10 September 2024 15:24

MERN

JS

M -> DB

E -> Framework (Nodejs)

N -> Runtime Environment (JS)

R -> Library .

ES6 (2015)

ECMA -> European Computer Manufacturing Association

ECMA 6

ES 6

Engine -> Compiler + Interpreter

Safari -> JS loops

I E -> Chakra

Firefox -> Spider Monkey

Brave /Crome -> V8 Engine

JS says

Functions are the No. 1 citizen .

Object

object

var a ;

arguments

function add(x, y)

Loosely Typed

int, float, double, long

char, string

Function add(int x, float y)

{

}

var a = 10;

a = " " ;

int a;

a = 10;

a = 10000;

a = "Praja" ;

GOD level / TOP Level

Object ( )

Object ( )
{

}

var obj1 = { }

Parent

var arr = [ ]

Array ( ) {
}

arr instance of Object
true

var a = 1000;

a++;

a = 1001

increment + assignment

a + 1
1001

a;

1000

a=a+1

a++

a+=1

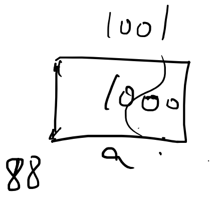
a=a+1

Immutability

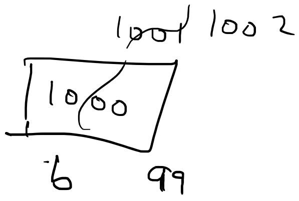
changes done in Either copied value or original value, does not affect the other

var a = 1000;

var b = a;



a++



b++

b++

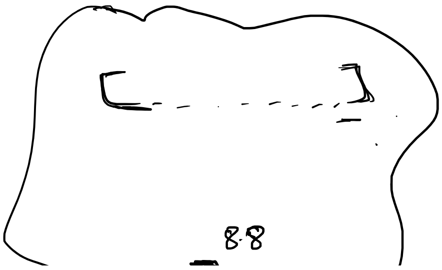
Immutability

number, string, boolean, bigInt,

var a;

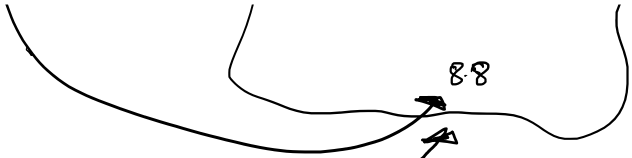
var arr1 = [ ]

arr2 = arr1;



var arr2 = arr1;

memory address , reference.



arr2.pop()

Mutability.

changes done in Either original value or copied value, will ultimately affect the other.

By car.

Keys.

Keys : —> car, Same car.

car → dent;

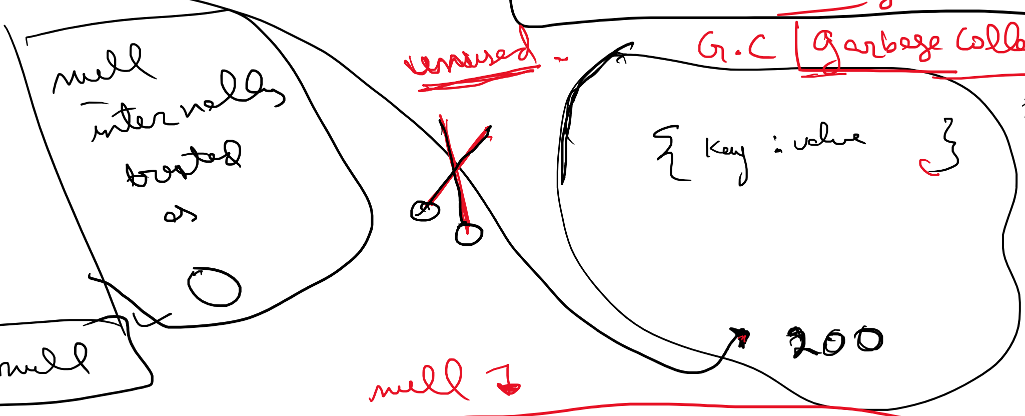
Original car damage.

var obj1 = {

}

type of obj1

'object'



G.C (Garbage Collection)

algo

Heap memory

obj1 = null

null ↓

To dereference an object

acts like scissor.

primitive type

'object' ← arr, obj

p = " "

p = ' '

p = `  `

'string'

var a = new String ("priya");

new → object

String

Explicitly

AUTOBOXING

" priya" ;

Powers,

methods,

Properties

var a = " priya" ;

a.touppercase();

'PRIYA' ;

normal String (value type)

Properties

convert value type.

undefined == null

== loose comparision

① Type Convert

② value checking

empty values

=== strict comparision

① Type check

② value check