

JAVA SCRIPT = JS

Console.log is used
to log (print)

```
Console.log (" ") ;
```

To connect html and JS
we write <script></script>
after Body tag src = "file name"

Variables In JS =

Variables are

Container for data

Null = knows what Inside in its blank value

Undefined = don't know what inside
in Undefined.

Dynamically Typed language JS:

Pehle se Batra

mai Parke variable me konfi value stored
hogi |coie bhi value store kar sakte hain.

(=) - Assignment operator

variable Rules: ① Variable name shouldn't keyword.

* a, A → Case Sensitive ①

* There should not be spaced in between variables, Undersore - _ & digits are allowed.

a - z both small or capital

② * 1st character only a letter or \$ or Underscore (-)

fullname = "Zohan";

- fullname = "2ohan";

\$ fullname = "2ohan";

④ Can't Start variable with number

* Reserved words cannot be variable names:

Reserved = fixed variable

like fixed word fullname me use nai hogा

full Name = Camel Case

First letter small next word 2nd
word Capital

Total Praise

let, const and Var:

In maximum case we use let ~~or~~^{and const} Var

let age = 24;

let fullName = "Zohan";

Var age = 24;

var fullName = "Zohan";

2015 before use Var not
now. Because it Reusable Variable

ES6 = Introduce In 2015

Const = Constant defined =

Constant value can't be changed
(fixed value) Const value.



age to the variable ki value nai
di let suppose age = a
its output will be undefined by
default.

Block means = { } curly Brackets
{ abc } one Block

use new Block if redefined var.

1 Block
Ex: { }

let a = 5 ;

console.log(a);

}

2 Block { }

let a = 10 ;

console.log(a);

Jaha se curly Brackets use hote
hain new Block Start hoga.

Data types in JS:-

Name = string data "text"

Numerical data = 123 any Number

Boolean = true, false.

Data Types divide in two categories.

Primitive data type:

* Number = let age = 24; → number type

* String =
let fullName = "Zohran";

* Boolean =
let isFollow = "true";
let isPass = "true";

* Undefined - Every variable is undefined

is there no value =

let x = x

let class; (no value has been given)

* Null =
let x = null → Absent of an object

let abc = null

Bigint = let x = BigInt("123")

Jaha Bigint hogi wala out but

me last me n. dikhi degg.....

Symbol

let y = Symbol ("Hello");

Output = y

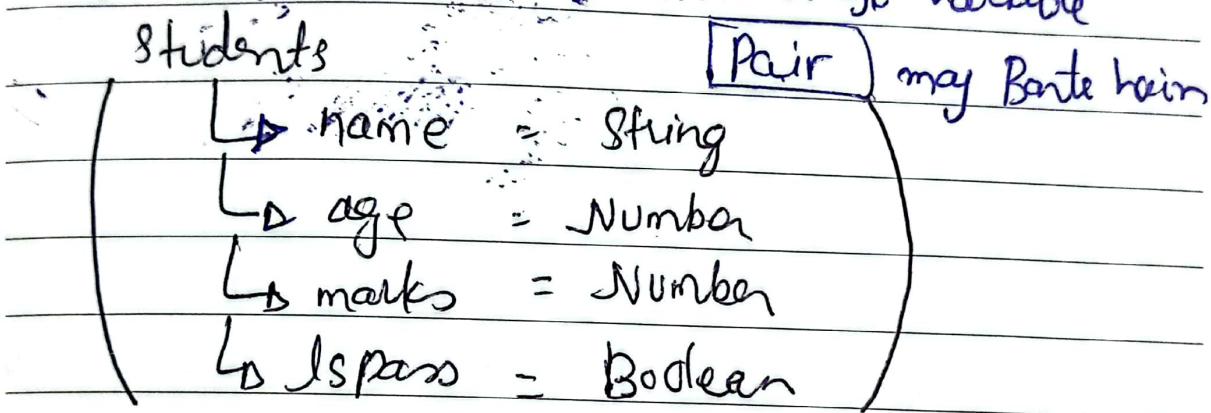
Symbol (Hello)

Non-Primitive data types / Composite

OBJEKT KE ANDAR HI CURRY ATA HY.

OBJECTS: IT'S A COLLECTION OF VALUES

IT STORE MULTIPLE VALUE IN A SINGLE VARIABLE



all value of collection is called object

Object mai key : value store hoti hain
 let Student = {

age : 24,

name : "Zohar"

{}

Array : store multiple value in single variable

Value written in Square brackets []

Syntax : let info = [5, "hina", "computer"];



FUNCTION:

function startWelcome()

Date

Object generally declare with

const Student = {

 fullName: "Zohan";

 key 4 age : 10, → value

 " CgPa = 8.1, ,

 isPass : True, ? ;

object collection of different variables.

Obj . key

obj [key]

When use . we don't use " lots

when use [] we need to use " double

'let can be update

const = Can't be update

But

const Obj . key can be update

Variable Scope = Block Scope Variable = if variable declare in block of code (in curly braces {}), it will alive in block and will not be accessible after curly braces.

Chapter #2 Operations and Conditional Statement



Comments in JS:

Comments neko execute

Comments \rightarrow Yeh line execute nahi hogi.

// Single line comment

/ *---*/ Multiple line comment

Operators

Use to perform some operation on data.

Arithmetic operator

+ , - , * , /

Modulus $\rightarrow \%$

Exponentiation \rightarrow Power $a^{***} b =$
 $a * * * a^b$

Unary operators: Needs single operator

\rightarrow Increment = ++

$a++$ (post increment)

\rightarrow Decrement = --

$++a$ (pre "")

$a--$ (post decrement)

$--a$ (pre decremented)

Assignment Operator:

Operate =

To assignment

= += -= *= % = ** =
 ↓

a += 1

a = a + 1

Comparison operators:

To compare two values

Equal to ==

Equal to & type ===

Not equal to !=

Not " " !==

\neq : ! \rightarrow not equal to in JS

$==$ to check data type as well

>, >=

<, <=



Logical Operations:

Expression

To evaluate multiple

Logical And & & = Both value True.
Condition 1 and 2 final True.

Logical OR || = if any 1 condition true answer
will be true.

Logical Not ! = if S/O give true it will
give false and if S/O give false
it will give true, it will
print opposite.

Conditional Statement:-

To implement
some condition in the code.

If Statement:-

To check any condition

```
let color;  
If ( mode == "dark-mode")  
color = "black";
```

* If else Statement

if → else

ager if condition true If Part Run

ager if condition false hoga to

else wala Part Run hoga.

MDN Docs =

MDN Web docs → Search documentation
and read.

Prompt = Temporary taken by user se Input lenak
Bhi Lelega. or print kore aje gya

alert (message) alert ("hello");
web page per Pop up message aye gya

PRINT / DISPLAY In JS

On Browser

Window.document.write("hello");

document.write ("hello"); to print on browser

in console = console.log ("Sehrish");

it will print on console window

Popup = window.alert ('Sehrish');

alert ("Sehrish"); it will open on

pop up

Taking Input from Users in JS :

Prompt : To ask the user for input

let answer = prompt ("Do you want to")

Arithmetic Operator

$a + b$, $4 + 5$

↓ ↓
(Operand) Operation

Tonneed $(+, -, \times, \%, -\%)$ (Modules / remainder)
two value

- * Increment } single operand.
- * decrement }

document.write ("(a" + "b)", " = ", a + b);
" (a/b)"

Modulus gives remainder

Unary Operators

Ternary Operators

condition? True output: false output;



Comparison Operators Answer on Boolean

$==$ (equal to) $==$ (equal to + same datatype)
 \neq (not equal to) \neq (not " ")

 $>$ \geq $<$ \leq

Logical Operators:

Logical AND $\&\&$ Both conditions should be true
 Logical OR $\|$ If one condition is true then it is true
 Logical NOT !

Conditional Operators:-

If Statement

If-else Statement

If-else If Statement

Ternary Operator:

If you want to check in one line then use Ternary operator

String:

- * String is a sequence of character used to represent a text.
- * It's a primitive data type.
- * we can create String by using template literals in single and double quotations.

Template literals \Rightarrow esc (escape button)

~~is a template induction~~



\n \Rightarrow next line

\t \Rightarrow for tab

\l = length

Space also count on Converting numbers' position / Index start with 0 in String.

Concatenation = Concat is used to joint two strings concat() add to variables.

Trim \Rightarrow To remove space from the data
To start and end

let str2 = " I am learning JS ";

Trim End \Rightarrow To remove space from end.

Trim Start \Rightarrow To remove space from start.

- * To Uppercase → change in uppercase
- * To LowerCase → change in lowercase

Replace → To replace ^{first} any word
 ("learning", "reading")

Replace all → Jaha Jaha wo replace
 word milega wo change kardega.

Include — To find there is word or not

Backtick

esc



Template literal

('my roll no is \$ {Variable} gkt').



Loops are used to execute a piece of code again and again

For Loop:

```
for (let i=1; i<=5; i++) {
```

```
    console.log ("Sehrlich");  
}
```

i = Block scope variable

```
for (let i=1; i<=5; i++) //  
{ document.write ("hello"); }
```

```
for (i=1; i<=50; i++)
```

```
{ document.write (i, "<br>"); }
```

Array: (Non primitive data types)

Store multiple value in single variable.



Date

Functions in JS

('Sohish'.toUpperCase)
functions definition
Parameter

functional call
argument

Arrow Function (ES6)

```
const mult = (p1, p2) => {  
    let ans = p1 * p2;  
    return ans;  
}
```

Task To perform both function and arrow function.