

~~1st class  
Mr. Hira~~

- ~~Introduction to JavaScript~~ \*  
• Javascript was invented by Brendon Eich in 1995.  
• It was developed for Netscape 2, and became the ECMA-262 standard in 1997.

- Ecma International (formally European Computer Manufacturers Association)  
European Computer Manufacturers Association) is an organization that develops standards in computer technology -  
• ESI to ESS (1997 to 2009)-  
• After that in 2015 (major changes to follow the rules & regulations) this is called EcmaScript / Es 2015 / ESB.  
• ESB is standard for javascript after that every year new changes come ES7, ES8, ES10 etc -  
• JS is light weight object oriented programming language -  
• use in form submit.  
• in client side validation -  
• popup / events on click -  
User :- • Client side execute browser - (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 - vscode, framework etc)

Script.js file kba kr heads or <body> dono me link kri skte -  
Lekin head section me link kong se nukсан ye hoga k memory file  
load hoga or work fast nahi hoga - is ge behin hik body section  
me closing tag se pahly js ki file link karain -

<title> first class JS </title>  
 <head>  
 <body>

<script src="script.js"> </script>  
 </body>  
 </html>

browser project  
 right click context menu  
 click inspect → inspect window  
 ne open console window & type  
 hoing ge console window & type  
 hoj, console window & type  
 from search engine  
 on reading script  
 known

Js k sb error console me find kar sktey-

F12 shortcut key for console window...

## VARIABLE || RULES / Declaration

### Variables

- var (old version)
- let
- const [modern JavaScript]

Variable banay se memory me 1 space reserve ho jati hai - koi ne kis value store kare kya & ye aik container ki form hoti - Container aik Variable hai container me jo date wo value urki

Let me value change b ho skti; lkn const ki value change nhi ho skti.

**DATA TYPES:** \* Number \* String \* boolean \* null \* undefined  
 \* Array \* Objects \* functions

## VARIABLE Rules :-

- ① Variable me ~~4~~ chizain likh skiy aF-\$b  
② dollar sign ③ underscore ④ alphabet capital & small ⑤ Number
- ② Variable kisi b number se start nahi hoga
- ③ Variable keywords nahi ho skiy like const, var, alert, console, etc
- ④ ye Variable ka nam nahi skhi skiy -  
String hamesha quotation " " me likhiy hain
- ⑤ Jis b j name milta us likhiy to camelcase me likhiy (fullName)
- ⑥ Boolean me True or False || Yes or No

## 3rd Class: 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 vary but memory location will always remain same.

### Naming Variables in JavaScript :-

#### Rules:

- Variable's name can't be any Keyboard-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 & underscore
- Variable name can't be start with digits (number) in 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 & 2<sup>nd</sup> word start with Capital letter -

egz fullName, rollNumber etc

#### TYPES OF VARIABLES:

- Var: (used before EcmaScript - Their type of variable can be declared again & again in js)-After Es6 in modern or advance javascript There 2 keywords use for declaration variables.

- \* let: its value can change any time in programming language and can declare and assign in 2 steps  
eg `let name; (declare)  
name = "Hello"; (assignment)`
- \* const: it is used for constant value (e.g. pie value) - its value cannot be changed - its value must be assigned at the time of declaration  
eg `const name = "Hina"; (declare & assign in same sentence)`

### VARIABLE SCOPE

1. Block scope variable: if variable declared 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 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");
```

## Taking input from Users in Js :-

\* **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 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.

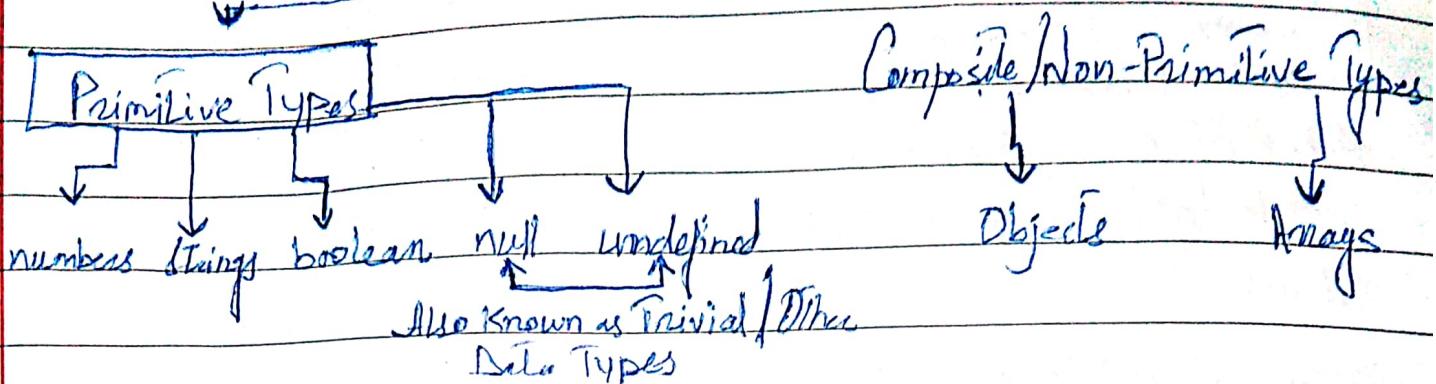
Variable and data type : js

let answer = prompt("Do u want to send payment Y/n?");  
in answer Variable value will be stored & you can print it.

```
let age = 45;  
console.log("age");  
console.log(age);  
let fullName = "Hina";  
console.log(fullName);  
let isPass = true;  
console.log(isPass);  
let roll = null;  
console.log(typeof roll);  
let lastName = prompt("enter last name:");  
window.console.log(lastName);  
window.document.write("Hello");  
document.write("World");  
console.log("Hello world");  
window.alert("Hina");  
let num = prompt("choose any number: 1-10");  
document.write(num);
```

# Variables & Data Types in JavaScript

## Data Types in JS



Primitive Data Types: To check dataType → Type of variable name

- number      let rollNo = 6;
- string      let name = "Hina";
- boolean     let isPass = true;
- undefined    let PercentAge;
- object(null) let class; null;

Non Primitive Data Types:

- 1- array \* Stores multiple value in single variable,
  - \* Values written in square brackets [ ].
- Syntax:

```
let info = ["Hina", "Tree"];
console.log(info);
```

Print  
 document.write(info);  
 document.write(info[1]);

```

// number
let age = 55;
document.write(age);
console.log(typeof age);
let name = "Amina";
name = "Kiran";
document.write(name);
console.log(typeof name);
let isPass = true;
document.write(isPass);
console.log(typeof isPass);
let clas;
document.write(clas);
console.log(typeof clas);
let abc = null;
document.write(abc);
console.log(typeof abc);

```

```

// Object
let item = {
    name: "lipstick",
    Price: 20,
    rating: 4,
    available: true,
    offer: 10,
}
console.log(item);

```

## ② Object:

- Stores multiple values in single variable-
- Values written in curly brackets {} in pairs with Keys-
- Syntax:

```

let student = {
    name: "Hima",
    rollno: 20,
    class: "Maths",
}

```

```

document.write(student);
document.write(student.rollno);
console.log(typeof student);

```

# OPERATORS IN JS :

Class 4th

- \* Arithmetic Operator
- \* Assignment Operator
- \* Comparison Operator
- \* Logical Operator
- \* Conditional Operator

function welcome()

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

function sum(a,b)
{
  let c = a+b;
  return c
}

welcome();
let answer = sum(4,5);
document.write(`answer`);
document.write(`type of sum`);
```

// Arithmetic Operators

let a = 9;

let b = 5;

// addition

document.write(a+b); //

## ARITHMETIC OPERATORS:

- + (Addition)
- (Subtraction)
- \* (Multiplication)

/ (Division)

% (modulus/remainder)

Exponentiation

- Increment
- Decrement

## Unary Operators

Post increment a++

Pre increment ++a

Post decrement a--

Pre decrement --a

## // Arithmetic operators

let a = 5; let b = 3;

// a++;

document.write(a--);

document.write(a); //

// b---;

document.write(b);

// a++, a = a + 1;

// a--, a = a - 1;

document.write(++a); // pre-increment

document.write(a); //

// addition, Subtraction, multiply, divide & Modulus

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

// document.write(a, "-", b, " = ", a-b, "<br>"); //

// document.write(a, "\*", b, " = ", a\*b, "<br>"); //

// document.write(a, "/", b, " = ", a/b, "<br>"); //

// document.write(a, "%", b, " = ", a%b, "<br>"); //

// document.write(a, "\*\*", b, " = ", a\*\*b, "<br>"); //

#.

## Assignment Operator (assign value)

=      a = 2;    left side = right

+ =      a += 4/a = a+4;

- =      a -= 4/a = a-4;

\* =      a \*= 4/a = a\*4;

% =      a %= 4/a = %4;

\*\* =      a \*\*= 4/a = a\*\*4;

// Assignment operator

let a = 8;

// a += 4;

// document.write(a); //

a \*\*= 3; // 8 \* 8 + 8

document.write(a); //

Class 5<sup>th</sup>OPERATORS IN JSComparison Operator $=$  (equal to) $==$  (equal to + same data type) $!=$  (not equal to) $!=$  (not equal to & data type) $>$  greater than     $>=$  greater than or equal to     $<$  less than     $<=$  less than or equal to

let a = 2; // number    b = "2"; // string

a != b    false

a != b    true

Comparison OperatorLogical Operator : (<sup>Condition</sup><sub>1 & 2 condition</sub>)

let a = 5; // number

let b = "5"; // String    5 == 5

if (a == b) / (a != b)  
{ document.write("Hello") }

- Logical AND &

- Logical OR ||

- Logical NOT !

let age = 23; // number

if (age &gt;= 18)

{ document.write("I can vote"); }

else { document.write("I can't vote") }

Conditional Operator:

(if age == 23; if (age &lt; 18)

{ document.write("I'm a child"); }

else if (age &gt; 60)

{ document.write("I'm old"); }

else { document.write("I'm young") }

Ternary Operator

condition? true output: false output;

e.g:

age &gt; 18? "adult"; "not adult";

## // Logical Operator

```
let a=5; let b=6;  
if (a>7 & b>5)  
    {alert("both condition are true");}  
else { alert ("one conditions True");}
```

```
if (a>7 || b>5)  
    {alert("Hello");}  
else {alert("welcome");}
```

```
if (!(a<b)) // 1 condition True  
    {alert("Hello");}  
else {alert("welcome");}
```

```
if (!a>b) // True
```

## // Ternary Operator / Logical

```
let age = 20; let result;  
result = age > 18 ? "adult": "not adult";  
alert(result);
```

## Assignments Using Operators

```
let name = "Hina";  
let tname = "Miss Orchard";  
let cla = "8th";  
let emarks = 78;  
let umarks = 89;  
let pmarks = 78;  
let cmarks = 58;  
let imarks = 90;  
let Total = emarks + umarks + pmarks + cmarks + imarks;  
let per = Total / 500 * 100;  
let grade;  
if (per <= 100 && per >= 89) // 90 - 100  
    {grade = "A+";}  
else if (Per <= 90 && Per >= 80) // 80 - 89  
    {grade = "A";}  
else if (Per <= 80 && per >= 70) // 70 - 79  
    {grade = "B";}  
else if (per <= 70 && per >= 60) // 60 - 69  
    {grade = "C";}  
else if (per <= 60 && per >= 50) // 50 - 59  
    {grade = "D";}  
else {grade = "F";}
```

```

11 now create marksched
document.write("Roll No:", rollno);
document.write("<br>Name:", name);
document.write("<br>Class:", clz);
document.write("<br>Teacher Name:", Tname);
document.write("<Table border=1>");
document.write("<tr><th>Subject</th><th>Marks Obtained</th><th>Set of Marks");
document.write("<tr><td>English</td><td>85</td><td>100</td>"); 
document.write("<tr><td>Urdu</td><td>78</td><td>100</td>"); 
document.write("<tr><td>Physics</td><td>92</td><td>100</td>"); 
document.write("<tr><td>Computer</td><td>88</td><td>100</td>"); 
document.write("<tr><td>Islamiat</td><td>75</td><td>100</td>"); 
document.write("<tr><td>Total</td><td>420</td><td>500</td>"); 
document.write("<tr><td>Percentage</td><td>84</td><td>100</td>"); 
document.write("<tr><td>Grade</td><td>A</td><td>100</td>");
```

Class 7th:

### Switch Statement / STRING Manipulations

```

• let reply = prompt("Do you want to continue...?");

• switch(reply)
  { case "Y":
    document.write("Continue");
    break;
  case "y":
    document.write("Continue");
    break;
  case "N":
    document.write("End");
    break;
  default:
    document.write("Wrong input");
  }
```

```

let str = "Hye! how r u? ";
let str2 = "I am learning AI ";
let str3 = "World"; // It is very easy language;
// let l = str.length; // Length
document.write(str.length);
document.write(str[18]); // // index
document.write(str + str3 + str2);
let newVar = str.concat(str3, str2); // concatenations
document.write(newVar);
// document.write(str.trim()); // remove space from start & ending
// document.write(str2.trimStart());
// (str2.trimEnd());
(str3.toUpperCase()); // change in upper case
(str2.toLowerCase()); // change in lower case
(str2.replaceAll("learning", "reading"));
(str3.includes("are.")));
document.write(str3.slice(5, 10));

```

## String Creation & manipulation

- let str1 = "I'm learning JS"; // double
- let str2 = 'I'm learning js'; // single
- let str3 = `I'm learning string Template`; // Template literal (adjacent T)
 ↪ key in keyboard -

## Template Literal

Template literals are a feature in JavaScript that were introduced with ES6. They give you more flexible & maintainable way of working with strings in JavaScript.

### How To use Template Literal?

- for next line
- for Tab (space)
- for print \ in string
- for write variable in string \$ {variable name}
- for double quotation "hello" 'hello'

## METHOD & LOOP /FOR, IN / WHILE / DO WHILE

Splice :-

Let array = [0, 1, 2, 3, 4, 5, 6, 7, 8];

→ .splice(1, 3, "a", "b") →

## String

**Class 8<sup>th</sup>**

- \* String is a sequence of characters used to represent a text.
  - \* It is a primitive data type -
  - \* We can make string by using Template Literals and in single double quotations-
    - Let str = "I am learning js"; // double
    - Let str = `I am learning js` ; // single
    - Let str = `I am learning string template` ; // template
- literal adjacent to 1 key in keyboard called backtick.

## Template literals / String Template

Template literals are a feature in Javascript that were introduced in ES6. They give you more flexible & maintainable way of working with strings in Javascript.

We can write variable in string if string is string Temp

Syntax:

- \* normal string : document.write("my roll no is "variable," if i fill
- \* Template literal : document.write(`my roll no is \${variable}`)

let num1 = 56;

let num2 = 66;

document.write(`The sum of \${num1} and \${num2} is \${num1 + num2}`)

browsed →

The sum of 56 and 66 is 122

## String manipulation

- \n use for print Text in new Line
- \t use for spaces (tab) more than single space in Text
- \" use for single \ in Text
- \\" use double quotations in string

## Some String Properties And Methods :

- Let str1 = "I am Learning JS";
- Let str2 = "CSS";
- Let str3 = "html";
- position      str[0], str[1] (first index start with 0)
- To find length      str.length
- To join strings      document.write(str1 + " " + str2)  
document.write(str1, " ", str2)
- by concat()      Let str4 = str.concat(str2) / str.concat(str2, str3)  
document.write(str4)

Let str = "I am Learning javascript";

- str.trim() // To remove space from start and end.
- str.replace() //
- str.toUpperCase() //
- str.toLowerCase() // change in lowercase
- str.replace("javascript", "html") // search word and replace case sensi  
tive
- str.includes("is") // search word is / return boolean / if not found  
return -1

## LOOPS

- To execute piece of code again and again -
- finite loop and infinite loop -
- Finite Loop (ending point)
- Infinite (not end) memory full / computer hang

## For Loop

using ~~roshan kumar~~

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

i is block scope variable. use for iteration

first step initialization.

2nd condition check jaisa Tak condition

true block of code execute-

3rd step update-

~~class~~

## ARRAY (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", "complex"]; console.log(info);
info = []; // To empty any array
```

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(); // remove word from last & return updated 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 -

### array.js

```
// for (let i=0; i<=1; i---)  
{}
```

```
let arr = [34, "Sadia Adil", "JavaScript", "computer"]
```

```
// let i = arr.length - 1;
```

```
// for (let i=0; i<=1; i++)
```

```
{
```

```
// document.write(arr[i], "<br>");
```

```
}
```

```
// for of Loop
```

```
for (let value of arr)
```

```
{
```

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

```
}
```

```
// let arr = [34, "Sadia", "Java", "computer"];
```

```
arr.push("html"); // item insert in end
```

```
document.write(arr);
```

### array - js

```

let arr = [34, "Sadia", "Java", "Computer"];
document.write(arr, "<br>");
arr.push("HTML"); // item insert in end
document.write(arr, "<br>");
arr.unshift("CSS"); // add item in start
document.write(arr, "<br>");
arr.shift(); // remove item in start
document.write(arr, "<br>");
arr.pop(); // remove item from end
document.write(arr, "<br>");
arr.join(); //
document.write(arr, "<br>");

let arr3 = arr.concat(arr2); // 
document.write(arr3, "<br>"); // document.write(arr3.indexOf("Computer"));
+-----+ +-----+ +-----+ +-----+
• book.indexOf("Hello"); // to find any word's position in array
• book.slice(startIdx, endIdx); // don't change in original array
// return a slice piece of array
• book.splice(startIdx, deleteCount, next); // change original array
// add, remove and replace.

```

e.g. let arr = [1, 2, 3, 4, 5, 6, 7];

arr.splice(2, 2, 55, 56); // 2 index per 3 // 3 will remove // update  
// Then 55 & 56

```
arr.splice(3, 0, 44); // if you don't want to delete any item // 3 index  
per joker add 44  
aa.splice(2, 3); // if you want to delete items // 2 index & items  
delete 3
```

### Class 11:

- A Javascript function is a block of code designed to perform a particular Task.
- → " . . . . . (Calls)"

### FUNCTIONS IN JS

• for write variable in "hello" "hello"  
• for double quotation "hello" "hello"

Class 10<sup>th</sup>: METHOD & LOOP FOR, IN / WHILE / DO WHILE

// Splice :-

```
let array = [0, 1, 2, 3, 4, 5, 6, 7, 8];
document.write(array.splice(1, 2, 98, 99), "  
");
document.write(array);
```

// For in Loop

const student =

```
{ name: "fatima",
```

```
rollno: 55,
```

```
Subject: "Computers"
```

```
for (let key in student)
```

```
{ document.write(key, student[key], "  
"),
  (key: ", student[key].
```

for in Loop method

// for of loop method

```
let arr = [1, 2, 3, 4, 5, 6];
for (let i of arr)
{ document.write(i, "<br>"); }
```

// for of loop

```
let arr = [1, 2, 3, 4, 5, 6];
for (let i of arr)
{ if (i % 2 == 0)
  { document.write(i, "<br>"); }}
```

// while loop for (let i=1; i<=10; i++)

```
let i=1
while (i<=10)
{ document.write(i, " <br> ");
  i++; }
```

// Do while loop method:

```
let i=21;
do
{ document.write(i, " <br> ");
  i++;
} while (i<=10); // false
```

// let i=1; let sum=0;

```
do
{ document.write(i, " <br> ");
  sum = sum + i;
  i++;
}
```

} while (i<=10); // false => document.write sum of 1 to 10 = "sum";

delete 3

## Class 11:

### Javascript Function

- A Javascript function is a block of code designed to perform a particular task.
- Javascript function is executed when "something" invokes it [calls it].

e.g. `document.write("china".toUpperCase());`

Function definition (define)

`function fname()`

{  
  block of codes  
}

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

`- function sum (p1, p2)  
{`

`ans = p1 + p2;`

`return ans;`

`}`

Function invoke (call)

`• fname();`

`fname (arg1, arg2)`

or

`fname (2, 3)`

`let ans = sum(2, 3);`

`console.log(ans);`

## Functions JS

// function define

```
function mult (P1, P2) {  
    let ans = P1 * P2;  
    return ans;
```

}

```
let a = 6;
```

```
let b = 8;
```

// call / invoke

```
let ans = mult (a, b);  
document.write (ans);  
let abc = mult (7, 4);  
document.write (abc);
```

~~know Function~~

```
const fname = () => {  
    fname ()
```

block of code

```
} // arrow function esc
```

```
const mult = (P1, P2, P3) => {
```

```
    let ans = P1 * P2 * P3;
```

```
    return ans;
```

```
}
```

```
let a = 9; let b = 8;
```

// call / invoke

```
let ans = mult (a, b, b);
```

```
document.write (ans);
```