LECTURE 1
"JAVA SCRIPT"

Date 20th January, 24

Javascript is the 1st Language.
- Invented by Brendan Eich in 1995
- Developed for Netscape 2, became ECMA-262 in 1997.
- ECMA = European Computer Manufacturer Association
- ES1 to ES5 (1997 to 2009)
- In 2015 this is called Ecma Script / Es 2015 / ES6.
- ES6 is the standard for Javascript. After came ES7, ES8, ES10 etc.
- Js is light weight object oriented Programing language
- Use in form submit, In client side validation.
- Popups / events on click

USES: 1) Client side execute / browser (Js query, React Js, Angular Js)
2) Website Server Side (node Js, Express Js)
3) Mobile Development (Hybrid App) (framework for mob dpp react native, phone gap etc)
4) Software Development (Electron Js, Ex-vscode, framework etc)

- We can use script tag in Javas & in any <body> & <head> tag.
- <script> alert("hello"); </script>
- Always use script tag in body tag because it has so much load in head.
- Link Javas files in end tag of body.
- Inspect → Console on web page to fix errors.

"LECTURE 2" VARIABLES 22nd January, 24

- Variables: Space is saved in memory in variables. (It's a container)
- There are 3 ways to declare variables in Javascript.
1) Var 2) Let 3) Constant.
Var used before 2015 ECMA-6

* F12 short form of (console)
* data can be diff but location will be same.

Let (its value can change any time & can declare & assign in 2 steps)
eg let name ; (declare)
Const (its value can't be changed & can't be written in 2 lines
eg:

Date _____

- let & Const variables use in modern js E-6
- Var rejected b/c (it can be declared every time).

1st Rule:-
1) Variable is case sensitive eg: Name or name (2 diff variable) → ($s $_ 4)
2) Can't be any key word eg: alert prompt etc. 3) consist of alphabet, num, dollar sign & underscore, 4) Variable can't be start with digit (number) its first variable., no space.
5) variable name should be match with content 6) Camel Case start with small letter but next word is Capital letter.

Data types: 1) Number 2) String 3) Boolean 4) Null
5) Undefined 6) Array 7) Objects 8) functions.

Let name = "Kinza";
name = "Hina"; console.log(name);
Const abc = "Hello"; (it can't be changed)

1 { let age = 45; console.log("age"); console.log(age); }
2 { let fullName = "Kinza"; console.log(fullName); }
3 { let isPass = true; console.log(isPass); }
4 { let roll = null; console.log(type of roll); }
5 let lastname = prompt("enter last name"); window.console.log(lastName);

"LECTURE 3"

29th January, 24

"Javascript :: Data Types (Primitive & Non-primitive)"

Variable Scope: 1) Block scope Variable: If variable declare in block of codes (in curly braces {}) will only alive in block & will not be accessable after curly braces.
2) Global Scope Variable: there variables used globally in whole program.

Print / Display in Js: 1) On browser: window.document.write("Hello");
2) In Console: Console.log("Hello");
3) popup: window.alert("Hello");

Taking input from users in Js: PROMPT: In Javascript we use the prompt() function to ask user for input. As a parameter we input text we want to display to user. Once user click the input value returned. we store user input in variable so that we can use the information in our program.

Let answer = prompt("Do you want to send payment y/n ?:");
& answer variable value will be store & u can print it.

Date _____

Primitive data types:

1) number — Let rollno = 56 document.write(rollno); console.log(typeof rollno);
2) String — Let name = "Nisha"; document.write(name); console.log(typeof name);
3) boolean — let = ispass = true; // (ispass); // (typeof ispass);
4) undefined — let = class; // (class); // (typeof class);
5) Object(null) — Let abc = null; (abc); // (typeof abc);

Non Primitive data type:

1) Array : store multiple data in single variable written in Square bracket [ ]

Syntax : let info = [5, "Nina", computer];
console.log (info);

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

2) Object : Store multiple value in single variable
- values written in curly brackets { } in pair with keys.
- Syntax : name = "Nina",

let student = { rollno : 23
Class: computer }

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

"CLASS 4: JS OPERATORS IN JS" 26/1/24 10:46 pm

- Operators in Js :- 1) Arithmatic Operator (Values are operand & result is operand)
2) Assignment Operator
3) Comparison Operator
4) Logical Operator
5) Conditional Operator.

- Arithmetic operator; Increment & decrement work on single value.

+ = operator.
4+5 = operand.

let a = 9;
let b = 5;
document.write(a,"+",b,"-",a+b);

(2) let a=4; let b=3;

Arithmetic Operator: Substraction: document.write(a,"-",b,"=",a-b);

(3) Multiply: document.write(a"*",b,"=",a*b);

(4) divide: document.write(a"/"b"=",a/b);

(5) Modulus: document.write(a"%",b,"=",a%b);

(6) Exponent: document.write(a,"**",b,"=",a**b);

Unary Operand a++, a=a+1; > document.write(a);

a--, a=a-1;

Post increment a++

Pre increment ++a

post decrement a--

pre decrement --a.

a+=4/a = a+4;

a-=4/a = a-4;

a*=4/a = a*4;

a%=4/a = a%4;

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

ASSIGNMENT OPERATOR:

a=2; left=right

* we use assign operator for big values.

CLASS 5: JS OPERATORS

- Comparison Operator: 1) ==(equal to) ===(equal to + same data type)

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

3) > greater than 4) >= greaterthan true 5) <less than

6) <=lessthan & equal to - true (*its result in boolean).

Logical Operator: Compare b/w 2 values.

1) Logical AND(&&)(both should be true) 2) logical OR(||)(only one can be true) 3) logical NOT(!)

Conditional Operator: 1) if statement 2) if-else statement

3) if-else if statement

Ternary Operator: condition, true output false output.

(it is used to check small operation.

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

Logical Operator:

let a=5;

let b=6;

if(a>2 && b>5)

{alert("both conditions are true");

}else{

alert("both conditions are not true");

}

if(a>7 || b>5)

{alert("hello");

}else{

alert("welcome");

}

if(!(a<b))

{alert("hello");

}else{

alert("welcome");

}

let age=20;

let result;

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

alert(result);

Ternary Operator.

"STRINGS | LOOPS"

CLASS 7: Date

SWITCH CASE:

```
let reply = prompt("Do you want to continue---");
Switch(reply)
{
Case "y":
docunt.write("continue");
 break;
Case "yes":
docunt.write("continue");
break;
Case "N":
docunt.write("end");
break;
Case "no":
docunt.write("end");
 break;}
```

STRING:

1) String is a sequence of character used to represent a text.
2) It is a primitive data type
3) We can create string by using template literals & in single & double quotations.

String Creation & manipulation:

```
let str1= "I am learning js"; // double.
let str2= 'I am learning js' // single.
let str3= `I am leaing string template`; // temp literal (adjacent to 1 key in keyboard. back tick.
```

Template Literal: Temp lit are a feature in Js that introduced with ES6. They give more flexible & maintainable way of working with strings in Js

template string & back tick.

Date

How to use template literals:
- for next line \n
- for tab (space) \t
- for print \ in string \
- for write variable in string. $(Variable Name)
- for double quotation "hello\" hello.

Properties of string:
- position / index starts with 0
- to find length

```
let str = "hello! how are you?";
let L = str.length; // length
length -> document.write(L);
index -> document.write(str[8]); // o
```

document.write(str + " " + str2)

newvar = str.concat(str2); -> concatination (when 2 strings are in same line).

let Str = " I am learning Javascript";
* Str.trim() -> to remove space from start & end.
* Str.trimstart() // * str.trimend() //
* str.upperCase() -> change in upper case * Str.lowerCase() -> change in lower case.
* str.replace("javascript", "Html") -> search word & replace (case sensitive)
* str.includes("is") -> search word is / if not found return -1.

Template literal Syntax

```
let roll no 56;
let name = "Kinza";
document.write(`my roll no is ${rollno} my name is ${name}`);
```

String Manipulation:

1) \n (for print text in new line)
2) \t (use for spaces (tab) more than single space in text)
3) \\ (use for single \ in text)
4) \" (use double quotations in string).

CLASS 08: SWITCH/BREAK Date 5th Feb, 24.

STATEMENT | STRING METHODS | FOR LOOP

Switch Case break:

```
Switch(reply)
{ case "y";
document.write("continue...");
break;
Case "yes":
document.write("continue");
break;
case "no":
document.write("end");
break;
default:
document.write("wrong input");}
```

STRING:

① let rollno=56; let name="Kenza";
document.write(`my rollno is ${rollno} my name is ${name}`);

② let num1=56;
let num2=66;
document.write(`the sum of ${num1}{ ${num2} is ${num1+num2}`);

③ Console.log("My name is \kenza \n my subject is computer");

\n : It is used to write text in next line.
\t : It is used to create 3 to 4 spaces b/w texts.
\" : to write qudations.
\\ : to write single slash.

Console.log(str2.trim()); (remove spaces from both sides)
Console.log(str2.trimEnd()); (removes spaces from End)
" (str2.trimStart()); (removes spaces from Start).

Loop: for(i=1; i<=5; i++)) It is used to write counting.
{ document.write("kenza");
}

```
for(i=1; i<=50; i++)
  { document.write(i,"<br>");
  }
```

Date _____

Q) Print a table of any number by taking user input.

```
let tabno = prompt(parseInt("Enter Tableno of your choice:"));
for(i=1; i<=12; i++)
{
document.write(`${tabno} x ${i} = ${tabno*i} <br>`);
document.write("<br>");
}
```

CLASS 9: "ARRAY METHOD { FOR OF LOOP"

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

ARRAY (PRIMITIVE DATA TYPE): []

① Store multiple values in single variable. ② Values written in square bracket
③ Values are seperated by comma ④ each position is called Index
⑤ Each value call through index number start with 0

eg: arr[0]; arr[1]

Syntax: let info=[5, "kinza", "Computer"];
console.log(info); info=[]; // to empty any array

```
let arr=[34, "sadia", "Javascript"];
document.write(arr[2], arr[1]);
document.write(arr.length);
let l=arr.length-1;
for(let i=0; i<=1; i++)
{
document.write(arr[i], "<br>");
}
```

FOR OF LOOP:

```
for(let i of arr)
{
document.write(i);
}
```

```
for(let value of arr)
{
document.write(value);
}
```

Date _____

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 updated array

⑥ Book.toString(); // to convert array in string.

⑦ Book.concat(book2); // to join 2 or 3 array in new array dont change original array.

⑧ book.indexOf("Urdu"); // to find any words position in array.

⑨ book.slice(start idx, end idx); // dont change in original array.
// return a slice piece of array.

10) book.splice(Start idx, del count, next); // change original array.
// add, remove & replace.

eg: let arr = [1,2,3,4,5,6,7];
arr.splice(2, 2, 55, 56); // 2 index pe 3 // 3 & 4 remove // replace
// them 55 & 56.

arr.splice(3, 0, 44); // if you dont want to delete any item // 3 index pe jaka add 44

aa.splice(2,3); // if you dont want add only delete items // 2 idx and delete 3 items.

CLASS 10: "METHODS & LOOP/FOR IN/ WHILE/ DO WHILE LOOP"

let array = [2, 3, 56, "kinza", 78, 65, 23, 81];
document.write(array.length);
document.write(array.slice(3));
// (array.splice(1,2,8));
// (array)

CLASS 10

Date

Object:

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

Syntax:

```
let student = {
    name = "Hina"
    rollno : 23
    class : "Computer"
    }
```

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

```
Const student=
{ name = "fatima";
  rollno = 55,
  subject = "Computer"
}
for (let key in student)
{
document.write(key,"<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:

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

* do while loop use to execute 1 time loop.

```
let i = 1;
let sum = 0;
do {
document.write(i);
sum = sum + 1;
i++;
} while (i <= 10);
document.write(sum);
```

CLASS 11 "FUNCTIONS" Date 14th Feb, 24

FUNCTION DEFINE | FUNCTION INVOKE | ARROW FUNCTION

- A javascript function is a block of code designed to perform a particular task.
- A javascript function is executed when "something" invokes it (calls it).

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;
}
```

```
abc(5,7);
function abc(p1,p2){
  document.write(p1*p2);
}
document.write(p1);
```

Function Invoke (call)

```
fname();
```

```
fname(arg1,arg2)
or
fname(2,3)
```

```
let ans = sum(2,3);
console.log(ans);
let a = 9;
let b = 8;
let ans = mult(a,b,6);
document.write(ans);
```

Arrow function (ES6)

```
const fname = () => {
    block of code
}
```

Syntax:

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

```
fname()
```

CLASS 12 was revision.

x —— x.