```
var x = 10;  }
var y = 2;   }

console (x + y)

console (x - y)

x - y
```

/n → forward slash

\n → backslash

\t → special char

\n

"\$"

"\n"

Relational Operators

<, >, <=, >=

$(10 < 12)$

$(13 > 5)$

$(5 < 2)$

$>$

operand1 $<$ operand2

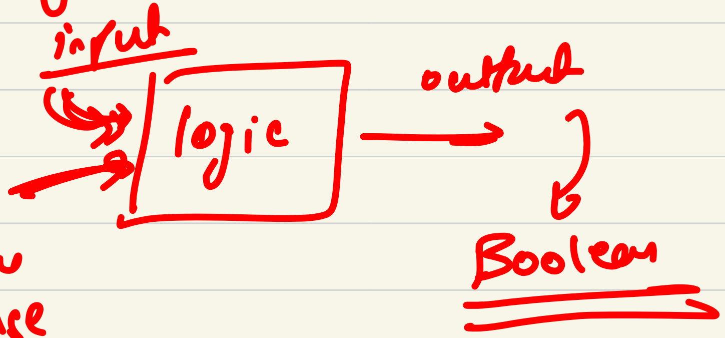
$>=$

$\boxed{<=}$

$\underbrace{(10 < 12)} \longrightarrow \underline{\underline{\text{true}}}$

$(5 < 2) \longrightarrow \underline{\underline{\text{false}}}$

Logical Operators



Boolean
logic
Gates

true
false

AND
OR
→ these need 2 operands

NOT → 1 operand

operand1 operand2

AN

AND GATE $\longrightarrow$ &&

OR GATE $\longrightarrow$ ||

NOT GATE $\longrightarrow$!

$$\begin{tikzcd} \text{operand 1} \arrow[d] & \text{operand2} \arrow[d] \\ \boxed{\text{boolean}} & \boxed{\text{boolean}} \end{tikzcd}$$

AND

X	Y	X AND Y
false	false	false
true	false	false
false	true	false
true	true	true

OR

X	Y	X OR Y
false	false	false
false	true	true
true	false	true
true	true	true

NOT

X	output
true	false
false	true

```
console.log( true && false)
```

$\rightarrow$ false

```
console.log( (10 > 5) && (6 < 3))
```

true && false

$\underbrace{\quad 10 \quad} \; \&\& \; \underbrace{\quad 6 \quad}$

||

Qn what values are falsy in JS ??

null
undefined
"" $\rightarrow$ empty string
+0, -0, NaN
false

} apart from these everything is truthy

→ <u>Coercion</u> (type <u>Interconversion</u>)

AND

X	Y	X AND Y
false	false	false
true	false	false
false	true	false
true	true	true

OR

X	Y	X OR Y
false	false	false
false	true	true
true	false	true
true	true	true

$(0 \&\& 6)$

$\downarrow$

$\underline{\underline{0}}$

NOT

X	output
true	false
false	true

In a AND gate, if the first input is false, then, it doesn't evaluate the second input and immediately returns the first input.

as well as if first input is true, then the Second input has to be evaluated & then second input is returned

In a OR gate, if the first input is true, then it doesn't evaluate the second input & immediately return the first input.

whereas,

if the first input is false then it returns the second input.

10 && 6

$\downarrow$

truthy

$\underline{\underline{\text{ft.}}}$

$(10 > 6)$ && $\underbrace{(6 < 7)}$ $\rightarrow$ $\underline{\underline{\text{true}}}$

```
console.log(6 && 10)
```

$\rightarrow \underline{\underline{10}}$

Numbers →

0

-0 directional

NaN

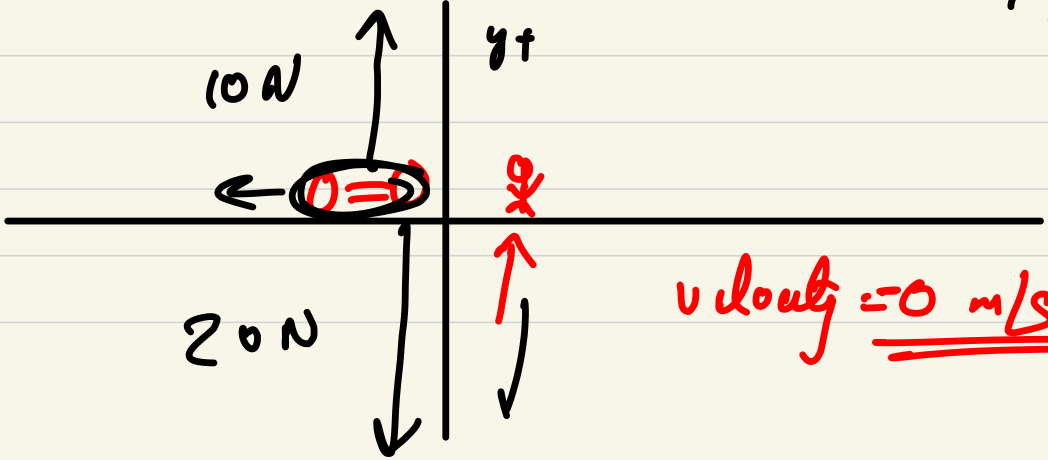
Infinity

-Infinity

Numbers

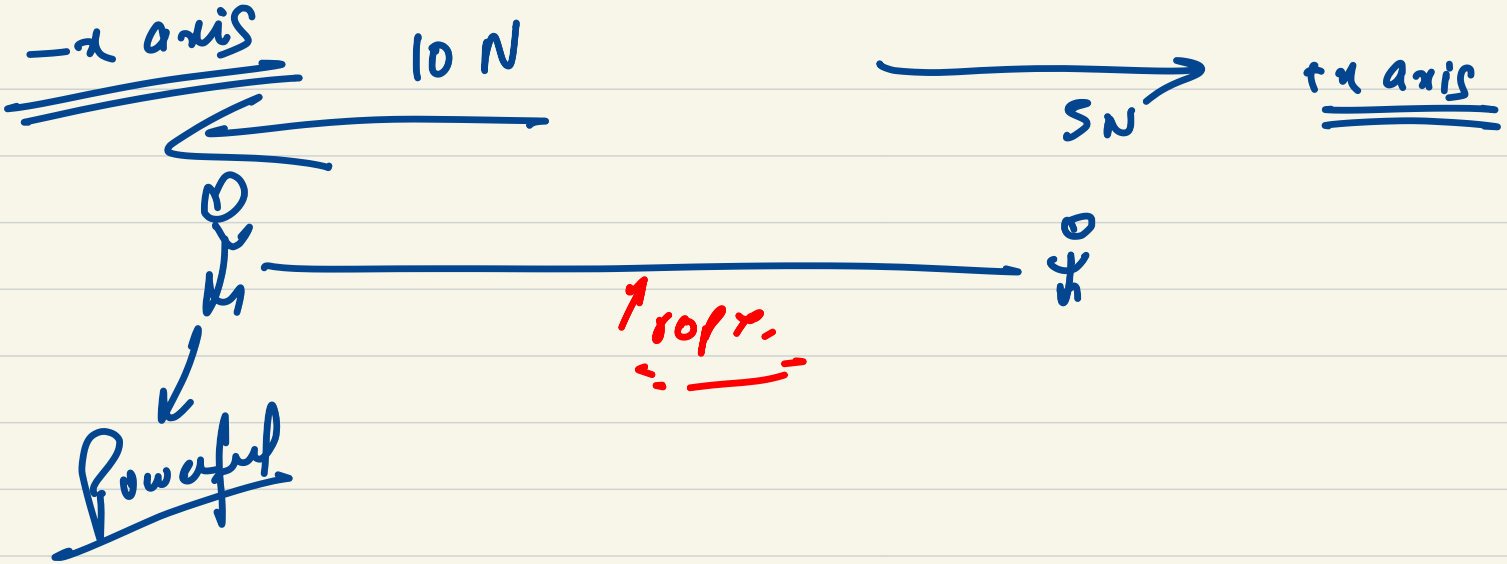
magnitude

direction



10 — 20

-10 N



Net force → 5 Newton in −ve x axis

$-5\,N$

NaN → <u>Not</u> <u>A</u> <u>Number</u>

0	1	2	3	4	5	6
ab	cd	or	xy	z	mn	a

→ Return the Bucket Number in which df string is <u>present</u>.

if there is a situation where you're bound to return a number, but there is no valid possible No. to return,

then we us <u>**NaN**</u>

Which is the only number in JS, which is not equal to <u>itself</u> ??

<u>**NaN**</u>

undefined / null

(→

"Sanket" / 2