== and !=

# == y !=

Los operadores == y != hacen coerción de tipo antes de comparar. Esto es malo porque hace que ' \f\r \n\t ' == 0 sea true. Esto puede ocultar errores de tipo.

Cuando compares con cualquiera de los siguientes valores, utiliza siempre los operadores === o !==, que no hacen coerción de tipos:

0 '' undefined null false true

Si quieres la coerción de tipo, utiliza la forma abreviada. En lugar de:

(foo != 0)

di simplemente:

(foo)

Y en lugar de

(foo == 0)

di:

(!foo)

Siempre es preferible utilizar los operadores === y !==. Existe la opción "No permitir == y != " (eqeqeq), que exige el uso de === y !== en todos los casos.