Data types:

1. Primitive

i. string

var str=”Rahul Ghosh”;

var str2=’Rahul Ghosh’;

ii. Number

var x=3.6;

iii. BigInt

var bigint=12346564546461651165;

iv.Boolean

var a=2;

var b=3;

var c=2;

a==b//return false

a==c//return true

v.Undefined

var x;

vi. Null

var x=null;

1. Non-premitive:

Var obj={

X:25,

Y:”Hello World!”,

Z:function(){

Return this.x;

}

}

Generally hoisting mane holo jokhn amra code likhchi suppose variable declare korar agei

Sei variable ta k print kora te chaichi then variable ta Jodi pore jekane khusi declare kora thake tao variable tar output pabo, but Jodi use strict(); likhe di then eta error debe.

== compares only values

===compares both values and types.

Let and const has limited scope mane let die define kora functions ke only declare kora block r modhyei access kora jay. Where as var is globally scoped.

String coercion:

Var x=3;

Var y=”3”;

x+y//returns 33

var x=24;

var y=”Hello”;

x+y//returns 24Hello

var x=3;

var y=”3”;

x-y//returns 0

javascript ekta dynamically typed language mane kono variable e value assign korar pore

in future again value assign korar time e amra different data type r value assign korte parbo as

javascript run time e variable r data types check kora hoy compile time e na kore.

NaN means Not A Number.

isNaN(“Hello”);//returns true

isNaN(123);//returns false

primitive data types der pass by value and non primitive data types der pass by reference.

Pass by reference mane address pass kora hche.

Pass by value:

Function modifyValue(x){

X=10;

Console.log(x);

}

a=5;

modifyValue(a);//10

console.log(a);//5

pass by reference:

function modifyObject(obj) {

obj.name = "John";

console.log("Inside function:", obj); }

let person = { name: "Alice" };

modifyObject(person); // Outputs: { name: 'John' }

console.log("Outside function:", person); // Outputs: { name: 'John' }

IIFE(Immediately invoked function):Also known as self invoking function

Runs as soon as its defined.

(function(){

//

})();

Characteristics of strict mode in javascript

1. Duplicate arguments are not allowed by developers.
2. In strict mode, you won't be able to use the JavaScript keyword as a parameter or function name.
3. The 'use strict' keyword is used to define strict mode at the start of the script. Strict mode is supported by all browsers.
4. Engineers will not be allowed to create global variables in 'Strict Mod

Higher order function:

Function inside a function,

Function higherOrderFunction(){

Return function(){

Console.log(“Inside”);

}

}

var x=higherOrderFunction();

console.log(x);//Inside

this keyword returns j object ta create kora hoyeche tar property

function Person(name) {

this.name = name;

}

const person = new Person('Alice');

console.log(person.name); // Outputs: Alice

call()::

function greet(greeting, punctuation) {

console.log(greeting + ", " + this.name + punctuation);

}

const person = { name: "Alice" };

greet.call(person, "Hello", "!"); // Outputs: Hello, Alice!

Apply()::

function greet(greeting, punctuation) {

console.log(greeting + ", " + this.name + punctuation);

}

const person = { name: "Alice" };

greet.apply(person, ["Hello", "!"]); // Outputs: Hello, Alice!

Call accepts arguments individually where as apply accepts arguments as an array.

The exec() method searches for a match in a specified string. If it finds a match, it returns an array containing the matched results; if no match is found, it returns null.

The test() method tests for a match in a specified string. It returns true if it finds a match and false otherwise

Both uses for RegEx or regular expression.

Currying::

Function add(a){

return function(b){

return a+b;

}

}

Add(3)(4)//return 7

Some advantages of external javascript are

1. It allows web designers and developers to collaborate on HTML and javascript files.
2. We can reuse the code.
3. Code readability is simple in external javascript.

Scope:

Global, function, block

Scope chain:

In javascript first the compiler searches the variable inside the local scope and if not found then searches it in outer scope, if not found then searches in global scope and if not found then a reference error is thrown.

Jokhn kono function er vitore arekta function define kora hoy then vitore thaka function outer scope e thaka function e declare kora variables k access korte pare eta kei closure bole.

function outerFunction() {

let outerVariable = 'I am from outer function';

function innerFunction() {

console.log(outerVariable);

}

return innerFunction;

}

const closureFunction = outerFunction();

closureFunction(); // Outputs: 'I am from outer function’

DOM(Document Object Module):

Dom puro web page ta k tree like structure e create kore dey jar modhye each node is an object representing a part of the document.

1.selecting elements:

getElementById

const element=document.getElementByID(‘myID’);

getElementByClassName

const element=document.getElementByClassName(‘myClass’);

getElementByTagName

querySelector: selects the first element that matches a css selector

const element=document.querySelector(‘.myClass’);

querySelectorAll:selects all elements that matches css selector

const element=document.querySelectorAll(‘.myClass’);

Creating elements::

Const elemet=document.createElemet(’div’);

Event handling::

Const button=document.getelemetById(‘myButton’);

Button.addEventListener(‘click’,function(){

alert(‘Button was clicked’);

}..mane jokhn amra oi button ta click korbo screen ekta pop us asbe jate message thakbe button was clicked and seta korar jnno amader addEventListener and alert() r help lagbe.

Mainly dom manipulation kora hoy interactive dynamic web applications create korar jnno.

same class multiple elements k assign kora jay mane suppose multiple elements are there jader same designs and functions hobe then tader k same class assign korbo

where as id use kora hoy kono single element k uniquely identify korar jnnno.

css e . die class choose kora and # die Id choose kora hoy.

An element can have multiple classes where as an element can have a single id.

Callback()::

Eta ekta function ja as an argument onno function e pass korano hoy and it is executed after some operation has been completed, callbacks use kore amra api theke data fetch kora, file read kora, event handing , etc kore thaki.

Function greet(name,callback){

Console,log(‘Hello’+name);

Callback();

}

Function goodbye(){

Console.log(‘Goodbye’);

}

Greet(‘Alice’,goodbye);// hello alice goodbye

Types of error:

Syntax error: ei khetre program execution er time e stop hye jabe.

Logical error: without any error program run kore jabe but desired result pabona.

**Benefits of Memoization**

* **Performance Improvement**: By caching the results of expensive function calls, subsequent calls with the same inputs return much faster.
* **Efficiency**: Reduces the number of redundant calculations, especially useful in recursive functions like Fibonacci or in dynamic programming problems.

The charAt() function of the JavaScript string finds a char element at the supplied index. The index number begins at 0 and continues up to n-1, Here n is the string length. The index value must be positive, higher than, or the same as the string length.