

## ¿Qué hace la etiqueta input de HTML y como se utilizan?

La etiqueta “<input>” se utiliza para crear controles interactivos para recepción de datos de usuarios en formularios basados en la web. Esta etiqueta permite obtener información en multitud de formas diferentes, según la naturaleza del dato a obtener.

Para utilizarlo lo podemos poner de la siguiente manera:

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

Donde especificamos que tipo de input es, su id y su nombre, pero para que la información sea procesada es necesario utilizar PHP por ejemplo:

```
<!DOCTYPE html>
<html>
<body>

<form action="/action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit">
</form>

</body>
</html>
```

**Desarrolla en tu página web un ejemplo de un formulario utilizando la etiqueta input.**

**Desarrolla en tu página web un muestrario de al menos 10 tipos de inputs diferentes.**

## ¿Qué es el lenguaje JavaScript?

JavaScript es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas, lo que le permite al usuario tener una mejor experiencia en el sitio web. Como lenguaje de scripting, por parte de un servidor, es una de las principales Tecnologías de la WWW.

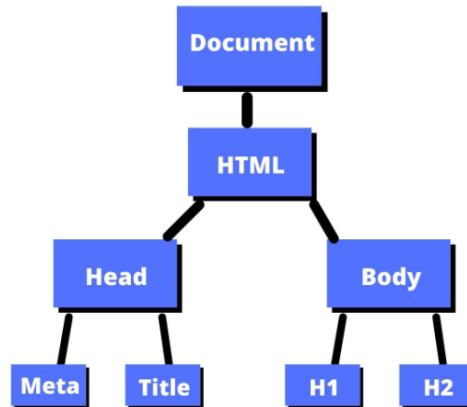
## ¿Para qué sirve el lenguaje JS?

Este lenguaje sirve para que las aplicaciones web sean más dinámicas, ya que los navegadores responden a la interacción de los usuarios, cambiando la distribución del contenido en la página web.

## ¿A qué se refiere el DOM (Document Object model) en el desarrollo Web?

El DOM hace referencia a la interfaz de programación que nos permite tanto crear, cambiar, o remover elementos del documento; como agregar eventos a esos elementos para poder aumentar el dinamismo de nuestra página.

Este modelo visualiza el documento HTML como un árbol de nodos como se muestra a continuación:



Donde, podemos acceder a estos elementos en el documento y hacer cambios a ellos usando JavaScript.

### ¿Cuál es la diferencia entre los archivos HTML, CSS y JS?

La diferencia entre estos tres archivos es la parte de la página web que se desea trabajar, esto lo podemos ver de la siguiente manera:

- HTML: se utiliza para definir el contenido de la página web en general.
- CSS: especifica el diseño de las páginas web.
- JavaScript: se usa para programar el comportamiento de la página

### ¿Para qué sirve la etiqueta <script>?

Un script en este caso lo podemos tomar como un archivo de texto con código Javascript, por lo que, para utilizarlo en una página, debemos indicar en nuestro HTML que vamos a cargar un script por lo que, para hacerlo, utilizamos la etiqueta <script>. Por ejemplo:

```
<script>
  alert('¡Hola!');
</script>
```

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

### ¿En qué consiste los 3 métodos para cambiar elementos tipo Tag Name, ID y Class Name con JS?

- **Por Tag name:** para poder cambiar elementos tipo Tag Name se puede conseguir mandando a llamar el elemento como

“elementos= document.getElementsByTagName(name);”

y sobre esta lista iterar y modificar los elementos como se requiera.

- **Por ID:** Este proceso es similar al anterior, solo que para poder iterar y modificar a nuestra conveniencia el elemento, es necesario llamarlo como

`"elemento = document.getElementById(id);"`

- **Por Class Name:** En caso de querer cambiar elementos del tipo class name, es necesario también mandar a llamar los elementos, solo que en este caso utilizamos

`"var elementos = document.getElementsByClassName(nombres)"`

ó

`"var elementos = elementoRaiz.getElementsByClassName(nombres)"`

**Elabora en Replit un programa básico "Hola mundo" con JS, describe el funcionamiento y elementos utilizados.**

### ¿Cuál es la sintaxis de la programación en JS?

Podemos decir que las normas básicas que definen la sintaxis de JavaScript son las siguientes:

- No se tienen en cuenta los espacios en blanco y las nuevas líneas
- Sí detecta y distingue entre mayúsculas y minúsculas.
- No se define el tipo de variables, no se tiene que indicar el tipo de dato que almacenará.
- No es necesario terminar cada sentencia con el carácter punto y coma.
- Se pueden incluir comentarios utilizando (`//`), o (`/*` y `*/`).

### ¿Como se declaran variables en JS?

Existen 3 maneras de declarar variables en JavaScript, las cuales utilizan una palabra diferente:

- **var:** Ésta es una abreviación de "variable" y es la manera "antigua" de declarar lo que estamos escribiendo, Por ejemplo:
  - `var x=1`

En este caso le estamos asignando un 1 a la variable "x"

- **let:** En este caso la palabra "let" nos permitirá declarar variables en JavaScript que son ciertas a nivel del bloque, que también es conocido como *block scoped*, lo que hace que este tipo de variables sólo sean ciertas dentro de un bloque. Por ejemplo:
  - Si declaramos `var x = 1` & `let y = 1` pero dentro de un bloque queremos realizar un cambio tal que:

```
if (true) {
```

```
    var x = 2;
```

```
    let y = 2;
```

```
}
```

cuando queramos pintar los elementos x e y, obtendremos resultados

diferentes. Cuando pintemos `console.log (x)`, obtendremos el número 2;

mientras que cuando pintemos `console.log (y)`, obtendremos el número 1. Esto

sucede porque la variable `let` que definimos en el bloque anterior solo es verdadera dentro del bloque, por lo que su cambio no se aplica de manera general.

- **const:** Esta instrucción nos permite establecer variables a nivel de bloque, que hace referencia a una constante por lo que para javascript es un tipo de variable inmutable. Ésta se utiliza de la siguiente manera:
  - `const x=5`

### ¿Como se declaran funciones en JS?

Para poder declarar una función, debemos de utilizar la palabra “function”, seguida de:

- El nombre de la función.
- Una lista de parámetros de la función, entre paréntesis y separados por comas.
- Las declaraciones de JS que definen la función, entre llaves.

Por ejemplo:

```
function square(number) {  
    return number * number;  
}
```

### ¿Como se llaman funciones en JS?

Para llamar la función se puede realizar escribiendo el nombre, seguido del argumento entre paréntesis, Por ejemplo:

```
square(5);
```

o

```
console.log(square(5));  
/* ... */  
function square(n) {  
    return n * n;  
}
```

¿Cómo declaras una función interna de un HTML con JS y como declaras un documento externo con JS?

La función interna se puede acceder sólo a partir de sentencias en la función externa. Esta función puede utilizar los argumentos y las variables de la función externa, mientras que la función externa no puede utilizar los argumentos y las variables de la función, esto se hace tal cual, creando una función dentro de otra, por ejemplo:

```
function addCuadrado(a, b) {  
  function cuadrado(x) {  
    return x * x;  
  }  
  return cuadrado(a) + cuadrado(b);  
}  
  
a = addCuadrado(2, 3); // retorna 13  
b = addCuadrado(3, 4); // retorna 25  
c = addCuadrado(4, 5); // retorna 41
```

Una de las maneras en las que podemos insertar JavaScript en HTML, es agregando JavaScript directamente a un archivo HTML utilizando la etiqueta `<script>` `</script>` que deben de envolver a todo el código JavaScript, estas etiquetas se pueden agregar entre las etiquetas de head o de body, en el caso de agregarlo en la sección de head permanecerá separado del contenido del archivo HTML, pero si se coloca dentro de la etiqueta de body ayudará a mejorar la velocidad de carga pues cargará primero el contenido del sitio y solo después de eso se procesará el JavaScript. Por ejemplo:

```
<!DOCTYPE html>  
<html lang="en-US">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  <script>JAVASCRIPT IS USUALLY PLACED HERE</script>  
  <title>Time right now is: </title>  
</head>  
<body>  
  <script>JAVASCRIPT CAN ALSO GO HERE</script>  
</body>  
</html>
```

Otra manera de incorporar el JavaScript es agregando el código a un archivo separado, por lo que, si queremos hacer referencia a él, tenemos que hacer referencia a este archivo, esto lo podemos hacer utilizando la etiqueta de "script" en donde especificamos la ruta del archivo en JS, por ejemplo:

```
<!DOCTYPE html>  
<html lang="en-US">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  <title>Time right now:</title>  
</head>  
<body>  
</body>  
<script src="js/myscript.js"></script>  
</html>
```

## ¿Cuáles ciclos de programación están disponibles en JS?

- For: Este ciclo se repite hasta que una condición especificada se evalúe como falsa, este es muy similar al ciclo for de C y su sintaxis es de esta manera:

```
for ([expresiónInicial]; [expresiónCondicional]; [expresiónDeActualización])  
  instrucción
```

- Do...while: esta instrucción se repite hasta que una condición especificada se evalúe como falsa y se usa de la siguiente manera:

```
do  
  expresión  
while (condición);
```

- While: esta declaración ejecuta sus instrucciones siempre que una condición especificada se evalúe como cierta, esta se utiliza como:

```
while (condición)  
  expresión
```

- For...in: esta instrucción itera una variable especificada sobre todas las propiedades e enumerables de un objeto. Para cada propiedad distinta, JavaScript ejecuta las instrucciones especificadas, esta instrucción la utilizamos de la siguiente manera:

```
for (variable in objeto)  
  instrucción
```

- For...of: esta declaración genera un bucle que se repite sobre objetos iterables, invocando un bucle de iteración personalizado con declaraciones que se ejecutarán para el valor de cada propiedad, en este caso se utiliza de la siguiente manera:

```
for (variable of objeto)  
  expresión
```

## ¿Cuáles funciones de condicional están disponibles en JS?

En Java script los condicionales disponibles son los siguientes:

- If: que especifica que un bloque de código se ejecutará si se cumple una función.

```
if (condition) {  
  // block of code to be executed if the condition is true  
}
```

- Else: la cual especificará que un bloque de código se ejecutará si esta condición mencionada es falsa.

```

if (condition) {
    // block of code to be executed if the condition is true
} else {
    // block of code to be executed if the condition is false
}

```

- }
  - Else if: se utiliza para especificar una nueva condición a probar si la primera condición es falsa.

```

if (condition1) {
    // block of code to be executed if condition1 is true
} else if (condition2) {
    // block of code to be executed if the condition1 is false and condition2 is true
} else {
    // block of code to be executed if the condition1 is false and condition2 is false
}

```

- }
  - Switch: esta condicional toma una sola expresión / valor como una entrada, y luego pasan a través de una serie de opciones hasta que encuentran una que coincida con ese valor, ejecutando el código correspondiente que va junto con ella.

```

switch (new Date().getDay()) {
    case 0:
        day = "Sunday";
        break;
    case 1:
        day = "Monday";
        break;
    case 2:
        day = "Tuesday";
        break;
    case 3:
        day = "Wednesday";
        break;
    case 4:
        day = "Thursday";
        break;
    case 5:
        day = "Friday";
        break;
    case 6:
        day = "Saturday";
    }
}

```

### Referencias:

input - HTML: Lenguaje de etiquetas de hipertexto | MDN. (2023, 27 noviembre). MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/HTML/Element/input>

¿Qué es JavaScript? - Explicación de JavaScript (JS) - AWS. (s. f.). Amazon Web Services, Inc.  
<https://aws.amazon.com/es/what-is/javascript/>

HTML Editors. (s. f.). [https://www.w3schools.com/html/html\\_editors.asp](https://www.w3schools.com/html/html_editors.asp)

Torres, J. T. (2022, 6 enero). ¿Qué es el DOM? El significado del Modelo de Objeto de Documento en JavaScript. freeCodeCamp.org. <https://www.freecodecamp.org/espanol/news/que-es-el-dom-el-significado-del-modelo-de-objeto-de-documento-en-javascript/>

Sintaxis. (s. f.). librosweb.es. Recuperado 29 de marzo de 2024, de  
<https://lenguajehtml.com/html/scripting/etiqueta-html-script/>

KeepCoding, R. (2023, 21 diciembre). Declarar variables en JavaScript | KeepCoding Bootcamps. KeepCoding Bootcamps. [https://keepcoding.io/blog/declarar-variables-en-javascript/#Como\\_declarar\\_variables\\_en\\_JavaScript](https://keepcoding.io/blog/declarar-variables-en-javascript/#Como_declarar_variables_en_JavaScript)

Funciones - JavaScript | MDN. (s. f.). MDN Web Docs.  
<https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Functions>

Funciones - JavaScript | MDN. (s. f.-b). MDN Web Docs.  
<https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Functions>

B, G., & B, G. (2023, 10 agosto). Cómo insertar JavaScript en HTML. Tutoriales Hostinger.  
<https://www.hostinger.mx/tutoriales/insertar-javascript-en-html/>

Bucles e iteración - JavaScript | MDN. (s. f.). MDN Web Docs.  
[https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Loops\\_and\\_iteration](https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Loops_and_iteration)

JavaScript Switch Statement. (s. f.). [https://www.w3schools.com/js/js\\_switch.asp](https://www.w3schools.com/js/js_switch.asp)

Document.getElementsByClassName() - referencia de la API web | MDN. (2023, 24 julio). MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/API/Document/getElementsByClassName>

Document: getElementById() method - Web APIs | MDN. (2023, 21 diciembre). MDN Web Docs.  
<https://developer.mozilla.org/en-US/docs/Web/API/Document/getElementById>

Document: getElementsByTagName() method - Web APIs | MDN. (2023, 2 noviembre). MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/API/Document/getElementsByTagName>