

■ Guía rápida: == vs === en JavaScript

Situación		¿Conviene usar == ?	¿Conviene usar === ?	Ejemplo
Comparar null y undefined como equivalentes	■ Sí	■ No (ignora undefined)		if (valor == null)
Comparar números con strings numéricos	■ Sí (flexible)	■ No (estricto)		"18" == 18
Validar booleanos contra 0 o 1	■ Sí (rápido)	■ Mejor === en lógica crítica		if (flag == 1)
Inputs de usuario (string o número)	■ Sí	■ Mejor con parseo estricto		"0" == 0
Detectar valores vacíos (espacios, 0, "")	■ Sí	■ Puede confundir		" " == 0
Array vacío con "" o 0	■ Posible	■ Poco claro		[] == ""
Comparar fechas con strings	■ Sí	■ No siempre consistente		new Date(...) == "Fri Aug 29 2025"
Comparar booleanos con strings	■ Sí	■ Peligroso		false == "0"
Lógica sensible (dinero, auth, seguridad)	■ Nunca	■ Siempre		saldo === 100
Código colaborativo / producción	■ Nunca	■ Siempre		usar ===

■ Regla de oro:

- Usa === casi siempre (seguro y claro).
- Usa == solo cuando conscientemente quieras permitir coerción de tipos.