

Guía completa del operador Spread en JavaScript

Esta guía modular cubre todos los usos modernos del operador Spread, con ejemplos claros, advertencias y buenas prácticas. Está organizada en archivos independientes para facilitar el estudio y la consulta.

🔗 Introducción

El operador Spread (`...`) permite expandir elementos de arrays, objetos o iterables en lugares donde se esperan múltiples elementos o propiedades. Es una herramienta clave para escribir código limpio, inmutable y expresivo.

📦 Índice de archivos

Nº	Tema	Archivo
01	Uso básico	01-uso_basico.js
02	Convertir estructuras	02-convertir_estructuras_parecidas_a_array_en_array.js
03	Clonación inmutable	03-clonacion_de_arrays_y_objetos.js
04	Spread en funciones	04-spread_en_funciones.js
05	Diferencia Rest vs Spread	05-diferencia_entre_rest_y_spread.js
06	Desestructuración inversa	06-destructuracion_inversa.js
07	Limitaciones del Spread	07-limitaciones_del_spread.js
08	Agregar elementos a arrays	08-agregar_elementos_en_un_array.js
09	Agregar propiedades a objetos	09-agregar_propiedades_a_un_objeto.js
10	Fusionar arrays	10-fusionar_arrays.js
11	Fusionar objetos	11-fusionar_objetos.js
12	Estructuras anidadas	12-estructuras_anidadas.js
13	Estructuras anidadas con Lodash	13-estructuras_anidadadas_con_libreria_lodash.js
14	Propiedades condicionales	14-propiedades_condicionales.js
15	Advertencias comunes	15-advertencias_comunes.js
00	Resumen global	spread-resumen.js

☑ Usos principales

- Expandir arrays en llamadas a funciones
 - Clonar arrays y objetos (shallow copy)
 - Fusionar arrays y objetos
 - Excluir propiedades con desestructuración inversa
 - Construir objetos con propiedades condicionales
 - Manipular estructuras anidadas (con ayuda de Lodash)
-

Buenas prácticas

- Usa Spread para mantener la inmutabilidad
 - Evita mutar objetos originales
 - Prefiere Spread sobre `Object.assign()` por legibilidad
 - Para estructuras profundas, usa `_.cloneDeep()` o `structuredClone()`
-

Advertencias comunes

- No puedes usar Spread en valores no iterables (ej. `...123`)
 - No copia métodos ni prototipos (ej. `new Date()`)
 - No confundas Rest (`...args` en definiciones) con Spread (`...args` en llamadas)
-

Recomendación final

Si estás trabajando con estructuras complejas, considera usar Lodash o `structuredClone()` para clonación profunda. Esta guía está pensada para ayudarte a dominar el operador Spread con ejemplos claros y aplicables.
