


Capítulo 3. Contenido

- 3: Inserción de Código JavaScript en HTML
 - 3.1. Etiqueta `<script>` en el Cuerpo del Documento
 - 3.2. Etiqueta `<script>` en el Encabezado
 - 3.3. Archivo Externo JavaScript 
 - Ejemplo de llamada en el `<body>` :
 - Ejemplo de llamada en el `<head>` :

3. Inserción de Código JavaScript en HTML

En HTML, puedes incluir código JavaScript de varias maneras:

3.1 Etiqueta `<script>` en el Cuerpo del Documento

Puedes incluir código JavaScript en la sección `<body>` de tu documento HTML utilizando la etiqueta `<script>`.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mi Página</title>
  </head>
  <body>
    <h1>Ejemplo de código JavaScript en el cuerpo del documento</h1>
    <script>
      function saludar() {
        alert("¡Hola, mundo!");
      }
      saludar();
    </script>
  </body>
</html>
```

3.2 Etiqueta `<script>` en el Encabezado

También puedes incluir código JavaScript en la sección `<head>` de tu documento HTML. En este caso, generalmente se coloca en un evento como `DOMContentLoaded` para asegurarse de que el DOM esté completamente cargado antes de ejecutar el código.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mi Página</title>
    <script>
      document.addEventListener("DOMContentLoaded", function () {
        function saludar() {
          alert("¡Hola, mundo!");
        }
        saludar();
      });
    </script>
  </head>
  <body>
    <h1>Ejemplo de código JavaScript en el encabezado</h1>
  </body>
</html>
```

3.3 Archivo Externo JavaScript

Para mantener tu código organizado, es una buena práctica incluir código JavaScript en archivos externos y luego enlazar esos archivos en tu HTML utilizando la etiqueta `<script>`. Para ello y como he comentado podríamos hacerlo desde la cabecera o Head o desde el Body, teniendo presente en este caso que se debe de insertar la llamada al script antes de cerrar la etiqueta `</body>`.

Ejemplo de llamada en el `<body>` :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mi Página</title>
  </head>
  <body>
    <h1>Ejemplo de archivo externo JavaScript</h1>
    <script src="mi_script.js"></script>
  </body>
</html>
```

El contenido de `mi_script.js` :

```
// mi_script.js
function saludar() {
  alert("¡Hola, mundo!");
}
saludar();
```

Ejemplo de llamada en el `<head>` :

La llamada de un fichero con código JavaScript en la cabecera es algo más delicada.

La etiqueta `<script>` en HTML tiene varios atributos que pueden afectar cómo se carga y ejecuta el código JavaScript en la página. Muestro algunos ejemplos, junto con los atributos que se pueden utilizar:

1. **src (Source)**: Este atributo se utiliza para especificar la fuente (URL) de un archivo JavaScript externo que se va a cargar y ejecutar en la página.

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

2. **async** : Cuando se agrega el atributo `async` a la etiqueta `<script>` , se indica que el archivo JavaScript se debe **cargar de forma asíncrona** mientras se sigue analizando la página HTML. Esto significa que la ejecución de la página no se bloqueará por la carga del script y se ejecutará tan pronto como esté disponible. Sigue existiendo un bloqueo en el renderizado pero menor que con el comportamiento normal. No se garantiza la ejecución de los scripts asíncronos en el mismo orden en el aparecen en el documento.

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

3. **defer** : Al agregar el atributo `defer` a la etiqueta `<script>`, se le dice al navegador que el archivo JavaScript se debe cargar de forma asíncrona, pero la ejecución del script se pospone hasta que se complete el análisis del documento HTML. Esto es útil cuando el orden de ejecución del script es importante y debe ser relativo al orden en que aparece en el documento. Además La ejecución de todos los scripts diferidos se realiza en el mismo orden en el que aparecen en el documento.

```
<script src="mi_script.js" defer></script>
```

4. **type** : El atributo `type` se utiliza para especificar el tipo de contenido del script. El valor común es `"text/javascript"`.

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

5. **charset** : Este atributo se utiliza para especificar la codificación de caracteres del archivo JavaScript, aunque en la mayoría de los casos no es necesario ya que se asume UTF-8 por defecto.

```
<script src="mi_script.js" charset="UTF-8"></script>
```

Es importante elegir la combinación adecuada de estos atributos según las necesidades de tu página y cómo deseas que se cargue y ejecute el código JavaScript. Por ejemplo, si necesitas que el script se ejecute tan pronto como esté disponible, puedes usar `async`. Si necesitas controlar el orden de ejecución, puedes usar `defer`.