

JAVAT Script

Data Types :-

1) Number

int data = 23
= =
1000 data 23

let data = (23)
= =
23.5 1210123
typeof (data) \Rightarrow Number

2) String :-

const name = "xyz"
= 'xyz'
=

const data = '1234'
= '1Ab'
= 'A'

C. log (typeof (data)) \Rightarrow String

3) Boolean :- true or false

```
const data = true
```

C. $\log(\text{dalu}) \Rightarrow \text{true}$

c. `log(typeof(data))` \Rightarrow boolean

4) BigInt :- large value stored by the BigInt data type

5) undefined :- — one of the data type of JS

`console.log(a) → undefined` ⇒ undefined → one of the variable
`var a = 10;` object has initialize by nothing
but it has memory location.

6) null :- One of the data type in JS

null \Rightarrow any object has represent by null that means it doesn't have any memory allocation.

let a = undefined

undefined

c.log(typeof(a)) \rightarrow undefined

let b = null

\rightarrow x

c.log(typeof(b)) \rightarrow object

7) symbol :- unique value - one of the data type \rightarrow symbol

NaN :- \rightarrow one of the result

(difficult to represent)

\rightarrow they indicate, whenever we can't represent any calculation result then we define that result by NaN

\rightarrow NaN \Rightarrow Not a Number

e.g. $10/0$ \Rightarrow NaN ✓

typeof(NaN) \Rightarrow Number

* Operator :-

1) Arithmetic Operator :- + - * / % ^

2) Relational Operator :- >, <, >=, <=, == \Rightarrow true/false

3) Logical Operator :- &&, ||, XOR, NOR $(\overline{a} = \overline{b} \& \& \overline{a} = \overline{c})$

4) Bitwise Operator $>_2 <_2, \&, |$

5) Assignment operator :- = e.g. $a = \underbrace{20}_{\leftarrow} c = a + b$

6) Increment / Decrement Op :- ++ --

Pre/Post

$\begin{array}{l} ++a \\ --a \end{array} \quad \begin{array}{l} a++ \\ a-- \end{array}$

H.W. $\left\{ \begin{array}{l} \text{e.g. let } a = 10 \\ \text{c. log (} \begin{array}{l} ++a \\ ++ \end{array} \text{)} \end{array} \right. \quad \left. \begin{array}{l} \text{let } b = a++ ; \\ = ++a ; \end{array} \right\} \quad \begin{array}{l} b = a-- \\ = --a \end{array}$

7) Short hand assignment operator :-

H.W. let $a = 10 \quad a = a + 1 ; \quad \underline{a += 1} \quad / \quad a -= 1$

8) Conditional Operator / Ternary Operator

Syntax:-

$$\frac{(\text{cond})}{\text{true/false}} \quad ? \quad \underline{\text{true}} : \underline{\text{false}}$$

Alternative for
if / else

e.g.

$$\begin{array}{c} \text{let } a = 10 \\ | \\ \text{let } b = 20 \\ | \\ | \\ (a > b) ? a : b \\ | \\ \underline{T/F} \end{array}$$

$$(a > b) : \text{c.log}(a) : \text{c.log}(b)$$

$\left. \begin{array}{l} \text{if}(a > b) \\ \{ \text{c.log}(a); \\ \} \\ \text{else} \\ \{ \text{c.log}(b); \} \end{array} \right\}$

9) Strictly Equality Operator :- =

Relational Operator \Rightarrow equal operator \Rightarrow =

e.g. let a = 10 // number | c.log(a == b) // false true
let b = '10' // string |

$== \Rightarrow$ equality operator \Rightarrow they just check the value
& convert into different data types
implicitly

e.g.

C. `log (a == b)` // false \Rightarrow they check not only but also
check the data type.

* Operations :-

1) number $\boxed{+}$ number / string / boolean / undefined / null

✓

(a) $\boxed{-}$ (b)
 $*$
 $/$

2) string + string / number / boolean / undefined / null

\ (a) -
* (b)
/

3) boolean + boolean / number / string / undefined / null
(a) ~~*~~ (b)
 /

4) undefined + undefined / number / string / undefined / null / boolean
(a) ~~*~~ (b)
 /

~

5) null + null / number / string / boolean / undefined
(a) ~~/~~ (b)
 /

Relational Op → >, <, >=, <=

* Template Literals :-

e.g. let a = 'boy' / 10 / true

console.log ('I am a = ', a) \Rightarrow I am a = boy

console.log (' I am a \${a} ') \Rightarrow I am a boy

Used in mostly react appln / react project

* functions :-

Version Control :- git / github





