

1.A. Arquitecturas y lenguajes de programación en clientes web

4.- Integración de código Javascript con HTML.

Ahora que ya conoces las herramientas que puedes utilizar para comenzar a programar en JavaScript, vamos a ver distintas formas de ejecutar JavaScript en nuestro navegador.

Una posible forma es usar el esquema javascript: en la barra de tu navegador. Actualmente pocos navegadores lo permiten por motivos de seguridad. A este tipo de "trozos" de código se les llama [bookmarklets](#) y permiten ejecutar pequeñas aplicaciones JavaScript sin necesidad de cargar un documento ya que la URL del bookmarklet se puede almacenar en los marcadores del navegador.

Sin embargo, lo común y lo que vamos a hacer en este módulo es **integrar el código JavaScript** en el código HTML.

Los navegadores web te permiten varias opciones de inserción de código de JavaScript.

Podremos insertar código usando las etiquetas `<script>` `</script>` y empleando un atributo `type` indicaremos qué tipo de lenguaje de script estamos utilizando:

Por ejemplo:

```
<script type="text/javascript">
// El código de JavaScript vendrá aquí.
</script>
```

Otra forma de integrar el código de JavaScript es incrustar un fichero externo que contenga el código de JavaScript. Ésta sería la forma más recomendable, ya que así se consigue una separación entre el código y la estructura de la página web y como ventajas adicionales podrás compartir código entre diferentes páginas, centralizar el código para la depuración de errores, tendrás mayor claridad en tus desarrollos, más modularidad, seguridad del código y conseguirás que las páginas carguen más rápido. La rapidez de carga de las páginas se consigue al tener el código de JavaScript en un fichero independiente, ya que si más de una página tiene que acceder a ese fichero lo cogerá automáticamente de la caché del navegador con lo que se acelerará la carga de la página. Para ello tendremos que añadir a la etiqueta `script` el atributo `src`, con el nombre del fichero que contiene el código de JavaScript. Generalmente los ficheros que contienen texto de JavaScript tendrán la extensión `.js`.

Por ejemplo:

```
<script type="text/javascript" src="tucodigo.js"></script>
```

Si necesitas cargar más de un fichero `.js` repite la misma instrucción cambiando el nombre del fichero. Las etiquetas de `<script>` y `</script>` son obligatorias a la hora de incluir el fichero `.js`. **Atención:** no escribas ningún código de JavaScript entre esas etiquetas cuando uses el atributo `src`.

Para **referenciar el fichero origen .js** de JavaScript dependerá de la localización física de ese fichero. Por ejemplo en la línea anterior el fichero `tucodigo.js` deberá estar en el mismo directorio que el fichero `.html`. Podrás enlazar fácilmente a otros ficheros de JavaScript localizados en directorios diferentes de tu servidor o de tu dominio. Si quieres hacer una referencia absoluta al fichero, la ruta tendrá que comenzar por `http://`, en otro caso tendrás que poner la ruta relativa dónde se encuentra tu fichero `.js`

Ejemplos:

```
<script type="text/javascript" src="http://www.tudominio.com/ejemplo.js"></script>
<script type="text/javascript" src="../js/ejemplo.js"></script>
```

(el fichero `ejemplo.js` se encuentra en el directorio anterior (`../`) al actual, dentro de la carpeta `js/`)

Cuando alguien examine el código fuente de tu página web verá el enlace a tu fichero `.js`, en lugar de ver el código de JavaScript directamente. Ésto no quiere decir que tu código no sea inaccesible, ya que simplemente copiando la ruta de tu fichero `.js` y tecleándola en el navegador podremos descargar el fichero `.js` y ver todo el código de JavaScript. En otras palabras, nada de lo que tu navegador descargue para mostrar la página web podrá estar oculto de la vista de cualquier programador.



A veces te puedes encontrar que tu script se va a ejecutar en un navegador que no soporta JavaScript. Para ello dispones de una etiqueta `<noscript>Texto informativo</noscript>` que te permitirá indicar un texto adicional que se mostrará indicando que ese navegador no soporta JavaScript.

Si estamos trabajando con **XHTML**, la sintaxis para insertar un código de **JavaScript** directamente en el XHTML es un poco diferente:

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

<!--><![CDATA[</><!--

// código de JavaScript a continuación

//--><!]]>

</script>
```

Los analizadores de XHTML son intolerantes a los elementos que no comienzan por `<` y a las entidades HTML que no comienzan por `&`, y estos caracteres como verás más adelante son ampliamente utilizados en JavaScript. La solución es encapsular las instrucciones de JavaScript en lo que se conoce como sección CDATA:

```
<![CDATA[

XHTML permite cualquier tipo de carácter aquí incluyendo < y el símbolo &

]]>
```

Como puedes observar el código es un poco complejo en este caso, razón de más para integrar el código de JavaScript en un fichero externo.

El código que escribes es legible por cualquier persona que tenga los conocimientos suficiente para acceder a él ya que no está protegido de ninguna forma. Seguramente estarás pensando en cómo puedes **proteger el código de JavaScript** que vas a programar, del uso fraudulento por otros programadores o visitantes a tu página: la respuesta rápida a esa pregunta es que es imposible hacerlo.

Para que el código de JavaScript pueda ejecutarse correctamente deberá ser cargado por el navegador web y por lo tanto su código fuente deberá estar visible al navegador. Si realmente te preocupa que otras personas usen o roben tus scripts, deberías incluir un mensaje de copyright en tu código fuente. Piensa que no solamente tus scripts son visibles al mundo, sino que los scripts del resto de programadores también lo son. De esta forma puedes ver fácilmente cuando alguien está utilizando al pie de la letra tus scripts, aunque esto no evita que alguien copie tu código y elimine tu mensaje de copyright.

Lo más que se puede hacer es que el código sea más difícil de leer o incluso hacerlo ilegible para un humano (a esto se suele denominar ofuscar el código). Las técnicas de ofuscación incluyen la eliminación de saltos de línea, espacios en blanco innecesarios, tabuladores, utilización de nombres ininteligibles en las funciones y variables, utilización de variables para almacenar trozos de código, uso de recursividad, etc. La forma más rápida de hacer todas esas tareas de ofuscación es utilizar un software que producirá una copia "comprimida" del script que has programado para facilitar su carga rápida.

Aquí tienes un par de proyectos para ofuscar tu código JavaScript:

- [JavaScript Ofuscator](#).
- [UglifyJS](#)

Lo mejor es que cambies ese paradigma y pienses de una manera diferente. En lugar de proteger tu código, lo mejor es promocionarlo y hacer ostentación de él, añadiendo comentarios útiles en el código fuente, publicándolo en tu blog o en webs de programación, etc. También liberando el mismo con una licencia de software libre que se adapte a tus intenciones. Por ejemplo una licencia **Creative Commons** para animar a la gente a que lo utilice, lo copie y lo mantenga público al resto del mundo. Así conseguirás mayor reputación como buen programador y la gente contactará contigo para más información, posibles trabajos, etc.

¿Te gustaría saber qué tipos de licencias tienes bajo Creative Commons?

[Explicación de licencias Creative Commons](#).

Créditos de las imágenes

HTML

Autoría: <https://pixabay.com/es/html5-html-tipo-de-archivo-teclado-386614/>

Licencia: Pixabay

Creative Commons

Autoría: Creative Commons

Licencia: CC-BY



