



Dime cómo funciona constructor en javascript

El constructor en JavaScript es un método especial que se utiliza dentro de una clase para inicializar objetos cuando se crea una nueva instancia usando la palabra clave `new`^[1] ^[2].

Qué hace el constructor

- El constructor asigna valores iniciales a las propiedades del objeto usando los parámetros que recibe^[3].
- Solo puede haber un método constructor por clase^[2].
- Se invoca automáticamente en el momento en que se crea la instancia.

Cómo se utiliza

```
class Animal {  
  constructor(nombre) {  
    this.nombre = nombre;  
  }  
  hablar() {  
    console.log(this.nombre + ' hace ruido.');//  
  }  
}
```

Al crear una instancia así: `const miAnimal = new Animal('Perro')`, el constructor asigna 'Perro' a la propiedad `nombre` del objeto. Así, cada objeto puede tener valores diferentes según los parámetros que se le pasan al constructor^[1] ^[3] ^[2].

Resumen visual

- El constructor es el método que pone las "propiedades iniciales" de cada objeto creado.
- Se ejecuta siempre al hacer `new Clase(...)`.
- Permite personalizar cada objeto con valores únicos desde el principio^[1] ^[3] ^[2].

**

La palabra constructor es reservada? Ponme otro ejemplo

Sí, la palabra "constructor" en JavaScript es una palabra reservada especial dentro de las clases. Solo puede haber un método llamado `constructor` en cada clase, y es el método que se ejecuta automáticamente cuando se crea una instancia de la clase con `new`. Si intentas definir más de un método con ese nombre, JavaScript lanzará un error de sintaxis^{[12] [13] [14]}.

Ejemplo diferente de uso del constructor

```
class Coche {  
    constructor(marca, modelo, año) {  
        this.marca = marca;  
        this.modelo = modelo;  
        this.año = año;  
    }  
  
    describir() {  
        console.log(`Este coche es un ${this.marca} ${this.modelo} del año ${this.año}.`);  
    }  
}  
  
const miCoche = new Coche("Toyota", "Corolla", 2020);  
miCoche.describir();  
// Salida: Este coche es un Toyota Corolla del año 2020.
```

En este ejemplo, el constructor recibe tres parámetros para inicializar las propiedades del objeto creado, y luego el método `describir` usa esos valores para mostrar una descripción^{[13] [15] [14]}.

**

Y el uso de \$?

El símbolo \$ en JavaScript no es un operador especial en sí mismo, sino que tiene varios usos importantes, dependiendo del contexto. Uno muy común es en las plantillas literales (template literals) para insertar variables o expresiones dentro de cadenas de texto. Se usa junto con las llaves {} dentro de comillas invertidas `.

Uso de \$ en plantillas literales para interpolar variables

```
let nombre = "Ana";  
console.log(`Hola, ${nombre}!`);  
// Salida: Hola, Ana!
```

Aquí, `${nombre}` inserta el valor de la variable `nombre` dentro de la cadena.

Distinción con el operador ternario ?

El operador ternario es otro símbolo que usa ? pero es totalmente distinto a \$. Sirve para hacer condicionales en una sola línea. Por ejemplo:

```
let edad = 20;
let resultado = edad >= 18 ? "Mayor de edad" : "Menor de edad";
console.log(resultado);
// Salida: Mayor de edad
```

Resumen

- \$ se usa en template literals para insertar valores dentro de cadenas con \${...}.
- ? es parte del operador ternario para condicionales cortas.

Si se quiere, puedo explicar más ejemplos de cada uno [\[21\]](#) [\[22\]](#) [\[23\]](#).

**

Esos mismos ejercicios pónmelos sin usar el \$

Claro, aquí están los ejemplos sin usar el símbolo \$, es decir, sin usar plantillas literales, sino con concatenación clásica de strings:

Ejemplo sin \$ para concatenar cadenas

```
let nombre = "Ana";
console.log("Hola, " + nombre + "!");
// Salida: Hola, Ana!
```

Ejemplo del operador ternario sin cambio (no usa \$)

```
let edad = 20;
let resultado = edad >= 18 ? "Mayor de edad" : "Menor de edad";
console.log(resultado);
// Salida: Mayor de edad
```

Aquí la diferencia principal es que en vez de usar \${nombre} dentro de backticks, se usa la suma de strings con + para incluir la variable dentro del texto [\[31\]](#) [\[32\]](#) [\[33\]](#).

**

1. <https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Classes/constructor>
2. <https://jscodez.com/javascript-constructor-de-clase/>
3. <https://www.luisllamas.es/javascript-constructor-en-clase/>
4. <https://www.youtube.com/watch?v=2BL6StoXuoo>

5. https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=826%3Aconstructores-javascript-crear-objetos-vacios-anadir-propiedades-y-metodos-singleton-ejemplos-cu01145e&catid=78&Itemid=206
6. image-1.jpg
7. <https://keepcoding.io/blog/constructor-de-una-clase-en-javascript/>
8. <https://es.javascript.info/constructor-new>
9. <https://jscodez.com/javascript-funcion-constructora/>
10. <https://www.luisllamas.es/javascript-funciones-constructoras/>
11. https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Function/Function
12. <https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Classes/constructor>
13. <https://www.luisllamas.es/javascript-constructor-en-clase/>
14. <https://keepcoding.io/blog/crear-clases-en-javascript/>
15. <https://jscodez.com/javascript-constructor-de-clase/>
16. <https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/super>
17. <https://es.js.org/sintaxis/palabras-reservadas>
18. <https://jscodez.com/tipos-super-clases/>
19. <https://lenguajejs.com/javascript/oop/clases/>
20. <https://ifgeekthen.nttdata.com/s/post/clases-en-javascript-MCLKITC44WXZYK45OERIYA24I?language=es>
21. https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/Conditional_operator
22. <https://lenguajejs.com/fundamentos/estructuras-de-control/operador-ternario/>
23. <https://www.luisllamas.es/javascript-condicional-ternario/>
24. https://www.youtube.com/watch?v=G_8rylgbRZw
25. <https://www.freecodecamp.org/espanol/news/como-usar-el-operador-ternario-en-javascript-explicado-con-ejemplos/>
26. <https://es.linkedin.com/pulse/descubre-el-operador-ternario-en-javascript-martinez-melchor-vqlIf>
27. <https://jscodez.com/javascript-operadores-ternario/>
28. <https://www.freecodecamp.org/espanol/news/como-utilizar-el-operador-ternario-en-javascript-explicado-con-ejemplos/>
29. <https://keepcoding.io/blog/operadores-ternarios-en-javascript/>
30. <https://asjordi.dev/blog/aprende-a-utilizar-el-operador-ternario-en-javascript/>
31. https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/Conditional_operator
32. <https://lenguajejs.com/fundamentos/estructuras-de-control/operador-ternario/>
33. <https://www.luisllamas.es/javascript-condicional-ternario/>