

# **VARIABLES IN JAVASCRIPT**

# VARIABLES

- **Javascript → 3 Type → Variable → Create**
- **Let**
- **Const**
- **Var**

A screenshot of the Visual Studio Code (VS Code) interface. The main area shows a code editor with a file named "index.js" containing the following JavaScript code:

```
1 let name = "MBS Coding";
2 let age = 25;
3 age = 30;
4 console.log(name, age);
```

The terminal below shows the output of running the script with "node index.js". The output is "MBS Coding" followed by "25" on the first run, and "MBS Coding" followed by "30" on the second run, indicating a change in the value of the variable "age". A yellow arrow points to the gear icon in the bottom left corner of the terminal panel.

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS>
node index.js
MBS Coding
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
MBS Coding 25
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
MBS Coding 30
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS>
```

<>    ⚡ 0 ⚡ 0    Ln 3, Col 10    Spaces: 2    UTF-8    CRLF    { } JavaScript    🎁 Finish Setup    🔍 Go Live    ✅ Prettier    📰

A screenshot of the Visual Studio Code (VS Code) interface. The main editor window displays the file `index.js` with the following code:

```
7 // 2. Const
8
9 const account = 1234;           █
10 // account = 200;
11 console.log(account);
```

The cursor is positioned at the start of the assignment operator in line 9. The status bar at the bottom shows the current position as "Ln 9, Col 22".

In the bottom right corner of the editor, there is a small preview window showing the code structure:

```
L 2 JS > index.js > ...
```

The terminal tab is selected, showing the following error message:

```
TypeError: Assignment to constant variable.
    at Object.<anonymous> (C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS\index.js:10:9)
    at Module._compile (node:internal/modules/cjs/loader:1706:14)
    at Object..js (node:internal/modules/cjs/loader:1839:10)
    at Module.load (node:internal/modules/cjs/loader:1441:32)
    at Function._load (node:internal/modules/cjs/loader:1263:12)
```

The terminal also lists two PowerShell sessions:

- [powershell]
- [powershell]

The bottom navigation bar includes icons for file operations, search, and other development tools.

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The main area displays a file named `index.js` with the following content:

```
L 2 JS > index.js > ...
15
16 var a = 10;
17 a = 20;
18 var a = 30;
19 console.log(a);
```

The code uses `var` to declare a variable `a`, which is initially set to `10`, then reassigned to `20`, and finally reassigned to `30`. The `console.log(a)` statement is intended to log the final value of `a`.

Below the code editor, the terminal tab is selected, showing the output of running the script:

```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
20
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
30
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS>
```

The terminal shows two executions of the script. The first execution logs `20` and the second execution logs `30`. A cursor is visible in the terminal window.

The status bar at the bottom provides information about the current file: `Ln 19, Col 16`, `Spaces: 2`, `UTF-8`, `CRLF`, and `{ } JavaScript`. It also includes icons for `Finish Setup`, `Go Live`, `Prettier`, and a bell icon.

The screenshot shows the Microsoft Visual Studio Code interface with the following details:

- Title Bar:** Shows the title "index.js" and a search bar containing "JAVASCRIPT".
- Left Sidebar:** Includes icons for File, Search, Open, Save, Find, Replace, and a Problems icon with a blue circle containing the number "1".
- Code Editor:** Displays the content of "index.js":

```
24
25  if(true){
26    var a = 20;
27 }
28 console.log(a);
```
- Terminal:** Shows the command "node index.js" being run and its output "20".
- Bottom Status Bar:** Shows file statistics ("Ln 28, Col 16"), encoding ("UTF-8"), line endings ("CRLF"), and language ("JavaScript"). It also includes status icons for "Finish Setup", "Go Live", "Prettier", and a bell.

A screenshot of the Visual Studio Code (VS Code) interface. The main area shows a file named "index.js" with the following code:

```
31
32 function fun(){
33     var c = 30;
34 }
35 console.log(c);
```

The code contains a syntax error at line 35 where "c" is used before it is defined. The terminal below shows the resulting error message:

```
ReferenceError: c is not defined
at Object.<anonymous> (C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS\index.js:35:13)
at Module._compile (node:internal/modules/cjs/loader:1706:14)
at Object..js (node:internal/modules/cjs/loader:1839:10)
at Module.load (node:internal/modules/cjs/loader:1441:32)
at Function._load (node:internal/modules/cjs/loader:1263:12)
at TracingChannel.traceSync (node:diagnostics_channel:328:14)
```

The status bar at the bottom indicates the code is in "JavaScript" mode, with line 35, column 16, and other file details.

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The main area displays a file named `index.js` with the following content:

```
41 if(true){  
42 |   let c = 90;  
43 }  
44 console.log(c);
```

The code consists of four lines. Lines 41 and 42 are part of an if block. Line 43 is the closing brace of the if block. Line 44 is a call to `console.log` with the variable `c`. The line numbers are aligned to the left of the code.

Below the code editor, the terminal tab is selected, showing the following stack trace output:

```
at Object.<anonymous> (C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS\index.js:44:13)  
at Module._compile (node:internal/modules/cjs/loader:1706:14)  
at Object..js (node:internal/modules/cjs/loader:839:10)  
at Module.load (node:internal/modules/cjs/loader:1441:32)  
at Function._load (node:internal/modules/cjs/loader:1263:12)  
at TracingChannel.traceSync (node:diagnostics_channel:328:14)  
at wrapModuleLoad (node:internal/modules/cjs/loader:237:24)  
at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:171:5)
```

The terminal also shows some status indicators at the bottom: Ln 44, Col 16, Spaces: 2, UTF-8, CRLF, { } JavaScript, Finish Setup, Go Live, Prettier, and a bell icon.

# **DATA TYPES IN JAVASCRIPT**

A screenshot of the Visual Studio Code (VS Code) interface. The top bar includes icons for file operations, a search bar containing "JAVASCRIPT", and window control buttons. The left sidebar features a file tree with "index.js" selected, a search icon, a refresh icon, and a notifications icon with a blue circle containing the number "1". The main editor area displays the following code:

```
L 2 JS > index.js
45
46 // Data Types in JS
47
48 // 1. Premitive Data Type
49
50 // 2. Non Premitive Data Type
```

The code uses syntax highlighting where comments are in pink and numbers are in light blue. The cursor is positioned at the start of the second line of code. Below the editor, the "TERMINAL" tab is active, showing a command prompt:

```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS>
```

The terminal also lists two PowerShell sessions in its sidebar.

At the bottom, status bars show "Ln 48, Col 27", "Spaces: 2", "UTF-8", "CRLF", "JavaScript", "Finish Setup", "Go Live", "Prettier", and a bell icon.

A screenshot of the Visual Studio Code (VS Code) interface. The top bar includes icons for file operations, a search bar containing "JAVASCRIPT", and window control buttons. The left sidebar features a file tree with "index.js" selected, a search icon, a folder icon, a preview icon, and a settings gear icon with a blue notification badge showing "1". The main editor area displays the following code:

```
L 2 JS > index.js
47
48 // 1. Primitive Data Type
49
50 // Number, String, Boolean, Undefined, Null, Bigint, Symbol
51
52 // 2. Non Primitive Data Type
```

The terminal at the bottom shows the command line prompt "PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS>". The status bar at the bottom provides information about the current file: "Ln 50, Col 60", "Spaces: 2", "UTF-8", "CRLF", "JavaScript", and "Finish Setup". It also includes links for "Go Live", "Prettier", and a bell icon.

The screenshot shows the Visual Studio Code (VS Code) interface. The top bar includes icons for file operations, a search bar containing "JAVASCRIPT", and a tab bar with multiple tabs. The main workspace displays a file named "index.js" with the following content:

```
L 2 JS > index.js
51
52 // 2. Non Primitive Data Type
53
54 // Array, Object, Function
```

The sidebar on the left contains icons for file operations, search, file tree, terminal, and settings. The bottom status bar shows the current file path "C:\Users\maches\Desktop\JAVASCRIPT\L 2 JS>", line and column information "Ln 54, Col 27", and encoding "UTF-8". It also includes icons for Prettier and Go Live.

A screenshot of the Microsoft Visual Studio Code (VS Code) interface. The main area shows a code editor with a file named "index.js". The code contains the following JavaScript:

```
53
54 let a = 10;
55 let b = 2.45;
56 console.log(typeof(a));
57 console.log(a, b);
58
```

The terminal below the editor shows the output of running the script with "node index.js". The output is:

```
PS C:\Users\maches\Desktop\JAVASCRIPT\L 2 JS> node index.js
10 2.45
PS C:\Users\maches\Desktop\JAVASCRIPT\L 2 JS> node index.js
number
10 2.45
PS C:\Users\maches\Desktop\JAVASCRIPT\L 2 JS>
```

The status bar at the bottom provides information about the current file: Ln 56, Col 24, Spaces: 2, UTF-8, CRLF, and the language is set to JavaScript. There are also icons for "Finish Setup", "Go Live", "Prettier", and a notification bell.

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The top bar includes standard icons for file operations, a search bar containing "JAVASCRIPT", and a tab bar with multiple open files. The main workspace displays an "index.js" file with the following content:

```
L 2 JS > index.js > ...
60
61 let c = "MBS Coding";
62 let d = 'Official';
63 console.log(typeof(c))
64 console.log(c,d);
65
```

The cursor is positioned at the end of the line "63 console.log(typeof(c))". Below the editor, the "TERMINAL" tab is selected, showing the command-line output of running the script:

```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
MBS Coding Official
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
string
MBS Coding Official
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS>
```

The terminal also lists two PowerShell sessions in the sidebar.

At the bottom, status bar information includes: Ln 63, Col 24, Spaces: 2, UTF-8, CRLF, { } JavaScript, Finish Setup, Go Live, Prettier, and a notification icon.

The screenshot shows a dark-themed instance of the Visual Studio Code (VS Code) code editor. The main workspace contains a single file named `index.js`. The code within the file is as follows:

```
L 2 JS > index.js > ...
67
68 let login = true;
69 let f = false;
70 console.log(login, f);
71
72 // 2. Non Primitive Data Type
```

The code uses the `console.log` function to output the values of the variables `login` and `f`. The output is visible in the integrated terminal at the bottom of the interface:

```
PS C:\Users\maches\Desktop\JAVASCRIPT\L 2 JS> node index.js
true false
```

The terminal also shows the command used to run the script (`node index.js`). The status bar at the bottom provides additional information: line 70, column 22, two spaces, UTF-8 encoding, CRLF line endings, and the current file type is listed as JavaScript.

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The left sidebar contains icons for file, search, file tree, and other development tools. The main editor area displays a file named `index.js` with the following content:

```
L 2 JS > index.js > ...
72 // Undefined
73
74 let user;
75 console.log(user);
76
77 // 2. Non Primitive Data Type
```

The cursor is positioned at the start of the `let user;` line. The terminal below shows the output of running the script with Node.js:

```
PS C:\Users\maches\Desktop\JAVASCRIPT\L 2 JS> node index.js
undefined
PS C:\Users\maches\Desktop\JAVASCRIPT\L 2 JS> []
```

The status bar at the bottom provides information about the current file: `Ln 74, Col 1 (29 selected)`, `Spaces: 2`, `UTF-8`, `CRLF`, `{ } JavaScript`. It also includes links for `Finish Setup`, `Go Live`, `Prettier`, and a notification bell.

The screenshot shows a dark-themed instance of Visual Studio Code (VS Code) with the following interface elements:

- Title Bar:** Includes icons for back, forward, search (with "JAVASCRIPT" typed), and various window control buttons.
- Left Sidebar:** Features a vertical toolbar with icons for file operations (New, Open, Save, Find, Replace, etc.), a search icon, a folder icon, a preview icon, a terminal icon, and a settings gear icon. A blue circular badge with the number "1" is visible on the preview icon.
- Editor Area:** Displays the "index.js" file content:

```
L 2 JS > index.js > ...
12 // undefined
13
14 let user;
15 console.log(user);| [I]
16 user=true;
17 console.log(user);
18 ?
```
- Bottom Navigation:** Tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is selected), and PORTS.
- Terminal Area:** Shows the command-line output:

```
PS C:\Users\maches\Desktop\JAVASCRIPT\L 2 JS> node index.js
undefined
true
PS C:\Users\maches\Desktop\JAVASCRIPT\L 2 JS> []
```
- Bottom Status Bar:** Shows file statistics (Ln 75, Col 19, Spaces: 2, UTF-8, CRLF), language mode (JavaScript), and status indicators for "Finish Setup", "Go Live", "Prettier", and a notification bell.

# **BIGINT IN JAVASCRIPT**

# **5. Big Int Data Type**

## **1. BigInt → BigInteger**

## 5. Big Int Data Type

2. Question → Below Code → Number → How much Byte → Memory ?

**let a = 10 ;**

**let b = 2.36;**

**Ans. This numbers like 10, 2.36 → 8 Byte of Memory → 64 Bit**

**Note : 64 Bit → Integer(10) + Floating Point Number(2.36)**

## **5. Big Int Data Type**

**3. But you have very big number → Which not fit → In 8 Byte**

**Ex. Our Memory size → 3 Bit But we want to store 13 into it**

**13 → Value =  $1101 (2^0 * 1 + 2^1 * 0 + 2^2 * 1 + 2^3 * 1) = 1 + 0 + 4 + 8 = 13$**

**So 13 → 3 Bit Not stored → Value = 1101 => 4**

## **5. Big Int Data Type**

**4. One number → Not stored 8 Bytes**

**Q. How we can stored that number in our system ?**

**Max Value →  $2^{53} - 1$**

**Min Value →  $-2^{53}$**

**Why ? → Because it stores → Both → Integer + Float**

## **5. Big Int Data Type**

**5. Note : 3 Bit → 8 Number Store**

**8 Number = 4 Number +ve & 4 Number -ve**

**0 1 2 3 => +Ve Number**

**-4 -3 -2 -1 => -Ve Number**

**Note : Highest Number => 3 & Lowest Number -4**

## **5. Big Int Data Type**

**6. 8 Bytes → 64 Bit**

**64 Bit =>  $2^{64}$  => Total Numbers Stores => True**

**But Here floating point number + integer**

 **That's why ans will be :**

**Max Value →  $2^{53} - 1$**

**Min Value →  $-2^{53}$**

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The main editor window displays a file named `index.js` with the following content:

```
// Bigint
let num = 1234567890987654321n;
console.log(num);

// 2. Non Primitive Data Type
```

The terminal at the bottom shows the output of running the script:

```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
1234567890987654321n
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS>
```

The status bar at the bottom indicates the following settings: Ln 84, Col 1 (50 selected), Spaces: 2, UTF-8, CRLF, JavaScript, Finish Setup, Go Live, Prettier, and a gear icon.

The screenshot shows a dark-themed instance of Visual Studio Code (VS Code) with the following interface elements:

- Title Bar:** Displays the title "JAVASCRIPT" and various window control icons.
- Sidebar:** On the left, there are several icons: a file icon, a search icon, a folder icon, a preview icon, a refresh icon, a problems icon (with a blue circle containing the number 1), and a settings gear icon.
- Code Editor:** The main area contains the following code in a file named "index.js":

```
87 // Null
88
89 let weather = null;
90 console.log(weather);
91
92 // 2. Non Primitive Data Type
```
- Terminal:** Below the editor, the "TERMINAL" tab is selected. It shows the command "node index.js" followed by the output "null".
- Bottom Status Bar:** Shows the current line and column ("Ln 89, Col 1"), the number of selected characters ("42 selected"), and encoding information ("UTF-8 CRLF"). It also includes status indicators for "JavaScript", "Finish Setup", "Go Live", "Prettier", and a "Decorations" icon.

## **UNDEFINED VS NULL :**

**Ex. MBS Train in pune ?**

**Current Train = Yes available → Current**

**Sorry MBS Train exist but current location not found → Null**

**→Please try after some time → Intentionally khali**

**MBS Train Not Exist → Undefined → Please Enter valid name**

## **UNDEFINED VS NULL :**

**Ex. Temperature of pune ?**

**Current Temp = 25 Deg Cel → Current**

**Sorry we cant fetch temperature of pune → Null → Please try after  
some time → Intentionally khali**

**Not Exist → Undefined → Please Enter valid name**

A screenshot of the Visual Studio Code (VS Code) interface. The main area shows the code editor with the file `index.js` open. The code contains several lines of JavaScript, including two `const` statements defining symbols and two `console.log` statements printing them. A comment `// 2. Non Primitive Data Type` is present. The terminal below shows the execution of the script, resulting in the output `Symbol(id)` twice. The sidebar on the left includes icons for file operations, search, and other development tools. The status bar at the bottom provides information about the current line (Ln 94, Col 1), character count (90 selected), and encoding (UTF-8).

```
94 const id1 = Symbol("id");
95 console.log(id1);
96 const id2 = Symbol("id");
97 console.log(id2);
98
99 // 2. Non Primitive Data Type
```

```
PS C:\Users\maches\Desktop\JAVASCRIPT\L 2 JS> node index.js
Symbol(id)
PS C:\Users\maches\Desktop\JAVASCRIPT\L 2 JS> node index.js
```

index.js

```
93
94 const id1 = Symbol("id");
95 console.log(id1);
96 const id2 = Symbol("id");
97 console.log(id2);
98 console.log(id1==id2);
```

L 2 JS > index.js > ...

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\maches\Desktop\JAVASCRIPT\L 2 JS> node index.js
Symbol(id)
Symbol(id)
false
PS C:\Users\maches\Desktop\JAVASCRIPT\L 2 JS>
```

Ln 94, Col 1 (114 selected) Spaces: 2 UTF-8 CRLF { } JavaScript Finish Setup Go Live Prettier

A screenshot of the Visual Studio Code (VS Code) interface. The top bar shows the title 'JAVASCRIPT' and various icons. On the left is a sidebar with icons for file, search, file tree, and settings. The main area shows a code editor with a file named 'index.js'. The code contains the following:

```
104 // Array
105
106 let arr = [10,12,14.4,"MBS Coding",true];
107 console.log(typeof(arr));
108 console.log(arr);
```

The terminal below shows the execution of the script:

```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
object
[ 10, 12, 14.4, 'MBS Coding', true ]
```

The status bar at the bottom indicates the current file is 'index.js', encoding is 'UTF-8', and the file has been modified ('Finish Setup'). Other status indicators include 'Ln 107, Col 26', 'Spaces: 2', 'CRLF', 'Prettier', and 'Go Live'.

The screenshot shows the Visual Studio Code (VS Code) interface. The main area displays the content of `index.js`:

```
111
112 let user = {
113   name:"MBS Coding",
114   account:12345,
115   age:25,
116   category:'general'
117 }
118 console.log(typeof(user));
119 console.log(user);
```

The terminal below shows the execution of the script:

```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
{ name: 'MBS Coding', account: 12345, age: 25, category: 'general' }
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
object
{ name: 'MBS Coding', account: 12345, age: 25, category: 'general' }
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS>
```

The status bar at the bottom indicates the current file is `index.js`, and the file has 119 lines and 19 columns. Other status items include `Spaces: 2`, `UTF-8`, `CRLF`, `JavaScript`, `Finish Setup`, `Go Live`, and `Prettier`.

A screenshot of the Microsoft Visual Studio Code interface. The main area shows a file named 'index.js' with the following code:

```
121 // Function
122
123 function add(){
124     console.log("Hello");
125 }
126 add();
```

The terminal at the bottom shows the output of running the script:

```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
Hello
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS>
```

The status bar at the bottom indicates the current line (Ln 124), column (Col 22), and encoding (UTF-8). It also shows icons for JavaScript, Finish Setup, Go Live, Prettier, and a settings gear.

The screenshot shows the Microsoft Visual Studio Code interface with the following details:

- File Explorer:** On the left, it shows a file tree with "index.js" selected.
- Code Editor:** The main area displays the following code in "index.js":

```
129
130 let s = function add(){
131   console.log("MBS Coding");
132 }
133 console.log(s);
134 s();
```

- Terminal:** Below the editor, the terminal tab is active, showing the output of running the script:

```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
[Function: add]
MBS Coding
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS>
```

- Status Bar:** At the bottom, the status bar shows: Ln 134, Col 5 | Spaces: 2 | UTF-8 | CRLF | { } JavaScript | Finish Setup | Go Live | Prettier.

The screenshot shows a dark-themed instance of Visual Studio Code (VS Code) with the following details:

- File Explorer:** On the left, it shows a single file named "index.js".
- Code Editor:** The main area contains the following code:

```
L 2 JS > index.js > ...
89
90 // Null
91
92 let weather = null;
93 console.log(typeof(weather));
94 // console.log(weather);
95
```
- Terminal:** Below the editor, the "TERMINAL" tab is selected, showing the output of running the script:

```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
bigint
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
object
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> []
```
- Sidebar:** The sidebar includes icons for File, Search, Find, Open, Recent, and Settings.
- Bottom Status Bar:** Shows file path (Ln 93, Col 30), encoding (UTF-8), line endings (CRLF), language (JavaScript), and status indicators for Finish Setup, Go Live, and Prettier.

# **1. Premitive Data Type : 7 P D T**

- **1. Number :** Integer(1, 2, 3, 4 , 5, 6, 7, 8, 9) + Floating point number( 1.1, 2.1, 3.1, 4.1, 5.1)
- **2. String :** MBS, Coding, Sachin
- **3. Boolean :** true, false
- **4. Undefined :**
- **5. Null :**
- **6. Bigint :**
- **7. Symbol :**

## **2. Non Primitive Data Type :**

- **1. Array :**
- **2. Object :**
- **3. Function :**

## Primitive Data Type is Immutable

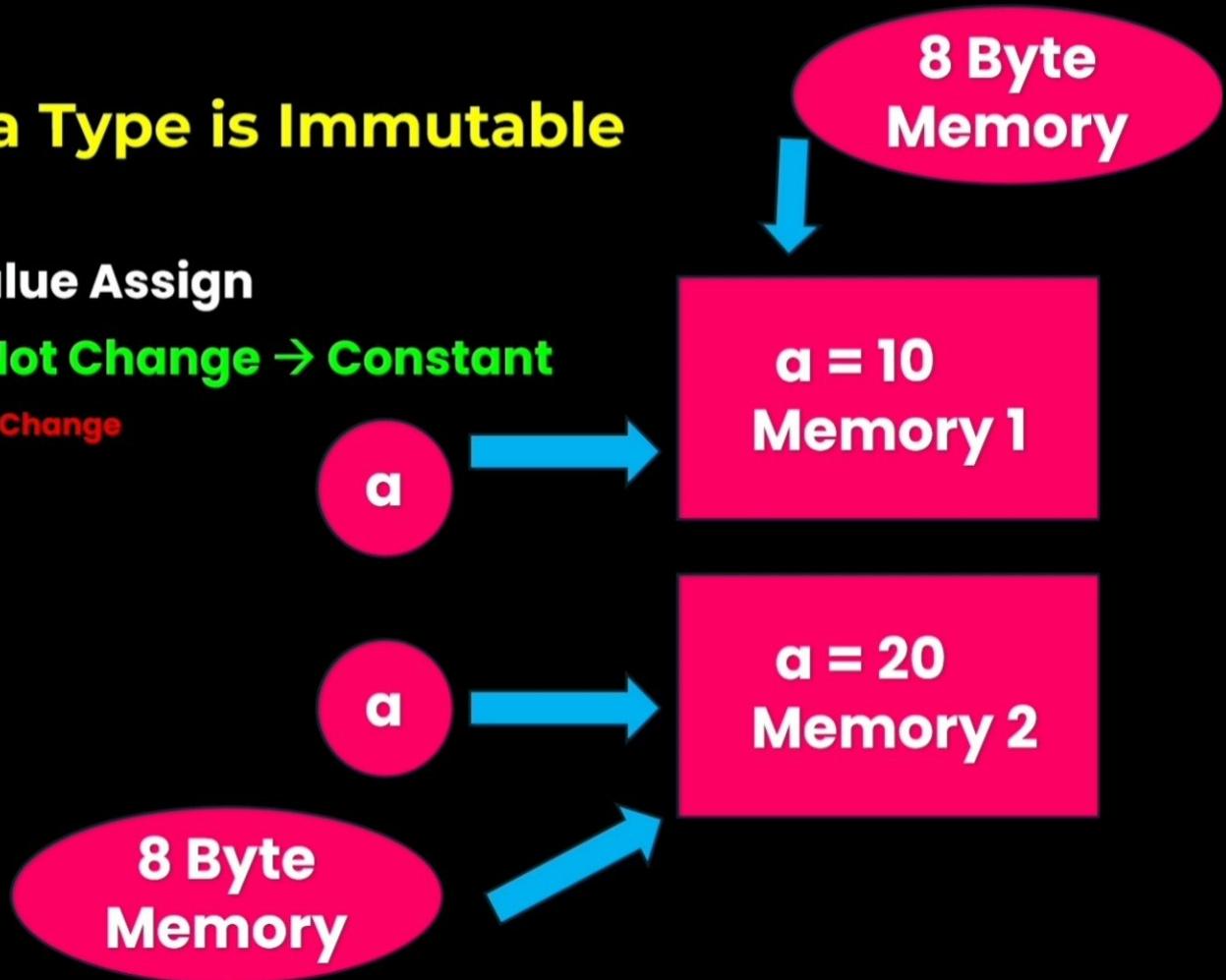
Variable → New Value Assign

Variable Value → Not Change → Constant

Pre Data 1 Time Create → Not Change

let a = 10;

a = 20;



## Primitive Data Type is Immutable

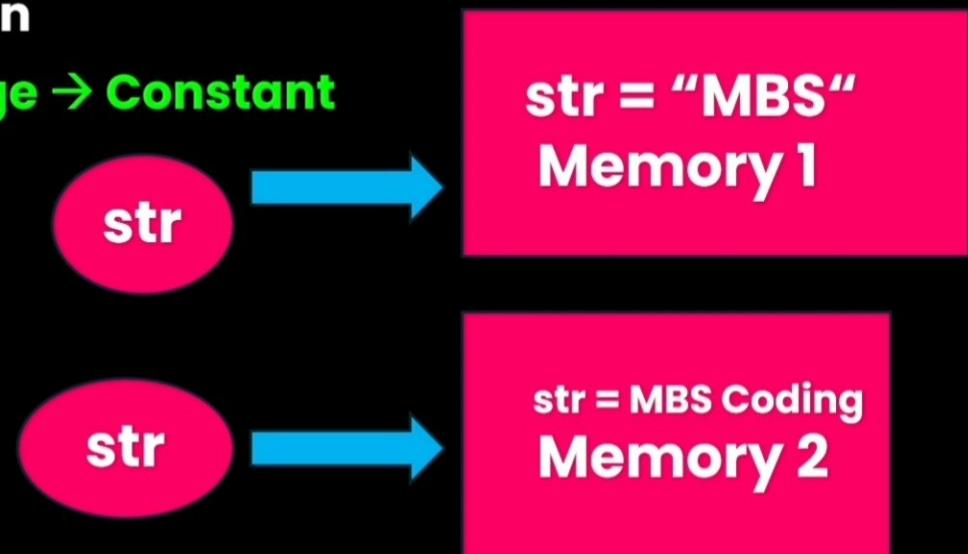
**Variable → New Value Assign**

**Variable Value → Not Change → Constant**

Pre Data 1 Time Create → Not Change

```
let str = "MBS" ;
```

```
str = "MBS Coding" ;
```



The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The top bar includes standard window controls and a search bar labeled "Q JAVASCRIPT". The left sidebar features a file tree with "index.js" selected, a search icon, a refresh icon, and a notifications icon with a blue circle containing the number "1". Below the sidebar are five tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined, indicating it is active), and PORTS. The main editor area contains the following code:

```
143
144 let a = 20;
145 console.log(a);
146 a = 30;
147 console.log(a);| █
148
149
```

The terminal below the editor shows the execution of the script and its output:

```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
20
30
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> █
```

At the bottom of the interface, there are status indicators: a file icon with "0", a circular icon with "0", and a triangle icon with "0". The status bar also displays "Ln 147, Col 16", "Spaces: 2", "UTF-8", "CRLF", "JavaScript", "Finish Setup", "(•) Go Live", "✓ Prettier", and a settings gear icon.

A screenshot of the Visual Studio Code (VS Code) interface. The main area shows a code editor with the file `index.js` open. The code contains a comment `// Immutability Proof`, a variable declaration `let str = "MBS Coding";`, and a line where the first character of the string is modified: `str[0] = "0";`. A cursor is positioned after the closing brace of the assignment. The code editor has a dark theme with syntax highlighting. To the right of the code editor is a vertical sidebar containing a tree view of project files and folders. Below the code editor is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is currently selected), and PORTS. The terminal window shows the output of running the script: 

```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js  
MBS Coding  
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS>
```

 At the bottom of the interface, there are status indicators and toolbars for navigating between files, switching tabs, and performing other operations.

L 2 JS > index.js > ...

150 // Immutability Proof

151

152 let str = "MBS Coding";

Q 153 str[0] = "0";

154 console.log(str);

155

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js  
MBS Coding  
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS>

powershell  
powershell

<> ⌂ 0 ⚠ 0 Ln 153, Col 12 Spaces: 2 UTF-8 CRLF { } JavaScript Finish Setup Go Live Prettier

A screenshot of the Visual Studio Code (VS Code) interface. The main area shows a code editor with the file `index.js` open. The code contains a section titled `// Immutability Proof`. The terminal below the editor shows the output of running the script with Node.js, displaying the string "MBS Coding". The status bar at the bottom provides information about the current file: line 152, column 23, spaces: 2, CRLF, and the language is set to JavaScript.

index.js

```
L 2 JS > index.js > str
149
150 // Immutability Proof
151
152 let str = "MBS Coding";
153 str = "MBS Coding Official";
154 str[0] = "O";
155 console.log(str);
156
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
MBS Coding
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS>
```

Ln 152, Col 23 Spaces: 2 CRLF { } JavaScript Finish Setup Go Live Prettier

The screenshot shows a Visual Studio Code (VS Code) interface. On the left, there's a vertical toolbar with icons for file operations, search, find, and settings. The main area displays a code editor with the following content:

```
159 let arr = [10,20,30,40];
160 arr[0] = 50;
161 console.log(arr);
162 arr.push(60);
163 console.log(arr);
```

The code uses `console.log` to print the state of the `arr` variable at two points: before and after modifying it. The output in the terminal window below shows the array changing from [10, 20, 30, 40] to [50, 20, 30, 40, 60].

Below the code editor, the tab bar shows "index.js" is the active file. The bottom navigation bar includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. The status bar at the bottom indicates the current line (Ln 159), column (Col 12), and file type (JavaScript). To the right, there's a terminal window showing the command prompt and the execution of the script.

A screenshot of the Visual Studio Code (VS Code) interface. The main area shows a code editor with a file named `index.js`. The code demonstrates how to change a property of an object:

```
L 2 JS > index.js > ...
164
165 // Object - Immutable => Change
166
167 let obj = {
168   name:"MBS",
169   age:25
170 }
171 console.log(obj.name);
172 obj.name = "MBS Coding";
173 console.log(obj);
```

The terminal below shows the output of running the script with Node.js:

```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
{ name: 'MBS Coding', age: 25 }
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> []
```

The sidebar on the left contains various icons for navigating through files, searching, and managing projects.

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The main area displays a code editor with a file named `index.js`. The code contains several lines of JavaScript, including variable declarations and a `console.log` statement. The terminal below shows the execution of the script, displaying the output in the console.

`index.js`

```
156
157 let a = 10;
158 let b = a;
159 b = 20;
160 console.log(a,b);|
161
162
```

TERMINAL

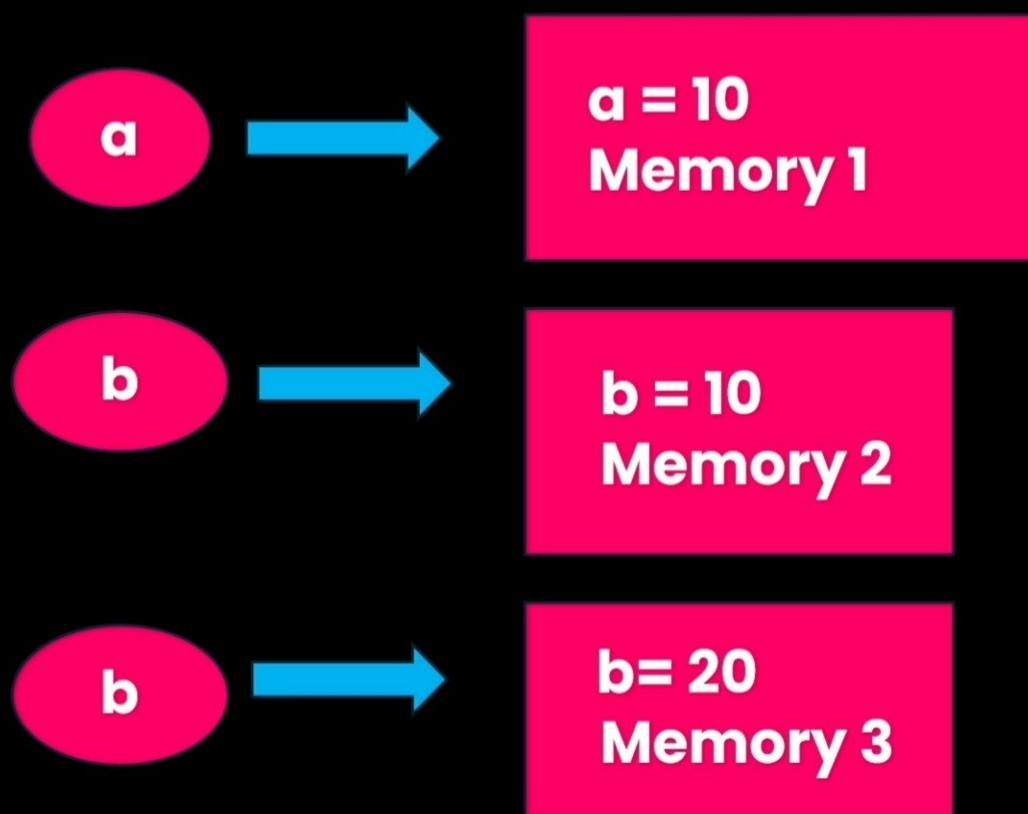
```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
MBS
{ name: 'MBS Coding', age: 25 }
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Ln 160, Col 18 Spaces: 2 UTF-8 CRLF { } JavaScript Finish Setup Go Live Prettier

## Primitive Data Type is Immutable

```
let a = 10;  
let b = a;  
b = 20;  
console.log(a,b);
```



index.js

```
156
157 // PD - Pass by value or Copy by value
158
159 let a = 10;
160 let b = a;
161 b = 20;
162 console.log(a,b);
163
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js  
10 20  
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS>

<> ⚡ 0 ⚡ 0 Ln 160, Col 11 Spaces: 2 UTF-8 CRLF { JavaScript Finish Setup Go Live Prettier

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The main area displays the content of a file named `index.js`. The code defines an object `obj` with properties `name` and `age`, logs it to the console, creates a copy `obj2`, and logs it again. The terminal below shows the execution of the script and its output.

```
L 2 JS > index.js > ...
184
185 let obj = {
186   name:"MBS Coding",
187   age:25
188 }
189 console.log(obj);
190 let obj2 = obj;
191 console.log(obj2);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + ...

```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
{ name: 'MBS Coding', age: 25 }
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS>
```

Ln 189, Col 18 Spaces: 2 UTF-8 CRLF { } JavaScript Finish Setup Go Live Prettier



A screenshot of a Microsoft Edge browser window. The address bar at the top contains the text "JAVASCRIPT". Below the browser is a dark-themed code editor interface.

The code editor shows a file named "index.js" with the following content:

```
L 2 JS > index.js > ...
188 }
189 console.log(obj);
190 let obj2 = obj;
191 console.log(obj2);
192 obj2.name = "MBS Coding Official";
193 console.log(obj2);
194 console.log(obj);|
```

The code uses `let` instead of `var` to declare `obj2`. When run in the terminal, it outputs two objects with the same properties: the original object and the modified one.

The terminal output is as follows:

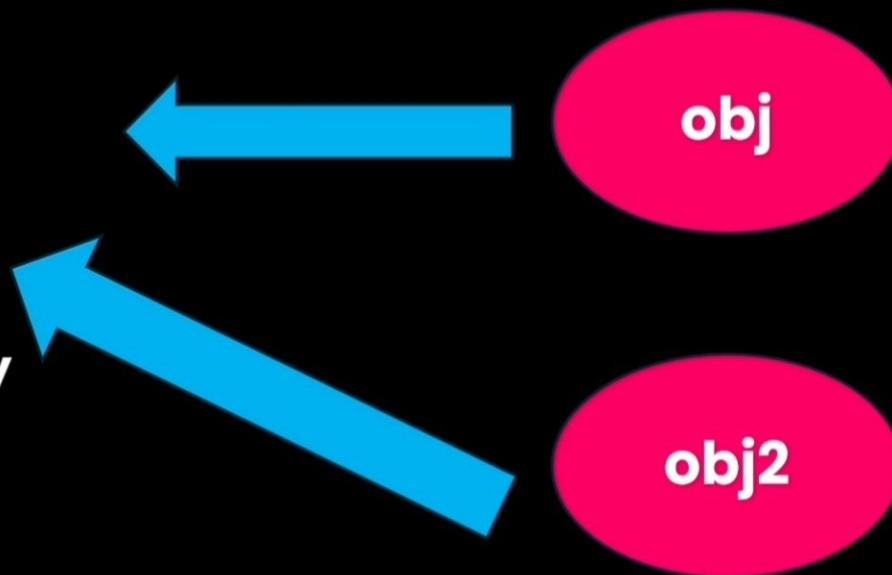
```
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> node index.js
{ name: 'MBS Coding', age: 25 }
{ name: 'MBS Coding', age: 25 }
{ name: 'MBS Coding Official', age: 25 }
{ name: 'MBS Coding Official', age: 25 }
PS C:\Users\mahes\Desktop\JAVASCRIPT\L 2 JS> []
```

The status bar at the bottom of the code editor displays the following information: Ln 194, Col 18, Spaces: 2, UTF-8, CRLF, { } JavaScript, Finish Setup, Go Live, Prettier, and a settings icon.

## Object Pass by Reference / Reference Copy :

```
{  
name: "MBS Coding",  
age:25  
}
```

**Not Created Separate Copy**  
**Share Same Data**



## **Why Pass by Reference?(Memory Optimization)**

- 1. My Code → Mere System Run → Mere Ram Run**
- 2. Object size → Real world api deal → 500 Elements  
Present : 500 Key value present**
- 3. 5 Mb First, 5 Mb Second, .... Size very big object**
- 4. Concept: Jinka Size Bahut Bada → Copy ✗, Same  
cheej refer**
- 5. Array : Object → 1000 Elements**

## **Why Pass by Reference?(Memory Optimization)**

- 6. Concept : Jinka size chota => Copy ✓**
- 7. Number = 8 byte, Boolean = 1 byte, String = Jitne uske andar character = Jitne size ka wah hogा = 3,4 byte**
- 8. Premitive data type size kam = Pass by value = new memory location → Copy ✓**
- 9. Non premitive data type → Not copy → If copy program crash → Big data present**
- 10. Thats why → Non Premitive sab ek cheej ko refer → Big but ek cheej ko refer memory consume less**