

1st Class of JS

Introduction to JavaScript

- * JavaScript was invented by Brendan Eich in 1995.
- * It was developed for Netscape 2 & became the Ecma-262 Standard in 1997.
- * European Computer Manufacturers Association Ecma International (formally European Computer Manufacturers Association) is an organization that develops standards in Computer & technology.
- * ES1 to ES5 (1997 to 2009)
- * After that in 2015 (major changes to follow the rules & regulations) it is called EcmaScript / ES2015 / ES6.
- * ES6 is standard for JavaScript after that every year new changes came ES7, ES8, ES9, ES10, etc.
- * JS is a light weight object oriented programming language.
- * Use in form Submit
- * in Client Side Validation
- * Popup / event on Click
- Uses
 - * Client Side execute / browsers. (JS query, React JS, angular JS)

* Website Server Side (node.js, Express.js)
Mobile Development (Hybrid App) (framework
for mobile app react native, Phone gap etc)

Software Development (Electron.js; Ex: VCode
framework etc)

for file like .js

<Script Src = "file name of JS"></script>

2nd Class of JS

Variable

* Variables just like a Container.

Example

a = 34 b = 56 a+b ← Variable
34 56 90 ← Value

* Variable declares space in memory.

is called variables declaration.

Three method of variables.

var, let, const are keyword

- Var old version of ECMAScript
- let New standard ECMAScript
- const In Modern JavaScript used let, const

Var not give error let give error)

let - not again declare (if we declare let same value

Const value not change

Const value pi = 3.142; fix value

Short cut of Console F12 key.

let is keyword we cannot declare with again
with let keyword we change value
of keyword don't write keyword again

(Semi Colon mean Sentence Complete.)

Example : let a = 5;

a = 10;

a = "hello"

last value is any Name.

last value is remain before value remove or
erase.

Console.log(name); any main print

Data types

P2-3

Number

String

boolean

null

undefined

Array

Objects

functions

Variable rules.

1) Variable Case Sensitive

Example: const abc = " ignra";

let Abc = " hello";

2) different Variable rules

Small letter is different variable

Capital letter is different variable

F1.2 Shortcut for Console

2. Variable names not should be keyword.
Some languages have keywords
for Example
Var , let , Const is keyword
We cannot var declare to variable name.
3. We can write in variable letter, small, capital
Number, dollar sing, - underscore.
There are four variable sing

Example

let a\$-\$4

4. We cannot start any variable from Number.
we only write number, Centre last, mid
but we Example

ab34

Data type

1. ~~The~~ Variable Convert Value in data type
- automatically

Example

let age = 45;

Console.log("age"); for Print age,

Console.log.(age); without " " Print value. 45

let fullName = "ijra"

Console.log("fullName")

Camel Case

We write "question" and
alphabates is called
String

```
let isPass = true;
```

```
console.log(isPass);
```

Console give value
"boden" use for yes/no
or true/false

We write mostly in keywords.

JavaScript in CamelCase.

for Example in map instead of
feelName: Name

Class 3 JavaScript

Data types, Primitive & Non-Primitive

Variables

- * Variables is just like a Container Variable is used to store information.
- * it reserves Space in memory: its data can very but memory location will always remains same

- * Variable's name can't be any keyword. eg alert Prompt etc
- * Variable is case Sensitive. Same name in Capital & small letters are different. eg Name or name (both are 2 different Variables)
- * Variables can be consist of alphabet numbers dollar Sign and underscore.
- * Variables name can't be start with digits (number) its first letter.
- * no space allowed.

AS a Good Programmer

- * Your Variable name should match with its Contents.
- * When you want 2 words join in Variable

name, So first word start with small letter ... E.g. 2nd word start with Capital letter.
e.g. fullName, rollNumber etc.

Types of Variables.

* Var

(Used before ECMAScript this type of Variable can be declare again and again in JS)
After ES6 in modern or advance JavaScript these 2 key words use for declaration variables.

* let

(its value can change any time in Programming language & can declare & assign in 2 steps.)

E.g. let name; (declare)

name = "Hello"; (assignment)

* Const

(it use for Constant value e.g. pi value)

its value can't be changed.

its a value must be assigned at the time of declaration.

E.g. const name = "Hina"; (declare and assign in same sentence)

Variable Scope

1 Block Scope Variable :-

* if variable declare in block of codes (in curly braces []), it will alive only in block & will not be accessible after curly braces.

2 Global Scope Variable:

These variables

are used globally in whole program.

Comments in JS

Single line: //let name = "hina";

multi-line: /* */

Print / Display in JS

On Browser: ← window.document.write("hina");

in Console: ← console.log("hina");

Popup: ← window.alert("hina");

document.write("iqra"); Print for browser

console.log("iqra"); Print in Console window

→ window.alert("iqra"); for Popup

alert("hina"); 2 types of alert

Taking input from users in J.S

Prompt:-

In JavaScript, we use the `prompt()` function to ask the user for input. As a parameter, we input the text we want to display to the user. Once the user presses "OK", the input value is returned. We typically store user input in a variable so that we can use the information in our program.

```
let answer = prompt("Do u want to send payment  
y/n?");
```

in `answer` variable value will be stored & you can print it.

Variables & Datatypes

in JavaScript

Date types in JS

Primitive types

Composite/Non Primitive

types

number, strings, boolean, null, undefined,

Object Arrays

Object part of array

Array inside of object

Primitive data types

To check data type
Type of VariableName

number

let rollNo = 58;

String

let name = "Iqra";

boolean

let isPass = true;

undefined

let Percentage; hot value

Object (null)

let Class = null;

function

Non Primitive Data types

1. Array :

- * Store multiple value in Single Variable
- * Values written in Square brackets []

Syntax

let info = [5, "Iqra", Computer];

• Console.log(info);

for .Print,

document.write (info);

document.write ('info [1]'); • Square bracket

info start value 0 to 1, 2, 3. that is

Called index number. in Array.

Array • Don't need keys of Array, only write part of object Array value in Array

2: Objects

Store multiple values in Single variable.
Values written in Curly brackets {} in
Pairs with keys
Syntax:

```
let Student = {  
    name: "iqra",  
    rollno: 58,  
    Class: "Computer"  
}
```

Separate for
function

- function have 2 function steps.

function called invoke.

Class 4: JavaScript Operators in js

Arithmetic Operator

Assignment Operator

Comparison Operator

Logical Operator

Conditional Operator

$a+b \leftarrow$ expression

$a+b, 4+5$ 4^5 is (operand)

2 Operand used for Add - Sub - multi
+ sing is . operators

Any text perform on data is called
operator

Arithmetic Operators

+ Use 2 values operand

+ (Addition)

- (Subtraction)

* (multiplication)

/ (Division)

% (modulus / remainder)

Unary Operators

Post increment $a++$

Pre increment $+a$

Post decrement $a--$

Pre decrement $--a$

Exponentiation

• Increment $++ \leftarrow$ Use for Singal
operator

• Decrement $-- \leftarrow$ Use for Singal
operator

Arithmetic Operators

let a = 9;

let b = 5;

//addition

document.write(a, "+", b, "=", a+b); //14

$$9 + 5 = 14$$

show in browser

write question "html tags
 tag for linebreak

2^x

3.

power

3^2

2 time x

4^4

4 time multiply

5^3

3 time x

Unary Operators

//a++, a=a+1; ← mean only 1 plus

//a--, a=a-1; ← mean only 1 subtraction

Assignment Operator (assign value)

Right side value stored in leftside variable
 $a = 2$; left side = right is called Assignment

$+ = a += 4 / a = a + 4;$

$- = a -= 4 / a = a - 4;$

$* = a *= 4 / a = a * 4;$

$% = a \% = 4 / a = \% 4;$

$** = a ** = 4 / a = a ** 4;$

$/ = a / = 4 / a = a / 4;$

JavaScript Class 5

Operations in JS

Comparison Operator

`==` (equal to)

`==` (equal to + same data type)

`!=` (not equal to)

`!=` (not equal to & data type)

`>` greater than

`>=`

`<` less than

`<=`

Comparison in data types or value

= of two types double `==` triple `===`

Comparison Operator give output result in boolean mean true/false

1 This sign is in Computer language is called! Not

`let a=2;` ^{d t} number

`let b="2";` ^{d t} string double equal check value
number

`a==b` true

check only ~~data~~ same Just like a number

`a==b` false

for perfect work `==` Same data type

`let a=2`

`let b="2"`

`a!=b` false

`a!=b` true check value & data type

JavaScript Class 5

Operators in JS

Comparison Operator

$==$ (equal to)

$!=$ (not equal to).

$>$ greater than

\geq

$<$ less than

\leq

$==$ (equal to + same data type)

$!=$ (not equal to \neq data type)

||

Comparison in data types or value

= of two types double $==$ triple $==$

Comparison Operators give output result in boolean mean true/false

! This sign is in Computer language is called ! Not

let a=2; numbers

let b="2"; string double equal check value number

a==b true check only ~~data~~ same Just like a number

a $==$ b false for perfect work == Same data type

let a=2

let b="2"

a!=b false

a $!=$ b true check value \neq data type

logical Not!
if ($!(a < b)$) give result false
inside Condition true outside write Not! Sing
! Nahi chata hai, base.)

logical Operator
logical And && Condition 1 && Condition 2
logical Or ||
logical Not !

Ternary Operator

2 values check in logical Operators then exec
Be must in logical && Condition 1, 2 true.
&& for execute Condition 1 and 2 should true
|| if any one Condition true return true.

if ($a > 2 \&\& b > 5$)

{

 alert ("both conditions are true");

}

else
 alert ("your conditions true") my² condition is
 if any 1 Conditions false give result false

logical || or

let a = 5;

let b = 6;

if ($a > 7 \text{ || } b > 5$)

{

 if any one Condition true

 Print it true

 } alert ("Hello");

else {

 } alert ("Welcome");

if 2 condition are false give result false.

Conditional Operator

if Statement

if-else Statement

if-else if Statement

if-else

Ternary Operator

condition? true Output:

false Output:

Eg:

age > 18? "adult": "not adult";

Comparison Operators

let a = 5;

let b = "5";

if (a == b)

{

document.write("hello")

} *is a block of code*

== double equal not check data type

consider double == equal

let a = 5; // *d.t.* number

let b = "5"; // *d.t.* string

if (a == b)

{

document.write("hello")

} *we write in blocks of code*

==== Triple equal check same data type gives result true

if data type not psi same give result empty

! Not. Kya karta hai ap kay Comparison ko Mat
let a=5; // d.t number
let a=5; // number
if (a!=b)
{
document.write("hello")
}

a is not equal to b give result empty because
not data type is same & not show result in browser

! Not give result Opposite

```
let a=5; // d.t number  
let b="5"; // String  
if (a!=b) 5=-5  
  
document.write("hello")  
}
```

a is not equal to b with data type.
5=-5 not equal because ^{this} data type "5" number,
and String then show result in browser
& print ("hello")

Use for Condition

Check

let age = 70
 if (age \geq 18) {
 less sign
 we can change age value
 document.write ("you are child");
 }

else if (age > 60)

{
 document.write ("you are old");
 }

else {

 document.write ("you are young");
 }

We use Comparison Operators in Condition Operators

> greaterthan

=

< less than

\leq

Ternary Operators

Only 1 line code

Check logical Condition in Ternary Operator

let age = 20; we can change value

let result;

Syntax result = age \geq 18? "adult": "not adult";

alert (result);

print in browser

Question? Mark for check condition.

JavaScript Class 6

Assignment Mark Sheet

JavaScript Class 7

String Manipulations

String

- * String is a sequence of characters used to represent a text.
- * It is a primitive data type
- * We can create string by using template literals E.g. Single & double quotations.

String Creation & manipulation

- let Str1="I am learning js"; //double
 - let Str2='I am learning js' // Single
 - let Str3=`I am learning String template` //template literal
- key : in keyboard called backtick.

Template literal

Template literals are a feature in JavaScript that were introduced with ESG. They give you a more flexible & maintainable way of working with strings in JavaScript.

How to use template literal

- for next line \n
- for tab (space) \t

Show Console.log

- * for Print } in String //
- * for Write Variable in String \$ (variable name)
- * for double quotation "hello" | "Hello"

let str = 'Hello! How r u?';

let num = 5;

document.write('Hello \$ {num}'); work for template literal.

Some String Properties & Methods

- * let str1 = "I am learning js";
- * let str2 = "CSS";
- * let str3 = "html";
- * Position/index start with 0 in string.
- * to find length str.length
- * to join strings
- * document.write(str1 + str2) / (str1, "", str2)
if we knew about length we write L use for length
let L = str.length;
- document.write(L); show length in browser
- if we knew about which word position.
we write document.write(str[9]);
use square bracket start go index position 0

+ Use for Concat

* by Concat()

let str4 = str.concat(str2) / str.concat(str2, str3).
document.write(str4)

Concat is a function or method use to join 2 string

Syntax:

let str = "Hello! world"

let str2 = "I am learning javascript";

document.write(str + str2) if you want space

str + " " (str + " " + str2) for space

let newvar = str.concat(str2, str3);

str.concat mean Join 1st String with 2nd
String depend on you you add many variable
add , quona

Concat mean in JavaScript write line wise

document.write(newvar);

let str = "I am learning javascript";

- str.trim() // to remove space from start & end.
- str.trimStart() // remove space from start
- str.trimEnd() // removes space from end
- str.toUpperCase() // change in Uppercase write in Capital letters
- str.toLowerCase() // change in lowercase write in small letters
- str.replace("javascript", "html") // search word & replace (case Sensitive)
- [str.includes("is")] search word is/if not found return false
document.write(str3.includes("are")); give result true & false

let str3 = "I am learning JavaScript Learning";
document.write(str2.replace("learning", "reading"));
replace only first word show on browser
I am reading javascript learning.
2nd learning word show as it is

Syntax: document.write(str2.replaceAll("learning", "reading"));
All mean when learning word found in text
replace it from reading

let str3 = "I am learning JS. it is very easy language";
document.write(str3.slice(5, 10)); 5 to 10 words pick
index start from 0

Class 8 JS Switch Statement

String Loop

Switch Case break These three words are in stat

- Let reply = prompt("Do you want to continue...");
 - Switch(reply), write variable with Switch
 - Case "Yes", & we write possible answer
document.write("Continue");
break;
- if value true & print Continue. & then do break
Break mean out of loop

Syntax if (reply == "y") {
document.write("Continue"); }

if & switch case do same work.

Just like a if else statement

default:

document.write("Wrong input"); } since give wrong input

Template literal / String Template.

Syntax:

- normal string: document.write("my roll no is ", variable);
 - template literal: document.write(`my roll no is \${variable}`);
- We use template literal \$ sing & variable write
in {} bracket complete code write in backtick

Loops

- to execute a piece of code again & again
- finite loop & infinite loop
- finite loop (ending point)
- infinite (not end) memory full / Computer hang.

for loop

```
for(let i=1; i<=5; i++) // i is block scope variable  
{  
    document.write("Hello");  
}
```

 | |
 | | | is for iteration/counting.
 | | | first step initialization
 | | | 2nd Condition check jab
 | | | tak condition true block
 | | | of code execute
 | | | 3rd step updation

```
for(let i=50; i>=1; i--) ← back work Counting  
{  
    document.write(i, "<br>");  
}
```

for table format

```
let tabno=prompt("Enter Table no. of your choice");
```

```
for(i=1; i<=12; i++)
```

```
{
```

```
    document.write(`${tabno} X ${i} = ${tabno * i}); //
```

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

```
}
```

Prompt change into integer is a number

1, 2, 3

Class 9 JS

Loop & Array

Array (Non Primitive data types)

- Store multiple value in Single Variable.
- Values written in square brackets [].
- Values separated by comma.
- Each position is called "index".
Each value call through index number. Start with 0
eg arr [0], arr [1].

Syntax:

```
let info = [5, "hina", "computer"]; console.log(info);  
info = [] // to empty array
```

```
let arr = [5.8, "Igra", "JavaScript"];  
document.write(arr[0]); // check for index, value we write in []  
let l = arr.length - 1; // for check length - 1 because length is 3  
for (let i = 0; i <= l; i++) // so we will do -1  
{  
    document.write(arr[i], "<br>");  
}
```

- for of loop

```
for (let value of arr);  
    { document.write(value); "<br>" tag use for new line  
    for of loop automatically print variable line
```

Use run with array

& we don't need write index number in for of loop

Properties & Methods in Array

- Let book = ["Maths", "English", "Urdu", "Physics", "Computer"];
1. book.length // length of array
 2. book.push("Chemistry"); // add word in array in the end
 3. book.unshift("Chemistry"); // add in start of array
 4. book.shift(); // remove a word from start
 5. book.pop(); // to remove word from last & return
update array
 6. book.toString() // convert array in String
 7. book.concat(book2) // join 2 or 3 array in new array
don't change original array
- syntax
arr.push("html"); // item inser in end:

let arr = [34, "Sara", "JavaScript", "Computer"];

document.write(arr.slice(2), "box"); return a slice
Piece of array

document.write(arr.splice(1, 2));

arr.slice

arr.splice

Class: 10 JS

Method & loop / for in / while / do
while loop

Object

store multiple value in single variable.

Values written in curly brackets {} in
Pairs with keys

Syntax:

```
let student = {
```

```
    name: "igra",
```

```
    rollno: 23,
```

```
    Class : Computer
```

```
}
```

```
document.write(student);
```

```
document.write(student.rollno);
```

- for in loop print keys

for of loop pick array and one by one

Print value

```
let arr [1,2,3,4,5,6];
```

```
if (i%2 == 0) {
```

Even number print

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

```
}
```

While & do while loop

- ~~With user~~ for loop just like a while loop

```
let i=1
```

```
while (i<=10)
```

```
{
```

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

```
i++
```

} - check condition in first after condition

- wrong not enter in loop, if condition right loop work

Do while loop

let i = 11

do

{

document.write(i, "Irra"
);

i++;

} while (i <= 10);

do give order

Do while loop must execute one time
do while loop

1) Print 1st time execute

2) Check Condition after

execute

if any condition do time print we use
do while loop.

Class: 11 JS function

Define / function invoke

function in JS.

A JavaScript function is a block of code designed to perform a particular task.

A JavaScript function is executed when "Something" invokes it (call it).

We recognized function have 2 () pa

function have two steps.

function in JS
function definition (define)

Function frame ()

```
{  
    document.write("igra");  
}
```

function invoke
(call)

frame (); Print your code
if you want 150 time

• frame ();

Function in JS

function definition

```
function fname (p1, p2)
{
    document.write (p1+p2)
}
```

p for parameters
in define function

function call

fname (arg1, arg2)
or

fname (2,3)

arg for arguments
in function call