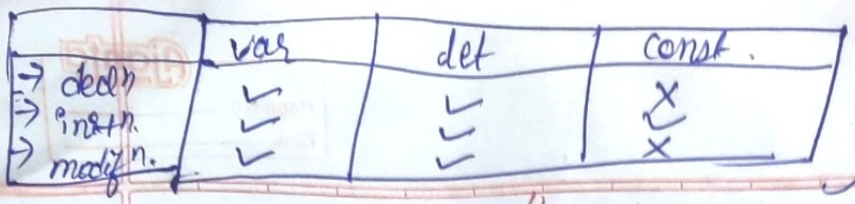

V.V.I. Important

```
x = 220 ; // modification.
b = 101 ;

document.writeln(x)
document.writeln(b)

</Script>
</body>
</html>
```

* static → changes allones 100% } don't confused.
* const → changes not allows }

18/11/23 Sat

Eg:- <h2> demo on const </h2>

```
<Script>
    const PI = 31.4 ; // varaible initialization (const)
    document.write()
// not allowed modification for eg: PI value can't change.
// Alwz prefered in capital letters for recognisations.
// PI = 1.6 ; Error. // modification not allowed.
const x ; // declaration not allowed. (msg will be missing initilalzer in const declearation.)
// int a = 10 ; // not allowed (unexpected initil")
// String name = "Venkatam" // not allowed.
```

non-changeble fixed.

due to this because only var, const, & let we can use.

Eg:- <!-- Example on JS variables.

⇒ loosely typed :-

JS did not provide any data types declaring var. & a varaible in JS can store any type of value. Hence Java script is loosely typed programme. We can use a variable directly without declaring it in JS., its called dynamic programming.

implicity / explicity.

any type of data / value we can assign. but possible one time only

means if are not assign the variable, system will define / assign.

Eg:- <head>
<title> JS </title>
</head>
<body>
<h2> demo on loosely typed & dynamic typed </h2>
<script>
var n; // var decln.
n = 10.56; // n = anything valid any type of no, string, boolen, etc, not restricted.
// loosely typed beej
document.write(n); // o/p → 10.56.

dynamically typed
// x = 100; // Assign. (x define inside the window)
document.write(n(x));
document.write(x * 15);
y = "apple"; // dynamically typed.
document.write(y); → // window.x

without writing window, also we can get value, not compulsory.

Both can we use but there is preference. that's it. nothing more.

#	var	let.
→	used for var decln.	→ same
→	Global decln. level.	→ func^n decln. level, block.
→	Since JS1. (old version of JS)	→ Since JS6 (ECMA script) ↳ new version
→	Supports Var. re-defination	→ not supported.
→	Supports var hosting.	→ Not supported.
→		

* const variable name also should be upper case.

Page No.
Date

<body>

Eg:- <h2> demo on var. Vs let </h3>

<Script>

{ var n = 100 ; // decln [first time]
document.write(n); // print our value.
/* for breaking line, not side by side, totally line break, we will use
 */

O/p
100

re-assign value.

{ document.write(n, "
");
var n = "rama" ; // re-constructing with string type. Or re-defination.

O/p
rama.

Or

let n = false ; // ERROR. (bcj in var we already decln 'n' variable so, we can't re-assign in diff. variable

var n = false ;
document.write(n, "
");

first use the variable than decln the variable

{ X = 101 ;
document.write(X, "
");
var X ; // hoisting.

~~1st decln than initi~~

→ first use than inform decln.

1st initn than decln only allowed.

{ Y = "mango" ;
document.write(Y, "
");
let Y ; //ERROR.
var Y ; // accepted. // hoisting.

* let & const doesn't support re-defination. only var supports.

20/11/23 Mon

Javascript datatypes :-

* In JS there are 2 types of datatypes:-

i) Primitive datatypes :-
ii) Non-primitive datatypes.

i) Primitive Datatypes :- • Primitive datatype allows data directly.
→ These datatypes allows us to store *only 1 value @ time.
→ These are popularly k/n as non-reference.
→ Stack Area.
→ Non-sharable.

ii) Non-primitive data types:-
→ reference / address.
→ N values @ time.
→ Heap Area.
→ Shareable.
→ reference.

Javascript has a 5 primitive data types:-
** (all datatypes starts with lowes case)

1) string
Eg:- "Siva", 'apple', `Hello`.

2) number
Eg:- 10, -25, 100.56, -3.7 22299999 etc

3) boolean
Eg:- true, false.

4) undefined
Eg:- value not assigned.

5) object.
Eg:- null.