



JavaScript

JS



GOBIERNO DEL PRINCIPADO DE ASTURIAS



2

¿Qué es JavaScript?

JavaScript es un lenguaje de programación para crear interactividad en HTML. El navegador lo ejecuta tras haber cargado el contenido del HTML y el CSS, por lo que su llamada se coloca al final de código HTML, antes del cierre del `</body>`

Puedes crear código JS de dos maneras:

1. **Incrustando** en el propio código HTML a través de una etiqueta

```
<p>&copy; Copyright 2022 - Company Internacional</p>  
</footer>  
<script> // código JS aquí </script>  
</body>  
</html>
```

2. **Enlazando** el código desde una archivo .js externo

```
<p>&copy; Copyright 2022 - Company Internacional</p>  
</footer>  
<script src="archivo.js"></script>  
</body>  
</html>
```

3

4

Comentarios

Se escriben como en CSS y PHP; y permiten añadir comentarios, o desactivar temporalmente fragmentos de código:

// comentario de linea

/*

Comentario multi linea

0 de párrafo

***/**

Recordatorio:

En HTML los comentarios se hacían de la siguiente manera:

<!-- comentario -->

5

Mostrar en pantalla: **alert()**;

Hay varias formas de sacar por pantalla el contenido que sea mostrados; contenidos como textos, variables, etc...

Alert es uno de los más sencillos.

Al utilizar Alert, se nos muestra un aviso en pantalla sobreimpreso en el navegador.

```
alert("Hola Mundo");
```

```
// otro ejemplo:
```

```
var nombre="Felipe";
```

```
alert(nombre);
```

6

Mostrar en pantalla: **document.write();**

Añade contenido al HTML

```
document.write("Hola Mundo");
```

```
// otro ejemplo:
```

```
var nombre="Felipe";  
document.write(nombre);
```

7

Mostrar en pantalla: **document.body();**

Añade a la etiqueta <body>, que sumado por ejemplo a:

innerHTML permite añadir texto en formato HTML

innerText permite añadir una cadena de caracteres.

```
document.body.innerHTML="<h1>Hola Mundo</h1>";
```

```
// otro ejemplo:
```

```
var nombre="Felipe";
```

```
document.body.innerText=nombre;
```

Práctico:

innerHTML reemplaza todo el contenido de una etiqueta por el nuevo. Así que `body.innerHTML` borrará todo el body si contaba con algún elemento previo.

Una solución es utilizar `append`, que añade el contenido al final, repitiendo lo preexistente.:

```
document.body.append("Hello World");
```

8

Mostrar en consola: **console.log()**;

El siguiente contenido aparece en la consola del navegador; un sitio oculto or defecto a los usuarios. Este metido nos puede servir para enviarnos pistas, como que ha funcionado un fragmento de código o ha llegado o se ha ejecutado hasta cierto punto. Para saber el valor de una variable que no se muestra en pantalla u otros casos.

```
console.log("Hola Mundo");
```

```
// otro ejemplo:
```

```
var nombre="Felipe";  
console.log(nombre);
```

Recordatorio:

Para sacar la consola en navegadores como Google Chrome:

Botón derecho sobre la página / Inspeccionar / Consola

9

Sacar por pantalla: **getElementById();**

getElementById(); es una función que selecciona el contenido de una etiqueta de HTML previamente nombrada con el ID que estemos especificando.

En el siguiente ejemplo **titulillo** que más adelante gracias a innerHTML sutituimos su contenido (el contenido de la etiqueta)

HTML

```
<h1 id="titulillo"></h1>
```

JS

```
document.getElementById('titulillo').innerHTML='Hola Mundo';
```

```
// otro ejemplo:
```

```
var nombre="Felipe";
```

```
document.getElementById('titulillo').innerHTML=nombre;
```

10

Sacar por pantalla: **querySelector()**;

querySelector(); es una evolución de getElementById, permitiéndonos seleccionar la primera etiqueta, clases, o id que encuentra

HTML

```
<h1></h1>  
<h2 id="subtitulo"></h2>  
<p class="texto"></p>
```

JS

```
document.querySelector('h1').innerHTML='El Quijote';  
document.querySelector('#subtitulo').innerHTML='M. de Cervant';
```

// otra manera de hacerlo mismo:

```
var texto = document.querySelector('.texto');  
texto.innerHTML='En un lugar de la Mancha';
```

11

if(){...} else{...}

if() es un elemento que compara un valor

HTML

```
<h1 id="mensaje"></h1>
```

JS

```
var nombre="Felipe";
```

```
if( nombre=="Felipe"){  
document.getElementById('mensaje').innerHTML="Bienvenido Jefe";  
}
```

```
else{  
document.getElementById('mensaje').innerHTML="¿Quién eres?";  
}
```

12

if(){...} else if(){...} else{...}

if() es un elemento que compara un valor

HTML <h1 id="mensaje"></h1>

JS

```
var nombre="Felipe";
```

```
if( nombre=="Felipe"){  
document.getElementById('mensaje').innerHTML="Bienvenido Jefe";  
}
```

```
else if( nombre=="María"){  
document.getElementById('mensaje').innerHTML="Hola Jefa";  
}
```

```
else{  
document.getElementById('mensaje').innerHTML="Quién eres";  
}
```

13

if(){...} anidados

if() es un elemento que compara un valor

JS

```
var nombre="Rosa";
```

```
var edad=55;
```

```
if( nombre=="Rosa"){  
    if( edad>=18){  
        alert("Hola Rosa, eres adulta");  
    }  
    else{  
        alert("Hola Rosa, aún eres menor");  
    }  
}  
else{  
    alert("No sé quien eres");  
}
```

14

if(){...} comprobación multiple

&& - se cumplen varias condiciones (y): `nombre=="Rosa" && edad>=18`

|| - se cumple una u otra (o): `nombre=="Rosa" || nombre=="Pepe"`

JS

```
var nombre="Rosa";
```

```
var edad=55;
```

```
if( nombre=="Rosa" && edad>=18){  
    alert("Hola Rosa, eres adulta");  
}  
else if(nombre=="Rosa" && edad!=18){  
    alert("Hola Rosa, aún eres menor");  
}  
else{  
    alert("No sé quien eres");  
}
```

15

function(){...}

Función es un conjunto de instrucciones empaquetadas que puede utilizarse varias veces.

JS

// definimos la función

```
function saludar(){  
    document.write("Hola Felipe");  
}
```

// llamamos a la función

```
saludar();
```

16

Llamar a una `function(){...}` desde HTML

Ahora vamos a llamar a la función cada vez que pulsemos sobre una etiqueta, y esta cambiará.

HTML

```
<h1 id="mensaje" onclick="saludar();" >Púlsame</h1>
```

JS

```
// definimos la función
```

```
function saludar(){  
    document.getElementById('mensaje').innerHTML="Hola Felipe";  
}
```


17

function(con_un_parámetro);

Parámetros son valores que utiliza la función para desarrollar sus instrucciones

JS

```
function saludar(nombre){  
    document.write("Hola" + nombre);  
}
```

// llamamos a la función

```
saludar("Felipe");  
saludar("María");  
saludar("9");
```

```
// Hola Felipe
```

18

function(con_varios_parámetros);

Parámetros son valores que utiliza la función para desarrollar sus instrucciones

JS

```
function saludar(nombre, apellido, edad){  
    document.write("<p>Hola " + nombre + " " + apellido + " tienes " + edad + " años.</p>");  
}
```

```
/* llamamos a la función  
 * (es muy importante el orden en el que le pasemos  
 * los valores del atributo: siempre nombre, apellidos, edad  
 */
```

```
saludar("Felipe", "Pérez", "47");  
saludar("María", "Suárez", "33");  
saludar("Jorge", "Vautista", "5");
```

```
// <p>Hola Felipe Pérez tienes 47 años</p>
```

19

Cambiar el Color de Fondo de la página

El siguiente ejemplo cambia el color de fondo de la web a rojo

HTML

```
<button onclick="cambiar();">Cambiar Color</button>
```

JS

```
// definimos la función
```

```
function cambiar(){  
    document.body.style.background="red";  
}
```

20

Cambiar Fondo con parámetros

El siguiente ejemplo permite cambiar el color de fondo de una página utilizando la misma función, a la que se le pasan diferentes parámetros. En este caso se utilizan unos parámetros que hacen referencia al color, de tres formas diferentes.

HTML

```
<button onclick="cambiar('red');">Rojo</button>  
<button onclick="cambiar('#00FF00');">Verde</button>  
<button onclick="cambiar('rgb(0,0,255)');">Azul</button>
```

JS

```
// definimos la función  
function cambiar(color){  
    document.body.style.background=color;  
}
```

Atención:

En el caso de que contemos con comillas dentro de comillas deberemos alternarlas entre simple y dobles:

```
onclick="cambiar('red');"
```

21

Bucle For()

El bucle for nos permite repetir una acción un número de veces. Esa cantidad de veces se ve recogida habitualmente en la variable "i", se llama i por iteración. Una iteración es cada uno de las repeticiones realizadas.

JS

```
var almacen = 'hola';

// bucle hola
for(var i = 0; i<=15; i++){

    almacen += "hola " + i;

}

document.write(almacen);
```

22

Bucle For() para recorrer un Array 1

JS

```
let alumnos = ['Fran', 'Lola', 'Irene', 'Miguel', 'Adrian', 'Celia']  
let almacen = '';
```

```
// bucle  
for(var i = 0; i<=5; i++){  
    almacen += "<li>" + alumnos[i] + "</li>";  
}
```

```
document.write(almacen);
```

23

Bucle For() para recorrer un Array 2

length cuenta la cantidad de elementos que tiene el array eh "humano", es decir, no empieza contado por 0. Si necesitamos utilizar length como recurso para ir enumerando los elementos del array (para que empiece por 0 y acabe por el número total de elementos menos uno) utilizaremos **length-1**

JS

```
let alumnos = ['Fran', 'Lola', 'Irene', 'Miguel', 'Adrian', 'Celia']  
let almacen = '';
```

```
// bucle  
for(var i = 0; i<=alumnos.length-1; i++){  
  
    almacen += "<li>" + alumnos[i] + "</li>";  
  
}
```

```
document.write(almacen);
```

Truco:

En la siguiente página lo usaremos:

Si ponemos en el for < en lugar de <=
`i<peliculas.length`

Si será necesario hacer un **length-1**

24

Bucle For() cartelera mezcla 2 arrays

Utilizaremos 2 arrays: uno para los títulos de las películas y otro para cargar las imágenes de cada cartel.

JS

```
// nombre de las películas
let titulos=['Titanic','Avatar','007','Batman']

// ruta de las imágenes (en este ejemplo no concuerdan con la película)
let carteles =
['https://picsum.photos/120','https://picsum.photos/532','https://picsum.photos/627','https://picsum.photos/427']

let almacen ='<ul>';

// bucle
for(var i = 0; i<peliculas.length; i++){

    almacen += "<li>";
    almacen += "<h2>" + titulos[i] + "</h2>";
    almacen += "<img src='"+carteles[i]+'>";
    almacen += "</li>";

}
almacen += "</ul>"
document.write(almacen);
```


25

Bucle For() cartelera array Multidimens.

Evolucionado el código anterior: un **Array Multidimensional** es un (o más) array dentro de un array.. En este caso para hacer referencia a un elemento nombramos el array padre, luego el hijo, y luego la posición. Ej: **peliculas[1][1]** en el caso del bucle: **peliculas[i][1]**

JS

```
// nombre de los proyectos
let peliculas=[
    ['Titanic', 'https://picsum.photos/120'] // 0 (nº peli)
    ['Avatar', 'https://picsum.photos/532', // 1 (nº peli)
    ['007', 'https://picsum.photos/627'] // 2 (nº peli)
    ['Batman', 'https://picsum.photos/427'] // 3 (nº peli)
    // 0 , 1 (nº dato)
];

// bucle
for(var i = 0; i<peliculas.length; i++){

    almacen += "<li>";
    almacen += "<h2>" + peliculas[i][0] + "</h2>"; // peliculas[pele][dato]
    almacen += "<img src='"+peliculas[i][1]+"'>";
    almacen += "</li>";

}

document.write(almacen);
```

26

Bucle For() cartelera con Objeto.

Este Array con formato de Objeto nos permite insertar la información desordenada y que haremos referencia la nombre de cada campo en lugar de a la posición de ocupa (como en casos anteriores)

JS

```
// nombre de los proyectos
let peliculas=[
  { "titulo": "Titanic", "imagen": "https://picsum.photos/120" },
  { "titulo": "Avatar", "imagen": "https://picsum.photos/532" },
  { "imagen": "https://picsum.photos/627" "titulo": "007", },
  { "titulo": "Batman", "imagen": "https://picsum.photos/427" }
];

// bucle
for(var i = 0; i<peliculas.length; i++){

  almacen += "<li>";
  almacen += "<h2>" + peliculas[i][titulo] + "</h2>";
  almacen += "<img src='"+peliculas[i][imagen]+"'>"; //alt='"+peliculas[i]+"'"

  almacen += "</li>";

}

document.write(almacen);
```

27

Otros bucles (for, while, do while)

Además del bucle FOR existen otros bucles como:

JS

```
// bucle For
for(i = 1; i <= 5; i++){
  console.log(i);
}
```

let i = 1: Inicializa el contador.
i <= 5: Condición para continuar el bucle.
i++: Incrementa el contador en cada iteración.

```
// bucle While
let i = 1;
while (i <= 5) {
  console.log(i);
  i++; // Incrementa el contador
}
```

El bucle continúa mientras la condición `i <= 5` sea verdadera

```
// bucle do While
let i = 1;
do {
  console.log(i);
  i++;
}
while (i <= 5);
```

28

más bucles para arrays

Además del bucle FOR existen otros bucles como:

JS

```
// bucle for...of para recorrer un array  
const colores = ['Rojo', 'Verde', 'Azul'];
```

```
for (const color of colores) {  
    console.log(color);  
}
```

```
// bucle for...in para recorrer un objeto  
const persona = { nombre: 'Juan', edad: 30, ciudad: 'Madrid' };
```

```
for (const clave in persona) {  
    console.log(`${clave}: ${persona[clave]}`);  
}
```

29

For each

Además del bucle FOR existen otros bucles como:

JS

```
// bucle for...of para recorrer un array  
const numeros = [1, 2, 3, 4, 5];
```

```
numeros.forEach((numero) => {  
    console.log(numero);  
});
```

```
const frutas = ['Manzana', 'Banana', 'Cereza'];
```

```
frutas.forEach((fruta, indice) => {  
    console.log(`Fruta ${indice + 1}: ${fruta}`);  
});
```

```
/*  
Fruta 1: Manzana  
Fruta 2: Banana  
Fruta 3: Cereza  
*/
```

30

Switch

Además del bucle FOR existen otros bucles como:

JS

```
const dia = 3;

switch (dia) {
  case 1:
    console.log("Lunes");
    break;
  case 2:
    console.log("Martes");
    break;
  case 3:
    console.log("Miércoles");
    break;
  case 4:
    console.log("Jueves");
    break;
  case 5:
    console.log("Viernes");
    break;
  default:
    console.log("Fin de semana o día inválido");
}
```

31

<https://codepen.io/ksizorCode/pen/JjaPLRW>

32

CSS

```
body{
    font-family: sans-serif;
    max-width:670px;
    margin:0 auto;
    background:coral;
}

ul{
    display:flex;
    gap:10px;
    flex-wrap:wrap;

    padding-left: 0;
    list-style:none;
}

li{ list-style:none; border-radius:20px; background: white; padding:5px;
    width:180px; text-align: center;
    box-shadow:3px 3px 18px;
}

li img{ width:170px; height:220px; object-fit:cover; border-radius:14px;}
```


33

prompt();

Sirve para insertar valores que se pueden asignar a una variable

JS

```
var nombre= prompt("¿Cuál es tu nombre?");
```

```
document.write("Tu nombre es "+nombre);
```

34

Método GET: Obtener datos via URL

A través de una URL se pueden pasar uno o varios datos con los que la web puede trabajar. La URL tendrá un formato similar a este: `index.html?titulo=Hola%20Mundo`

`http://www.midominio.com/index.html?titulo=Hola%20Mundo`

HTML

```
<h1></h1>
```

JS

```
// Obtiene el texto del parámetro "titulo" en la URL
const titulo = new URLSearchParams(window.location.search).get("titulo");

// Si hay un título en la URL, lo escribe como H1
if (titulo) {
  document.querySelector('h1').textContent = decodeURIComponent(titulo);
}
```

35

[https://www.youtube.com
/watch?v=LB0hVng5rk8](https://www.youtube.com/watch?v=LB0hVng5rk8)

36

https://www.w3schools.com/css/css_rwd_grid.asp

<https://code.visualstudio.com/docs/editor/emmet>

<https://code.visualstudio.com/docs/editor/userdefinedsnippets>

<http://diana-adrienne.com/>