JAVASCRIPT

2 types of datatypes :

Defined and undefined

Defined --- Number, string, boolean, BigInt(literal, function)

(character is not present, only string even for single character)

Number --- integer, floating values, infinity, NaN

integer, floating values --- success

infinity, NaN ---- error

NaN is a number.

To not show NaN and and display something else use isNan() method

a= 9007199254740992n //literal

a=BigInt("6472847357238732850326"); //function

N means it is big int value

BigInt works with only BigInt

Here type casting is at runtime

Use backticks for multiline strings and ' '," " for single line

"A is "+a+" B is"+b;

Easy way is : `A is ${a} and B is ${b}`; //string interpolation

True--1, false--0; //10+true; --- 11

Number+string--string

BigInt+string--string

Number+boolean--number

BigInt+Number=typeError

Hoisting---Variable declared at top

function show() { var t; //this is declared at top of funtion

t=10; //this is assignment

ggg = 1000; //declared at top of tab(attached with tab), global

}

Block scope--->function scope---->global scope----->reference error

Object is a collection of different/ same values.

var Amit = {id:1001, name:"Amit"} --- object created

ASSIGNMENT-1

Q1: Why dot use with value type?

Ans: Impilictly primitive type gets converted to equivalent reference type(called wrapper). After autoboxing, it is able to access properties and methods.

Q2: What is null and where we use it? Hint: typeof

Ans: Null means removing reference that was stored in the variable. The variable is no longer able to access the object, since no refence. This is done to make an object free, which then becomes available for garbage collection.

Primary types should not be made null, else they become objects.

Q3: typeof?

Ans: typeof is an operator that gives the datatype of a variable.

Eg: var a = 100;

typeof a;//number

Q4: Null vs undefined?

Ans: Null removes the reference of an object, making it eligible for garbage collection.

Undefined simply means no value is assigned to a variable. They are not same.

Q5: == vs ===?

Ans: == is loose comparison operator. It first does type conversion from one type to another, then compares the vales.

=== is strict comparison. It checks the type of both the values. If the type is same, then compares the values.

Q6: Prime number compute function?

// take input from the user

const number = parseInt(prompt("Enter a positive number: "));

let isPrime = true;

if (number === 1) {

console.log("1 is neither prime nor composite number.");

} else if (number > 1) {

// looping through 2 to number-1

for (let i = 2; i < number; i++) {

if (number % i == 0) {

isPrime = false;

break;

}

}

if (isPrime) {

console.log(`${number} is a prime number`);

} else {

console.log(`${number} is a not prime number`);

}

}

// check if number is less than 1

else {

console.log("The number is not a prime number.");

}

Q7: Armstrong number compute function?

Ans:

let sum = 0;

const number = prompt('Enter a three-digit positive integer: ');

let temp = number;

while (temp > 0) {

// finding the one's digit

let remainder = temp % 10;

sum += remainder \* remainder \* remainder;

// removing last digit from the number

temp = parseInt(temp / 10); // convert float into integer

}

// check the condition

if (sum == number) {

console.log(`${number} is an Armstrong number`);

}

else {

console.log(`${number} is not an Armstrong number.`);

}

ASSIGNMENT-2

Ques 4:

const date = new Date();

let day = date.getDate();

let month = date.getMonth();

let year = date.getFullYear();

let currentDate = `${month}-${day}-${year}, ${month}/${day}/${year} or ${day}-${month}-${year}, ${day}/${month}/${year}`;

console.log(currentDate);

Ques 3:

const date = new Date();

const curr = date.getDay();

const dayNames = {1:"Monday",2:"Tuesday",3:"Wednesday",4:"Thursday",5:"Friday",6:"Saturday",0:"Sunday"};

console.log(`Today is : ${dayNames[curr]}`);

console.log(`Time is ${date.getHours()} - ${date.getMinutes()} - ${date.getSeconds()}`);

function x(...args) {

console.log(args);

}

function x(a,b,c,d,...args) {

console.log(args);

}

undefined

x(10,20,30,40,50);

ASSIGNMENT -3

Ques 1:

function calculate() {

let length = Number(window.prompt("Enter length"));

let breadth = Number(window.prompt("Enter breadth"));

console.log("Perimeter of rectangle is :"+(2\*(length+breadth)));

console.log("Area of rectangle is :"+(length\*breadth));

}

calculate();

//anonymous funtion

function calc(){

var add = function(x,y) {

return x+y;

}

var subtract = function(x,y) {

return x-y;

}

return {add,subtract};

}

calc().subtract(3,2);

//arrow functions

const add=(a,b)=>{

console.log("I am arrow");

a+b;

}

ASSIGNMENT-4

1. const salaries = [10000,20000,30000,5000];

var sum = salaries.reduce((accumulator,currentValue) => {

if(currentValue > 10000) {

accumulator = accumulator + currentValue;

}

return accumulator;

},0);

console.log(sum);

1. const salaries = [10000,20000,30000,5000];

var maxSalary = salaries.reduce((acc, cur) => {

if(cur > acc) {

acc = cur;

}

return acc;

},0);

console.log(maxSalary);

1. const salaries = [10000,20000,30000,5000];

var count = 0;

var countSalary = salaries.reduce((acc, cur) => {

if(cur > 10000) {

count++;

}

return count;

},0);

console.log(countSalary);

1. const employee = [{name:"Ram",salary:10000}, {name:"Tom", salary:35000}, {name:"Rohan",salary:25000}, {name:"Komal",salary:45000}];

employee.sort((a,b) => a.salary - b.salary);

employee.sort((a,b) => a.name.localeCompare(b.name));

employee;

1. const salaries = [10000,20000,30000,5000];

newSal = salaries.map((e) => e + 0.10);