Javascript developed by **Brendan Eich**, Netscape,1995. Mocha->Livescript->javascript

- Why name as script – it execute as the page loads

*- <script type=”text/javascript”>* type=”text/javascript” is not needed

- External js files can be stored as cache in browsers

-’use strict’; to use only latest functionality of javascript versions

-whitespace, case-sensitive

|  |  |  |  |
| --- | --- | --- | --- |
| Javascript reserved keywords | | | |
| break | as | any | switch |
| case | if | throw | else |
| var | number | string | get |
| module | type | instanceof | typeof |
| finally | for | enum | export |
| while | void | this | new |
| null | super | catch | let |
| static | return | true | false |

**Variables**: represents name for a memory block

var name=”rajesh”; //scope dependent for a function or window object

let name=”rajesh” //used as block scope, cannot be redeclared

const NAME=”rajesh” //cannot be reassigned, redeclared, requires declaration, immutable

Template literal: can use ${name} inside string.   
 var name = ‘rajesh’; console.log(“my name is ${name}”);

**Datatypes:**

1.Number - integer & floating point

2. String - one or more characters

3. Boolean - true/false

4. Null - Unknown but (null==undefined) is true , (null==0) is false

5. Undefined – Known by non - defined

6. Object - collections of variables and properties

7. Symbol – unique identifiers

**Object:**

var user = new Object(); // "object constructor" syntax

var user = {}; // "object literal" syntax

user.name=’rajesh’;

delete user.name;

-for object keyname, no restrictions for reserved words, bu \_\_proto\_ will not work

-check key existense in object *“key” in obj;*

|  |  |
| --- | --- |
| function makeUser(age) {  return {age: age};} | function makeUser(age) {  return {age};} |

**for key….in** - *for(key in object){………}* //this loop run in an ascending ordered manner for number keys

**Comparing object** : obj1={}; obj2={}; obj1==obj2; //false obj1===obj2; //false

obj1={age:22}; obj2=obj1; obj1==obj2;//true obj1===obj2;//true

**Const object:**

Array:

var arr[“name”,’age’,’job’]=[“rajesh”,”12”,’vetti’]; console.log(arr[“name”]); //rajesh

[1,2,3].includes(2); //return true false

[1,2,3].findIndex (2); //return true false

String:

“Hello world”.include(“world”); //return true false

“Hello world”.startswith(“H”, index); //return true false

“Hello world”.endswith(“d”, index); //return true false

“sorry ”.repeat(100); //print sorry 100 times

Conditions:

if

if..else

nested if else if

switch case

Loops:

for

while

do...while

Loop control:

break:

continue:

Function:

function functionName(){...body....}

functionName(); //function calling

Rest parameter(variable arguement):

method1: function functionName(..params){console.log(params[0]; console.log(params[1];)}

functionName(1,2,3); //function calling

method2: function functionName(i,j){...body...} functionName(...arrayName);

Arrow function:Es6

let functionName = (parameter)=>{...function body.....}

Default parameter:

function functionName(i=1){....function body.....}

let objName={name:"rajesh",age:"22",job:"it"};

let {name,age,job}=objName;

console.log(name); //rajesh

ES5:

function Cricketer(name,age,position){

this.name=name;this.age=age;this.position=position;

}

Cricketer.prototype.changePosition=function(position){

this.position=position;

}

var cricketer = new Cricketer(“rajesh”,”22”,”batting”);

console.log(cricketer);

crickter.changePosition(“bowler”);

console.log(crickter);

ES6:

class Cricketer {

constructor(name,age,position){

this.name=name;this.age=age;this.postion=position;

}

editPosition(position){

this.position=position;

}

}

let crickter = new Crickter(“rajesh”,”2”,”batting”);

console.log(crickter);

crickter.editPosition(“bowler”);

console.log(cricketer);