

# Qué es VBScript?

[Tutorial de VBScript](#)  
[Siguiente](#)

---

Microsoft Visual Basic Scripting Edition, el miembro más reciente de la familia de lenguajes de programación Visual Basic, suministra una automatización activa a una amplia variedad de entornos, incluidas las secuencias de comandos del cliente de Web en Microsoft Internet Explorer y las secuencias de comandos del servidor de Web en Microsoft Internet Information Server.

## Fácil de utilizar y de aprender

Si ya conoce Visual Basic o Visual Basic for Applications, VBScript le será muy familiar. Incluso si no conoce Visual Basic, una vez que lo haya aprendido podrá programar con toda la familia de lenguajes Visual Basic. Aunque pueda aprender acerca de VBScript en estas páginas Web, no le enseñarán cómo programar. Para comenzar con la programación, consulte los manuales *Step by Step* disponibles en Microsoft Press.

## ActiveX Scripting

VBScript se comunica con las aplicaciones de host mediante ActiveX™ Scripting. Con ActiveX Scripting, evita que los exploradores y otras aplicaciones host necesiten escribir código especial de integración para cada componente de la secuencia de comandos. ActiveX Scripting permite a un host compilar una secuencia de comandos, obtener y llamar puntos de entrada, y administrar el espacio de nombres disponible para el programador. Con ActiveX Scripting, los distribuidores de lenguajes pueden crear run times de lenguaje estándar para la secuencia de comandos. Microsoft proporcionará soporte de tiempo de ejecución para VBScript. Microsoft está trabajando con varios grupos de Internet para definir el estándar de ActiveX Scripting de modo que se puedan intercambiar los motores de automatización. ActiveX Scripting se utiliza en Microsoft® Internet Explorer y en Microsoft® Internet Information Server.

## VBScript en otras aplicaciones y exploradores

Como programador, puede licenciar su implementación de origen de VBScript sin cargo para utilizarla en sus productos. Microsoft proporcionará implementaciones binarias de VBScript para la API de Windows® de 32 bits, la API de Windows de 16 bits y la de Macintosh®. VBScript se integra con exploradores de World Wide Web. VBScript y ActiveX Scripting también se pueden utilizar como un lenguaje de automatización general en otras aplicaciones.

## Visual Basic Script

Cuando alguien construye un programa, lo escribe en texto, obteniendo lo que se llama código fuente. Para que el programa lo pueda ejecutar un usuario hay que convertirlo en binario éste paso se llama compilar. Al compilarlo obtenemos un archivo de extensión .exe que podremos ejecutar directamente. (No hablamos aquí del programa objeto ni del linkado, otros pasos intermedio, a veces necesarios).

Un SCRIPT es un programa en código fuente, en textos, que no necesita ser compilado para funcionar. La aplicación que lo lee es capaz de ejecutarlo al leer sus códigos textuales.

El Visual Basic Script, VBScript, es un lenguaje basado en Visual Basic para construir Scripts.

Más tarde lo podremos utilizar con las páginas ASP

Voy a distinguir VBScript en tres niveles:

1.- VBScript Simple, son programas con las ordenes "sencillas" de Visual Basic FOR NEXT, CLS, IF THEN ELSE, INPUTBOX, MSGBOX, DO LOOP, SELECT CASE, SUB, DATE, TIME, NOW, SQR, RND, SIN, MID, LEFT, RIGHT, LEN, CHR etcétera. Podemos realizar textualmente un programa, un archivo con estos comandos (y los que faltan) y luego guardarlo con la extensión .vbs.

Cuando pulsemos sobre el archivo, se ejecutará.

Por ejemplo creamos con el Bloc de notas un archivo de texto y escribimos:

```
nom = INPUTBOX("Escribe tu nombre")
edad = INPUTBOX("Escribe tu edad")
IF edad <18 THEN
MSGBOX(nom & " eres menor de edad")
ELSE
MSGBOX(nom & " eres mayor de edad")
END IF
```

Guárdalo como edad.vbs. Ejecútalo. Fácil ¿verdad?, pues bien quién conozca las órdenes de Visual Basic, puede hacer Scripts de éste tipo.

(Nota: Cuidado cuando guardes el archivo, no vayas a guardarlo como edad.txt.vbs)

(Esto de poner un archivo .txt.vbs es un truco que utilizan los troyanos, pero eso es otra historia en la que no entramos ahora.)

2.- VBScript con Objetos. Estos son programas como el comentado en el apartado anterior, con las órdenes del Visual Basic, pero además se insertan Objetos. Un Objeto en éste caso es: (dicho de forma poco académica) un archivo que nos da la posibilidad de tener más comandos.

Fíjate en este VBScript. Cuando lo ejecuta crea un Objeto. Este Objeto creado nos da la posibilidad de manejar archivos. En este caso vamos a crear un archivo llamado C:\amigo.doc. Esto empieza a ser peligroso pues lo mismo que puede crear un archivo, lo puede borrar de manera que puede dañar al sistema.

Así que con los Objetos podemos controlar el sistema. Por cierto, el VBScript utiliza una tecnología de Windows llamada ActiveX, que lo que trata es de aprovechar los archivos y los recursos de Windows. Para evitar que un programa de éste tipo dañe a nuestro sistema, lo mejor es anular para que no se ejecuten éstos tipos de Aplicaciones.

Como ejemplo crea este archivo, guárdalo como creador.vbs y ejecútalo.

```
Set fso = CreateObject("Scripting.FileSystemObject")
Dim fl
If (fso.FileExists("C:\amigo.doc" ) ) Then
b=MsgBox("El archivo ya existe.")
Else
Set fl = fso.CreateTextFile("C:\amigo.doc", True)
c=MsgBox("He creado el archivo amigo.doc en el disco C:")
End If
```

3.- VBScript en página HTML. Esto sería poner códigos de VBScript en páginas HTML de manera que al activarse la página se activa también el código VBScript que está insertado dentro de ella. (El VBScript en páginas web es como el JavaScript pero tal vez menos potente). Bueno, no tan menos potentes, pues muchos virus y gusanos están escrito en VBS de tal manera que pueden entrar en nuestro ordenador a través de correos o páginas web y desconfigurarnos el sistema. ¿Recuerdas el I LOVE YOU?, pues era un virus VBS que solía venir por e-mail. Hay que tener cuidado que en una página no nos inserten códigos VBS. De todos modos el Internet Explorer tiene un sistema de seguridad, que hemos de configurar para que cada vez que un código VBS quiera entrar en nuestro sistema, nos avise. En el documento 50.htm nos informa sobre como configurar el sistema de seguridad. En nuestro caso los programas VBS que vamos a realizar son inocuos, por lo cual nos es conveniente activar la entrada de ActiveX (o sea, que actúen los VBS). Por supuesto los VBS insertados en páginas web también pueden hacer uso de los Objetos.

Aquí presento un sencillo ejemplo de VBS insertado en una página web.

```
<HTML>
<HEAD> <TITLE>Adivina</TITLE> </HEAD>
<BODY>
Se trata de que el ordenador (mediante VBS) cree un número, que hemos de adivinar.<br>
<SCRIPT LANGUAGE="VBSCRIPT">
```

```
Randomize  
x=CInt(1 + Int(Rnd()*19))
```

```
Do  
t=CInt(InputBox("Introduce un número del 1 al 12"))  
If x<t Then r=MsgBox("Mi número es más pequeño.")  
If x>t Then r=MsgBox("Mi número es mayor.")  
Loop Until x=t  
MsgBox("Has acertado")  
</SCRIPT>
```

<br>

Ya veo que has acertado. Espero que te des cuenta de que trata esto de VBScript en paginas web.

</BODY>

</HTML>

Aquí puedes ver su funcionamiento, pulsa [adivina.htm](#)

---

Otra de la forma de evitar que en tu ordenador no funcionen los VBScript es quitar la Aplicación que los ejecuta, pero de esta forma te dejará el ordenador sin esa posibilidad que muchas veces es muy agradecida, para ello:

- 1) Mi PC/Panel de control.
- 2). Pulsa en: Agregar o quitar programas.
- 3). Instalación de Windows.
- 4). Marca: Accesorios.
- 5). Pulsa en: Detalles.
- 6). Busca hacia abajo: Windows Scripting Host y desmárcalo.

En fin, tu sabrás ..... no te lo recomiendo.

La aplicación que hace funcionar a los VBScript es : wscript.exe, si buscas esa aplicación y la renombas 000wscript.exe, no te funcionarán los vbs.

Esto lo puedes hacer para que no te metan gusanos en vbs.

## Agregar código de VBScript a una página HTML

[Tutorial de VBScript](#)  
[Anterior](#) | [Siguiente](#)

---

Puede usar el elemento SCRIPT para agregar código de VBScript a una página HTML.

### La etiqueta <SCRIPT>

El código de VBScript se presenta en una página HTML dentro de una pareja de etiquetas <SCRIPT>. Por ejemplo, un procedimiento que comprueba una fecha de envío aparece como:

```
<SCRIPT LANGUAGE="VBScript">
<!--
    Function PuedeEntregar(Dt)
        PuedeEntregar = (CDate(Dt) -
Now()) > 2
    End Function
-->
</SCRIPT>
```

Las etiquetas <SCRIPT> de comienzo y fin rodean el código. El atributo LANGUAGE indica el lenguaje de secuencia de comandos. Necesita especificar el lenguaje porque los exploradores pueden utilizar otros lenguajes de secuencia de comandos. Observe que la función `PuedeEntregar` aparece entre etiquetas de comentario (<!-- y -->). Esto evita que los exploradores que no entienden la etiqueta <SCRIPT> muestre el código.

Como el ejemplo es una función general —no está vinculado a ningún control de formulario en particular— puede incluirlo en la sección HEAD de la página:

```
<HTML>
<HEAD>
<TITLE>Haga su pedido</TITLE>
<SCRIPT LANGUAGE="VBScript">
<!--
    Function PuedeEntregar(Dt)
        PuedeEntregar = (CDate(Dt) -
Now()) > 2
    End Function
-->
</SCRIPT>
</HEAD>
<BODY>
...
```

Puede utilizar procedimientos en código en cualquier lugar de una página. Puede colocar procedimientos en las secciones BODY y HEAD.

Sin embargo, puede que desee colocar todos los procedimientos en la sección HEAD para mantener todo el código junto. Mantener el código en la sección HEAD asegura que todo el código se lee y decodifica antes de que se necesite debido a una llamada dentro de la sección BODY.

Una excepción destacable a esta regla es que puede desear proporcionar formularios en línea con código de secuencia de comandos para responder a eventos de objetos del formulario. Por ejemplo, puede incluir código de secuencia de comandos para responder a un clic en un botón en el formulario:

```
<HTML>
<HEAD>
<TITLE>Probar eventos de botón</TITLE>
</HEAD>
<BODY>
<FORM NAME="Formulario1">
  <INPUT TYPE="Button" NAME="Boton1"
VALUE="Clic">
  <SCRIPT FOR="Boton1" EVENT="onClick"
LANGUAGE="VBScript">
    MsgBox "¡Botón presionado!"
  </SCRIPT>
</FORM>
</BODY>
</HTML>
```

La mayoría del código aparecerá en procedimientos **Sub** o **Function** y sólo se llamará cuando el código que ha escrito provoca su ejecución. Sin embargo, puede escribir código de VBScript fuera de procedimientos, pero dentro de un bloque SCRIPT. Este código se ejecuta sólo una vez, cuando se carga la página HTML. Esto le permite inicializar los datos o cambiar dinámicamente la apariencia de su página Web cuando se carga.