Appendix B. Bad Parts

# ==

JavaScript tiene dos conjuntos de operadores de igualdad: === y !==, y sus gemelos malvados == y !=. Los buenos funcionan como cabría esperar. Si los dos operandos son del mismo tipo y tienen el mismo valor, entonces === produce true y !== produce false. Los gemelos malos hacen lo correcto cuando los operandos son del mismo tipo, pero si son de tipos diferentes, intentan coaccionar los valores. Las reglas por las que lo hacen son complicadas y poco memorables. Estos son algunos de los casos interesantes:

'' == '0' // false  
0 == '' // true  
0 == '0' // true  
  
false == 'false' // false  
false == '0' // true  
  
false == undefined // false  
false == null // false  
null == undefined // true  
  
' \t\r\n ' == 0 // true

La falta de transitividad es alarmante. Mi consejo es que nunca utilices los gemelos malvados. En su lugar, utiliza siempre === y !==. Todas las comparaciones que acabamos de mostrar producen false con el operador ===.

[anterior](apb_1.html)[Subtema 2 de 2: (Ver todo)](apb.html)