18
4) 75
4
35
32
3

(1.30hr)

11/05/23

* Modulo operator (%) ↳ Modulo gives Remainder

```
Let x = 75;
let y = 4;

console.log(75%4)

console.log(x%y)  ⇒ Output - 3.

Console.log("Remainder is", x%y) :
                    Remainder is 3.
```

+ Exponentiation operator (**) ↳ power

Ex

```
Let a = 20;      ⇒   a**b = 20^7
let b = 7;                  =
```

Ex

```
Let a = 3 ;
let b = 10;                 Output
                            ↳ 59049.
console.log(a**b)
```

Square root of 16

```
  a = 16
Console.log("square root is", a**0.5);
              = 4.
```

* when we add two strings gets output as string.

8

9 * when we add two numbers get output as

10 NOTE

* when adding number to number output will be number (addition)

11

Imp * when am adding string with string output will be string

1

* when am adding string with number output will be string

2

3 Concatenation

(Joining)

4

* Boolean data types

5

Booleanmeans true . or False

6

* Relational & comparision operator *

(1) Partially accepted]
(2) wrong Ans]

* Runtime error
* compile time error

* let x = "Aman";
let y = "Singh";] concatenation (joining)

console.log(x+y);

x + y => AmanSingh ;

togivespace

console.log (x + " " + y)) → Aman_Singh

Console.log (x, y) → (seperating)
(,) Quamma

* "\n"
→ (used for next line)

Ex let x = " 'Masai school'";
let y = " Transformation in education";

console.log(x + "\n" + y) ;
→ Next line.

Output

Masaischool
Transformation in education

→ It Basicaly helps you to move to next line

~~Concatenation~~

~~* Ration~~

* Relational operator:-

Imp

* These operators works in case of numbers.

↳ > (greater than)

↳ >= (greater that Equal to)

↳ < (less than)

↳ <= (less than Equal to)

Note :- The output of relational operators will always be boolean (In the form of true or false)

(*) (1) > (Greater than)

* when we get output true

If my first value is strictly greater than second

~~Flase~~ False

If my first value is less than or equal to second

MARCH - 2023

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Ex:

```
let a = 20;
let b = 20.
console.log(a>b)
```

output:-

```
False.
```

Note: Inorder to get true as output the first value should always be greater than the second value.

~~Ex Let Did sunil pass exam,~~

```
let sunil_mark = 36;
let passing_mark = 35;
```

method1

```
console.Log(sunilmarks >passing_marks)';
```

method2

```
Let Result = sunil_marks> passing_mark;

console.log(Result);
```

WK 08 (057-308)

SUNDAY 26

MARCH
APRIL

② >= (Greaterthan Equal to)

true :- when first value is greater or equal to the second.

False : when the first value is smaller.

③ < (less than)

true :- when First value is strictly less than the second.

False :- when the First value is greater or equal to second.

④ ≤ (less than Equal to)

true ; when First value is less than or Equal to second .

False : when First value is greater the second.

MARCH - 2023						
M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

* (=) ⇝ single equal to
used for value assigning
to variable.

* **Comparision operators**

↳ ~~com~~ operators works with numbers as well as strings

8

9

10 ↳ == (Double equal to)

11 ↳ != (not equal to)

12 ↳ === (tripple equal to)

1 ↳ !== (not Double equal to)

→ output gives in the form of Boolean

2

3 (i) == (Double Equal to)

4 true: when both values are equal

5 False :- when both values are not equal.

6 Ex:-
```
1) console.log(10==10); // true
2) consol.log(5==7); // False
3) console.log("Aman"=="Aman"); true
4) console.log("Ajay"=="ajay"); False.
```

② != (not equal to)

It is opposite of ==.

true: when values are not equal.

false: when values are equal

③ tripple Equal to (===)

→ ~~looks~~ ~~equal to~~ works: ~~looks~~ exactly (==)

Exception.

Incase of ==

2==2 ; true

In case of ===

2===2; false

== (Double Equal)	=== (tripal equal)
8 true: when LHS = RHS	true: LHS = RHS.
False: when LHS ≠ RHS	False: LHS ≠ RHS.
10 * will not check the datatype	* will check the data type.

11

Ex:-

console.log (10 == 10); true

1

console.log (10 === 10); true

2

console.log (5 == 7); False

3

console.log (5 === 13); False

4

Exception:

5

* console.log (2 == "2"); true

6

* console.log (2 === "2"); False

└→ Bcoz also check

④ !== (not Doubleequal)

8 ↳ this is opposite of tripple.equal to. (===)

9

Ex

10

console.log(2!=="2"); output. true

11

12 == — T (equal)

1 F (not equal)

2

!= — T (not equal

3 F (equal)

4

=== — T (when equal)

5 F (not equal)

*also check for data type

6