



APRENDERAPROGRAMAR.COM

HTMLELEMENT  
JAVASCRIPT. MÉTODOS  
FOCUS, BLUR. SUBMIT  
FORMULARIOS: DETENER  
ENVÍO. ONSUBMIT  
RETURN. EJEMPLOS.  
(CU01181E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

**Resumen:** Entrega nº81 del Tutorial básico "JavaScript desde cero".

Autor: César Krall

## SIGNIFICADOS DE FOCUS Y BLUR

Hemos de distinguir varios posibles usos de focus y blur. Como sabemos, focus hace alusión a que un elemento HTML toma el foco en la página web (bien porque el usuario haga click sobre él, bien porque al pulsar la tecla de tabulación el foco llegue hasta él). Tomar el foco indica que se puede escribir sobre el elemento (si es un campo de texto de un formulario) o que el elemento está seleccionado (si es otro tipo de elemento como una imagen).



Vamos a llamar la atención sobre el hecho de que focus y blur pueden tener distintos significados o usos:

- a) Eventos que podemos capturar para disparar un código de respuesta (esto ya lo hemos estudiado en apartados anteriores del curso).
- b) Un método aplicable a objetos de tipo Window (esto ya lo hemos estudiado en apartados anteriores del curso).
- c) Un método aplicable a objetos de tipo HTMLElement (nodos del DOM) para hacerles obtener o perder el foco, siempre que dicho objeto sea susceptible de tener el foco. Hay elementos como un input de un formulario que pueden tener el foco y otros como un span que no pueden tener el foco.

La sintaxis para establecer el foco en un elemento que admite el foco es:

```
nombreDelElemento.focus();
```

La sintaxis para hacer que un elemento que tenía el foco lo pierda es:

```
nombreDelElemento.blur();
```

Escribe el siguiente código y comprueba sus resultados. Comprueba cómo en él se hace uso por un lado de los eventos focus y blur, y por otro lado del método focus para establecer el foco cuando carga la página en el primer elemento input de tipo text que existe en el formulario.

Fijate también en cómo se manejan los eventos blur y focus para hacer que cuando un input de tipo text recibe el foco, su color de fondo cambie a amarillo, mientras que cuando pierde el foco, su color de fondo cambia a blanco.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">
<style type="text/css"> body {margin-left:30px; font-family: sans-serif;}
    .estiloForm {background-color: #f3f3f3; border: solid 2px black; margin-left:20px; width: 330px; padding:10px; }
    .estiloForm label {display: block; width: 120px; float: left; text-align:right; margin-bottom: 35px; padding-right: 20px;}
    br {clear: left;} input[type="submit"], input[type="reset"] {margin:25px 5px 10px 5px;}
</style>
<script type="text/javascript">
window.onload = function () {
    var elegido = false;
    var elementosInput = document.getElementsByTagName("input"); //Elementos input
    for (var i=0; i< elementosInput.length; i++) {
        if (elementosInput[i].type == 'text' && elegido==false) {var elegidoParaFoco = elementosInput[i];
elegido=true;}
        if (elementosInput[i].type == 'text') {
            elementosInput[i].addEventListener('blur', ponerFondoBlanco);
            elementosInput[i].addEventListener('focus', ponerFondoAmarillo);
        }
    }
    elegidoParaFoco.focus(); elegidoParaFoco.style.backgroundColor='yellow';
}

function ponerFondoBlanco() {this.style.backgroundColor = 'white';}
function ponerFondoAmarillo() {this.style.backgroundColor = 'yellow';}
</script>
</head>
<body><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplos JavaScript</h3>
<div class="estiloForm">
<form name="formularioContacto" method="get" action="#">
<label>Tratamiento</label>
<input type="radio" name="tratamiento" id="tratarSr" value="Sr."/>Sr.
    <input type="radio" name="tratamiento" id="tratarSra" value="Sra."/>Sra.<br/>
    <label>Nombre</label><input id="nombre" name="nombre" type="text"/><br/>
<label>Apellidos</label><input id="apellidos" name="apellidos" type="text"/><br/>
<label>Dirección</label><input id="direccion1" name="direccion1" type="text"/><br/>
<label>Ciudad</label><select id="ciudad" name="ciudad">
    <option value="">Elija opción</option>
    <option value="Mexico">México D.F. (MX)</option>
    <option value="Madrid">Madrid (ES)</option>
    <option value="Santiago">Santiago (CL)</option>
</select><br/>
    <label>Preferencias</label><input name="Libros" type="checkbox" />Libros
    <input name="Películas" type="checkbox" />Películas
    <input type="submit" value="Enviar"/> <input type="reset" value="Cancelar"/>
</form>
</div>
</body></html>

```

Cuando en una página web el formulario es el elemento principal, se considera una buena práctica (para facilitar que el usuario introduzca los datos) establecer el foco sobre el primer elemento del formulario una vez la página haya cargado.

## SIGNIFICADOS DE SUBMIT

Al igual que focus y blur pueden ser eventos o métodos de objetos, submit puede ser un evento pero también un método de un formulario. Como sabemos, submit hace alusión al envío de un formulario.

Podemos utilizar el evento submit para retrasar el envío del formulario hasta después de haber procesado el evento (de esta forma podemos hacer operaciones previas al envío, por ejemplo una validación). Una vez procesado el evento, procederíamos al envío del formulario. Para que el formulario se envíe o no según los resultados de las operaciones previas al envío añadiremos el parámetro para capturar el evento en la función manejadora y aplicaremos el método preventDefault() que hemos estudiado en anteriores entregas del curso.

Este sería un ejemplo de sintaxis:

```
document.forms['nombreFormulario'].addEventListener('submit', funcionManejadora);  
...  
...  
function funcionManejadora(evObject) {  
    evObject.preventDefault(); //Anulamos la acción de defecto  
    if ( ... ) { //Tareas a realizar si se cumple la condición  
        document.forms['formularioContacto'].submit(); } //Se envía el formulario  
    else { //Tareas a realizar si no se cumple la condición  
        ... } //No se envía el formulario  
}
```

Escribe este código y comprueba sus resultados.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">  
  <style type="text/css"> body {margin-left:30px; font-family: sans-serif;}  
    .estiloForm {background-color: #f3f3f3; border: solid 2px black; margin-left:20px; width: 330px; padding:10px;}  
    .estiloForm label {display: block; width: 120px; float: left; text-align:right; margin-bottom: 35px; padding-right: 20px;}  
    br {clear: left;} input[type="submit"], input[type="reset"] {margin:25px 5px 10px 5px;}  
  </style>  
<script type="text/javascript">  
  window.onload = function () {  
    document.forms['formularioContacto'].addEventListener('submit', avisarUsuario);  
  }  
  
  function avisarUsuario(evObject) {  
    evObject.preventDefault();  
    var nuevoNodo = document.createElement('h2');  
    nuevoNodo.innerHTML = '<h2 style="color:orange;">Enviando el formulario...</h2>';  
    document.body.appendChild(nuevoNodo);  
    var retrasar = setTimeout(procesaDentroDe2Segundos, 1000);  
  }  
</script>
```

```
function procesaDentroDe2Segundos() {
document.forms['formularioContacto'].submit();
}

</script>
</head>

<body><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplos JavaScript</h3>
<div class="estiloForm">
<form name="formularioContacto" method="get" action="http://aprenderaprogramar.com">
<label>Tratamiento</label>
<input type="radio" name="tratamiento" id="tratarSr" value="Sr."/>Sr.
<input type="radio" name="tratamiento" id="tratarSra" value="Sra."/>Sra.<br/>
<label>Nombre</label><input id="nombre" name="nombre" type="text"/><br/>
<label>Apellidos</label><input id="apellidos" name="apellidos" type="text"/><br/>
<label>Dirección</label><input id="direccion1" name="direccion1" type="text"/><br/>
<label>Ciudad</label><select id="ciudad" name="ciudad">
<option value="">Elija opción</option>
<option value="Mexico">México D.F. (MX)</option>
<option value="Madrid">Madrid (ES)</option>
<option value="Santiago">Santiago (CL)</option>
</select><br/>
<label>Preferencias</label><input name="Libros" type="checkbox" />Libros
<input name="Peliculas" type="checkbox" />Películas
<input type="submit" value="Enviar"/> <input type="reset" value="Cancelar"/>

</form>
</div>
</body>
</html>
```

El resultado esperado es que cuando se pulsa el botón enviar del formulario, debajo de este aparezca el mensaje "Enviando el formulario...". Para hacer aparecer este mensaje hemos añadido un nodo al DOM y establecido su contenido con innerHTML.

Nota: para que el efecto sea visible hemos introducido un pequeño retraso forzado en el envío del formulario usando un setTimeout. Hemos hecho esto para que nos dé tiempo a ver el texto (si la conexión a internet que usemos es muy rápida y no establecemos una pequeña demora no nos daría tiempo a ver el mensaje, ya que este desaparece en cuanto comienza a cargarse la página de destino del formulario). Este retraso no tiene una utilidad "práctica", únicamente lo hemos introducido para poder ver cómo podemos dar lugar a la ejecución del evento submit usando el método submit() en el momento en que nosotros deseemos.

## EJERCICIO Y EJEMPLO

Otra forma de detener el envío de un formulario se basa en que si el código de respuesta a un evento submit es el valor booleano false, no se producirá el envío del formulario. En otro caso, sí se producirá el envío. Examina el siguiente código y responde a las cuestiones que se muestran a continuación:

```
<!DOCTYPE html>
<html><head><script>
function validarForm() {
  var x = document.forms["elForm1"]["nombre"].value;
  if (x==null || x=="") { alert("El formulario no puede enviarse sin rellenar el nombre");
    return false; }
}
</script></head>
<body><form style ="margin:30px;" name="elForm1" action="http://aprenderaprogramar.com" onsubmit="return
validarForm()" method="get">
Nombre: <input type="text" name="nombre">
<input type="submit" value="Enviar">
</form>
</body></html>
```

- a) ¿Qué cometido cumple este código?
- b) ¿Qué ocurre si en lugar de onsubmit="return validarForm()" escribes onsubmit="validarForm()"? ¿Por qué?
- c) Si x no es null ni es vacío, ¿qué devuelve la función validarForm()? ¿Qué implicaciones tiene esto en relación al envío del formulario?

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros [aprenderaprogramar.com](http://aprenderaprogramar.com).

**Próxima entrega:** CU01182E

**Acceso al curso completo** en [aprenderaprogramar.com](http://aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:  
[http://aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=78&Itemid=206](http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=78&Itemid=206)