

Desarrollo Web en Entorno Cliente (DWEC)

# ETIQUETA SCRIPT. COMPORTAMIENTO HTML

---

26 de septiembre de 2018

## ENUNCIADO

Realiza las siguientes lecturas y contesta a las siguientes cuestiones:

- [async vs defer attributes](#)
  - [Where should I put <script> tags in HTML markup?](#)
  - [Etiqueta <NOSCRIPT>, Curso práctico HTML5](#)
1. Ventajas de incorporar el script desde un fichero externo frente a hacerlo en línea (código javascript directamente en el fichero HTML)
  2. Atributos aplicables a la etiqueta script
  3. Atributos por defecto.
  4. Atributos booleanos. Qué implican.
  5. Sitio recomendado para colocar la etiqueta script.
  6. Etiqueta noscript: utilidad, atributos y dónde colocar

## 1.-VENTAJAS DE INCORPORAR EL SCRIPT DESDE UN FICHERO EXTERNO.

Cuando un navegador carga un sitio web ocurren 6 pasos en este:

1. Obtiene o recupera la página. Por ejemplo un index.html
2. El navegador comienza a analizar el HTML
3. Se encuentra una etiqueta **script**.
4. El navegador solicita el archivo script y detiene la carga del HTML hasta obtener este.
5. Después de un rato el script se descarga y se ejecuta.
6. El analizador continua con el resto de código HTML.

El punto número 4 impide que el usuario siga cargando su página HTML hasta que el script sea validado y ejecutado, por lo cual es una mala experiencia para él.

Por lo cual si nuestro código javascript está en un archivo aparte .js el analizador cargará toda la página HTML antes de ser modificada por nuestros scripts de javascript.

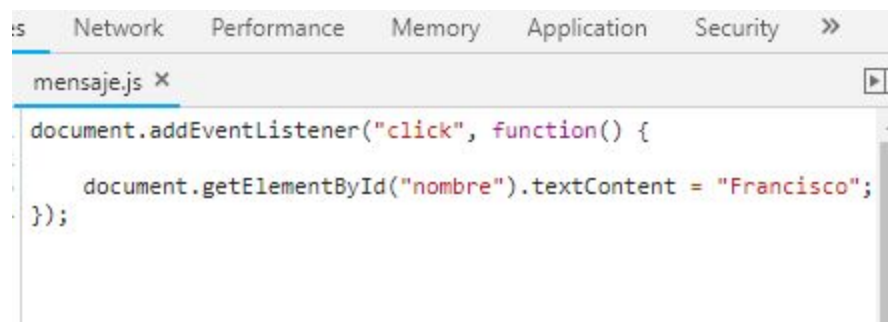
Según este código HTML tenemos un archivo **mensaje.js** con una función al hacer **click**.

El analizador tiene la página web correspondiente cargada, y el script no se cargará hasta que no activemos el evento.

### Primera pasada del analizador.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Prueba</title>
    <script type="text/javascript" src="mensaje.js"></script>
  </head>
  <body>
    <p id="nombre">Click para saber mi nombre!</p>
  </body>
</html>
```

### Segundo analiza el archivo javascript y lo carga



```
document.addEventListener("click", function() {
  document.getElementById("nombre").textContent = "Francisco";
});
```

Tercero el HTML está cambiado por el script.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Prueba</title>
    ... <script type="text/javascript" src="mensaje.js"></script> == $
  </head>
  <body>
    <p id="nombre">Francisco</p>
  </body>
</html>
```

Una recomendación anticuada era colocar los scripts en la parte inferior del body

## 2.-ATRIBUTOS APLICABLES A LA ETIQUETA SCRIPT.

1. **crossorigin** → Indica si la petición hecha a un servidor externo debe presentar credenciales **CORS**<sup>1</sup> o no. Este atributo resulta útil al incrustar programas ubicados en otros servidores que soportan acceso cross-origin, para permitir reporte de errores enriquecido. Existen dos valores:
  - a. **anonymous**: las peticiones CORS para el elemento tendrán la etiqueta "omit credentials" establecida.
  - b. **use-credentials**: las peticiones CORS para el elemento no tendrán la etiqueta "omit credentials" establecida.
2. **integrity** → Permite a los navegadores comprobar que el recurso recuperado no ha sido manipulado.
3. **charset** → Codificación de caracteres. Actualmente obsoleta ya que es requisito que todos los programas sean interpretados en UTF-8
4. **importance** → Indica la importancia relativa de la secuencia de comandos. Los consejos de prioridad se delegan usando los valores :
  - a. auto → ninguna preferencia
  - b. high → alta
  - c. low → baja
5. **language** → parecido al type, indica el lenguaje del script sin embargo este no está estandarizado. *Mejor usar el type.*

<sup>1</sup> Uso compartido de recursos entre orígenes  
Estándar de W3C donde un servidor modera la directiva de origen.

### 3.-ATRIBUTOS POR DEFECTO.

1. src → Indica la URL donde se aloja el archivo javascript.

```
<script src="mensaje.js"></script>
```

2. type → El tipo de contenido que representa el script. Existen tres tipos

**text/javascript** le afecta charset, async y defer

**module** programa módulo<sup>2</sup>, no le afecta charset y defer

**cualquier otro** → imágenes, videos, audios...

```
<script type="text/javascript"></script>
```

### 4.-ATRIBUTOS BOOLEANOS. QUÉ IMPLICAN.

1. **nomodule** → Atributo booleano que indica que el programa no debe de ser ejecutado que poseen soporte para programas modulares. Por ejemplo HotJava.
2. **async** → Instruye a los navegadores a ejecutar los programas tan pronto como les sea posible, pero sin bloquear el proceso de carga del documento.

```
<script src="mensaje.js" async=></script>
```

3. **defer** → Indica que los programas deberían ser ejecutados una vez que el documento ha sido completamente cargado.

### 5.-SITIOS RECOMENDADOS PARA COLOCAR LA ETIQUETA SCRIPT

Como hemos comentando anteriormente en el apartado 1, la posición de los scripts es importante a la hora de cargar una página web para ellos, deberemos de introducir estos, siempre que se pueda, en la parte inferior de nuestro código html5, pero siempre antes de </body>.

Sin embargo no siempre se puede, por ejemplo con el uso de **document.write("")**; La posición de esta es esencial a la hora de cargar la página web.

---

<sup>2</sup> Subprogramas con el fin de hacerlo más legible.

## 6.-ETIQUETA NOSCRIPT: UTILIDAD, ATRIBUTOS Y DÓNDE COLOCAR.

La etiqueta noscript es complementaria de la etiqueta de script, esta se utiliza cuando ciertos navegadores no soportan la etiqueta script o están desactivados.

Unos navegadores actuales que no soportan la etiqueta script son:

*Internet explorer.*

*Opera Mini.*

*Uc Browser for Android*

*Samsung Internet*

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			63						
			67		10.3				
		61	68	11.1	11.2				4
11	17	62	69	12	11.4	todas	69	11.8	7.2
	18	63	70	TP	12				
		64	71						
			72						

La posición de la etiqueta depende de su utilidad, podemos colocar en el head o en el cuerpo del código:

1. **head** → Si se utiliza dentro de esta etiqueta es para que afecte a ciertas otras como <link>, <style>, <meta>... y así, el navegador los interprete aunque no permita la ejecución de script.
2. **body** → Si se indica en el cuerpo del código es para avisar al usuario la necesidad de tener JavaScript para ver el contenido de esta.

 Prueba



Click para saber mi nombre!

**No me veo en Google Chrome, pero si en internet explorer!!!!**

La etiqueta <noscript> no posee ningún tipo de atributos.