



APRENDERAPROGRAMAR.COM

JAVASCRIPT INTERNO
<SCRIPT TYPE=
"TEXT/JAVASCRIPT">.
INTÉRPRETE JAVASCRIPT .
FUNCIÓN EJEMPLO
BÁSICO. (CU01109E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

Resumen: Entrega nº9 del Tutorial básico “JavaScript desde cero”.

Autor: César Krall

INCLUIR JAVASCRIPT EN WEBS.

Hemos visto que podemos incluir JavaScript “embebido” dentro de una línea de HTML. Pero esto será poco práctico cuando tengamos que escribir pequeños programas (sería muy confuso entremezclarlo todo con el HTML). Por eso vamos a ver dos formas adicionales para escribir JavaScript de forma clara y ordenada: JavaScript interno delimitado por etiquetas, o JavaScript en archivos externos.



JAVASCRIPT INTERNO DELIMITADO POR ETIQUETAS (TAGS)

De la misma forma que el código CSS puede aparecer agrupado entre etiquetas `<style> ... </style>`, el código JavaScript irá entre etiquetas `<script> ... </script>` tal y como mostramos a continuación:

```
<head>
...
<script type="text/javascript">
function nombreDeFuncion1() {
... aquí el código de la función
}
function nombreDeFuncion2() {
... aquí el código de la función
}
...
</script>
</head>
```

Dentro de las etiquetas `<head> ... </head>` incluiremos una etiqueta de apertura de declaración de script `<script type="text/javascript">`, a continuación colocaremos la definición de una, dos o más funciones JavaScript y terminaremos cerrando con la etiqueta `</script>`.

En muchas páginas web y muchos libros verás que se escribe `<script> ... </script>` sin hacer uso de la definición de type. El motivo para esto es que JavaScript se ha convertido en un estándar reconocido por todos los navegadores y ya no se considera necesario especificar `type="text/javascript"`. En versiones antiguas de HTML se consideraba necesario, pero en las actuales no es necesario especificar `type="text/javascript"`. Todos los navegadores actuales reconocerán JavaScript sin necesidad de la especificación type.

En algunas páginas web antiguas te podrás encontrar incluso otros tipos de sintaxis como `<script language="JavaScript">....</script>`.

Los motivos por los que se introdujo la especificación type son varios:

- a) Existían (y existen) distintos lenguajes de script, es decir, además de javascript nos podemos encontrar otros lenguajes. El uso de type se concibió para indicarle al navegador qué lenguaje era el que usaba el script. Por ejemplo type="text/vbscript" indicaría lenguaje VBscript, type="text/tcl" indicaría lenguaje TCL, etc.
- b) Fue parte de la especificación oficial de algunas versiones de HTML.

Desde el momento en que todos los navegadores reconocen JavaScript sin necesidad de la especificación type, usarla o no resulta indistinto y no vamos a preocuparnos más por este asunto. Por otro lado ten en cuenta que la etiqueta <script> y su contenido no es lenguaje JavaScript, sino lenguaje HTML.

Una cuestión importante e interesante que ha surgido al ver cómo se define JavaScript es que el código JavaScript se escribe normalmente dentro de funciones. Hablaremos sobre las funciones con más detalle más adelante. De momento, simplemente consideraremos que una función es un fragmento de código que tiene un nombre, y que puede ser invocado desde algún punto del código HTML para ser ejecutado. Para escribir una función especificaremos de momento: `function nombreDeLaFuncionAqui { código ... }` Las funciones introducen algo interesante: cuando tengamos que hacer algo repetidas veces, no tendremos que escribir el código varias veces, sino simplemente invocar a la función para que se ejecute.

Veamos un ejemplo. Escribe este código y guárdalo en un archivo de extensión html:

```
<html>
<head>
<title>Portal web - aprenderaprogramar.com</title> <meta charset="utf-8">
<script type="text/javascript">
function mostrarMensaje1() {
  alert('Bienvenido al curso JavaScript de aprenderaprogramar.com');
}
function mostrarMensaje2() {
  alert('Ha hecho click sobre el párrafo inferior');
}
</script>
</head>
<body>
<div>
<p>Aquí un párrafo de texto situado antes de la imagen, dentro de un div contenedor</p>

<p onclick="mostrarMensaje2()">Aquí otro párrafo de texto. JavaScript es un lenguaje utilizado para dotar
de efectos dinámicos a las páginas web.
</p>
</div>
</body>
</html>
```

Visualiza la página. El resultado esperado será que al hacer click sobre la imagen se muestre un mensaje (el mensaje 1), mientras que al hacer click sobre el párrafo inferior se mostrará otro mensaje (el mensaje 2). Aquí vemos cómo hemos definido un script. Nos podemos hacer algunas preguntas como:

¿Puede escribirse un script en otro lugar distinto del comprendido entre las etiquetas <head> ... </head>?

La respuesta es que sí. La mayoría de los navegadores aceptarán que el script esté en casi cualquier parte. Prueba a cambiar el código y ponerlo en distintos lugares y comprueba los resultados. Pero poniendo el código fuera de <head> ... </head> podemos tener problemas con algunos navegadores, o podemos tener problemas en algunos casos en que el código deba guardar cierto orden. Para no tener problemas pondremos siempre los scripts dentro de las etiquetas <head> ... </head>.

¿Pueden escribirse varios scripts en una misma página web?

La respuesta es que sí. Por ejemplo esto sería válido:

```
<script type="text/javascript">
function mostrarMensaje1() {
alert('Bienvenido al curso JavaScript de aprenderaprogramar.com');
}
</script>
<script type="text/javascript">
function mostrarMensaje2() {
alert('Ha hecho click sobre el párrafo inferior');
}
</script>
```

Aquí en vez de un script con dos funciones, tenemos dos scripts cada uno de ellos con una función. ¿Cuál es mejor opción, un script o dos scripts? Recordar que JavaScript es un código que es interpretado por el navegador (en concreto por una parte del navegador a la que se denomina precisamente intérprete JavaScript). Al escribir dos scripts estamos obligando al intérprete a leer más líneas, y considerar dos unidades de scripts en lugar de una. Esto ralentiza la carga de la página web y su ejecución, por tanto normalmente incluiremos el código dentro de un único script, excepto cuando justificadamente necesitemos o sea razonable hacerlo de otra manera. En páginas web avanzadas es posible que encuentres muchos scripts que salen de muchos sitios diferentes.

¿Qué extensión puede tener un script?

La que nosotros queramos. Puede ser desde una línea hasta miles de líneas.

Próxima entrega: CU01110E

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