

# Funciones del lenguaje

# Funciones del lenguaje

- Al igual que pasa con otros lenguajes, VBS incluye una librería con funciones predefinidas del lenguaje
- El usuario puede hacer uso de ellas en cualquier momento del programa
- Algunas de ellas son:

# Funciones del lenguaje

- Para la creación de arrays de forma más rápida y sencilla, VBS incluye la función
- **Array (valor1, valor2, valor3...)**
  - Devuelve un array con los elementos que se han pasado

# Funciones matemáticas

# Funciones del lenguaje

- **Abs(número)**
  - Devuelve el número sin signo
- **Fix(número)**
  - Devuelve la parte entera del número, truncando los decimales.
- **Int(número)**
  - Devuelve la parte entera del número.
  - Si es negativo devuelve el primer negativo igual o menor que encuentre

# Funciones del lenguaje

```
numero = -3.9  
numero2 = -3.2
```

```
num = fix(numero)  
num2 = fix(numero2)
```

```
inte = int(numero)  
inte2 = int(numero2)
```

```
response.write("<table border>")  
response.write("<tr bgcolor='pink'>")  
response.write ("<td>valor</td><td>fix</td><td>int</td></tr>")  
response.write ("<tr><td>"&numero&"</td><td>"&num&"</td><td>"&inte&"</td></tr>")  
response.write ("<tr><td>"&numero2&"</td><td>"&num2&"</td><td>"&inte2&"</td></tr>")  
response.write("</table>")
```

valor	fix	int
-3,9	-3	-4
-3,2	-3	-4

# Funciones del lenguaje

- **Round(numero, decimales)**
  - Redondea el número y lo devuelve con tantos decimales como se indique
- **Sqr(numero)**
  - Devuelve la raíz cuadrada del número indicado
- **Rnd()**
  - Devuelve un número aleatorio.
  - Hay que incluir **RANDOMIZE** antes de usarlo

# Funciones del lenguaje

```
dim numeros(10)

for i = 0 to 10
    numeros(i) = rnd()
next

for i= 0 to 10
    response.write (numeros(i) & "<br>")
next
```

```
dim numeros(10)

RANDOMIZE

for i = 0 to 10
    numeros(i) = rnd()
next

for i= 0 to 10
    response.write (numeros(i) & "<br>")
next
```

0,7055475  
0,533424  
0,5795186  
0,2895625  
0,301948  
0,7747401  
1,401764E-02  
0,7607236  
0,81449  
0,7090379  
4,535276E-02

0,6601221	0,6076776
0,9040753	0,1082836
0,7644399	1,118976E-02
0,8441737	0,525799
0,3099894	0,9994883
0,734991	0,4633693
0,1801401	0,6043802
0,1217313	2,088594E-02
0,1533419	0,940253
0,4268571	0,6129991
0,6887701	0,4897192



# Comprobaciones

# Funciones del lenguaje

- **Typename(variable)**
  - Devuelve el nombre del tipo de dato asociado a la variable
- **Isarray (variable)**
  - Devuelve **verdadero** o **falso** dependiendo de si la variable es o no un array
- Como esta, hay muchas más
  - Isdate
  - Isnumeric
  - Isnull
  - isempty

```

fel = "25/12/2014"
num = 98
cad = null
arra = array (1, 2, 4)

isfec = isdate(fel)
isarr = isarray(fel)
isnum = isnumeric(fel)
vacia = isnull(fel)

```

variable	fecha	array	numero	vacia
25/12/2014	Verdadero	Falso	Falso	Falso
98	Falso	Falso	Verdadero	Falso
	Falso	Falso	Falso	Verdadero
array	Falso	Verdadero	Falso	Falso

# Funciones del lenguaje

- **Formatcurrency (numero, decimales, cero\_decimal, negativos\_parentesis, agrupar)**
  - Devuelve el número pasado como una moneda, coloca el símbolo de la moneda local
  - Permite indicar:
    - Número de **decimales** que mostrar
    - Si se quiere completar con **ceros** los decimales que falten
    - Si se quiere que los **negativos** aparezcan entre paréntesis
    - Si se quiere que aparezcan **agrupados** de 3 en 3
  - -1 = sí
  - 0 = no
- **Formatnumber (numero, decimales, cero\_decimal, negativos\_parentesis, agrupar)**
  - Igual, pero sin asociar a tipo moneda

# Manejo de fechas

# Funciones del lenguaje

- **Date()**
  - No recibe nada, devuelve la fecha **actual** del sistema
- **Cdate(cadena)**
  - Recibe como parámetro una cadena que representa una fecha
  - Devuelve la fecha pasada, pero convertida al tipo de dato DATE
- **Day(fecha), month(fecha), year(fecha)**
  - Reciben una fecha y devuelven el valor correspondiente al día, el mes y el año de dicha fecha

# Funciones del lenguaje

- **Dateadd(intervalo, cantidad, fecha)**
  - Devuelve el resultado de sumar un **periodo de tiempo** a una fecha concreta
  - **Intervalo** es una cadena de texto que indica la unidad de tiempo que queremos añadir
  - **Cantidad** es la cantidad de unidades del tiempo indicado que queremos añadir

# Funciones del lenguaje

Cadena	Significado
yyyy	Año completo
q	Trimestre
m	mes
d	día
w	semana
h	hora
n	minuto
s	segundo



# Ejercicios

- Mostrar por pantalla la fecha actual, así como el resultado de sumar 2 trimestres al día de hoy

```
<%
```

```
fe = date()  
response.write("hoy es "&fe)
```

```
fe = dateadd("q", 2, fe)  
response.write("<br>dentro de 2 trimestres será "&fe)
```

```
%>
```

# Funciones del lenguaje

- **Datediff (intervalo, fecha1, fecha2)**
  - Devuelve la diferencia entre las dos fechas, expresada en el tipo de periodo que hayamos indicado en intervalo

```
fe1 = "25/12/2014"
```

```
fe2 = "13/05/2015"
```

```
dias = datediff ("d", fe1, fe2)
```

```
meses = datediff ("m", fe1, fe2)
```

```
anios = datediff ("yyyy", fe1, fe2)
```

```
trimestres = datediff ("q", fe1, fe2)
```

periodo	valor
fecha1	25/12/2014
fecha2	13/05/2015
dias	139
meses	5
años	1
trimestres	2

# Funciones del lenguaje

Función	Definición
Day(fecha)	Recibe una fecha y devuelve el número del día del mes
Month(fecha)	Recibe una fecha y devuelve el número del mes
Monthname(mes, abrevia)	Recibe un numero del 1-12 y devuelve el nombre del mes. Puede ser abreviado (true-false)
Weekday (fecha, primer_dia)	Recibe una fecha y el primer día de la semana. Devuelve número de día de la semana
Weekdayname(num_dia, abreviado, primer_dia)	Recibe un número (1-7) y devuelve el nombre del día correspondiente. Puede ser abreviado (true-false)

# Funciones del lenguaje

- **Weekday** y **weekdayname** reciben, además de una fecha, una constante indicando el día por el que debe considerarse que empieza la semana.
- La constante debe ser una de las siguientes:

Constante	día
VBSUNDAY	Domingo
VBMONDAY	Lunes
VBTUESDAY	Martes
VBWEDNESDAY	Miércoles
VBTHURSDAY	Jueves
VBFRIDAY	Viernes
VBSATURDAY	Sábado

# Ejemplo weekday y weekdayname

<8

```
hoy = date()  
num_hoy = weekday (hoy, VBMONDAY)  
nombre_hoy = weekdayname (num_hoy, true, 2)  
  
response.write (nombre_hoy)
```

8>

# Manejo de horas

# Funciones del lenguaje

- **Time()**
  - Devuelve la hora del sistema en formato **hh:mm:ss**
- **Timeserial(horas, minutos, segundos)**
  - Recibe 3 números y los convierte a formato hora.
- **Timevalue(fecha)**
  - Recibe una fecha del sistema, y extrae la parte de la hora.
- **Hour(hora), minute(hora), second(hora)**
  - Reciben una hora y devuelven el valor correspondiente a la hora, los minutos o los segundos de dicha hora

# Manejo de Cadenas



# Antes de empezar

- En VBScript las cadenas de caracteres no empiezan por el carácter 0 si no por el carácter 1.
- Para indicar el inicio de la cadena habrá que hacerlo indicando la posición 1

# Funciones de lenguaje

- **Len(cadena)**

- Recibe una cadena y devuelve el número de caracteres que tiene dicha cadena

- Funciones de **creación** de cadenas

- **String(numero, carácter)**

- Genera una cadena compuesta por el carácter tantas veces como indique el número

- **Space(numero)**

- Genera una cadena compuesta por tantos espacios en blanco como indique el número

# Funciones del lenguaje

- Funciones de modificación del contenido:
  - **Lcase(cadena)**
    - Recibe una cadena y la convierte a **minúsculas**
  - **Ucase(cadena)**
    - Recibe una cadena y la convierte a **mayúsculas**
  - **Rtrim(cadena)**
    - Recibe una cadena y la devuelve sin los espacios en blanco que tenga a la **derecha**
  - **Ltrim(cadena)**
    - Recibe una cadena y la devuelve sin los espacios en blanco que tenga a la **izquierda**
  - **Trim(cadena)**
    - Recibe una cadena y la devuelve sin los espacios en blanco que tenga a izquierda y derecha

# Funciones del lenguaje

- Búsqueda en cadenas
  - **Instr(comienzo, cadena1, cadena2)**
    - Busca la primera aparición de cadena2 en cadena1.
    - **Comienzo** indica desde qué carácter hay que empezar a buscar
    - Si se omite comienzo, empezará desde el principio
    - Devuelve la **posición** donde encuentra cadena2
  - **Instrrev (cadena1, cadena2, comienzo)**
    - Igual que la anterior, pero comienza la búsqueda por el **final**
    - El valor que devuelve es contando desde el principio

# Funciones del lenguaje

- **Strreverse (cadena)**
  - Devuelve la cadena al revés.
- **Replace(cadena1, cadena2, cambia\_por, comienzo, veces)**
  - Busca **cadena2** dentro de **cadena1** desde el **comienzo** indicado
  - La cambia por **cambia\_por**
  - Si **cadena2** estuviera más de una vez dentro de **cadena1**, la cambiará **veces** veces

# Funciones del lenguaje

- **Left (cadena, longitud)**
  - Devuelve una cadena compuesta por los caracteres que hay a la izquierda de la cadena.
  - Se tomarán **longitud** caracteres
- **Right (cadena, longitud)**
  - Devuelve una cadena compuesta por los caracteres que hay a la derecha de la cadena.
  - Se tomarán **longitud** caracteres
- **Mid (cadena, inicio, longitud)**
  - Devuelve una cadena compuesta por los caracteres que desde **inicio** hasta **longitud**.

# Funciones del lenguaje

- **Strcomp (cadena1, cadena2)**
  - Compara las dos cadenas que recibe
  - Si son iguales devuelve 0
  - Si la primera es mayor que la segunda devuelve 1
  - Si la segunda es mayor que la primera devuelve -1
  - Si alguna cadena contiene null, devuelve null
- **NOTA:** compara alfabéticamente

# Funciones del lenguaje