Word vs Keyword ☑
Var Const let ☑

- Hoisting :-
    Variable and functions are hoisted which
means there declaration is moved on the top
of code.

For Example :-

```
Console.log (a)
Var a = 12;  // Javascript convert it into

    Var a;
    a = 12;
```

Here, declaration is moved on the top of code
that look like :-

```
Var a;
Console.log (a)
a = 12;
```

This is Known as Hoisting.

- <u>Not defined</u> :-

For example :-
                Console.log (b)   Line :-01

Here, b is not defined.

- <u>Undefined</u> :-

For example :-
                Console.log (b)
                Var b = 12; → Javascript convert it

1.   Var b;   Line :-01
2.   Console.log (b)   Line :-02
3.   b = 12;   Line :-03

It display Undefined.
This is called Hoisting.

- Types in js :-

    • Primitive & Reference.

<u>Primitives</u> :- Number, Null, String, boolean, Undefined.
<u>Reference</u> :- [ ] ( ) { }

Note :- Aisi Koi bhi value jisko copy Karne par
real copy nahi hota, balki us main value Ka
Reference pass ho jaata hai, use hum reference
Value Kahte hai, aur jiska Copy Karne par
real copy ho jaaye wo primitives type value
hoti hai

For example:-

            Var a = 12;   Result :- a is 12
            Var b = a;   b is 12
            b = b+2;   After b is 14 & a is 12.

For example :-

            Var a = [1, 2, 3, 4];
            Var b = a;
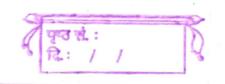   b. pop ();

Result :- Note :- a ke pass hai khud Ka [1,2,3,4] or
                  b ke pass hai a Ka [1,2,3,4]

        This is called Reference

        b value is [1,2,3] &
        A value is [1,2,3]

When variable contain values with brackets i.e [], (), {} This is Reference we can't copy it but Real Copy whow to do it will discuss later.

IF value like Var a = 4, This is primitive. Bracket hai to voh Reference hai or jis mai bracket nai hai wo Primitive hai Simple.

* Conditional :- if else else-if

// jab bhi bat agar magar par aayegi, yaa fir baat aayegi aisa hua to ye karo waisa hua to ye karo

if (True or False) Only Two values are there either True or False

* loops:-
    loop ka matlab repeat.

    for (var i=0; i<11; i++) {
        Console.log (i);
    }

Functions :-

functions mainly teen kam ke liye hote hai
1. Jab aapka code ap turant nahi chalana chaahte future mein chalaana chaahte ho.
2. jab aapka code aap reuse karna chahte ho
3. jab aap code chalana chaahte ho har baar with different data

function ka matlab aap kuchh code ko likh kar koi naam de sakteh ho and baad mein usey use kar sakte ho with that name as many times.

Function = code ko naam dena

```
function abcd () {
    // function statement.
}
```

1.
```
function hellobolo() {
    Console.log ("Hello");
}
hellobolo();     Day :-01  } Reuse.
hellobolo ();    Day :-02
```

⌐—→ Parameter,

```
Function abcd (a) {
    Console.log (a);
}
```

abcd (12); ⌐—→ Arguments.

arguments = real value jo hum dete hai fnc chalaate waqt

Parameter = Variables jinme value store hoti hai arguments waali.

- **arrays:-** hum ek variable mein ek value store kar paate hai par jab humein ek se jaada value store karni ho tab fir use hota hai array ka, matlab ki aapko freedom deta hai ek se jaada value use karne ki.

Array :- Group of values.

$$0 \quad 1 \quad 2 \quad 3 \quad 4 :- \quad Index$$
$$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$$

Var arr = [1, 2, 3, 4, 5];

```
arr.push(8);      // Add Number i.e 1,2,3,4,5,8
arr.pop();        Remove last Number. i.e 1,2,3,4
arr.unshift(0);   // Add New member i.e 0,1,2,3,4,5
arr.shift()       // Remove member from start i.e 2,3,4,5.
arr.splice(2,1)   // Remove no from between i.e.
         3 Remove Result:- 1,2,4,5,8.
```
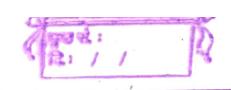
Push, Pop, shift, Unshift, Splice.

Object :-

ek se jaada bande Ki baat Ki to hua array, ek bande Ke baare mein Saari baat Ki to hua object.

object hai ek bande Ki details Ko hold Kaana, in a Key value pair

1. Blank obj :-

   Var a = {};

2. Filled obj :-

   Var a = {

   age: 24,
   name: "Harsh",
   email: "RanpariyaKeshav",
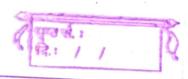   contact: "9499xxxx"

   }

How to access a value :-

a.name :- Access a Value.

Result :-

Harsh.

- Props vs method.

      ┌──→ properties.

Var Watch = {

Props {
      Name: "Rolex",
      price: "16K",
      Color: "black",

method ← Kuchhbolo: Function () { }
cz itz
prop Ki value
Function hai.

}

object Me aandr asi propertie jish Ki value
Function hai use Method Kehte hai.

- <u>Updating object props</u>:-

Watch. Name = "Rolex di Mario"
Console.log (Watch)

Result:-
    Name :- Rolex di Mario
    Price :- 16K
    Color :- black.

                <u>End of</u>

        <u>Java Script Learn What Matters</u>