



Departamento de Lenguajes y
Sistemas Informáticos



Universitat d'Alacant
Universidad de Alicante

Visual Basic Script

Programación en Internet
Curso 2005-2006

Programación en Internet – Curso 2005-2006

Introducción (1)

- Exclusivo productos Microsoft (estándar de programación)
- Indicado para programadores de Visual Basic
- Dónde: Internet Explorer, ASP, Office y Windows Scripting Host

2

Programación en Internet – Curso 2005-2006

Introducción (y 2)

Aplicación anfitriona	1.0	2.0	3.0	4.0	5.0	5.1	5.5	5.6
Microsoft Internet Explorer 3.0	x							
Microsoft Internet Information Server 3.0		x						
Microsoft Internet Explorer 4.0			x					
Microsoft Internet Information Server 4.0			x					
Microsoft Visual Studio 6.0				x				
Microsoft Internet Explorer 5.0					x			
Microsoft Internet Explorer 5.01						x		
Microsoft Windows 2000						x		
Microsoft Internet Explorer 5.5							x	
Microsoft Windows Millennium Edition							x	
Microsoft Internet Explorer 6.0								x
Microsoft Windows XP								x

3

Programación en Internet – Curso 2005-2006

Comentarios

- **Dos estilos:**
`Rem comentario`
`' comentario`
- **Actúan hasta el final de la línea**

4

Bloques de código

- No existe {...}
- Los bloques vienen definidos por las sentencias:

```
If a < 3 Then  
    b = 5  
End If
```

- Para ejecutar varias instrucciones:

```
a = 1: b = 5: c = a + b
```

5

Tipos de datos

- No es necesario declarar las variables:

```
Dim var1, var2, var3
```

- Si se quiere forzar la declaración:

```
Option Explicit
```

(tiene que ser la primera instrucción)

- Tipo de dato: Variant
- Subtipos: Empty, Null, Boolean, Byte, Integer, String, ...

6

Tipos de datos

- VarType (variable)

Empty = 0	Currency = 6
Null = 1	Date = 7
Boolean = 11	String = 8
Byte = 17	Object = 9
Integer = 2	Error = 10

7

Constantes

- **Palabra reservada** Const
Const unaCadena = "Algo"
Const unNumero = 49

8

Operadores

- Aritméticos
- De comparación
- Lógicos
- Concatenación de cadenas

9

Operadores aritméticos

- Exponenciación: ^
- Negación: -
- Multiplicación y división: * y /
- División de número entero: \
- Módulo (resto): Mod
- Suma y resta: + y -

10

Operadores de comparación

- Igualdad: =
- Desigualdad: <>
- Menor que: <
- Mayor que: >
- Menor o igual que: <=
- Mayor o igual que: >=
- Equivalencia de objetos: Is

11

Operadores lógicos

- Negación: Not
- Y: And
- O: Or
- O exclusiva: Xor
- Equivalencia: Eqv
- Implicación: Imp

12

Operadores de concatenación de cadenas

- Concatenación: + y &

13

Sentencias

- Condicionales
- De repetición
- De manipulación de objetos

14

Sentencias condicionales

- Selección simple:

```
If condicion Then
  instrucciones
[ElseIf condicion Then
  instrucciones]
[Else
  instrucciones]
End If
```

15

Sentencias condicionales

- Selección múltiple:

```
Select Case expresion
  Case expresion
    instrucciones
  [Case Else
    instrucciones]
End Select
```

16

Sentencias de repetición

- Repetición con contador:

```
For contador = ini To fin Step inc
  instrucciones
Next
```

17

Sentencias de repetición

- Repetición con condición final:

```
Do
  instrucciones
Loop While condicion
```

- Repetición con condición inicial:

```
Do While condicion
  instrucciones
Loop
```

18

Sentencias de repetición

- Repetición con condición final:

Do

instrucciones

Loop Until condicion

- Repetición con condición inicial:

Do Until condicion

instrucciones

Loop

19

Sentencias de repetición

- Salir de un bucle:

- Exit For

- Exit Do

20

Sentencias de manipulación de objetos

- Recorrido en una colección:

```
For Each elemento In coleccion
  instrucciones
Next
```

- Objeto por defecto:

```
With objeto
  instrucciones (.propiedad)
End With
```

21

Procedimientos y funciones

- Procedimiento:

```
Sub procedimiento (arg1, arg2, ...)
  instrucciones
End Sub
```

- Función:

```
Function funcion (arg1, arg2, ...)
  instrucciones (funcion = valor)
End Function
```

22

Procedimientos y funciones

- Interrumpir la ejecución:
 - `Exit Sub`
 - `Exit Function`

23

Tratamiento de cadenas

- `LCase(cad)`: convierte a minúsculas
- `UCase(cad)`: convierte a mayúsculas
- `Left(cad, lon)`, `Right(cad, lon)`,
`Mid(cad, inicio, lon)`: devuelve subcadena
- `Trim(cad)`, `LTrim(cad)`, `RTrim(cad)`: elimina espacios en blanco
- `StrReverse(cad)`: devuelve la inversa
- `Replace(cad, cadVieja, cadNueva)`: busca y sustituye una cadena por otra
- `Join(array, delimitador)` : une las componentes de un array
- `Split(cadena, delimitador)`: divide una cadena en trozos

24

Tratamiento de fechas

- Subtipo: Date
- Now: fecha y hora
- Date: fecha
- Time: hora
- Second(hora) , Minute(hora) , Hour(hora) : desde 0
- Day(fecha) , Month(fecha) , Year(fecha) : desde 1

25

Funciones matemáticas

- Las usuales:
 - Abs(n) : valor absoluto
 - Cos(n) : coseno
 - Hex(n) : convierte a hexadecimal
 - Int(n) , Fix(n) : redondeo
 - Log(n) : logaritmo natural
 - Sqr(n) : raíz cuadrada
 - ...

26

Gestión de errores

- `Err`: objeto intrínseco que proporciona información sobre el último error
- **Propiedades:**
 - `Description`: cadena con la descripción del error
 - `HelpContext`: identificador de un tema en el fichero de ayuda
 - `HelpFile`: fichero de ayuda
 - `Number`: código de error (0 no error)
 - `Source`: objeto o aplicación que originó el error
- **Métodos:**
 - `Clear`: limpia el error producido
 - `Raise`: genera un error en tiempo de ejecución

27

Gestión de errores

- Cualquier error en tiempo de ejecución (ejemplo: división por 0), genera un mensaje de error y detiene la ejecución del código
- `On Error Resume Next`: **activa el manejo de errores**
 - No se genera el mensaje de error
 - La ejecución continúa en la siguiente sentencia posterior a la que causó el error
 - Importante: no es global, se tiene que activar también en las funciones y procedimientos
- `On Error GoTo 0`: **desactiva el manejo de errores**

28

Objetos de Script Runtime

- Objeto `Dictionary`
- Objeto `FileSystemObject`

29

Objeto Dictionary

- Almacena información en forma de parejas (clave, elemento) – (*key*, *item*)
- Se instancia con `Scripting.Dictionary`
- Propiedades:
 - `Count`: devuelve el número de elementos
 - `Item(clave)`: devuelve o asigna un nuevo elemento
 - `Key(clave)`: modifica una clave existente
- Métodos:
 - `Add`, `Exists`, `Items`, `Keys`, `Remove`, `RemoveAll`

30

Objeto FileSystemObject

- Permite gestionar unidades, directorios y archivos
- Se instancia con `Scripting.FileSystemObject`
- Se compone de distintos objetos:
 - `FileSystemObject`: objeto principal
 - `Drive`: representa una unidad
 - `Drives`: colección de todas las unidades del sistema
 - `File`: representa un fichero
 - `Files`: colección de ficheros en un directorio
 - `Folder`: representa un directorio
 - `Folders`: colección de directorios en un directorio
 - `TextStream`: permite leer o escribir en ficheros de texto