

JAVA Script

CLASSTEAM® Page No.

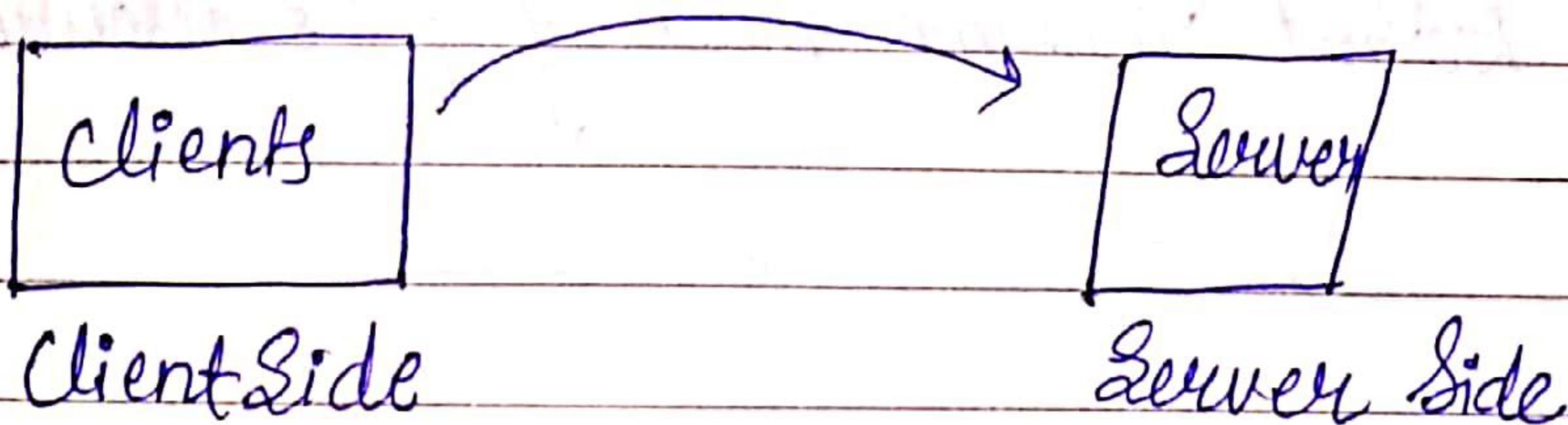
Date

JAVA Script Basic - I

1. JavaScript (JS) is the most popular lightweight, interpreted compiled programming language. It can be used for both Client-Side as well as Server-Side development.

JavaScript is also known as Scripting language for Web pages.

2. JavaScript is used to create interactive Web pages.
3. It is a high-level, dynamically typed language that is interpreted.



→ History of Java Script :-

It was created in 1995 by Brendan Eich at Netscape company. It was called "Mocha" → "Mona" → LiveScript
↑
Javascript. (1996)

It is designed to run as a scripting language in a host environment i.e. Browser.

* NO Compiler is Needed.

* Applications of JavaScript :-

- Web Development → Angular JS
- Web applications → React JS
- Server applications → Node JS
- Games
- SmartWatches
- Mobile Applications.

Ques - Why JavaScript is known as a lightweight programming language?

Ans Because, it has low CPU usage.
it has a minimalist syntax.
Very easy to learn (C++, Java).
Do not consume much of CPU's Resources.

A. Note:- JavaScript requires Engines in browser.

ex:- V8 → in Chrome and Opera.
Chakra → in Microsoft Edge.
Spider Monkey → in Firefox.

Adding JavaScript file in two ways:-

(1) Internal JavaScript :-

By using `<script>` --- `</script>` tag, it can be either be placed inside the `<head>` or `<body>` tag according to the requirement.

example:-

```
<html>
  <head>
    <title> Javascript </title>
  </head>

  <body>
    <!-- Javascript Code -->
    <script>
      console.log ("Welcome JavaScript");
    </script>
  </body>
</html>
```

Note:- Best way to use of JavaScript `<script>` tag is using before closing the `</html>` tag for easy parsing / rendering JS code.

Ex:- `<body>`

--- HTML Code ---

`<script>` --- `</script>`
`</body>`

→ (Just before).

* Understanding Basic Script Code :-

`<script>` → Comment

// Line below is used to print on the console window.

`console.log("Namaste Duniya");`

`<script>` is used to store client side instructions.

used to print.

↓
terminate instruction

(2) External JavaScript :-

By using

`<script src="file-name.js"></script>`.

It can be place either inside `<head>` or `<body>` tag.

example:-

(1)

`<html>`
`<head>`

`<title> External JavaScript </title>`
`<script src="main.js"></script>`

`</head>`
`<body>`
`</body>`
`</html>`

(2)

`<html>`
`<head>`
`<title> JS </title>`

`</head>`
`<body>`

`index.js`

`console.log("Namaste");`

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

A. If we want to run JS Code in VS Code Terminal.

- Open Terminal.
- Type "node index.js".

To check current path, type "pwd"

present working directory

Homework Question:-

Java

(1) Strongly typed language and variable must be declared first to use in the program.

(2) It is an Object-Oriented programming language.

(3) Java applications can run in any virtual machine or browser.

JavaScript

Loosely typed language and has more relaxed syntax and rules.

It is an object-based scripting language.

JS Code used to run only in the browser but now it can run on the server via Node.js.

→ Java is a Standalone language.

→ Mainly used for Backend.

→ Java uses more memory.

→ Requires a JDK to run the code.

→ Statically typed language

Contained within a webpage and integrates with its HTML content.

JS is used for frontend and backend both.

JavaScript uses less memory.

Requires any text editor or Browser Console.

Dynamically Typed Language.

Variables in JavaScript :-

Variables are named memory locations. It is the basic unit of storage in a program.

Rules to declare a Variable :-

- Case-Sensitive (-)
- Only begin with a letter, underscore or "\$" symbol.
- Cannot contain Space or Hyphen (-)
- A variable name cannot be a reserved keyword.

- Use only Camel Case

→ ageOfStudent.

example ⇒ let(f=5) X

We can declare variables in JavaScript in three ways.

(1) JavaScript var keyword :-

Syntax :-

`var variable-name = "abcd";` "Variable Value"
↓ Re-declaration
keyword. ex:- `var a=5;` is Possible
`var a="Ankit";`

(2) let keyword :-

Syntax:- `let varname = "variable-value";`

ex:- `let a=5;` In 'let' same declaration
~~`let a="Ankit";`~~ of variable again
→ already is NOT possible declared.

(3) const keyword :-

ex:- `const num = 500;`

const refers to fixed value permanently to a variable.

`if → const a=12;` J ERROR
`a=50;`

→ Assignment to a Constant Variable.

Important :-

Property	<u>Var</u>	<u>let</u>	<u>const</u>
Scope	Global & function scope	Block	Block scoped.
Updation	Value can be changed.	Value can be changed.	Value cannot be changed (Fixed)
Re-declaration	Can be Redeclared.	Cannot be Redeclared.	Cannot be Redeclared.
Hoisting	Hoisted at top	Hoisted at top	Hoisted at top

Data Types in JavaScript

* Primitive Types

- (i) String
- (ii) Number
- (iii) Boolean
- (iv) Undefined
- (v) Null
- (vi) Symbol

→ String → Sequence of characters.

ex:- let str1 = "Hello";
let str2 = "Welcome";
console.log(str1 + " " + str2);

→ number → Hold decimal values as well as without decimals.

ex:- let x = 450;
let y = 10.8;
console.log("Sum is "+x+y);

→ Boolean → True or False.

→ Undefined :- It refers to Variable whose "value is not defined"

→ Null :- Value is null.

ex:- let x = null;
console.log(x);

* Reference Types (Non-Primitive)

- (i) Objects
- (ii) Arrays
- (iii) Functions. (Further Study)

All are objects.

→ Objects :- It may be defined as an unordered collection of related data, in the form of "key:value" pairs.

ex:-
let school = {
name: 'Delhi School',
location: 'Delhi',
year: '1971',
};

Accessing → [school.name] → ('Delhi School')

using dot notation

→ [school['location']] → (Delhi)
using Bracket Notation

→ Array :- Collection of similar types of data

↳ used to contain a List of items.

ex:- let names = ['love', 'Rahul', 'Sangam'].

Indexing 0 1 2

* names[0] = love ✓

* names[1] = Rahul ✓

* names[2] = Sangam ✓

* names[3] = undefined ✓ Not A Error
But undefined.

* names[3] = "Chetan"; → Add Chetan in the Array.

* names[1] = "Babbar"; Babbar Replaces Rahul.

* console.log(names);

↳ 'love', 'Babbar', 'Sangam', 'chetan'.

Operators in JavaScript

- Arithmetic :- +, -, *, /, %, **, ++, --
- Assignment :- +=, -=, *=, /=, %=, **=, ^=
- Comparison :- >, <, >=, <=, ==, !=, ==, !=
- Bitwise :- >>, <<, \oplus , \otimes , \sim , \wedge
- Logical :- BB (AND), B || (OR), ! (NOT).

Note :- Strict Equality Operator (==) :-
This operator is used to compare the equality of two operands with type.

ex:- let num = 1; → number, 1
let str = '1'; → string, 1
console.log(~~(num == str)~~) (num == str)
↳ [false], ↳

If both value & data are equal, then it gives true.

* Loose Equality Operator (==)

This operator only checks the value But not its type.

ex:- let num = 1;
let str = '1';

console.log (num == str) ↳ [True]

* Logical Operator with Non-Booleans

$(\text{true} \text{ || } \text{true}) \rightarrow \text{true}$

$(\text{true} \text{ || } \text{false}) \rightarrow \text{true}$

$(\text{false} \text{ || } \text{false}) \rightarrow \text{false}$

$(\text{false} \text{ || } \text{'Ankit'}) \rightarrow \text{Ankit}$

$(\text{false} \text{ || } \text{1 T}) \rightarrow \text{1}$

$(\text{false} \text{ || } \text{5 T}) \rightarrow \text{5}$

$(\text{false} \text{ || } \text{1} \text{ / } \text{5}) \rightarrow \text{1}$

$(\text{false} \text{ || } \text{5 T} \text{ / } \text{1}) \rightarrow \text{5}$

$(\text{true} \text{ & } \text{5}) \rightarrow \text{5}$

$(\text{true} \text{ & } \text{'Ankit'}) \rightarrow \text{Ankit}$

$(\text{true} \text{ & } \text{5} \text{ & } \text{2}) \rightarrow \text{2}$

$(\text{5} \text{ & } \text{6} \text{ & } \text{false}) \rightarrow \text{F}$

} first & True Value
first time Ankit,
immediately point
out

last task check
true value out
all true value, first
true 2011 / 1st 2011
false 2011

→ Operators Precedence :- use brackets solving easily. Normal
to learn table.

Control Statements

→ if - else if - else .

→ switch case.

Syntax :-

switch (expression) {

case value : —

break;

! !

! !

default : -

}

Falsey Values

* Undefined

* Null

* 0

* false

* '' (Empty)

* NaN (Not a Number)

Truthy Values

Anything except falsy.

Loops :-

→ For loop :- finite No. of iterations.

for (initialization ; condition ; update){

}

→ while loop :- Unknown NO. of Iterations.
initializations
while (condition)
{

 }
 Updation

→ do-while loop :- [at least 1 time execute]

Syntax:-

Initializations

do
{

 }

Updation
 {

 While (Condition);