

AWEEN ANJUM BUKHARI

Class: JavaScript

Roll No: JS2-067

TEACHER: Ma'am HINA IMTIAZ

MENTOR: SIR FURQAN BEHLIM

(Notes → JavaScript)

Class 1:-

"Introduction of Javascript"

- * JavaScript was invented by **Brenden Eich** in 1995
- * It was developed for **Netscape 2** and became the **ECMA-262** standard in 1997.
- * **European Computer Manufacturers Association (ECMA) International** is an organization that develop standards in computer and technology.
- * **ES1 — ES5 (1997 – 2009).**
- * After that in 2015 (major changes to follow the rules and regulations) this is called (**EcmaScript / ES2015 / ES6**) (Imp)
- * **ES6** is standard for javascript after that every year new changes came **ES7, ES8, ES10 etc**
- * **Js** is a light weight object oriented programming languages
- * use in form submit \Rightarrow min. 3 letters name is must in it
- * in client side validation
- * popup / events on click.

Uses :-

- * **client side execute / browser** (Js query, React Js, angular Js) app names.
- * **Website server-side** (Node Js, Express Js).
 \downarrow
use at **server side**

Mobile development (Hybrid App) (framework)
for mobile app react nativ, phone gap etc.
Software development (Electron Electron.js,
Ex-vscode, framework etc)

As we can use `<tags>` in Html and Css
In Css we use

```
<head>  
  <title>document</title>  
<style>  
  </style>  
</head>
```

But in Js we can also use tag and that is
`<script>` tag we can tie it in head
section and also in body section as
well.

```
<head>  
  <title> </title>  
  <script> alert("hello 2"); </script>  
</head>  
  
<body> OR  
<script>  
  alert("hello 1"); </script>  
</body>
```

(alert is use for
its Pop up
Pop up)

"How to link Js file with html file.
create new file as (script.js)

```
<head>  
  <title> </title>  
  <script src="script.js"></script>  
</head> * (may b memory will loaded)  
If we use defer as an attribute in our script opening tag the browser will read our js file in last  
OR
```

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

* must use before body closing tag

* we can start coding without any tag use in our js file

* We can't execute file our js file just like

* this attribute will read html file first then read script in the end
to our html file to execute or open live server

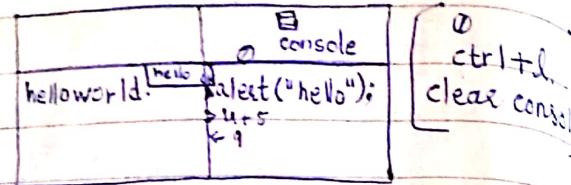
* `<script defer src="script.js"></script>`
we can use defer so js file will read at the end but big organizations didn't write it.
we must use js file in body tag.

Class 2 :-

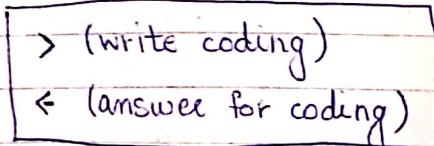
Variables || Rules/Declaration & Assignment

"How to open console window"

Browser → write click → inspect debugging + errors



- * In console window we can write our js code.
- * Also if we write our codes anywhere in html, js our errors will show on console window.



- * name must be of three letters minimum.
- * Special characters \$ @ # %
- * pop up / event on click we use js
We use

internal Js

We can write script on the head and also body

External Js

We make another file and ^{link} JS file before </body> closing body tag.

- * We declare our variables in 3 ways in JS
- * var, let, const
- * var was used before 2015 before ES6

Old version

var
(before ES6)

Modern JS

let, const
(In modern JS we use const and let)

Variables ⇒ creates space in memory as a name of variable.

* Variable is a container which stores a value.

Var

* We can declare var again and again and that is its week-point.

Let

* We can't declare let again n again const, const will not change in any case. Value should be constant in it.

"=" is called assign operator.

a = "Aween" ⇒

variable ⇒ store in lift variable location

a = "Aween"
stored → Assign operator → Value.
in variable memory

if Const will written without any value
it will must give error.

* 3 key words for declare variables

Var var ko reject kr dia gaya q k yeah
bar bar declare krnay pee koi error nahi da.

Let we can change value again and
again.

Const the value should be constant in it.

* Variable Rules

- Case Sensitive
- No Key words used as a variable.
- Don't start with a number & variable must start with 3 things.
- (letter, number, -, \$)

1:- const abc = "hello"
let ABC = "hi".

2:- key words ko hum variable nahi takh sakta as a name.

3:- Variable must not start with number.

4:- letter, numbers, -, \$ used as a variable

& we can start variable name with these as will other than number.

* data types

best way to write
variable is Camel

Case

Yes or No, true or false

empty k datatype is

Number

String name = "Aween";

Boolean isPass = true;

null rollNo = "";

undefined.

array

objects

functions

Camel Case

e.g (isPass)

starts from small

letter the next

letter starts with

cap.

Js file :-

→ let age = 45; // numbers //

✓ console.log(age); → as it is print kray ga.

X console.log("age");

→ let fullName = "Aween"; // String //

console.log(fullName);

→ let isPass = true; // boolean //

console.log(isPass);

document.write(isPass);

→ let rollnum = "?"; → this place should b empty thots datatype is // null //

Variables :-

Var

let

const

Class 3:-

Data types + Primitive + Non-primitive

Variables :-

- Variables is just like a container.
- Variables is used to store information.
- it reserves space in memory. its data will vary but memory location will always remains same.

e.g namak ya cheeni ka
container. jo k same
rehta hai lakin jo main
us main dal saktey
hain.

Naming Variables in JS:-

Rules:-

- Variables name can't be any keyword.
e.g alert, prompt etc.
- Variable is case sensitive. Same name in capital and small letters are different e.g:- name or Name (both are 2 different variables).
- Variables can be consist of alphabet, number, dollar sign and underscore.
e.g (aA, 45, \$, _)
- Variable name can be start with digits (numbers) its first letter.

- No space allowed in variable.

As a good programmer:-

- Your variable name should match with its contents.
- When ~~variable name~~ you wants 2 words join in variable name, so first word start with ~~start~~ small letters and 2nd word start with capital name or letter

- Name, full Name, roll Numbers etc
- Mostly variables can be write in Camel Case e.g rollNo=45;

Types Of Variables:-

→ Var :- (Used before EcmaScript, this type of variable can be declare again and again in JS). After ESG in modern and advance Javascript these two words use for declarative variables.

→ let :- its value can change in anytime in programming language and can't declare and assign in 2 steps
e.g

let name; (declare)

name = "hello"; (assignment)

Const:- (it use for constant value e.g. value). its value can be changed.

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

e.g const name = "Aween";

(declare and assign in same sentence)

* Comments in JS

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

Multi-line: /* */ this line will not execute.

Variable Scope:-

1- Block Scope Variable

If variable declare in block of codes (in curly braces {}).

it will alive only in block and will not be accessible after curly braces.

2- Global Scope Variable

These variables used globally in whole programme.

* Print/Display in JS

- On Browser

its an object
window.document.write("hina");

- In Console

console.log("hina");

- popup

window.alert("hina");

window — object

document — Sub-object

write ("hello world"),

Taking Input from user 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 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.

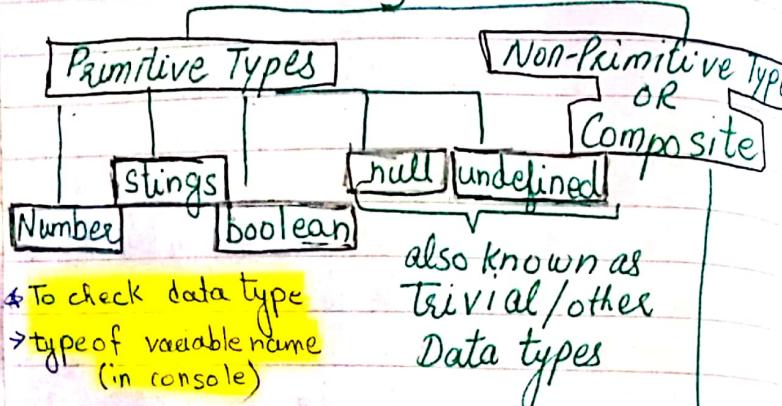
* let answer = prompt

(do you want to send payment y/n?:'

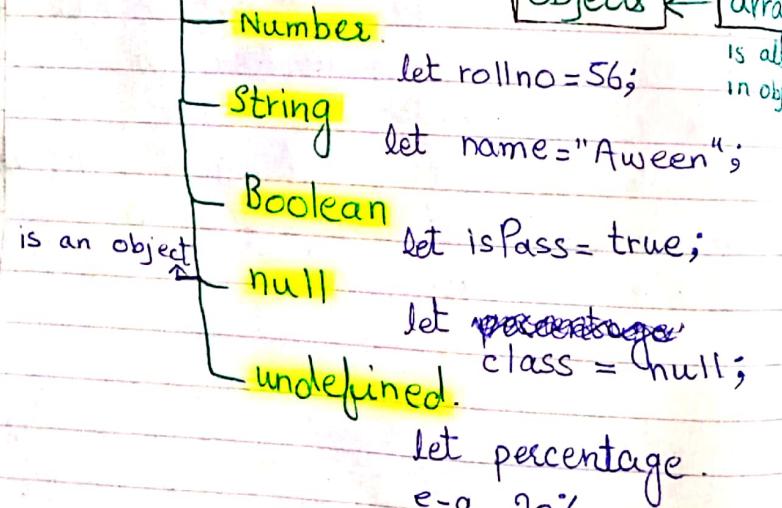
* In answer variable value will be stored and you can print it.

Variables & Datatypes in JS.

Data Type in JS.



Primitive Types -



We can't use keywords as a variable name.

Non Primitive Data Types

- 1- **Array**:- (array will be shown as object)
- Store multiple values in single variable with commas in b/w.
 - Values written in square brackets []

Syntax:-

```
let info = [15, "Aween", "Javascript"];
console.log(info);
```

To Print

document.write(info);

OR

document.write(info[1]);

multiple data store in commas.

index Posi: one will be declared

Aween

2- Object :-

- Store multiple value in single variable
- Values written in curly brackets {} in (pairs with Keys)

Syntax:-

```
let student = {
  name: "Aween",
  rollno: 67,
  class: "Javascript"
}
```

To Print

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

Class 4:-

Operators in JS.

operator (use to perform some operation on data.)

→ Arithmetic operator.

→ Assignment operator.

→ Comparison operator.

→ Logical operator.

→ Conditional operator.

Function :- (object hai).

interpreter will not read our function unless we call it in. or we invoke it

invoke ⇒ call → we can call it in HTML
as well as JS.

we define function as "function" and its name. otherwise it will not execute.

e.g. function welcome()

JS {

file document.write("welcome");

① welcome(); (another way is)

② function sum(a,b) {

 let c = a+b;

 return c

}

welcome();

let answer = sum(4,5);

document.write("answer");

document.write(typeof welcome);

document.write(typeof sum);

view
welcome function
a function

Class 5:

Operators (K1 2nd class)

3- Comparison Operators:- comes in bracket

To compare 2 values. If they are equal then start working. If they are not equal then start working also.

4- Logical Operators :

both cond. should be same

AND \Rightarrow we can check 2 conditions
In it e.g: (cond1) & (cond2)

|| (logical OR) \Rightarrow if one cond true then it will work
() || () also check 2 condition

! (logical NOT) \Rightarrow if condition is true then ! will become it (not true)
*(عَنْ تَوْكِيدِ الْمُؤْمِنِ الْمُؤْمِنِ)

5- Conditional operators

if statement working in block of code
if-else statement else
if-else if statement

Smarter way to learn JS
mdn documents
W3 school

geeksforgeeks.org
Step by step guide

Variables

Control flow

Loops

Operators

Functions

Events

Arbav

string

Class 6.

Assignment #1

```

let rollno = 58;
let name = "Aveen";
let tname = "Maam Hina";
let cla = "8th";
let emarks = 78;
let umarks = 91;
let mmarks = 95;
let cmarks = 88;
let pmrks = 90;
let total = emarks + umarks + mmarks + cmarks + pmrks;
let per = total / 500 * 100;
let grade;
if (per <= 100 && per >= 89)
    {grade = "A+";}
else if (per <= 90 && per >= 80)
    {grade = "A";}
else if (per <= 80 && per >= 70)
    {grade = "B";}
else if (per <= 70 && per >= 60)
    {grade = "C";}
else if (per <= 60 && per >= 50)
    {grade = "D";}
else {grade = "U";}

```

"quotation"
(" ")

("
Roll No:", rollno)

// now create marksheet //

```

document.write("Roll No:", rollno);
document.write("Name:", name);
document.write("Class:", cla);
document.write("Teacher Name:", tname);
document.write("<table border=4>");
document.write("<tr><th>Subjects </th><th>Marks Obtained </th><th>Percentage </th></tr>");
document.write("<tr><td>English </td><td>", emarks, "</td><td>");
```

→ document.write(" "); <td>100 </td></tr>);
 → document.write(" ");
 → document.write(" ");
 → document.write(" ");
 → document.write(" ");
 → document.write("Total </td> <td> ", total, "</td><td>");
 → document.write("500 </td></tr>);
 → document.write("<tr><td> Percentage </td><td> ", per, "</td><td>");
 → document.write(" </td></tr>);
 → document.write("<tr><td> grade </td><td> ", grade, "</td><td>");
 → document.write(" </td></tr>);

html file :-

```
<body bgcolor="light blue">
```

Class 7:-

"String" Manipulations

5- conditional operator (maam will check).

Break/Switch statement

```
let reply = prompt("Do you want to continue");
switch(reply){
```

```
case "y":
```

```
document.write("continue");
```

```
break;
```

```
case="yes":
```

```
document.write("continue");
```

```
break;
```

```
case "N":
```

```
document.write("end");
```

```
break;
```

```
case="no":
```

```
document.write("end");
```

```
break;
```

```
default:
```

```
{document.write("wrong input");}
```

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 and a single and double quotations.

String Creation and Manipulation

```
let str1= "I am learning js"; //double
```

```
let str2= 'I am learning js'; //single
```

```
let str3= `I am learning string template`; //template literal (adjacent to 1 key in Keyboard)
```

* template literal is also called backtick

Template literal.

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

How to use template literal.

- for next line \n
- for tab (space) It 4-space means tab
- for print \ in string //
- for write variable in string \${variable name}
- for double quotation "hello" hello

String Manipulation Methods

- let str1 = "Learning is".
• 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) / ~~or~~
 join (str1, " ", str2)
• to concat() (\rightarrow always str)
 let str4 = str.concat(str2) / str.concat(str1)
 document.write(str4)
• str.trim() // to remove space from string
• str.trimStart()
• str.trimEnd()
• str.toUpperCase() // change in uppercase
• str.toLowerCase() // change in lowercase
• str.replace("javascript", "html") // search word
 (^{only change 1st appearance}) and replace (case sensitive)
• str.includes("is") // search word is / if
 not found return -1
 -1 \Rightarrow false
 +1 \Rightarrow true

Sometimes it returns -1 or +1 which also means true or false.

index 5 and 10 & index
1 15
word 10 words part (konsa to kee)
utha kee tayy

- str.slice(5, 10) // 5 is index and 10 is

used to pick which

slice 10 words / part (konsa to kee)
utha kee tayy

Class 8:

Switch / Break Statement Loop.

* Switch is seems like if else condition.
Previous class do have it
in JS file

* String:-

Backtick is below the esc key on
s. keyboard.

Template Literal / String Template.

Template literals are a feature in
javascript that were introduced with E6. They
give you a more (flexibility) flexible and
maintainable way of working with strings
in javascript.

We can write variable in string if
string is string template.

Syntax:-

• normal string:

```
document.write("my roll no is", variable,);
```

• template literal:

```
document.write(`my roll no is ${variable} abc`);
```

String Manipulation

\n use for print text in new line

\t use for space (tab) more than single ^{space}
text

\\ use for single \ in text

\" use double quotations in string.

Some String Properties & Methods.

- 1- let str1 = "I am learning js";
- 2- let str2 = "CSS";
- 3- let str3 = "html";
- 4- Position || str[0], str[1] first index start with 0.
- 5 to find length || str.length
- 6 to join strings || document.write(str1 + " " + str2)
|| document.write(str1, " ", str2)
- 7 by concat() || let str4 = str.concat(str2)/
str.concat(str2, str3);
document.write(str4);
- 8 remove space from start & end || str.trim()
- 9 str.trimStart()
- 10 str.trimEnd()
- 11 str.toUpperCase() change in uppercase
- 12 str.toLowerCase() change in lowercase
- 13 str.replace("Javascript", "html") || search
word and replace (case sensitive)
- 14 str.includes("is") || space word is / return
boolean.

Loops:-

- to execute a piece of code again and again
- finite loop and infinite loop
 - finite loop (ending point)
 - In finite (not end) memory full/computer hang. we don't use it

For loop

- i is block scope variable.
- use for iteration/counting
- first step initialization
- 2nd condition check jab tak condition true block of code execute.
- 3rd step update.

Syntax

```
for (i=1; i<=15; i++)  
{  
    document.write("Aleena<br>")  
    document.write(i) // it will show numbers  
    (1-15)
```

Class 9 :-

loop & array

- " " → quotation
- backward counting :-
- number.html
- <head>
 <title> number </title>
- </head>
- <body>
 <h2> Backward Counting </h2>
- <script src="number.js"></script>
- </body>
- </html>

number.js

```
for (let i=20; i>=1; i--) {
  document.write(`<br>${i}`);
}
```

Screen View

Backward counting

20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1

Array:- (Non primitive data types)

- Store multiple values in single variable
- Values written in square brackets []
- Values separated by commas.
- Each position is called index * (index position starts from 0).
- Each value call through index number starts with 0.

e.g:- arr[0]

arr[1]

Syntax:-

```
let info=[5,"Aween","html"];
```

```
console.log(info);
```

```
info=[]; // to empty any array
```

* if we are writing string we must use ":" (quotation)
 * if we use numbers we don't need quotation.

Properties and Methods in Array

```
let arr=[67,"Aween","html","css","Javascript"];
```

1. arr.length // length of array

2. arr.push ("python"); // add word in array in the end

3. arr.unshift ("python"); // add word in start of array

4. arr.shift (); // remove a word from start

5. arr.pop (); // to remove word from last and return updated array

6. arr.toString(); // convert array into string.

7. arr.concat (arr2); // join 2 or 3 array into new array and don't change original array.

8. arr.indexOf (css); // to find any word's position in array.

9. arr.slice (start index, end index); // don't change in original array

// return a slice piece of array.

10 arr.splice (startidx, delcount, next); // change original array // add, remove and replace.

Example:-

* let arr=[1,2,3,4,5,6,7];

arr.splice (2,2,55,56); // 2 index per 3, 3 and 4 remove, replace them with 55 and 56.

OR

`arr.splice(3, 0, 44);` // if you don't want to delete item, we add 3rd index as 44
OR

`B arr.splice(2, 3);` // if you don't want to add only delete items // 2 idx and delete 3 items.

Practice

- create an array to store different items (minimum 6) vegetables / fruits / companies etc.
- 1. find length of array
- 2. print all items of array by using both for loop / for of loop
- 3. remove first item from array.
- 4. remove 3rd item and ^{add} 2 items is this place
- 5. add any item in the last of array.
- 6. remove item from end.

array.js :-

```
let arr[67, "Aween", "Javascript"];
document.write(arr);
```

if we want to print on 0 index
`document.write(arr[0]);`

if we want 2 index
`document.write(arr[2]);`

for of loop

`document.write(arr[2], arr[1]);`

but if we want to write more values like 50 values or 100 values. and want to find any values length then we use.

`document.write(arr.length);`

For Simple loop // let l = arr.length - 1; Screen View

for (let i=0; i<l, i++) {
 document.write(arr[i]);}

For of loop *Play automatically with array Screen View

let arr=[67, "Aween", "Javascript", "html", "css", "js"];
let l=arr.length-1; *loop will work for 3 times then we write.
for (let i=0; i<l; i++) {
 document.write(i); *arr.length-1
 value;
 value, "
");

// Push Method // let arr[67, "Aween", "html", "css", "js"]
arr.push("python"); in the end Screen View
67, Aween, html, css, js, python
document.write(arr);

// unshift Method //

```
let arr=[67,"Aween","html","css","Javascript"];
arr.unshift("python") // add item in start
document.write(arr,"<br>");
```

E // shift method //

```
let arr=[67,"Aween","html","css","Javascript"];
arr.shift(); // remove item from start
document.write(arr,"<br>");
```

// pop method //

```
let arr=[67,"Aween","html","css","Javascript"];
arr.pop(); // remove item from end
document.write(arr,"<br>");
```

// toString method //

```
let arr=[67,"Aween","html","css","Javascript"];
arr.toString(); // convert array into string
document.write(arr,"<br>");
```

// Concat method //

```
let arr=[67,"Aween","html","css","Javascript"];
let arr2=["eng","urdu","maths"];
let arr3=arr.concat(arr2); // join 2nd & 3rd array
document.write(arr3); // new array and don't change original array.
```

OR

document.write(arr3=arr.concat(arr2),"
"),

// indexOf method //

```
let arr=[67,"Aween","html","css","Javascript"];
arr.indexOf("css"); // to find any word's position
document.write(arr); in array
```

OR

document.write(arr.indexOf("css"));

// slice method //

```
let arr=[67,"Aween","html","css","Javascript"];
arr.slice(2);
document.write(arr.slice(2),"<br>"); // don't change
in original array and return a slice piece of array.
```

(piece)

// Splice method //

(1st index, 2nd index)
let array=[2,3,56,"hina",78,65,23,89];
document.write(array.slice(3)); OR
index=7
length=8
document.write(array.slice(3,6));

document.write(array.splice(2,2,99,66));
index remove we can add 1,2,3
any num according

Class 10 :- Object [Method & Loop / For In, While / do While loop]

- Store multiple value in single variable.
- Value written in curly brackets {} in pair with keys

Syntax

```
(const) let student = {  
    or  
    name: "Aman",  
    rollNo: 67,  
    class: "javascript",  
}
```

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

* "Const" variable
doesn't change its
values but we can
change their keys

loops :-

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

For loop :- We define starting and ending points of our range.

- i is a block scope variable. Use for iteration counting.

• 2nd condition check jab tak condition true block of code execute

- 3rd step updation

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

For of / For In loop :-

- Print odd numbers of this array

array = [34, 45, 67, 95, 24, 12, 1, 5, 6, 8, 9, 79]

by using (for of) loop.

• Create an object and print keys and data with it by using for in loop.

* For of loop array ko pakaeta hai aur one by one us ki value ko print kewala jaay ga.

* For in method main hai jo key waha k la raha hai.

* In for and while (jab tak condition true nahi ho gi wo loop main enter hi nahi hogा).

While loop / do while loop :-

do while loop :-

* Pehlay print phir dobara print krna
K lia condition check agar condition
true hoti tou print again hota aur
agar false tou programme end.

condition bad main check hogi

Class 11:-

Functions

- * A JS (Javascript) function is a block of code. designed to perform a particular task.
- * A JS (Javascript) function is executed when "something" invokes it (call it).
[Also we can call it again n again.]

• Ques: If function abc() has no parameters then what is its name?

Function abc ()
 ↓
function ↓
 name
 of function
 → Parameters.