots a=$ Начало формы <input type="text" value="15:35:29"/> Конец формы | Время |
| Timer | $a = \text{timer} \dots a=$ Начало формы <input type="text" value="56129,65"/> Конец формы | Количество секунд истекших после полуночи |
| Now | $a = \text{now} \dots a=$ Начало формы <input type="text" value="06.02.2009 15"/> Конец формы | Дата и время |
| Hour(выражение), Minute(выражение), Month(выражение), Second(выражение), Year(выражение), Weekday(выражение) | $a = \text{hour}(\text{now}) \dots$ и т.д. | Час Минута Месяц Секунда Год День недели (вск=1) |
| Atn (число) | $a = \text{atn}(0.37)$ | Арктангенс |
| Tan (число) | $a = \text{tan}(0.37)$ | Тангенс |

| | | |
|-------------|---------------|---|
| Cos (число) | a = cos(0.37) | Косинус |
| Sin (число) | a = sin(0.37) | Синус |
| Exp (число) | a = exp(0.37) | Экспонента |
| Log (число) | a = log(0.37) | Логарифм |
| Sqr (число) | a = sqr(0.37) | Квадратный корень |
| Rnd (число) | a = rnd(1) | Случайное число 0..1 |
| Randomize | Randomize | Установка генератора случайных чисел в исходное состояние |

Операторы используемые в выражениях

| |
|---|
| Привожу просто список, надеюсь все и так ясно: + And / = Eqv ^ > <= Imp <> \ < => Mod * - Or & Xor |
|---|

Текстовые Функции

| | | |
|---|-------------------------------------|--|
| Instr (начало, строка, искомая подстрока) | If Instr(1,mail,@) then ... | Возвращает номер символа в строке с которого начинается исходная подстрока |
| Lcase (строка) | a=Lcase("ПРИВЕТ") ... a="привет" | Преобразование символов строки в строчные буквы |
| Ucase (строка) | a=Ucase("ПРИВЕТ") ... a="ПРИВЕТ" | Преобразование символов строки в заглавные буквы |
| Left (строка,N символов) | a=Left("Привет",3) ... a="При" | Левая часть строки длиной N символов |
| Right (строка,N символов) | a=Right("Привет",3) ... a="вет" | Правая часть строки длиной N символов |
| Len (строка) | a=Len("Привет") ... a=6 | Длина строки |
| Ltrim (строка) | a=Ltrim(" Привет ") a="Привет " | Удаляет начальные пробелы |
| Rtrim (строка) | a=Rtrim(" Привет ") a="Привет" | Удаляет конечные пробелы |
| Trim (строка) | a=Trim(" Привет ") a="Привет" | Удаляет начальные и конечные пробелы |
| Mid (строка, начало, N символов) | a=Mid("Привет",3,2) a="ве" | Часть строки с позиции "начало" и длиной N |

| | | |
|--------------------------------|------------------------------------|--|
| | | символов |
| Left(строка,N символов) | a=Left("Привет",3) a="При" | Левая часть строки длинной N символов |
| Space (N) | a=Space(5) | Строка из N пробелов |
| String (N символов, символ) | a=String(5,"A") a="AAAAA" | Строка из N символов |
| Ltrim (строка) | a=Ltrim(" Привет ") a="Привет " | Удаляет начальные пробелы |

Ветвление и циклы

| | | |
|---|---|--|
| Call имя_событие | Call but_onclick | Вызов подпрограммы |
| Do ... Loop Until условие | Do a=a+1 Loop Until a>=5 | Выполняет код хотя бы один раз, пока условие не будет истинным |
| Do ... Loop While условие | Do a=a+1 Loop While a<5 | Выполняет код хотя бы один раз, пока условие справедливо |
| Do Until условие ... Loop | Do Until a>=5 a=a+1 Loop | Повторяет выполнение кода, пока условие не станет истинным |
| Do While условие ... Loop | Do While a<5 a=a+1 Loop | Повторяет выполнение кода, пока условие справедливо |
| For счетчик=начало To конец Step шаг ... next | For i=1 to 5 a=a+1 next | Цикл (заметьте: пишется "next" а не "next i") |
| IF условие Then действие1 Else действие2 End if | IF a>1 Then alert"больше" | Выполнение кода при выполнении условия |
| Select Case X Case значение1 действие Case значение2 действие2 ... End select | Select Case X Case 10 alert"x=10" Case 20 alert"x=20" End select | Выборочное выполнение кода при совпадении значения |
| While условие действие Wend | While a<5 a=a+1 Wend | Выполнение кода пока условие истинно |

Объявление функций и процедур

| | | |
|--|--|-----------------------|
| Dim переменная | Dim a | Объявление переменной |
| Sub имя_событие код End sub | Sub but_onclick Alert x End sub | Подпрограмма |
| Function имя(параметры) код End function | Function(x,y) x=x*y End function | Функция |
| Exit ... | Exit Do (For, Function, Sub) | Досрочный выход |

4. Вызов сценариев в VBScript

Веб страница с активным содержимым, всегда делится как бы на две части: Первая, это html код самой страницы содержащий необходимые элементы управления (ссылки, кнопки, формы для ввода информации и т.д.), и Вторая, сценарии на скрипте, которые начинают работать при различных событиях происходящих с элементами управления. Это можно назвать главной отличительной особенностью программ для веб страниц, от обычных исполняемых приложений.

Рассмотрим основные приемы вызова сценариев, или процедур.

Сценарий вызываемый неявно:

```
<html>
<head>
  <script language="VBScript">
    Sub but_onclick
      Alert "Здравствуй мир!"
    End sub
  </script>
</head>

<body>
  <form>
    <input type="button" name="but">
  </form>
</body>
</html>
```

Такой способ самый простой. Указываем имя оператором name="имя" в теге характеризующим элемент управления, это же "имя" пишем в вызываемой процедуре и через знак подчеркивания пишем событие, которое должно произойти для исполнения сценария.

Вызов сценария определяемый в элементе управления:

```
<html>

<head>
  <script language="VBScript">
    Sub scenariy
```

```

        Alert "Здравствуй мир!"
    End sub
</script>
</head>

<body>
    <form>
        <input type="button" name="but" _
            onclick="scenariy" language="VBScript">
    </form>
</body>
</html>

```

Такой способ используется в Java Script. Среди его преимуществ, перед первым методом можно отметить то, что один и тот же сценарий можно вызывать из разных элементов управления.

Автоматический вызов сценария:

```

<html>
<body>
    <script language="VBScript">
        Alert "Здравствуй мир!"
    </script>
</body>
</html>

```

Код программы в этом случае выполнится сразу после загрузки документа.

Вызов из другой процедуры:

```

<html>

<head>
    <script language="VBScript">
        Sub but_onclick
            Call butalert("Здравствуй мир!")
        End Sub

        Sub butalert(a)
            Alert a
        End Sub
    </script>
</head>

<body>
    <form>
        <input type="button" name="but" _
            onclick="scenariy" language="VBScript">
    </form>
</body>
</html>

```

Задание 3.

Написать программу, которая бы случайным образом загадывала число от 1 до 1000, и предлагала Вам его угадать. При неправильном предположении, программа должна выводить сообщение о том, больше

загаданное число или меньше. Ведется подсчет попыток. В случае победы выводится поздравление.

Запустить программу в браузере и продемонстрировать ее работу.

Литература

1. Конспект лекций.