



Dime conceptos básicos de javascript como estudiante de bachillerato. Para ejervicios tipo contar letras repetidas, cambiar color de fondo o tamaño letra.

JavaScript es un lenguaje de programación que se usa para dar interactividad a páginas web, como cambiar colores, responder a botones, o procesar texto ingresado^[1] ^[2] ^[3].

Conceptos clave de JavaScript

- **Variables:** Sirven para guardar cualquier dato. Se declaran con `let`, `var` o `const`. Por ejemplo:

```
let nombre = "Ana";
```

- **Tipos de datos:**

- Números: `let numero = 7;`
- Cadenas (texto): `let palabra = "Hola";`
- Booleanos: `let esMayor = true;`
- Arrays (listas): `let colores = ["rojo", "azul", "verde"];`

- **Operadores:**

- Suma: `3 + 2`
- Concatenar texto: `"Hola " + "mundo"`
- Comparaciones: `==`, `!=`, `<`, `>`

- **Estructuras de control:**

- Condicionales:

```
if (edad >= 18) {  
  console.log("Mayor de edad");  
} else {  
  console.log("Menor de edad");  
}
```

- Bucles (para repetir acciones):

```
for (let i = 0; i < 5; i++) {  
  console.log(i);  
}
```

```
}
```

- **Funciones:**

Bloques que ejecutan tareas y se pueden reutilizar.

```
function saludar(nombre) {  
  alert("Hola " + nombre);  
}  
saludar("Ana");
```

- **Interacción con la web (DOM):**

- Cambiar color de fondo:

```
document.body.style.backgroundColor = "yellow";
```

- Cambiar tamaño de letra:

```
document.body.style.fontSize = "24px";
```

Ejemplo: contar letras repetidas

Código simple para contar cuántas veces aparece una letra en un texto:

```
function contarLetra(cadena, letra) {  
  let contador = 0;  
  for (let i = 0; i < cadena.length; i++) {  
    if (cadena[i] === letra) {  
      contador++;  
    }  
  }  
  return contador;  
}  
console.log(contarLetra("banana", "a")); // Salida: 3
```

Estos conceptos te permiten hacer ejercicios básicos como los que mencionas: manipular el fondo, contar letras, cambiar tamaño de letra y mucho más en el navegador^{[1] [2] [3]}.

✱

Amplíame las interacciones con DOM. Explícame qué es, cómo se utiliza y ejemplos sencillos

El DOM (Document Object Model) es la estructura que representa y organiza todos los elementos de una página web como si fueran objetos en un árbol^{[17] [18] [19]}. Permite que JavaScript acceda y cambie el contenido, el diseño y la estructura de una página de forma dinámica.

Qué es el DOM

El DOM es una "interfaz de programación" que traduce el HTML de una página en una estructura de datos.

Cómo se utiliza el DOM

Para trabajar con el DOM en JavaScript se usan métodos que permiten seleccionar elementos, crear nuevos elementos o modificarlos. Los más usados son:

- **Seleccionar elementos**

- Por ID:

```
let titulo = document.getElementById("miTitulo");
```

- Por clase o etiqueta:

```
let cajas = document.getElementsByClassName("caja");  
let parrafos = document.getElementsByTagName("p");
```

- Selección avanzada:

```
let boton = document.querySelector("#enviar");  
let todos = document.querySelectorAll(".item");
```

- **Modificar elementos**

- Cambiar texto:

```
titulo.textContent = "Nuevo título";
```

- Cambiar estilos:

```
titulo.style.color = "blue";  
titulo.style.fontSize = "20px";
```

- Cambiar atributos:

```
boton.setAttribute("disabled", true);
```

- **Crear o eliminar elementos**

- Crear nuevo:

```
let nuevoDiv = document.createElement("div");  
nuevoDiv.textContent = "Soy nuevo";  
document.body.appendChild(nuevoDiv);
```

- Eliminar:

```
nuevoDiv.remove();
```

- **Eventos (responder a acciones)**

- Escuchar clic:

```
boton.addEventListener("click", function() {  
    alert("Has hecho clic!");  
});
```

Ejemplos sencillos de interacción DOM

- **Cambiar color de fondo con un botón**

```
<button id="colorBtn">Cambiar color</button>  
<script>  
    document.getElementById("colorBtn").onclick = function() {  
        document.body.style.backgroundColor = "lightgreen";  
    };  
</script>
```

- **Aumentar tamaño de letra al presionar**

```
<p id="texto">Hola mundo</p>  
<button onclick="document.getElementById('texto').style.fontSize='32px'">  
    Grande  
</button>
```

- **Contar letras en un campo de texto**

```
<input id="entrada">  
<button id="contarBtn">Contar "a"</button>  
<script>  
    document.getElementById("contarBtn").onclick = function() {  
        let texto = document.getElementById("entrada").value;  
        let contador = 0;  
        for(let letra of texto) {  
            if(letra === "a") contador++;  
        }  
        alert('Cantidad de "a": ' + contador);  
    }  
</script>
```

Gracias al DOM puedes crear páginas web dinámicas que responden a lo que hace el usuario^[17]
^[18] ^[19].

Explícame innerHTML

La propiedad `innerHTML` en JavaScript se utiliza para obtener o modificar el contenido HTML, incluyendo etiquetas, dentro de un elemento del DOM [\[26\]](#) [\[27\]](#) [\[28\]](#).

Qué es innerHTML

- Es una propiedad de los elementos HTML (como `<div>`, `<p>`, ``, etc.) que permite

- Cuando se usa para modificar, reemplaza todo el contenido interno del elemento por el nuevo código HTML que le asignes [\[26\]](#) [\[28\]](#).

Cómo se utiliza

- Para obtener el contenido HTML de un elemento:

```
let contenido = elemento.innerHTML;
```

Esto devuelve todo el código HTML dentro de ese elemento [\[26\]](#) [\[28\]](#).

- Para modificar el contenido y las etiquetas internas de un elemento:

```
elemento.innerHTML = "<b>Texto en negrita</b><br><i>Texto en cursiva</i>";
```

Cambia todo por el nuevo HTML dado en la asignación [\[26\]](#) [\[27\]](#).

Ejemplos sencillos

- **Modificar texto de un párrafo**

```
<p id="miParrafo">Texto original</p>
<button onclick="document.getElementById('miParrafo').innerHTML='Nuevo <b>texto</b>'"
```

El botón cambia el texto y puede añadir etiquetas (en este caso, la palabra 'texto' en **negrita**) [\[29\]](#) [\[30\]](#).

- **Crear una lista con innerHTML**

```
<ul id="miLista">
  <li>Elemento 1</li>
</ul>
<button onclick="document.getElementById('miLista').innerHTML='<li>Nueva 1</li><li>Nu
```

El botón reemplaza todos los ítems por nuevos elementos [\[29\]](#).

Advertencias

- Si se agrega contenido HTML proveniente de un usuario sin comprobarlo, puede introducir código dañino (inyección de scripts) ^[27].
- Si el elemento tenía otros nodos dentro, serán eliminados y sustituidos por el nuevo contenido.

`innerHTML` facilita la edición dinámica del contenido web de forma sencilla y rápida ^{[26] [27] [28]}.

✱✱

<ul id="miLista">

- Elemento 1

 Actualizar lista

Amplíame esta línea, explícame la sintaxis como estudiante de bachillerato

La línea que mencionas utiliza JavaScript y la propiedad `innerHTML` para cambiar el contenido de una lista cuando se pulsa un botón ^{[36] [37] [38]}.

```
<button onclick="document.getElementById('miLista').innerHTML='<li>Nueva 1</li><li>Nueva
```

Explicación paso a paso

- `onclick=""`:
Es un atributo del botón que indica qué debe suceder cuando se hace clic. Aquí, ejecutará la acción que hay dentro de las comillas ^[37].
- `document.getElementById('miLista')`:
Sirve para "encontrar" el elemento de la página cuyo `id` es "miLista" (en este caso, la lista ``) ^[39].
- `.innerHTML`:
Es una propiedad que permite cambiar el contenido HTML del elemento seleccionado. Todo lo que esté entre las etiquetas `` `` (los ``) será reemplazado por lo nuevo ^{[36] [38]}.

```
- **`='<li>Nueva 1</li><li>Nueva 2</li>':**
```

Indica que el nuevo contenido serán dos elementos de lista: "Nueva 1" y "Nueva 2". El texto y la etiqueta `` se escriben como una cadena (texto entre comillas simples) ^[37].

Sintaxis general del ejemplo

- Escoge el elemento con `getElementById`.
- Cambia el contenido interno usando `innerHTML = 'nuevo contenido HTML'`.

Ejemplo explicado

Al pulsar el botón, el navegador busca el elemento con id `'miLista'`, borra sus elementos

- `Nueva 1`

- `Nueva 2`

Por eso, la lista cambia en la pantalla inmediatamente y muestra los nuevos elementos^[36] ^[37] ^[38].

Esta forma es ideal para modificar rápidamente listas, textos o cualquier contenido en la

Explicación de la sintaxis

- **`document.getElementById('miLista')`**: Selecciona el elemento del documento HTML con el identificador `miLista` (en este caso, una lista ``)^[37] ^[39].

- **`**.innerHTML**`**: Es una propiedad que permite leer o modificar el contenido HTML de ese

- **`**= ' Nueva 1Nueva 2'**`**: Reemplaza el contenido actual de la lista con

- Todo esto se ejecuta automáticamente al hacer clic en el botón gracias a `onclick`^[37].

¿Cómo funciona?

- Antes de pulsar el botón: La lista tiene `Elemento 1`.

- Al pulsar el botón:
 - Se selecciona la lista por su id.
 - Se cambia el contenido entre las etiquetas `` por el texto y etiquetas nuevas que escribiste como cadena.
 - El navegador actualiza la página y muestra "Nueva 1" y "Nueva 2" en la lista^[37] ^[39].

Esta técnica te permite modificar el contenido de listas, párrafos o cualquier elemento del DOM usando JavaScript^[36] ^[38].

1. https://developer.mozilla.org/es/docs/Learn_web_development/Getting_started/Your_first_website/Adding_interactivity
2. <https://www.tusclases.co/blog/conceptos-basicos-empezar-entender-javascript>
3. <https://centrogeo.github.io/JSvis/08-Conceptos-básicos-de-JS.html>
4. <https://www.youtube.com/watch?v=1gIVfFxj8a4>
5. <https://www.youtube.com/watch?v=xnWtGNiG2lg>
6. <https://es.javascript.info>
7. <https://www.youtube.com/watch?v=QoC4RxNls5M>
8. <https://lenguajejs.com/fundamentos/introduccion/conceptos-previos/>
9. https://www.youtube.com/watch?v=Eh-s-n_6zNQ
10. <https://www.educaciontrespuntocero.com/tecnologia/programar-en-javascript/>
11. <https://www.youtube.com/watch?v=z95mZVUcJ-E>
12. <https://blogsaverroes.juntadeandalucia.es/plataformaeiv/files/2016/09/manual-javascript.pdf>
13. <https://www.youtube.com/watch?v=RqQ1d1qEWIE>
14. <https://www.aluracursos.com/blog/guia-de-javascript>
15. https://www.aulascript.com/evaluar/ejercicios_basicos_1.htm
16. <https://www.um.es/docencia/barzana/DAWEB/Lenguaje-de-programacion-JavaScript-1.pdf>
17. <https://www.freecodecamp.org/espanol/news/que-es-el-dom-el-significado-del-modelo-de-objeto-de-documento-en-javascript/>
18. https://developer.mozilla.org/es/docs/Web/API/Document_Object_Model
19. <https://www.arsys.es/blog/dom-document-object-model-que-es-y-como-funciona-en-javascript>
20. <https://www.inesem.es/revistadigital/informatica-y-tics/dom>
21. <https://lenguajejs.com/dom/introduccion/que-es/>
22. https://es.wikipedia.org/wiki/Document_Object_Model
23. https://developer.mozilla.org/es/docs/Web/API/Document_Object_Model/Introduction
24. <https://travesia.mcu.es/items/d3108f7c-aa42-4967-8576-e6e1d5a859bd>
25. <https://www.youtube.com/watch?v=Jh-LUQMwtRk>
26. <https://developer.mozilla.org/es/docs/Web/API/Element/innerHTML>
27. <https://lenguajejs.com/dom/contenido/innerHTML/>
28. <https://keepcoding.io/blog/propiedad-innerhtml/>
29. <https://www.digitallearning.es/intro-programacion-js/innerHTML.html>
30. <https://academiasanroque.com/manipulacion-del-dom-con-javascript-uso-intermedio-de-innerhtml/>
31. <https://www.youtube.com/watch?v=s7ykocU8Nog>
32. <https://www.youtube.com/watch?v=b7VYNQGSkAo>
33. <https://es.linkedin.com/advice/1/what-differences-between-innerhtml-dom-methods-vg4ff?lang=es&lang=es>
34. <https://www.aspclasico.com/innerHTML-para-cambiar-el-contenido-de-un-elemento-html-33/>
35. <https://es.stackoverflow.com/questions/486056/usar-innerhtml-correctamente-en-javascript>
36. <https://developer.mozilla.org/es/docs/Web/API/Element/innerHTML>

37. <https://www.digitallearning.es/intro-programacion-js/innerHTML.html>
38. https://www.mejorprogramacion.com/curso_javascript/8-innerhtml/
39. <https://www.digitallearning.es/intro-js/getelementbyid-innerhtml.html>
40. https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=818%3AinnerHTML-javascript-modificar-texto-o-html-pej-div-en-tiempo-real-this-acceder-al-nodo-actual-cu01138e&catid=78&Itemid=206
41. <https://es.stackoverflow.com/questions/486056/usar-innerhtml-correctamente-en-javascript>
42. <https://www.youtube.com/watch?v=b7VYNQGSkAo>
43. <https://lenguajejs.com/dom/contenido/innerHTML/>
44. https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=819%3Aejemplo-innerhtml-javascript-ejercicio-resuelto-modificar-campos-de-tabla-dinamica-editar-filas-cu01139e&catid=78&Itemid=206
45. <https://academiasanroque.com/ejercicio-de-javascript-modificando-contenido-con-innerhtml/>