

MATERIAL DE CONSULTA - EXAMEN 1ª EVALUACIÓN DWEC

Desarrollo Web en Entorno Cliente | 27 de noviembre de 2025

IMPORTANTE: Este material contiene únicamente referencia técnica de métodos y propiedades menos frecuentes. La sintaxis básica de JavaScript (variables, funciones, condicionales, bucles, DOM básico, eventos, localStorage básico) debe conocerse sin consulta de material.

1. MÉTODOS DE ARRAY AVANZADOS

arr.splice(inicio, cantidad, ...elementos)

- Elimina elementos y/o inserta nuevos
- Modifica el array original
- Devuelve array con elementos eliminados
- Ejemplo: arr.splice(2, 1) elimina 1 elemento desde índice 2

arr.unshift(elemento)

- Añade elemento(s) al INICIO del array
- Modifica el array original
- Devuelve la nueva longitud

arr.reverse()

- Invierte el orden de los elementos
- Modifica el array original

arr.some(callback)

- Devuelve true si AL MENOS UN elemento cumple la condición
- No modifica el array original

arr.every(callback)

- Devuelve true si TODOS los elementos cumplen la condición
- No modifica el array original

arr.lastIndexOf(elemento)

- Devuelve la ÚLTIMA posición donde aparece el elemento
- Devuelve -1 si no lo encuentra

2. MÉTODOS DE STRING POCO FRECUENTES

str.slice(inicio, fin)

- Extrae una porción del string
- No modifica el string original
- Acepta índices negativos (desde el final)

str.substring(inicio, fin)

- Similar a slice() pero NO acepta índices negativos
- No modifica el string original

str.lastIndexOf(texto)

- Devuelve la ÚLTIMA posición donde aparece el texto
- Devuelve -1 si no lo encuentra

str.repeat(n)

- Repite el string n veces
- Ejemplo: "hola".repeat(3) → "holaholahola"

str.padStart(longitud, relleno) / str.padEnd(longitud, relleno)

- Rellena el string hasta alcanzar la longitud indicada
- Ejemplo: "5".padStart(3, "0") → "005"

3. OBJETO MATH (MÉTODOS MENOS COMUNES)

Math.ceil(x) → Redondea hacia ARRIBA
Math.trunc(x) → Elimina los decimales (sin redondear)
Math.abs(x) → Valor absoluto (convierte negativo a positivo)
Math.pow(base, exp) → Potencia (equivalente a base ** exp)
Math.sign(x) → Devuelve 1, -1 o 0 según el signo del número

4. BOM - BROWSER OBJECT MODEL

Window:

window.innerWidth → Ancho del viewport (área visible)
window.innerHeight → Alto del viewport
window.scrollX → Píxeles desplazados horizontalmente
window.scrollY → Píxeles desplazados verticalmente

Location:

window.location.href → URL completa actual
window.location.hostname → Nombre del dominio
window.location.pathname → Ruta de la URL
window.location.search → Query string (?param=valor)
window.location.reload() → Recarga la página

History:

history.back() → Vuelve a la página anterior
history.forward() → Avanza a la página siguiente
history.go(n) → Navega n páginas (negativo=atrás)

Navigator:

navigator.userAgent → Información del navegador
navigator.language → Idioma del navegador

5. DOM - MÉTODOS MENOS COMUNES

elemento.insertBefore(nuevo, referencia)

- Inserta 'nuevo' ANTES del elemento 'referencia'
- Sintaxis: padre.insertBefore(elementoNuevo, elementoReferencia)

elemento.replaceChild(nuevo, viejo)

- Reemplaza el elemento 'viejo' por 'nuevo'

elemento.cloneNode(profundo)

- Clona el elemento
- Si profundo=true, clona también todos sus hijos

elemento.closest(selector)

- Busca el ancestro más cercano que coincida con el selector
- Incluye el propio elemento en la búsqueda

6. OBJETO DATE (MÉTODOS PRINCIPALES)

Obtener componentes de fecha:

fecha.getFullYear() → Año (4 dígitos)
fecha.getMonth() → Mes (0-11, enero es 0!)
fecha.getDate() → Día del mes (1-31)
fecha.getDay() → Día de la semana (0-6, domingo es 0)
fecha.getHours() → Hora (0-23)
fecha.getMinutes() → Minutos (0-59)
fecha.getSeconds() → Segundos (0-59)

Establecer componentes:

fecha.setFullYear(año) / fecha.setMonth(mes) / fecha.setDate(día) / fecha.setHours(hora)

7. DRAG & DROP API (SOLO PARA EJERCICIO BONUS)

Atributo HTML:

`draggable="true"` → Hace el elemento arrastrable

Eventos (en orden de ejecución):

`dragstart` → Cuando empieza a arrastrarse

`drag` → Mientras se arrastra (continuo)

`dragenter` → Cuando entra en una zona de drop válida

`dragover` → Mientras está sobre una zona de drop (continuo)

`dragleave` → Cuando sale de una zona de drop

`drop` → Cuando se suelta en una zona de drop

`dragend` → Cuando termina el arrastre

Objeto `event.dataTransfer`:

`event.dataTransfer.setData(tipo, dato)`

- Guarda información sobre el elemento arrastrado
- Ejemplo: `e.dataTransfer.setData('text/html', elemento.id)`

`event.dataTransfer.getData(tipo)`

- Recupera la información guardada
- Ejemplo: `let id = e.dataTransfer.getData('text/html')`

`event.dataTransfer.effectAllowed` → Define el efecto: 'move', 'copy', 'link'

`event.dataTransfer.dropEffect` → Define el efecto visual

CRÍTICO: En `dragover` SIEMPRE usar `event.preventDefault()`

En `drop` SIEMPRE usar `event.preventDefault()`

Sin `preventDefault()` en `dragover`, el `drop` NO funcionará

8. CONVERSIÓN DE TIPOS (CASOS ESPECÍFICOS)

`isNaN(valor)`

- Devuelve true si NO es un número

`Number.isNaN(valor)`

- Más estricto, solo true si es NaN exacto

`isFinite(valor)`

- Devuelve false si es Infinity o -Infinity

`Boolean(valor) o !!valor`

- Convierte a true/false

9. SPREAD OPERATOR CON OBJETOS

`let copia = {...objetoOriginal}` → Copiar objeto

`let combinado = {...objeto1, ...objeto2}` → Combinar objetos

`let nuevo = {...objetoViejo, nuevaProp: valor}` → Añadir propiedades

10. DESESTRUCTURACIÓN (DESTRUCTURING)

Arrays:

`let [primero, segundo] = [1, 2, 3]`

- `primero = 1`, `segundo = 2`

Objetos:

`let {nombre, edad} = {nombre: "Ana", edad: 25}`

- Extrae las propiedades `nombre` y `edad`

RECORDATORIOS IMPORTANTES

- `Array.isArray(variable)` → Comprueba si es un array
- `elemento.dataset.atributo` → Accede a atributos data-* del HTML
- `setTimeout(función, milisegundos)` → Ejecuta función tras X milisegundos
- `setInterval(función, milisegundos)` → Ejecuta función cada X milisegundos
- `clearTimeout(id)` / `clearInterval(id)` → Cancela un timeout/interval