

Tutorial de JavaScript

Insertar y eliminar elementos (nodos) en una página web de forma dinámica...

[\(index.php\)](#)
[\(document.php\)](#)
[\(formularios.php\)](#)
[\(objetos.php\)](#)

Acceder desde Javascript a elementos HTML

Para acceder a elementos HTML usaremos los siguientes métodos del Objeto **Document()**:

- **getElementById()**: nos permite acceder a elementos HTML a los que hemos asignado un **identificador** con el atributo HTML **id** (no deben haber más de uno con el mismo).
- **getElementsByName()**: para acceder a elementos HTML que tienen un atributo HTML **name** (pueden ser varios, como en la imagen de ejemplo).
- **getElementsByTagName()**: permite acceder a todos los elementos de una determinada etiqueta HTML.

En la siguiente imagen te mostramos un ejemplo:

The image shows an HTML document with the following structure:

```

01. <html>
02.   <head>
03.     <title>Tutorial JavaScript</title>
04.     <link rel="stylesheet" type="text/css" href="estilos.css" />
05.     <script src="ejemplos.js" type="text/javascript"></script>
06.   </head>
07.   <body>
08.     <div id="div1">Este es el contenido del DIV</div>
09.
10.     <form id="miFormulario" name="miFormulario" method="post" action="#" >
11.
12.       <label for="opc11">Opción 1A</label>
13.       <input type="radio" id="opc11" value="11" name="grupo1" />
14.
15.       <label for="opc12">Opción 1B</label>
16.       <input type="radio" id="opc12" name="grupo1" value="12" name="grupo1" checked />
17.
18.       <label for="opc13">Opción 1C</label>
19.       <input type="radio" id="opc13" name="grupo1" value="13" name="grupo1" />
20.
21.     </form>
22.
23.     <span>Este es el contenido del primer SPAN</span>
24.     <span>Este es el contenido del segundo SPAN</span>
25.
26.   </body>
27. </html>
  
```

Annotations in the image:

- A box points to `document.getElementById("div1")...` with the text: "document.getElementById("div1")..."
- A box points to `name="grupo1"` in the radio button attributes with the text: "document.getElementsByName("grupo1")... (devuelve un array con los elementos que tengan el nombre indicado)"
- A box points to `` tags with the text: "document.getElementsByTagName("span")... (devuelve un objeto NodeList con los los elementos encontrados)"

Usamos cookies para ofrecerte una experiencia mejorada, el continuar navegando supone que aceptas su uso

Una vez que sabemos cómo acceder a los elementos HTML podremos usar sus propiedades, métodos y eventos:

En el siguiente ejemplo, desde JavaScript accedemos al elemento 'div1' y con la propiedad **innerHTML** modificamos el contenido del mismo. Además, cambiamos el texto que se muestra en el botón 'btnEnviar':

```
01. document.getElementById("div1").innerHTML = "<p style='color:yellow;'>Introduzca los datos</p>";
02. document.getElementById("btnEnviar").value = "Enviar los datos";
```

Podemos usar **innerHTML** para obtener y asignar el contenido HTML de cualquier etiqueta, incluidas **<html>** y **<body>**:


```
01. // Inserta código HTML en la página web (elimina lo existente):
02. document.body.innerHTML = "<p id='parrafo1'>Haz clic en este <a id='enlace' href='https://informaticapc.
03.                               <p id='parrafo2'><span style='color:red;'>Cursos de informática gratis</span>
04.
05. // Usando el operador '+' se añade el siguiente código HTML al existente:
06. document.body.innerHTML += "<p id='parrafo3' >Curso de JavaScript</p>";
07. document.body.innerHTML += "<p id='parrafo4' ><input type='text' id='texto1' value='El texto' /></p>";
08. document.body.innerHTML += "<p id='parrafo5' ><input type='button' id='boton1' value='Esto es un botón'";
```

Continuando con el ejemplo anterior, usaremos **innerHTML** para acceder a los elementos HTML que hemos creado:

```
01. alert("Pulsa para modificar la página");
02.
03. document.getElementById("e").innerHTML = "enlace a informaticapc.com";
04. document.getElementById("parrafo2").innerHTML = "CURSOS GRATIS";
```

 **Descargar ejemplo** (./ejemplos/10_acceder_pagina_web/03_documento_acceder.zip)

La propiedad **innerHTML** no forma parte del **estándar W3C** (organismo que desarrolla estándares para la Web), pero es soportado por la mayoría de los navegadores web.

 En versiones antiguas del navegador web Internet Explorer para usar **getElementsByName()** se debía usar el **id** del elemento HTML, en lugar del atributo **name**.

Asimismo, no era posible acceder elementos HTML usando **getElementsByName()**, si se insertaban dinámicamente usando **appendChild()**.

 Usamos cookies para ofrecerte una experiencia mejorada, el continuar navegando supone que aceptas su uso

Crear elementos de forma dinámica


Cerrar

Más información

Con el método **createElement()** del objeto **Document()** podemos insertar elementos HTML al final del cuerpo de la página web, o de otro elemento HTML.

En el siguiente ejemplo te mostramos cómo **crear un enlace e insertarlo en la página**:

```
01.  var enlace1 = document.createElement("A");
02.
03.  // Atributos o Propiedades:
04.  enlace1.id = "enlace_1";
05.  enlace1.innerHTML = "Cursos de informática gratis";
06.  enlace1.href   = "https://informaticapc.com/";
07.  enlace1.target = "blank";
08.
09.  // Insertar el enlace al final del body:
10.  document.body.appendChild( enlace1 );
```

 **Descargar ejemplo** (./ejemplos/10_acceder_pagina_web/04_documento_dom_crear_1.zip)

En este otro ejemplo creamos dinámicamente un botón, le asignamos algunas propiedades, definimos un evento **onclick()**, y lo insertamos en un **<div>** cuyo atributo **id** es 'div1':

```
01.  var boton1 = document.createElement("input");
02.
03.  // Atributos o Propiedades:
04.  boton1.id   = "boton_2";
05.  boton1.type = "button";
06.  boton1.value = "Blog de informática y tecnología";
07.
08.  // Evento:
09.  boton1.onclick = function(){
10.      window.location.href = "http://www.informaticapc.com/";
11.  }
12.
13.  // Insertar el botón al final de 'div1':
14.  document.getElementById("div1").appendChild( boton1 );
```

Podemos indicar si el nuevo elemento puede insertarse como padre de otro usando **parentNode.insertBefore()**.

Supongamos que dentro de un **<div>** tenemos un **** cuyo atributo **id** es 'span1': si accedemos a 'span1' como hemos visto antes, e insertamos dinámicamente otro ****, éste quedaría dentro del primero. En cambio, del siguiente modo se insertaría fuera de él, quedando al final del **<div>**.


Usamos cookies para ofrecerte una experiencia mejorada, el continuar navegando supone que aceptas su uso

Lo puedes ver en el siguiente ejemplo:

Cerrar

Más información

```
01.   var span2 = document.createElement("span");
02.
03.   // Atributos o Propiedades:
04.   span2.id   = "span_2";
05.   span2.style.backgroundColor = "red";
06.   span2.innerHTML = "Contenido del span 2";
07.
08.   // Insertar 'span2' fuera de 'span1':
09.   document.getElementById("span1").parentNode.insertBefore( span2 );
```

 **Descargar ejemplo** (./ejemplos/10_acceder_pagina_web/05_documento_dom_crear_2.zip)


 **Vea también:** [Eventos en JavaScript (<https://informaticapc.com/tutorial-javascript/eventos.php>)]

Eliminar elementos del DOM

Para borrar un elemento del DOM (y los que hayan dentro de él) debemos acceder al elemento HTML o nodo padre dentro del que se encuentra, y usar el método **removeChild()** del Objeto **Document()** para eliminarlo.

Por ejemplo, para eliminar un **<div>** cuyo **id** es 'div1', haríamos lo siguiente:

```
01.   // Accedemos al elemento que deseamos borrar:
02.   var objDiv = document.getElementById("div1");
03.
04.   // Obtenemos el nodo padre del elemento:
05.   var objNodoSuperior = objDiv.parentNode;
06.
07.   // Borramos el elemento:
08.   objNodoSuperior.removeChild( objDiv );
```

 **Descargar ejemplo** (./ejemplos/10_acceder_pagina_web/06_documento_dom_eliminar.zip)

Dimensiones de un elemento HTML


Con las propiedades **offsetHeight** y **offsetWidth** podemos obtener el alto y ancho de un elemento HTML (incluidos bordes y relleno, pero no márgenes).

Usamos cookies para ofrecerte una experiencia mejorada, el continuar navegando supone que aceptas su uso

Cerrar

Más información

```
01. <html>
02.
03.     <head>
04.         <title>Curso de JavaScript</title>
05.         <script language="JavaScript">
06.
07.             function cargado()
08.             {
09.                 alert( "offsetHeight: " + document.getElementById("span1").offsetHeight + "\n" +
10.                     "offsetWidth: " + document.getElementById("span1").offsetWidth
11.                 );
12.             }
13.
14.         </script>
15.     </head>
16.
17.     <body onload="cargado();">
18.
19.         <span id="span1" style="background-color:yellow;height:40px;width:100px;">Curso de JavaScript</s
20.
21.     </body>
22.
23. </html>
```


 **Descargar ejemplo** (./ejemplos/10_acceder_pagina_web/07_documento_dom_dimensiones_elemento.zip)




Las dimensiones obtenidas dependerán del navegador web.



offsetWidth no forma parte del estándar W3C, aunque es soportada por la mayoría de los navegadores web más usados.

 **Vea también:** [Dimensiones del Documento (document.php#dimension1)] - [Scroll en una página web (ventana-del-navegador-web.php#scroll)] - [Posición de la ventana (ventana-del-navegador-web.php#posicion)]

 **Vea también:** [Posición de un elemento HTML (document-crear-elementos-dom.php#posicion)]



Posición de un elemento HTML

Usamos cookies para ofrecerte una experiencia mejorada, el continuar navegando supone que aceptas su uso

Cerrar


Más información


Las propiedades **offsetTop** y **offsetLeft** nos permiten obtener la distancia entre los bordes superior e izquierdo de un elemento HTML, con respecto a dichos bordes principales del documento (no del área visible), teniendo en cuenta el scroll realizado.

En el siguiente ejemplo obtenemos la posición del elemento HTML ****:

```
01. <html>
02.
03. <head>
04. <title>Curso de JavaScript</title>
05. <script language="JavaScript">
06.
07.     function cargado()
08.     {
09.         alert( "offsetTop: " + document.getElementById("span1").offsetTop + "\n" +
10.             "offsetLeft: " + document.getElementById("span1").offsetLeft
11.             );
12.     }
13.
14. </script>
15. </head>
16.
17. <body onload="cargado();">
18.
19. <div id="div1" style="background-color:red;
20.     width:300px;
21.     margin-top:80px;
22.     margin-left:100px;
23.     padding-top:0px;
24.     padding-left:0px;">
25.
26. <span id="span1" style="background-color:yellow;
27.     height:40px;
28.     width:100px;">Curso de JavaScript</span>
29.
30. </div>
31.
32. </body>
33.
34. </html>
```

 **Descargar ejemplo** (./ejemplos/10 acceder pagina web/08 documento dom posicion elemento.zip)
Usamos cookies para ofrecerte una experiencia mejorada, el continuar navegando supone que aceptas su uso

 **Vea también:** [Dimensiones del Documento (document.php#dimension1)] - [Scroll en una página web (ventana-del-navegador-web.php#scroll)] - [Posición de la ventana (ventana-del-navegador-web.php#posicion)]

 **Vea también:** [Dimensiones de un elemento (document-crear-elementos-dom.php#dimension2)]

 (index.php)  (document.php)  (formularios.php)  (objetos.php)



Aviso legal y Políticas (<https://informaticapc.com/cookies-y-privacidad.php>)

La mayor parte de los contenidos de esta web se encuentran registrados en la Propiedad Intelectual y bajo licencia Creative Commons: puede distribuirlos, copiarlos y exhibirlos **haciendo clara referencia a esta web**. No puede modificarlos ni obtener beneficio económico directo o indirecto sin nuestro consentimiento.

Usamos cookies para ofrecerte una experiencia mejorada, el continuar navegando supone que aceptas su uso

Cerrar

Más información