

# JavaScript

## 📌 Objetivo de la clase

- Introducir el lenguaje JavaScript básico.
  - Dar herramientas para agregar **funcionalidad a páginas web**.
  - Servir como base para validar formularios y otras interacciones.
- 

## 📘 ¿Qué es JavaScript?

- Lenguaje de scripting **interpretado** (no necesita compilación).
  - Diseñado por Netscape para dar **interactividad a páginas HTML**.
  - No es Java.
  - Corre del lado del cliente (en el navegador).
  - No requiere licencia para usar.
- 

## 📦 Estructura del Código JS

- Puede estar embebido en HTML usando `<script>`:

```
html CopiarEditar <script type="text/javascript"> document.write("Hola  
Mundo"); </script>
```

- Puede ubicarse en:
    - El `<head>` (funciones).
    - El `<body>` (ejecución inmediata).
    - Archivos externos `.js`.
  - Se pueden usar varios `<script>` por página.
  - `<noscript>` muestra un mensaje si JS está deshabilitado.
- 

## 💡 Usos comunes de JavaScript

- Validar formularios.
- Capturar eventos (clics, entradas de teclado, etc.).

- Modificar elementos HTML dinámicamente.
  - Detectar el navegador.
  - Ejecutar en segundo plano (Ajax).
- 

## Sentencias y Comentarios

- Una sentencia JS es una instrucción para el navegador.
  - Se pueden terminar con `;` (buena práctica).
  - Comentarios:
    - Línea: `// comentario`
    - Bloque: `/* comentario */`
- 

## Variables

- Declaración: `var nombreVariable;`
- Tipado dinámico: una misma variable puede tener distintos tipos.
- Ámbito:
  - Global (en toda la página).
  - Local (dentro de funciones).

### Tipos comunes:

- Números: `var n = 1;`
  - Strings: `var s = "texto";`
  - Arrays: `var dias = ["Lunes", "Martes"];`
  - Booleanos: `var b = true;`
- 

## Concatenación y Conversión

- Se usa `+` para concatenar strings.
  - `"5" + 1 → "51"`
  - `parseInt("5") + 1 → 6`
- Caracteres especiales en strings:
  - `\n` nueva línea, `\t` tab, `\'` comilla simple, `\\"` comilla doble.

## Operadores

### Aritméticos:

Operador	Ejemplo	Resultado
 suma	<code>x = y + 2</code>	<code>x = 7</code>
 multiplicación	...	...

### Asignación:

Operador	Ejemplo	Equivalente
<code>+=</code>	<code>x += y</code>	<code>x = x + y</code>

### Comparación:

Operador	Significado
<code>==</code>	igual en valor
<code>===</code>	igual en valor y tipo
<code>!=</code>	distinto
<code>&lt;, &gt;, &lt;=, &gt;=</code>	comparaciones numéricas

### Lógicos:

Operador	Descripción
<code>&amp;&amp;</code>	AND
<code>,</code>	
<code>!</code>	NOT

## Cuadros de diálogo (Pop-up)

- `alert("Hola")` → muestra un mensaje.
- `prompt("Ingresá tu nombre", "Juan")` → solicita un valor.

- `confirm("¿Estás seguro?")` → true o false según elección.
- 

## Recursos y enlaces

- [W3Schools JS](#)
- Libro online - Introducción a JavaScript (PDF)
- [ECMA estándar](#)

## TP1 JS

### ◆ FUNCIONES EN JAVASCRIPT

- **Definición:** Bloques de código reutilizable.
- **Sintaxis:**

```
javascript CopiarEditar function nombre_funcion(par1, par2) { // instrucciones return valor; }
```

- **Acceso:** A variables locales, parámetros y globales.
  - **Ubicación:** Puede estar en el `<head>` o archivo `.js`.
- 

### ◆ SENTENCIAS CONDICIONALES

#### `if - else`

```
javascript CopiarEditar if (condición) { // si se cumple } else { // si no se cumple }
```

#### `switch`

- Se usa para comparar un valor específico.
- Importante usar `break` para evitar que continúe con otros `case`.

```
javascript CopiarEditar switch (n) { case 1: // instrucciones break; case 2: // instrucciones break; default: // por defecto }
```

## ◆ BUCLES

- `for` : con índice.
- `while` / `do-while` : según condición.
- `for-in` : recorre propiedades de un objeto o índices de un array.

```
javascript CopiarEditar for (var i = 0; i < 10; i++) { ... } while (condición) { ... } do { ... } while (condición); for (prop in objeto) { ... }
```

## `break` y `continue`

- `break` : interrumpe el bucle.
- `continue` : salta a la siguiente iteración.

## ◆ EVENTOS

Permiten capturar interacciones del usuario:

### En elementos:

- `onclick` , `ondblclick` , `onmouseover` , `onmouseout` , `onmousemove` , `onmousedown` , `onmouseup`

### En teclado:

- `onkeydown` , `onkeypress` , `onkeyup`

### En formulario:

- `onfocus` , `onblur` , `onchange` , `onselect` , `onsubmit` , `onreset`

### En `body` :

- `onload` , `onunload` , `onresize`

## ◆ OBJETOS ÚTILES

### `navigator`

- `appName` , `userAgent` , `appVersion` , `systemLanguage`

### `window`

- Métodos: `alert()`, `prompt()`, `confirm()`, `resizeTo()`, `moveTo()`, `close()`, `open(url)`
  - Sub-objetos:
    - `window.history`: `.back()`, `.forward()`, `.go(n)`
    - `window.location`: `.reload()`, `.replace(url)`
- 

## ◆ DOM (Document Object Model)

- Representa el HTML como un árbol de nodos.

### Tipos de nodos:

- `Document`, `Element`, `Attr`, `Text`, `Comment`

### Acceso a elementos:

```
javascript CopiarEditar document.getElementById("id"); document.getElements  
sByTagName("tag"); document.getElementsByName("name");
```

## ◆ RECURSOS RECOMENDADOS

- [W3Schools JavaScript](#)
- Libro gratuito en PDF
- [ECMAScript Standard](#)

### TP2 JS

## ◆ FUNCIONES EN JAVASCRIPT

- Se definen con `function nombre_funcion(parámetros)`.
  - Usan `return` para devolver un valor.
  - Tienen acceso a:
    - Parámetros
    - Variables locales
    - Variables globales
  - Se suelen incluir en archivos externos o en `<head>`.
-

## ◆ SENTENCIAS CONDICIONALES

### if - else

```
js CopiarEditar if (condición) { // bloque si verdadero } else { // bloque si falso }
```

### switch

- Evalúa un valor contra múltiples casos.
- `break` evita que se ejecuten los siguientes `case`.

```
js CopiarEditar switch(n) { case 1: // código break; case 2: // código break; default: // código si no coincide ningún caso }
```

## ◆ BUCLES (LOOPS)

### for

```
js CopiarEditar for (var i = 0; i < n; i++) { // código }
```

### while

```
js CopiarEditar while (condición) { // código }
```

### do-while

```
js CopiarEditar do { // código } while (condición);
```

### for-in

- Itera sobre propiedades de objetos o índices de arrays.

```
js CopiarEditar for (var clave in objeto) { // código }
```

### **break** y **continue**

- **break** : interrumpe el bucle.
- **continue** : salta al siguiente ciclo sin ejecutar el resto del código actual.

## ◆ EVENTOS EN JAVASCRIPT

### Básico

```
html CopiarEditar <button onclick="accion()>Click</button>
```

### En **<body>**

- **onload** , **onunload** , **onresize**

### De teclado

- **onkeydown** , **onkeypress** , **onkeyup**

### De mouse

- **onclick** , **ondblclick** , **onmousedown** , **onmousemove** , **onmouseout** , **onmouseover** , **onmouseup**

### Formularios

- **onblur** , **onchange** , **onfocus** , **onreset** , **onsubmit** , **onselect**

## ◆ OBJETOS ÚTILES

### **navigator**

- **.appName** , **.userAgent** , **.appVersion** , **.systemLanguage**

### **window**

- Métodos: `alert()`, `prompt()`, `confirm()`, `resizeTo()`, `moveTo()`, `close()`, `open()`

### `window.history`

- `.length`, `back()`, `forward()`, `go(n)`

### `window.location`

- `.reload()`, `.replace(URL)`

## ◆ DOM (DOCUMENT OBJECT MODEL)

### Concepto

- Representa el documento HTML como un **árbol de nodos**.

### Tipos de Nodos

- `Document`, `Element`, `Attr`, `Text`, `Comment`

### Acceso a elementos

```
js CopiarEditar document.getElementById("id") document.getElementsByName("name") document.getElementsByTagName("tag")
```

## ◆ REFERENCIAS Y LINKS ÚTILES

- [W3Schools JavaScript](#)
- Tutorial completo en Wikilearning (antiguo)
- [Especificación oficial ECMAScript \(ECMA-262\)](#)
- Libro gratuito: *Introducción a JavaScript* por Javier Eguíluz – Descargar PDF

## ◆ Conceptos Básicos de Objetos en JavaScript

- JavaScript es un lenguaje **orientado a objetos**.
- Un **objeto** tiene propiedades (datos) y métodos (funciones).

- Acceso:
    - `objeto.propiedad`
    - `objeto.metodo()`
- 

## ◆ Objeto String

- Creación: `var a = "cadena"` o `var a = new String("cadena")`

- Propiedades:

- `length` → largo del string.

- Métodos comunes:

- `indexOf("texto")` → posición del texto o `1` si no lo encuentra.
    - `replace("a", "b")` → reemplaza.
    - `substr(inicio, largo)` → substring.
    - `toUpperCase()` / `toLowerCase()`
    - `a[pos]` → carácter en la posición `pos`.
- 

## ◆ Objeto Array

- Creación: `var A = new Array()` o `A = ["uno", "dos"]`

- Propiedades:

- `length` → cantidad de elementos.

- Métodos comunes:

- `concat(otroArray)`
    - `join("|")` → une con separador.
- 

## ◆ Objeto Date

- Creación: `var d = new Date()`

- Getters:

- `getFullYear()`, `getMonth()`, `getDate()`, `getDay()`
  - `getHours()`, `getMinutes()`, `getSeconds()`

- Setters:

- `setFullYear()`, `setMonth()`,  `setDate()`, etc.
-

## ◆ Objeto Math

- Propiedades:

- `Math.PI`

- Métodos:

- `Math.random()`
    - `Math.pow(x, y)`
    - `Math.sqrt(x)`
    - `Math.round(x)`
- 

## ◆ Objeto Document

- Permite interactuar con el HTML

- Propiedades:

- `document.body` , `document.title`

- Métodos importantes:

- `document.write()`
    - `getElementById("id")`
    - `getElementsByTagName("tag")`
    - `getElementsByName("name")`
- 

## ◆ Objetos de Formulario

### ✓ Input Text:

- `value` , `size` , `tabIndex`
- Métodos: `focus()` , `blur()` , `select()`

### ✓ Textarea:

- Igual que Input, más: `rows` , `cols`

### ✓ Checkbox & Radio:

- Propiedad principal: `checked` (booleano)

### ✓ Select:

- `options[]`, `selectedIndex`, `length`
  - Métodos: `focus()`, `blur()`, `select()`
- 

## ◆ Elementos div y span

- `innerHTML` → contenido del elemento.
- `style` → acceso y modificación de estilos CSS.

```
js CopiarEditar document.getElementById("id").style.color = "red"; document.body.style.backgroundColor = "#FFCC80";
```

## Fuentes útiles

- [W3Schools JS](#)
- [Tutorial completo - Wikilearning](#)
- [Libro de Javier Eguíluz Pérez \(PDF\)](#)

## TP3 JS

 **Código completo (Ejercicios 9, 11 y 12 en una sola página):**

```
html CopiarEditar <!DOCTYPE html> <html lang="es"> <head> <meta charset="UTF-8"> <title>Práctico 3 - JavaScript</title> <style> body { font-family: Arial; margin: 20px; } section { margin-bottom: 40px; } textarea { width: 100%; } #chatBox { border: 1px solid #ccc; padding: 10px; height: 200px; overflow-y: auto; } </style> </head> <body> <h1>Práctico 3 - JavaScript</h1> <!-- Ejercicio 9 --> <section> <h2>Ejercicio 9: Manipulación de Strings</h2> <input type="text" id="inputMensaje" placeholder="Ingrese un mensaje..." size="60"> <button onclick="procesarMensaje()">Procesar</button> <div id="resultados9"></div> </section> <!-- Ejercicio 11 --> <section> <h2>Ejercicio 11: Fechas y Edad</h2> <h3>Parte a)</h3> <p><strong>Fecha y hora actual:</strong> <span id="fechaActual"></span></p> <p><strong>Tiempo hasta el parcial:</strong> <span id="tiempoRestante"></span></p> <h3>Parte b)</h3> <button onclick="calcularEdad()">Calcular Edad</button> <p id="resultadoEdad"></p> </section> <!-- Ejercicio 12 --> <section> <h2>Ejercicio 12: Chat</h2> <form onsubmit="enviarMensaje(); return false;"> <label>Usuario: <input type="text" id="nombreUsuario" required></label><br><br> <textarea id="mensajeChat" rows="3" placeholder="Escriba su mensaje..." required autofocus></textarea><br> <button type="submit">Enviar mensaje</button> </form> <h3>Mensajes:</h3> <div id="chatBox"></div> </section> <script> // Ejercicio 9 function procesarMensaje() { const mensaje = document.getElementById("inputMensaje").value; const resultado = document.getElementById("resultados9"); let texto = ""; texto += `<p><strong>Mayúsculas:</strong> ${mensaje.toUpperCase()}</p>`; texto += `<p><strong>Minúsculas:</strong> ${mensaje.toLowerCase()}</p>`; texto += `<p><strong>Substring (posición 5, largo 10):</strong> ${mensaje.substr(5, 10)}</p>`; if (mensaje.toLowerCase().includes("hola")) { texto += `<p><strong>Contiene "hola":</strong> hola que tal</p>`;} const sinEspacios = mensaje.replace(/\s+/g, ""); const sinHola = mensaje.replace(/hola/gi, ""); texto += `<p><strong>Sin espacios y sin "hola":</strong> ${sinHola.replace(/\s+/g, "")}</p>`; // Traducción const traducciones = { "my": "mi", "hello": "hola", "name": "nombre", "is": "es", "good": "bueno" // Agregá más si querés }; let traducido = mensaje; for (let palabra in traducciones) { const regex = new RegExp(`\\b${palabra} + \\b`, "gi"); traducido = traducido.replace(regex, traducciones[palabra]); } texto += `<p><strong>Traducción:</strong> ${traducido}</p>`; resultado.innerHTML = texto; } // Ejercicio 11 - Fecha actual y cuenta regresiva function mostrarFechaYTiempo() { const ahora = new Date(); document.getElementById("fechaActual").textContent = ahora.toLocaleString(); const parcial = new Date("2025-06-25T10:00:00"); // Cambiar fecha del parcial si hace falta const diferencia = parcial - ahora; if (diferencia > 0) { const dias = Math.floor(diferencia / (1000 * 60 * 60 * 24)); const horas = Math.floor((diferencia / (1000 * 60 * 60)) % 24); const minutos = Math.floor((diferencia / (1000 * 60)) % 60); const segundos = Math.floor((diferencia / 1000) % 60); document.getElementById("tiempoRestante").textContent = `${dias} días, ${horas} hs, ${minutos} min, ${segundos} seg`;} else { document.getElementById("tiempoRestante").textContent = "¡El parcial ya pasó!"}} setInterval(mostrarFechaYTiempo, 1000); // Ejercicio 11 - Edad function calcularEdad() { const nombre = document.getElementById("nombreUsuario").value; const edad = document.getElementById("edad").value; const resultado = document.getElementById("resultadoEdad"); const resultadoDiv = document.getElementById("resultados9"); const resultadoP = document.createElement("p"); resultadoP.textContent = `Tu nombre es ${nombre} y tienes ${edad} años.`; resultadoDiv.appendChild(resultadoP); resultado.innerHTML = `Tu edad es ${edad}`; }
```

```
arEdad() { const año = prompt("Ingrese su año de nacimiento:"); const actual = new Date().getFullYear(); const edad = actual - parseInt(año); document.getElementById("resultadoEdad").textContent = `Su edad es ${edad}`; }
// Ejercicio 12 - Chat function enviarMensaje() { const user = document.getElementById("nombreUsuario").value.trim(); const msg = document.getElementById("mensajeChat").value.trim(); if (user && msg) { const chat = document.getElementById("chatBox"); chat.innerHTML += `<p><strong>${user}</strong>: ${msg}</p>`; document.getElementById("mensajeChat").value = ""; document.getElementById("mensajeChat").focus(); } } </script> </body> </html>
```

## ✓ ¿Qué hace esta página?

- **Ejercicio 9 (Strings):**
  - Convierte a mayúsculas, minúsculas.
  - Extrae subcadena.
  - Verifica si contiene "hola".
  - Elimina espacios y la palabra "hola".
  - Traduce palabras clave al español.
- **Ejercicio 11 (Fechas):**
  - Muestra la fecha y hora actual en vivo.
  - Cuenta regresiva para el parcial.
  - Calcula la edad con `prompt`.
- **Ejercicio 12 (Formulario tipo Chat):**
  - Formulario para ingresar nombre y mensaje.
  - Muestra el chat en pantalla como una conversación.
  - Limpia el textarea y posiciona el cursor automáticamente.

## TP4 JS

### Resumen y aspectos clave que encontraste:

- **LocalStorage y SessionStorage** son objetos del navegador para almacenar datos en clave/valor (strings).
- **LocalStorage**: Persiste indefinidamente, aunque cierres el navegador.
- **SessionStorage**: Persiste solo durante la sesión actual (se borra al cerrar el navegador).

- Ambos pueden almacenar entre 5MB y 10MB, mucho más que las cookies (4KB).
  - No envían la información al servidor con cada petición (a diferencia de las cookies).
  - Los datos están almacenados en el cliente, por dominio.
  - Solo almacenan strings, pero se puede usar JSON para guardar objetos o arreglos.
  - Métodos importantes:
    - `setItem(clave, valor)`
    - `getItem(clave)`
    - `removeItem(clave)`
    - `clear()`
    - `key(index)`
    - `length`
  - No son seguros para datos confidenciales.
  - No son iterables directamente, pero se puede iterar con `Object.keys()` o usando índices con `key(index)`.
- 

**¿Querés que te arme ejemplos prácticos con código para que los uses y entiendas mejor cómo funcionan?**

Por ejemplo:

- Cómo guardar y recuperar datos con LocalStorage
  - Cómo guardar objetos usando JSON
  - Cómo limpiar el almacenamiento
  - Cómo listar todas las claves almacenadas
-