data types are of 2 types.

1. primitive data type
 a. number.
 b. string.
c. boolean(true/false).
d. null.
e. Undefined.

2. non primitive data type.
 a. object.
 b. array
 c. function

d.

default value given by JS is undefined

var x = 10

Declaring

value

{

name : 'mahesh'

age : 24

pet : "Puppy"

}

{

name'.

age

pet : null

}

== → value

=== → value + datatype.

Input → datatype

output → boolean

$5 == 5 \longrightarrow$ true.

Type Coercion.

$\overset{N}{5} == \overset{S}{\underbrace{"5"}_{5}} \longrightarrow$ true

$\longrightarrow 5 == "5" \longrightarrow$ false

type coercion---> conversion of a number which is present inside a string to number.

Array ⟶ mutability

String ⟶ immutability

Let, Const ⟶ { }

Let, Const ⟶ global

var x

let → global, block scope

→ re-assigned

const → global, block scope

→ cannot be re-assigned

```
let a = 10

a = 20
```

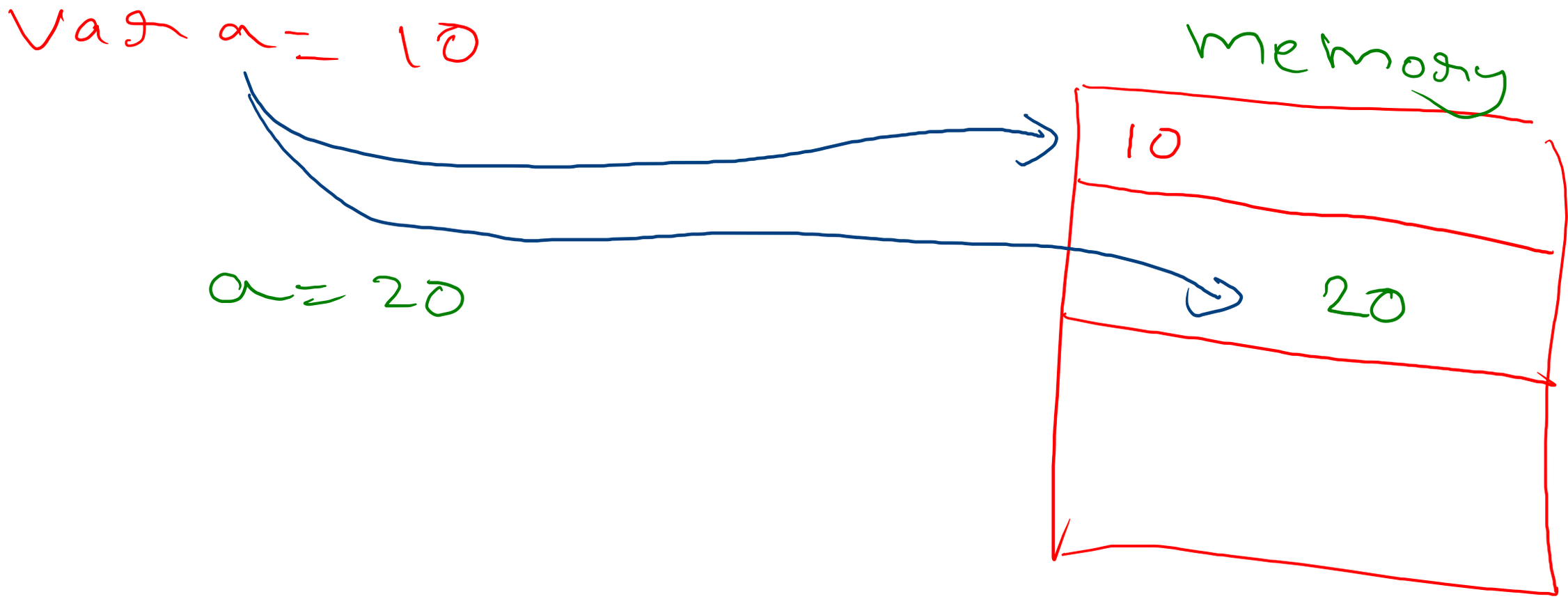
→ `console.log(a)`

→ 20

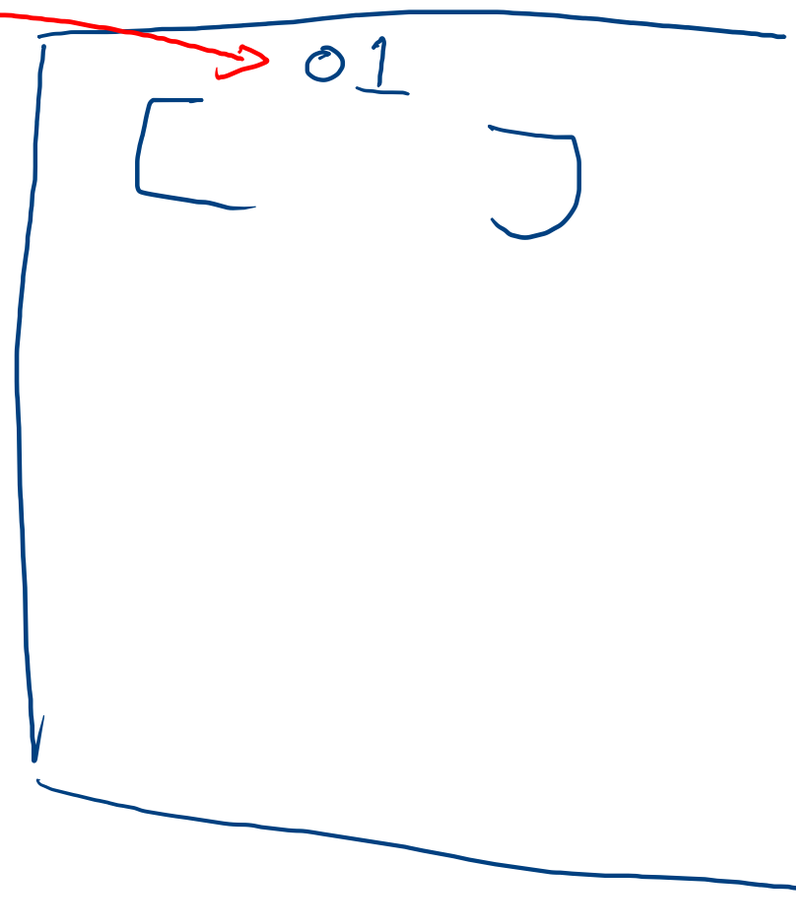
```
const a = 10

a = 20
```

`console.log(a)`

→ Error


Var a = 10
memory
10
a = 20
20

var a = [ ]

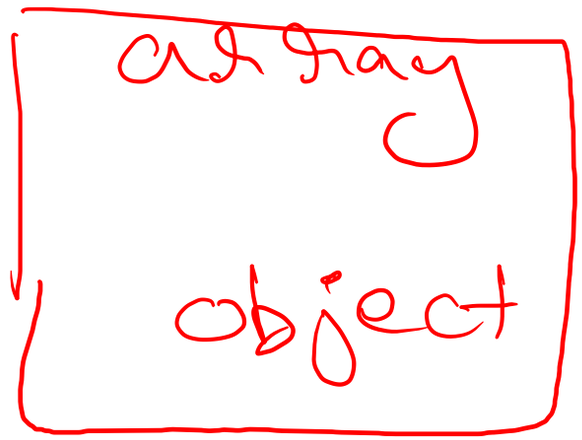


primitive

a

↓

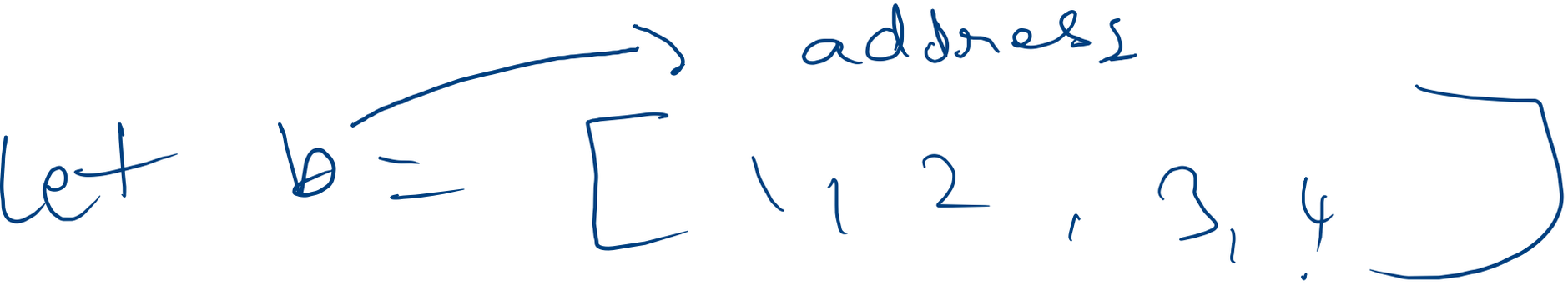
value



↓

address


a =
" string "


address
let b = [1, 2, 3, 4]

address

```
let a = [1, 2, 3];

let b = a; // address.

b[0] = "hi";
console.log(a);
```

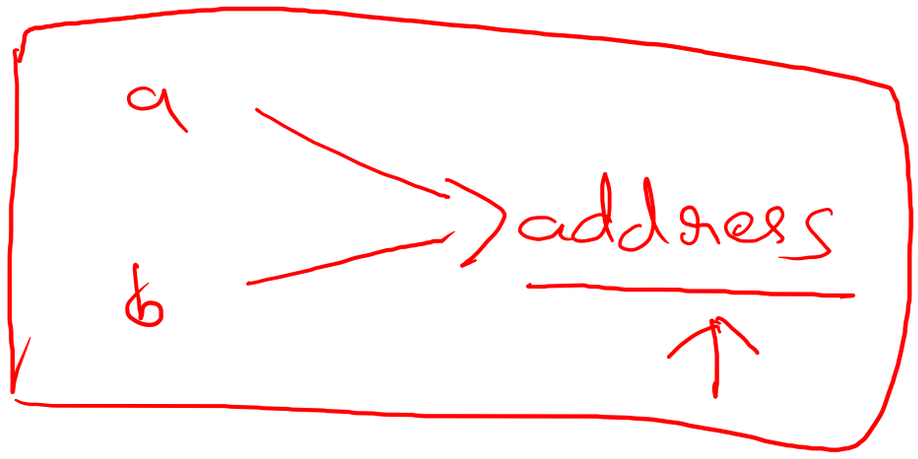
```
let x = 10;

let y = x; // 10

x = 20;

console.log(y);
```

array

object

↓

address



X = 10

Y = x // 10

x ⟶ 20

Array

vs

String

↓

Mutability

↓

Immutability