ToBoolean

↳ The ToBoolean abstract operation converts the given type to a Boolean value. ToBoolean works a bit differently when compared to ToString or ToNumber. It maintains a list of values which when received as an argument returns False. And everything apart from the list of values returns True.

List of falsy values →

null
undefened
+0
-0
NaN
"" (empty string)
false

if we get any one of these values in the argument we return false

else we return true.

Q ⇒ How can we test the ToBoolean operation?

→ We can use logical NOT operator (!)

!a

oldValue = ToBoolean(a)

if (oldValue is True)
    returns False

else
    returns True

What is the difference between == and === ?

→ (== operator only checks value

=== operator checks value as well as type)

100% wrong

Actually both == and === checks the types but the difference is both of them do something different after checking the type.

So == does type checking and if types are same it calls ===.

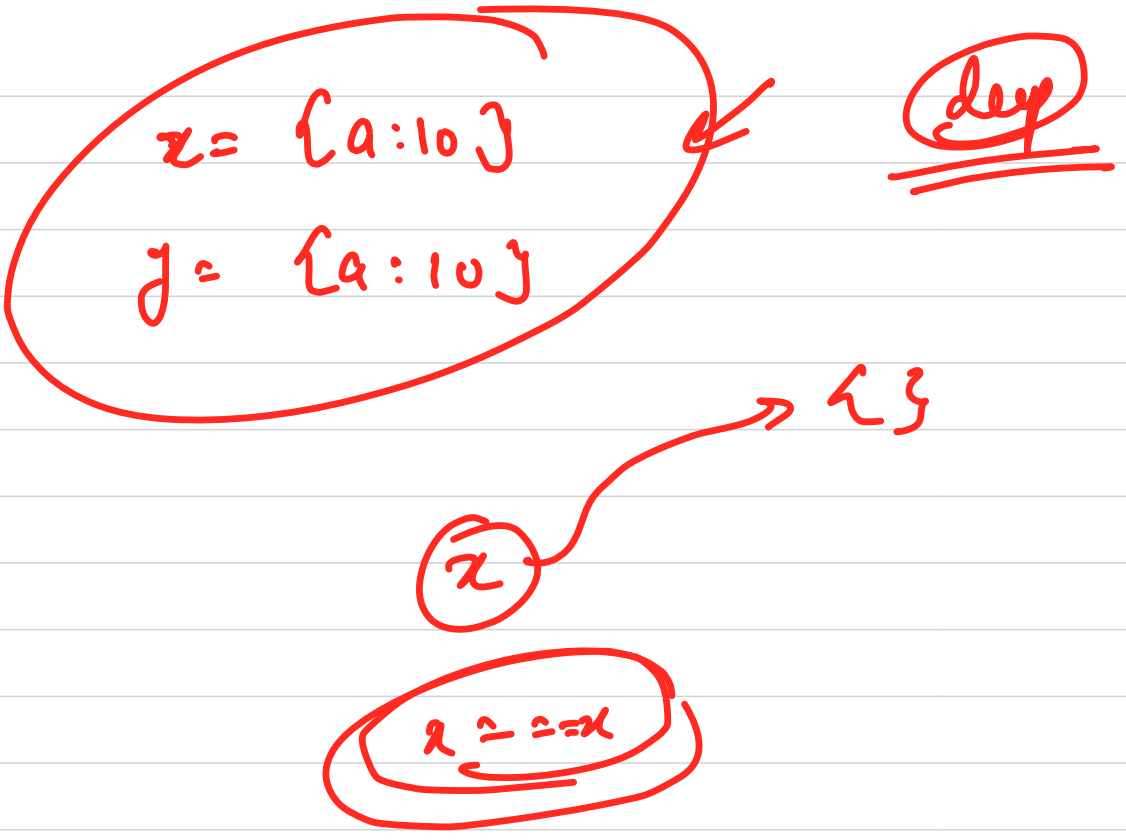
=== checks types & if types are not same return false.

The main difference is

→ abstract equality (==) does coercion if types are not same.

→ strict equality (===) never does coercion.

1) In JS NaN is the only primitive value not equal to itself.


x = {a:10}
y = {a:10}
deep
{}
x
x === x

REPL → Read
Evaluate
Print
Loop