

# Tutorial ASP.NET: Ejecutar código JavaScript en ASPNET

En el primer capítulo del Tutorial de desarrollo web con ASP.NET vimos una introducción y los requisitos para el desarrollo, en el segundo creamos nuestra primera aplicación web en ASP.NET y en el tercero vamos a ejecutar código JavaScript en ASPNET.



Fernando Giardina

diciembre 21 2010

Agregaremos funcionalidad JavaScript a la ya codificada en C#. Una forma sencilla de implementar JavaScript en ASP.NET es agregando un atributo al control del servidor que queremos que ejecute una función JavaScript.

Dentro del **TAG HEAD** escribimos lo siguiente.

```
1 | <!--Codigo JavaScript-->
2 | <script language="javascript" type="text/javascript">
3 |     function fnAceptar() {
4 |         alert('El Contenido del TextBox es: ' + document.get
```

```

5 |     document.getElementById("txtNombre").value = '';
6 | }
7 | </script>

```

Ahora necesitamos indicarle a **ASP.NET** que necesitamos ejecutar esa función desde el botón.

```

1 | <script runat="server">
2 | protected void Page_Load(object sender, EventArgs e)
3 | {
4 |     this.btnAceptar.Attributes.Add("OnClick", "javascrip
5 | }
6 | </script>

```

Aquí asignamos la función `fnAceptar` al evento `OnClick` del botón. El código completo quedaría de esta forma.

```

01 | <!--Directiva-->
02 | <%@ Page Language="C#" %>
03 | <!--Codigo en linea-->
04 |
05 | <script runat="server">
06 | protected void Page_Load(object sender, EventArgs e)
07 | {
08 |     this.btnAceptar.Attributes.Add("OnClick", "javascri
09 | }
10 | </script>
11 |
12 | <!--HTML para dibujar los controles en pantalla-->
13 | <html xmlns="http://www.w3.org/1999/xhtml">
14 | <head id="Head1" runat="server">
15 |     <title>Mi primera aplicacion - Maestros del Web</ti
16 |     <!--Codigo JavaScript-->
17 |     <script language="javascript" type="text/javascript
18 |     function fnAceptar() {
19 |         alert('El Contenido del TextBox es: ' + documen
20 |         document.getElementById("txtNombre").value = ''
21 |     }
22 |     </script>
23 | </head>
24 | <body>
25 | <form id="form1" runat="server">
26 |     <div>
27 |         <asp:TextBox ID="txtNombre" runat="server"></as
28 |         <asp:Button ID="btnAceptar" runat="server" Text

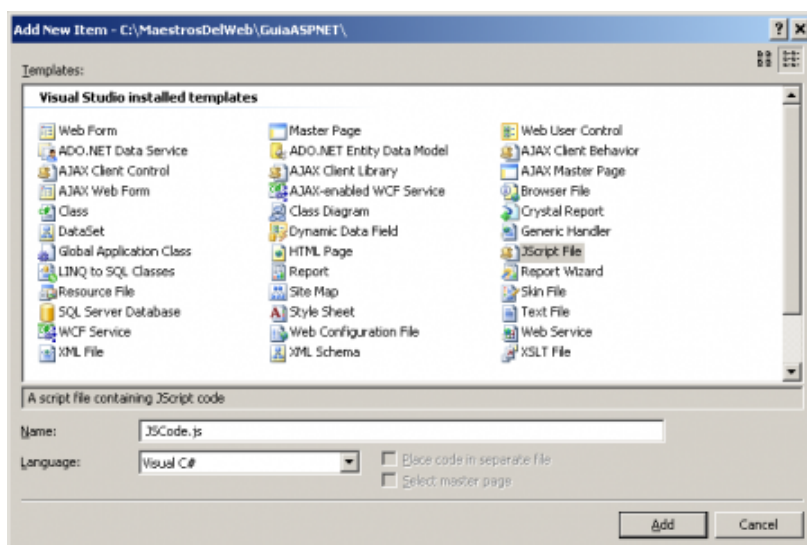
```

```

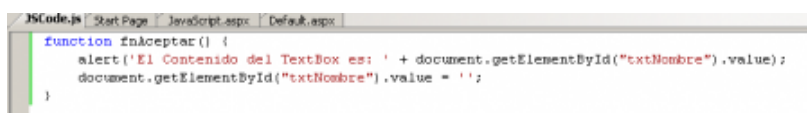
29         <br/>
30         <asp:Label ID="lblResultado" runat="server" ></
31     </div>
32 </form>
33 </body>
34 </html>

```

Hasta el momento vimos como conviven **código C#** y **JavaScript** en un mismo archivo **ASPX**. Ya es hora de separar el código en distintos archivos para mejorar el mantenimiento y lectura. Vamos a agregar un archivo JavaScript a nuestro proyecto **ASPNET**, mover el código JavaScript a ese archivo y referenciarlo desde nuestra página **aspx**.



Llamaremos a nuestro archivo **JSCode.js**. Abrimos el archivo y copiamos el código JavaScript que inicialmente escribimos en nuestra página **aspx**.



Ahora eliminamos el código JavaScript de la página **aspx** y hacemos referencia al archivo **.js** dentro del tag **HEAD**. De esta forma evitamos llenar nuestro **aspx** con código que hace difícil la lectura. También se puede separar el código C# de una forma similar al que acabamos de ver con JavaScript, pero lo veremos en unos capítulos más adelante.

## ClientScript.RegisterStartupScript

Una forma de usar código JavaScript desde una página **ASP.NET** es utilizando el método

**RegisterStartupScript** de la clase **ClientScript**, que nos permite registrar un script en tiempo de ejecución para su posterior utilización.

**Ejemplo:** Creamos la función en C#

```

01 | private void RegistrarScript()
02 | {
03 |     const string ScriptKey = "ScriptKey";
04 |     if (!ClientScript.IsStartupScriptRegistered(this.Ge
05 |     {
06 |         StringBuilder fn = new StringBuilder();
07 |         fn.Append("function fnAceptar() { ");
08 |         fn.Append("alert('El Contenido del TextBox es:
09 |         fn.Append("document.getElementById(\"txtNombre\
10 |         fn.Append("}");
11 |         ClientScript.RegisterStartupScript(this.GetType
12 |         ScriptKey, fn.ToString(), true);
13 |     }
14 | }

```

Es la misma función usada anteriormente **fnAceptar()** ahora registrándola desde ASP.NET. Luego hacemos la llamada a la función desde el **Page\_Load()**.

**RegistrarScript();**

Sin necesidad de un archivo **.js** ni escribir el código JavaScript en la sección de HTML.

Código completo:

```

01 | <!--Directiva-->
02 | <%@ Page Language="C#" %>
03 | <!--Codigo en linea-->
04 |
05 | <script runat="server">
06 | protected void Page_Load(object sender, EventArgs e)
07 | {
08 |     if (!IsPostBack)
09 |     {
10 |         this.btnAceptar.Attributes.Add("OnClick", "java
11 |         RegistrarScript();
12 |     }
13 | }
14 | private void RegistrarScript()
15 | {
16 |     const string ScriptKey = "ScriptKey";

```

```
17     if (!ClientScript.IsStartupScriptRegistered(this.Ge
18     {
19         StringBuilder fn = new StringBuilder();
20         fn.Append("function fnAceptar() { ");
21         fn.Append("alert('El Contenido del TextBox es:
22         fn.Append("document.getElementById(\"txtNombre\
23         fn.Append("}");
24         ClientScript.RegisterStartupScript(this.GetType
25         ScriptKey, fn.ToString(), true);
26     }
27 }
28 </script>
29 <!--HTML para dibujar los controles en pantalla-->
30 <html xmlns="http://www.w3.org/1999/xhtml">
31 <head id="Head1" runat="server">
32     <title>Mi primera aplicacion - Maestros del Web</ti
33 </head>
34 <body>
35 <form id="form1" runat="server">
36     <div>
37         <asp:TextBox ID="txtNombre" runat="server"></as
38         <asp:Button ID="btnAceptar" runat="server" Text
39         <br/>
40         <asp:Label ID="lblResultado" runat="server" ></
41     </div>
42 </form>
43 </body>
44
45 </html>
```

## Capítulos del Tutorial

- [Capítulo 1: Tutorial de desarrollo web con ASP.NET](#)
- [Capítulo 2: Primera aplicación web en ASP.NET](#)
- [Capítulo 3: Ejecutar código JavaScript en ASPNET](#)
- [Capítulo 4: Archivos Web.Config y Global.asax](#)

0	0	2	
Compartir	Twitter	+1	Compartir



Fernando  
Giardina

## Fernando Giardina

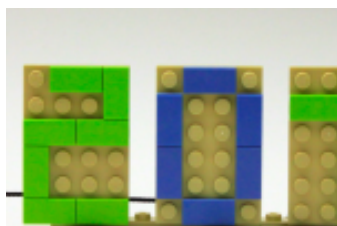
En 1997 comienzo a desarrollar en lenguaje C y Oracle programando IVRs, automatización de procesos y sistemas de CRM. Continuo con JAVA, Perl y PHP hasta que en el año 2002 me inicio en la tecnología .NET desarrollando aplicaciones web, desktop y mobile.

<http://www.metalstudio.com.ar>

## Otros artículos que podrían interesarte



[Cómo contar historias increíbles en Internet](#)



[10 Posts inspiradores de Maestros para guardar en PDF](#)



[12 formas de promocionar tu empresa desde el cover de Facebook](#)



[Derechos digitales en América Latina, los gobiernos te están observando \(yespiando\)](#)



Añade un comentario...

☐ Publicar también en Facebook

Publicar como **Javier J Solis Flores** ▼

Comentar

Plug-in social de Facebook



**Espantado**

Capo, una pregunta:  
¿Como hacer referencia al archivo .js dentro del tag HEAD ?  
Disculpa la ignorancia.



**Espantado**

Ya lo resolví  
Gracias de todas maneras!



**shylvia**

ola como puedo llamar la funcion script que esta en un archivo jscrip.js desde un boton programado en visual 2010



**angelomigue**

excelente editorial gracias maestrosdelaweb gracias por ayudarnos

<http://www.juegospa.com/> visitenla



**Tutorial ASP.NET: Utilización de código detrás del modelo o código en línea**

[...] Capítulo 3: Ejecutar código JavaScript en ASPNET [...]



**yense**

Buena man pero no se si puedes colocar como mandar datos a otra pagina y recojerla  
por ejemplo eso se hace con PHP y jquery de la siguiente manera

```
$.get("darvalores.php",{callback});
$.post("darvalores.php",{callback});
```

y bueno lo quiero hace de la misma manera pero en ASPx y C# con el mismo JQUERY

Saludos



**angel**

gracias me han ayuado mucho para mi web  
<http://www.juegodelocos.com/>  
muchas gracias



**angel7**

gracias me ayudaron mucho para resolver los problemas que tenia mi web... se los agradece  
<http://www.juegodelocos.com/>  
gracias