

NaN (Not a Number)

typeOf NaN; → // number

↳ It doesn't actually mean not a number, rather it gives notion of an invalid number.

linear search → -1

"Sanket" → ToNumber

NaN === NaN

→ false



NaN == NaN

NaN is the only primitive value in JS that doesn't follow identity property and is not equal to itself.

function compute() {

...

returns val;

}

let x = compute();

Now suppose result of compute function can be NaN. And we have to detect if x is NaN or not ??

$(x === x) \rightarrow \underline{\underline{\text{false}}}$ X

{ } === { }

$x === \text{NaN}$

is NaN → this function has some issues.

this is NaN function coerces the argument before checking for NaN. Now string is not NaN, right ?? but when we convert string like "abc" to a number it becomes NaN & hence is NaN returns true.

Qⁿ When to use == & when to use === ??

While developing apps, we might encounter multiple programming lang. Every lang might have different properties.

Based how the language executes the code we can categorise languages in the following:-

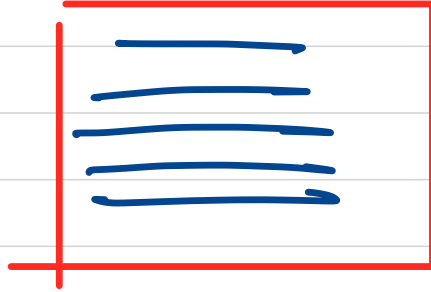
1) Compiled → C++

2) Interpreted → =====

Scopes

3) Hybrid →

Java



test.java

compile



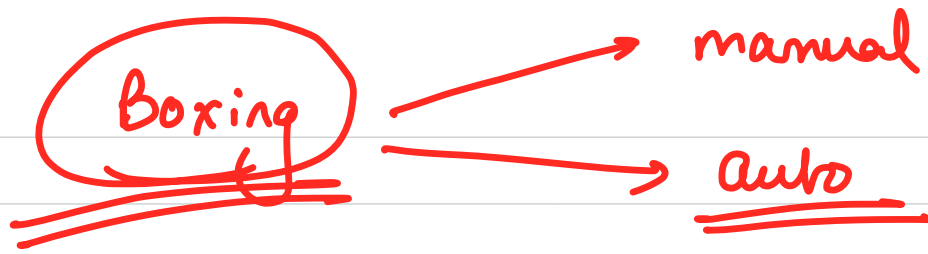
ByteCode

Java
is compiled

9 → number → primitive

"abc" → String





It is a phenomenon in JS when it tries to convert primitives into objects:-