**JAVA SCRIPT**

To **print** 🡪 console.log(variable\_name);

**Run code** terminal: node filename.js

**Variable:**

const accountId = 144553

let accountEmail = "abc@gmail.com"

var accountPass = "1234@"

* var, let and const is variable
* Prefer not to use var,

because of issue in block scope and functional scope

Print as a table multiple variable 🡪 console.table([variable1, variable2, variable3]);

const accountId = 144553

let accountEmail = "abc@gmail.com"

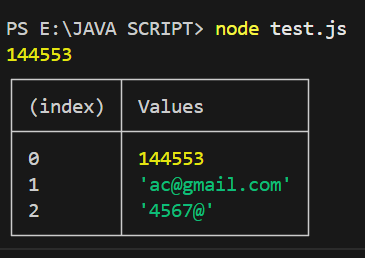
var accountPass = "1234@"

console.log(accountId);

accountEmail = "ac@gmail.com"

accountPass = "4567@"

console.table([accountId, accountEmail, accountPass]);

**output:**

**Datatype:**

"use strict"; this will treat all JS code as a newer version

Number range : 253 if number is larger than this we use **bigint**

string 🡪 ""

boolean 🡪 ture/false

null 🡪 standalone value

symbol 🡪 uniqe

console.log(typeof "wasi"); to get the type

"use strict";

let name = "wasi"

let age = 3

let isLoggedIn = true

console.log(typeof "wasi");

output:



**Datatype conversion confusion**

"use strict";

let score = "33"

console.log(typeof "score");

let valueInNumber = Number(score)

console.log(typeof valueInNumber);

console.log(score);

let result = "ab33b";

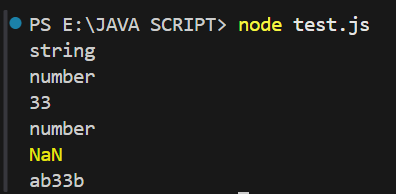
let valueInresult = Number(result);

console.log(typeof valueInresult);

console.log(valueInresult);

console.log(result);

output:



"33" 🡪 33

"33ab3" 🡪 NaN

""

"use strict";

let val = 1

let booleanVal = Boolean(val)

console.log(booleanVal);

Output: true

For Boolean

* 1 🡪 true and 0 🡪 false
* "" 🡪 false
* "wasi" = true

**Oparetions:**

* +true 🡪 1
* Increment:

let gamecounter = 100

gamecounter++;

console.log(gamecounter);

output : 101

**Comparison of datatypes:**

== work differently and <,>, <=, >= work differently.

Comparisons convert NULL to a number, treating it as 0.

null >= 0 🡪 true.

null > 0 🡪 false.

**Datatype of JS:**

* Primitive: 7 type of JS. Call by value.

1. String
2. Number
3. Boolean
4. null
5. undefined
6. Symbol
7. BigInt

const bigNumber = 3456543576654356754n

* Reference(Non Primitive):

1. Array
2. Objects

const heros = ["shaktiman", "naagraj", "doga"];

let myObj = {

    name: "Wasi",

    age: 22,

}

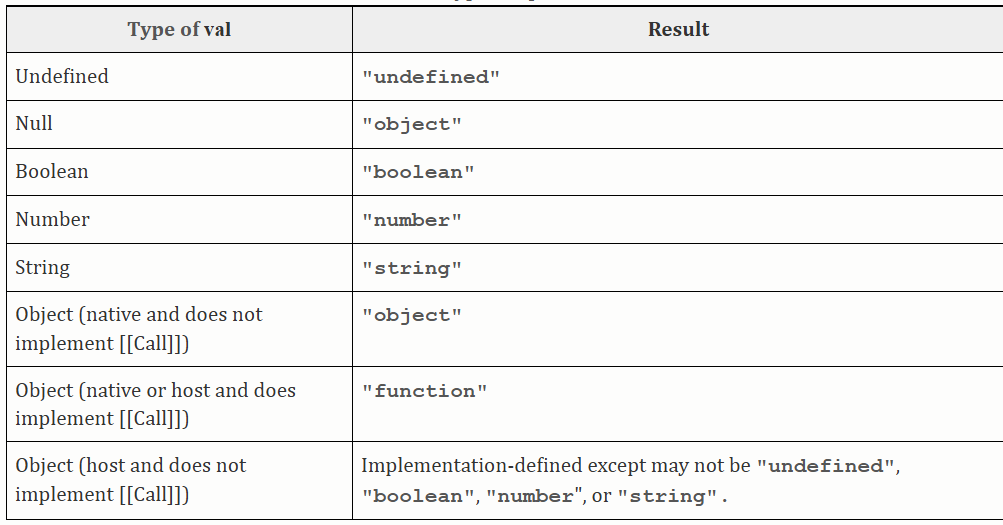
1. Functions

const myFunction = function(){

    console.log("Hello world");

}

console.log(typeof anotherId);



**Stack and Heap Memory:**