

VISUAL BASIC SCRIPT

1	NOZIONI GENERALI.....	2
1.1	Tipi di dati, variabili e operatori	2
1.2	Oparatori	4
2	CONTROLLO DEL FLUSSO.....	5
2.1	Cicli	5
2.2	Frase condizionali	5
2.3	Gestione degli errori	6
3	ELEMENTI DEL LINGUAGGIO	7
3.1	Commenti.....	7
3.2	Array	7
3.3	PROCEDURE E FUNZIONI	7
3.4	Assegnamento	8
3.5	Gestione delle stringhe	8
3.6	Funzioni varie	8
3.7	Input e Output	9
4	PROGRAMMAZIONE DEGLI EVENTI	11
4.1	Oggetti fondamentali	11
4.2	Implementazione di un evento in una pagina HTML.....	12
5	ESEMPLI.....	13
5.1	1 + 1 = 2.....	13

1 NOZIONI GENERALI

Il Visual Basic Script è un linguaggio capace di gestire in maniera puntuale tutti gli elementi inclusi in una pagina Web. La sua versatilità e facilità di utilizzo lo hanno reso celebre, anche se non viene supportato da alcuni importanti browser. Per Netscape, comunque, è disponibile un plug-in presso <http://www.nscompasslabs.com/>.

Un uso senza dubbio importante del VBS è quello all'interno di form HTML per la convalida dei dati. Così facendo non si appesantisce il lavoro del server che, altrimenti, tramite script CGI avrebbe dovuto assolvere anche a questa funzione. Contrariamente ad altri linguaggi di script il Visual Basic Script non distingue tra maiuscole e minuscole.

Si procederà con l'analisi delle caratteristiche del linguaggio.

1.1 Tipi di dati, variabili e operatori

1.1.1 Tipi e dichiarazioni

VBScript è un linguaggio debolmente tipizzato; in particolare possiede un unico tipo: Variant. Ciò non toglie che sia possibile utilizzare i suoi sottotipi che assicurano la compatibilità e la correttezza dei dati.

Sottotipo	Descrizione
Boolean	Può assumere solo i due classici valori "true" e "False"
Byte	E' un intero senza segno a 8 bit.
Currency	E' ottimizzato per gestire in modo preciso i valori valutari. Può assumere valori compresi tra -922.337.203.685.477,5808 a 922.337.203.685.477,5807
Date	E' un numero in virgola mobile a 64 bit che può rappresentare una data da 1/1/100 al 12/31/9999. Ogni numero intero rappresenta un giorno, i numeri decimali l'ora.
Double	E' un numero a virgola mobile a precisione doppia a 64 bit. L'intervallo di validità va da -1,79769313486232°308 a -4,94065645841247°-324 per i numeri negativi e da -1,79769313486232°308 a 4,94065645841247°-324 per i positivi
Integer	E' un valore intero con segno di un singolo byte che rappresenta un intervallo da -32.768 a 32.767.
Long	E' un valore intero con segno di due byte il cui intervallo numerico va da -2.147.483.648 a 2.147.483.647.
Object	E' un riferimento del puntatore di indirizzo a 32 bit a un'istanza di un oggetto OLE. La sintassi è la seguente: <code>Set NomeOggetto = New TipoOggetto</code>
Single	E' un numero a virgola mobile a precisione singola a 32 bit e rappresenta un intervallo che va da -3,402823°38 a -1,401298°-45 per i numeri negativi, mentre va da 3,402823°38 a 1,401298°-45 per i positivi.
String	E' una serie continua di caratteri a lunghezza variabile.

La definizione di una variabile può essere implicita (la variabile viene definita al suo primo utilizzo) o esplicita tramite l'istruzione "Dim". E' buona norma eliminare la possibilità di definire implicitamente le variabili inserendo l'istruzione "Option Explicit" all'inizio dello script.

Per controllare il sottotipo di una variabile è sufficiente richiamare la funzione "VarType()" passandogli come parametro la variabile stessa. Il valore restituito ne indicherà univocamente il

VISUAL BASIC SCRIPT

sottotipo. Di seguito è riportata la tabella dei valori restituibili dalla funzione e le costanti utilizzabili in VBS

Costante	Valore
vbEmpty	0
vbNull	1
vbInteger	2
vbLong	3
vbSingle	4
vbDouble	5
vbCurrency	6
vbDate	7
vbString	8
vbObject	9
vbError	10
vbBoolean	11
vbVariant	12
vbDataObject	13
vbByte	17
vbArray	8192

Per dichiarare esplicitamente una variabile è sufficiente (come accennato) utilizzare l'istruzione "Dim" come esemplificato di seguito.

```
| Dim NomeVariabile1, NomeVariabile2, NomeVariabile3
```

1.1.2 Visibilità delle variabili

Una variabile in VBS ha tre possibili ambiti: pagina, script e procedura.

Le variabili a livello di script vengono dichiarate all'interno dello script e sono disponibili per tutte le procedure dello script stesso. La dichiarazione avviene come nell'esempio precedente o sostituendo l'istruzione "Dim" con l'istruzione "private", come nell'esempio qui di seguito.

```
| Dim NomeVariabile1, NomeVariabile2, NomeVariabile3  
| private NomeVariabile4, NomeVariabile5
```

Le variabili all'interno di procedura vengono dichiarate all'interno delle procedure e rimangono visibili solo nella procedura in cui sono state dichiarate.

Le variabili a livello di pagina vengono dichiarate in uno script utilizzando l'istruzione "global" come mostrato di seguito.

```
| global NomeVariabile4, NomeVariabile5
```

Il VBS non supporta tipi strutturati e nemmeno costanti.

1.2 Operatori

A seconda del sottotipo del dato possono essere applicati operatori differenti. Laddove possibile il VBS si occupa, tra l'altro, di convertire il sottotipo in quello più compatibile all'operatore applicato.

1.2.1 Operatori aritmetici

Operatore	Definizione
-	Negazione unaria
*	Moltiplicazione
/	Divisione
\	Divisione intera
Mod	Modulo aritmetico
+	Addizione
-	Sottrazione
^	Esponente

1.2.2 Operatori di stringa

Operatore	Definizione
&	Concatenamento
+	Concatenamento tra stringhe

1.2.3 Operatori di confronto

Operatore	Definizione
=	Uguaglianza
<>	Ineguaglianza
<	Minore di
>	Maggiore di
<=	Minore o uguale a
>=	Maggiore o uguale a
Is	Equivalenza dell'oggetto

1.2.4 Operatori logici

A	B	Not A	A And B	A Or B	A Xor B	A Eqv B	A Imp B
V	V	F	V	V	F	V	V
V	F	F	F	V	V	F	F
F	V	V	F	V	V	F	V
F	F	V	F	F	F	V	V

2 CONTROLLO DEL FLUSSO

2.1 Cicli

2.1.1 Do... Loop

A condizione iniziale

```
DO [WHILE|UNTIL condizione]
  [istruzioni]
  [EXIT DO]
LOOP
```

A condizione finale

```
DO
  [istruzioni]
  [EXIT DO]
LOOP [WHILE|UNTIL condizione]
```

2.1.2 For... Next

```
FOR contatore = inizio TO fine STEP passo
  [istruzioni]
  [EXIT FOR]
NEXT
```

2.1.3 While... Wend

```
WHILE condizione
  [istruzioni]
WEND
```

2.2 Frasi condizionali

2.2.1 If... Then... Else

A frase singola

```
IF condizione THEN istruzione
```

Composta

```
IF condizione THEN
  [istruzioni]
[ELSEIF condizione THEN]
  [istruzioni]
[ELSE]
  [istruzioni]
END IF
```

2.2.2 Struttura Select Case

```

SELECT CASE espressione
  [CASE valore
    [istruzioni]]
  [CASE valore1, valore2, ...
    [istruzioni]]
  [CASE valoreinizio TO valorefine
    [istruzioni]]
  [CASE ELSE
    [istruzioni]]
END SELECT

```

2.3 Gestione degli errori

2.3.1 Istruzione On Error

L'errore in VBS viene gestito unicamente tramite l'istruzione

```
On Error Resume Next
```

ed il conseguente controllo del valore dell'oggetto ERR. Il programma con controllo dell'errore abilitato sarà del tipo:

```

On Error Resume Next
[istruzione]
If Err then
  [istruzioni]
Else
  [istruzioni]
End If

```

2.3.2 Oggetto ERR

L'oggetto ERR viene istanziato quando si verifica un errore di qualsiasi tipo. Si può accedere alle sue proprietà e ai suoi metodi per controllare il flusso di controllo del programma.

Proprietà

Description	Stringa che contiene una descrizione dell'errore
HelpContext	Intero lungo che punta al contesto in un file di guida
HelpFile	Stringa che contiene il percorso al relativo file di guida
Number	ntero lungo che indica il codice dell'errore
Source	Il nome dell'oggetto che provoca l'errore

Metodi

Raise	Provoca l'esecuzione di un errore specificato. La sintassi di chiamata è la seguente: ERR.Raise(numero[, sorgente, descrizione, helpcontext, helpfile])
Clear	Elimina l'oggetto ERR

3 ELEMENTI DEL LINGUAGGIO

3.1 Commenti

E' possibile inserire commenti in due modi differenti.

```
'questo è un commento VBS
REM anche questo è un commento VBS
REM in ogni caso l'istruzione deve essere ripetuta all'inizio di
' ogni riga di commento
```

3.2 Array

Definizione di un array statico

```
| DIM nomearray(dimensione1[, dimensione2[, ...[, dimensioneN]])
```

Definizione di un array dinamico

```
| DIM nomearray()
```

Operazioni che possono essere compiute su un array dinamico

```
| DIM nomearray()           `dimensionamento e creazione
| REDIM nomearray(dimensione) `ridimensionamento di un array
| REDIM PRESERVE nomearray(nuovadimensione) `mantiene i dati precedenti
| ERASE nomearray           `eliminazione dell'array
```

3.3 PROCEDURE E FUNZIONI

Dichiarazione di una procedura

```
| SUB nomeprocedura ([arg1[, arg2[, ...[, argN]])]
|   [istruzioni]
| END SUB
```

La chiamata ad una procedura può avvenire in due modi equivalenti

```
| nomeprocedura arg1, arg2, ..., argN
| CALL nomeprocedura ([arg1[, arg2[, ...[, argN]])]
```

Dichiarazione di una funzione

```
| FUNCTION nomefunzione (arg1, arg2, ..., argN)
|   [istruzioni]
|   [RETURN valorerestituito]
| END FUNCTION
```

Il valore di una funzione si estrapola come segue

```
| variabile = nomefunzione(arg1, arg2, ..., argN)
```

3.4 Assegnamento

L'operazione di assegnamento varia a seconda che si tratti di una variabile o un oggetto

```
[LET] nomevariabile = valore
SET nomeoggetto = altrooggetto
```

3.5 Gestione delle stringhe

E' possibile utilizzare molte funzioni per le elaborazioni delle stringhe presenti già in VB e VBA. Successivamente ne viene presentata la lista.

Asc	InstrB	Right	Rtrim
AscB	Len	RightB	Trim
AscW	LenB	Mid	CStr
Chr	Lcase	MidB	StrComp
ChrB	Ucase	Space	
ChrW	Left	String	
Instr	LeftB	Ltrim	

I postfissi "B" e "W" stanno per "Byte" e "Word"

3.6 Funzioni varie

3.6.1 Funzioni di verifica del sottotipo

Queste funzioni (esclusa VarType già discussa) restituiscono sempre un valore booleano. "True" se la variabile passatagli come argomento è del tipo indicato dalla funzione, "false" altrimenti.

IsArray	IsEmpty	IsNumeric	VarType
IsDate	IsNull	IsObject	

3.6.2 Funzioni di conversione

Abs	CDate	CStr	Int
Asc	CDbl	DateSerial	Oct
Chr	CInt	DateValue	Sgn
Cbool	CLong	Fix	TimeSerial
CByte	CSng	Hex	TimeValue

3.6.3 Funzioni matematiche

Atn	Tan	Sqr
Cos	Exp	Randomize
Sin	Log	Rnd

3.6.4 Data e ora

Now	DateValue	Year	Hour
Cdate	Day	Time	Minute
Date	Month	TimeSerial	Second
DateSerial	WeekDay	TimeValue	

3.6.5 Elaborazione delle stringhe

Left	Right	Mid	Space
String	InStr	Replace	Split
Join			

In particolare:

```
Replace(Stringa, Trova, Sostituisci, Inizio, Conta, TipoConfronto)
Split(Espressione, Separatore, Conta, TipoConfronto)
Join(ArrayStringa, Separatore)
```

3.7 Input e Output

3.7.1 Funzione MsgBox

```
[valore =] MsgBox(prompt, pulsantieicone, titolo, fileguida, contestoguida)
```

Valori possibili di “pulsanti” (gruppo A)

Valore	Pulsante	Gruppo
0	Ok	A
1	Ok, Annulla	A
2	Termina, Riprova, Ignora	A
3	Si, No, Annulla	A
4	Si, No	A
5	Riprova, Annulla	A
16	"X"	B
32	"?"	B
48	"!"	B
64	"i"	B
0	Primo	C
256	Secondo	C
512	Terzo	C
768	Quarto	C
0	Applicazione	D
4096	Sistema	D

A “pulsanti” è possibile assegnare qualunque combinazione del tipo A + B + C + D

Valori restituiti da “MsgBox”:

Valore	Pulsante premuto
1	Ok
2	Annulla
3	Termina
4	Riprova
5	Ignora

6	Si
7	No

3.7.2 Funzione InputBox

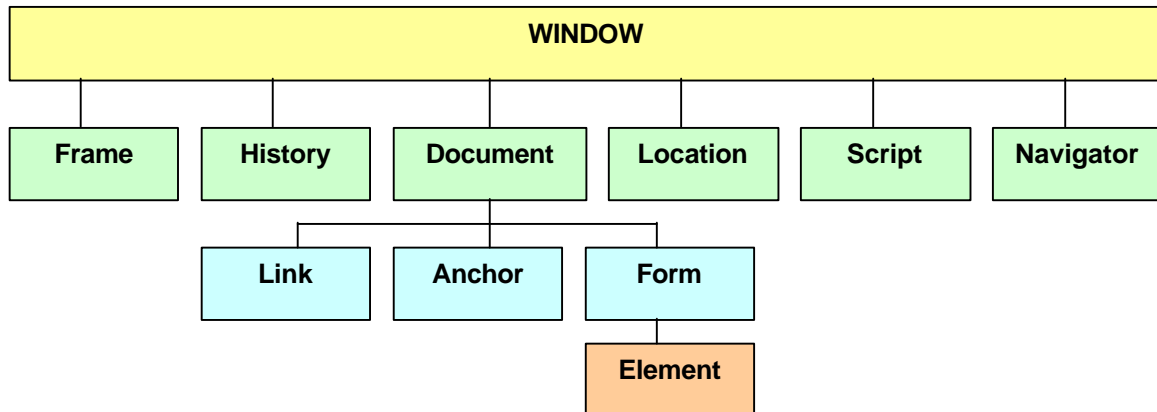
La funzione InputBox ha un uso più semplice della precedente.

```
| Variabile = InputBox(prompt, titolo, default, xpos, ypos, fhelp, chelp)
```

4 PROGRAMMAZIONE DEGLI EVENTI

4.1 Oggetti fondamentali

Gerarchia degli oggetti in Internet Explorer



4.1.1 Oggetto Window

Proprietà

Frames	Location	Name	Parent
--------	----------	------	--------

Metodi

Open(file,window)	Prompt(esp,def)	Close()	Navigate(URL)
-------------------	-----------------	---------	---------------

Eventi

OnLoad	OnUnLoad
--------	----------

4.1.2 Oggetto Document

Proprietà

BgColor	FgColor	Referrer	LastModified
---------	---------	----------	--------------

Metodi

Open()	Write(str)	Close()
--------	------------	---------

4.1.3 Oggetto Frame

E' un array indicizzato del numero di frame su una pagina.

4.1.4 Oggetto History

Proprietà

Length

Metodi

Go(n)

Forward(n)

Back(n)

4.1.5 Oggetto Location

Proprietà

Href

Protocol

Host

Hostname

4.1.6 Oggetto Navigator

Proprietà

appCodeName

AppName

AppVersion

UserAgent

4.2 Implementazione di un evento in una pagina HTML

Ogni tag HTML che possieda l'attributo "name" può essere collegato agli eventi VBScript. Ad esempio:

```
<font color="#FF0000"><a name="pippo" >BENVENUTI</a></font>
...
<SCRIPT LANGUAGE="VBScript">
  sub pippo_onClick()
    msgbox "pippo"
  end sub
</SCRIPT>
```

5 ESEMPI

5.1 1 + 1 = 2

```

<HTML>
<HEAD>
<TITLE>VBScript OptionButton Events</TITLE>
</HEAD>
<SCRIPT LANGUAGE="VBScript">
  Dim OkToTest, c
  Sub Window_OnLoad      'Can't set controls before page created
    optNum.Item(0).Value = False 'Clear default selection
    txtStatus.Value = "Your answer please!"
  End Sub
  Sub CanTest
    OkToTest = True
    For c = 0 To 2 'Walk through all buttons
      If optNum.Item(c).Checked Then 'If we've found the selection
        Exit For 'Answer found, stored in c
      End If
    Next
    txtStatus.Value = "Answer " & c + 1 & " Selected" 'Add 1 because
buttons start at 0
  End Sub
  Sub cmdTest_OnClick
    If OkToTest Then
      FindAns
    Else
      txtStatus.Value = "Must Select an Answer!"
    End If
  End Sub
  Sub FindAns
    Select Case c 'Test if it's the right answer
      Case 0, 2
        txtStatus.Value = "Wrong!"
      Case 1
        txtStatus.Value = "Correct!"
    End Select
  End Sub
</SCRIPT>
<BODY>
One Plus One Equals:<P><P>
<INPUT TYPE="Radio" NAME="optNum" size=10 OnClick="CanTest">One<P>
<INPUT TYPE="Radio" NAME="optNum" size=10 OnClick="CanTest">Two<P>
<INPUT TYPE="Radio" NAME="optNum" size=10 OnClick="CanTest">Three<P><P>
<INPUT TYPE="Button" NAME="cmdTest" VALUE="Test"><P>
<INPUT TYPE=TEXT NAME="txtStatus" size=20>
</BODY>
</HTML>

```