

Execution context

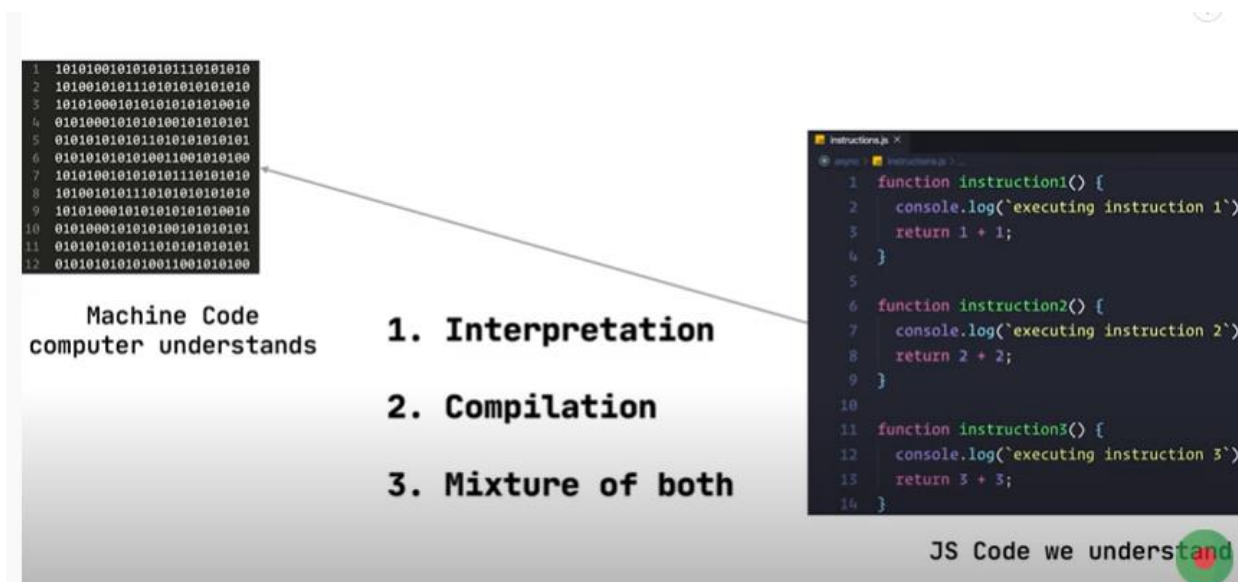
How to run js code?

How JavaScript code runs



.....engine js code k compile kore machine code a porinoto kore tokhon amder computer sheta bujte pare then amra output dekte pai.

Engine er bitor ki vabe js code ki vabe machine code compile hoi?



Line by line code nei machine code porinoto kore

1 10101001010101110101010

2 10100101011101010101010

3 10101000101010101010010

4 01010001010100101010101

5 01010101011010101010101

6 01010101010011001010100

7 10101001010101110101010

8 10100101011101010101010

9 10101000101010101010010

10 01010001010100101010101

11 01010101011010101010101

12 01010101010011001010100

Compiler

```
var myNumber = 10;
var myString = 'hello';
var myArray = [1, 2, 3];
var myObject = {name: 'John', age: 30};
var myFunction = function() {
  console.log('Hello World!');
};
myFunction();
```

✓ Fast Process

✗ System Crash & Hard to Debug

Pura code ekbare niya machine code porinoto kore fele

Interpreter

```
var myNumber = 10;
var myString = 'hello';
var myArray = [1, 2, 3];
var myObject = {name: 'John', age: 30};
var myFunction = function() {
  console.log('Hello World!');
};
myFunction();
```

✓ Easy to Debug

✗ Slow Process

+

Compiler

```
var myNumber = 10;
var myString = 'hello';
var myArray = [1, 2, 3];
var myObject = {name: 'John', age: 30};
var myFunction = function() {
  console.log('Hello World!');
};
myFunction();
```

✓ Fast Process

✗ System Crash & Hard to Debug

8

Just-in-time (JIT) Compiler

Compiler & interpreter er shomonoi JIT compiler.

JIT compiler ki vabe kaj kore?

JIT - Compiler

Interpretation

Machine Code

JIT - Compiler

Interpretation

Machine Code

```
function instruction1() {
  console.log('executing instruction 1');
  return 1 + 1;
}

function instruction2() {
  console.log('executing instruction 2');
  return 2 + 2;
}

function instruction3() {
  console.log('executing instruction 3');
  return 3 + 3;
}
```

JIT part interpretation er kaj kore and compiler machime code porinoto kore. JIT amder code take as it is rekhe dai. Jokhon ami kono function invoke/call kori , jamon instruction1 call korlam tokhon jit Just in time shatake machine code porinoto korbe and execute kore dibe. Akhon Suppose instruction2 call holo & error holo ai shomoi jit part bolbe wait I have just in time compiled it and error so tmi akhane thamo tai programmer k bole dite parbey tar mane easy to debug and fast enough. So best of both work. Aita e hossa JIT compiler er boishisto.

What is execution context?

JS tar je responsibility she tar interpretation kore amder code k machine code porinoto kore. Aita e js engine er kaj. Engine tar kaj k shohoj korer jonno complexity komanor jonno amder code k part part bag kore ekta ekta kore kaj kore. Ai choto choto part gulo ka execution contex bola hoi.

****Execution context hose vibino object. Aita 2 rokom hoi global & function execution context.**

2 tai hose different different object. Ai object gulur maddome e js bujte pare kontar por kontate run korte hobe. Kontake ki vabe run korte hobe. Egulur modda e lukiya ase hoisting, closer, lexical scope aita buja mane js ki vabe kaj korse buja.

1.Global execution context: Amder code jokhon run howa suro hoi prothome ekta loding/creation face diya jai. Prothomai amder code execute hoiya jai na. Shober prothome ekta global execution context object toiri hoi. Tar modda window ... etc thake. Then next step execution.

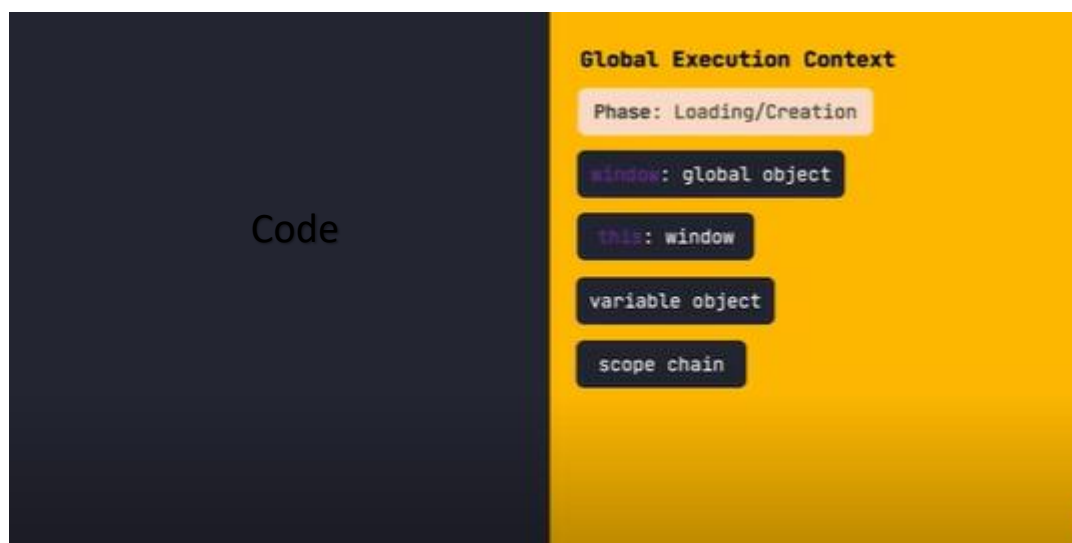


Fig: Loding mode



Fig: Execution mode

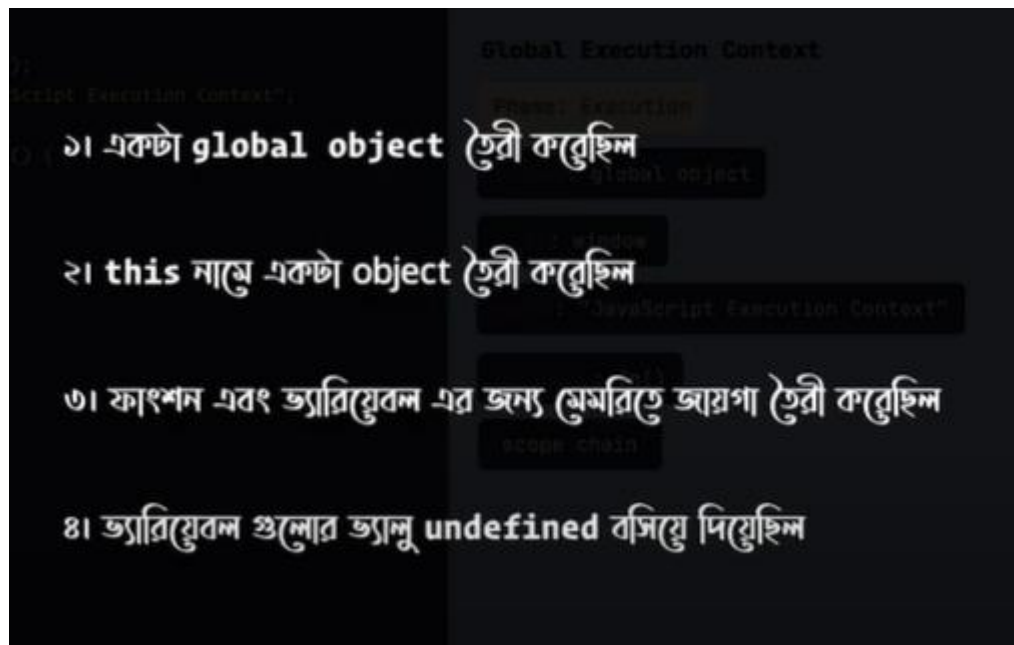


Fig: Global Execution Context

2. Function Execution context: Global Execution context er por function Execution context toiri hoi.

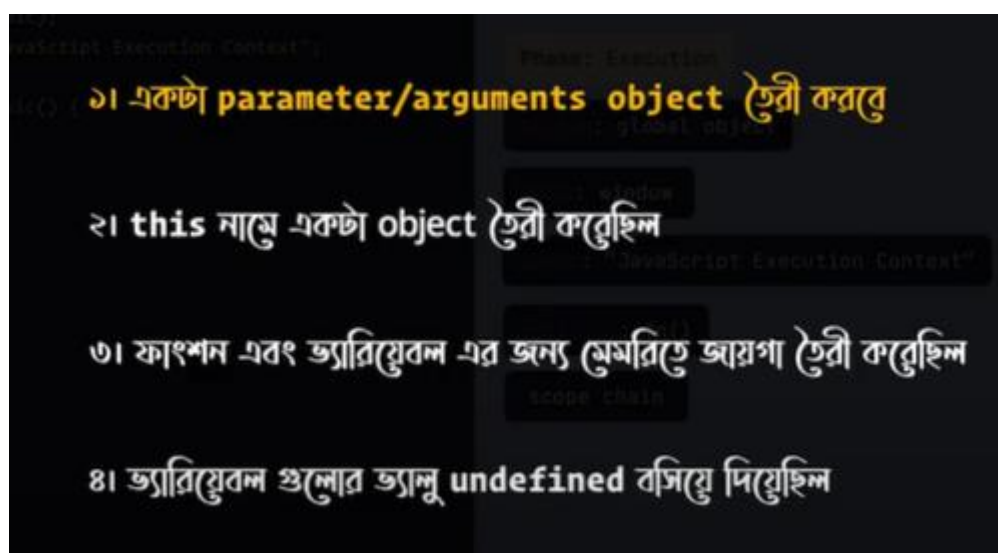


Fig: Function Execution context

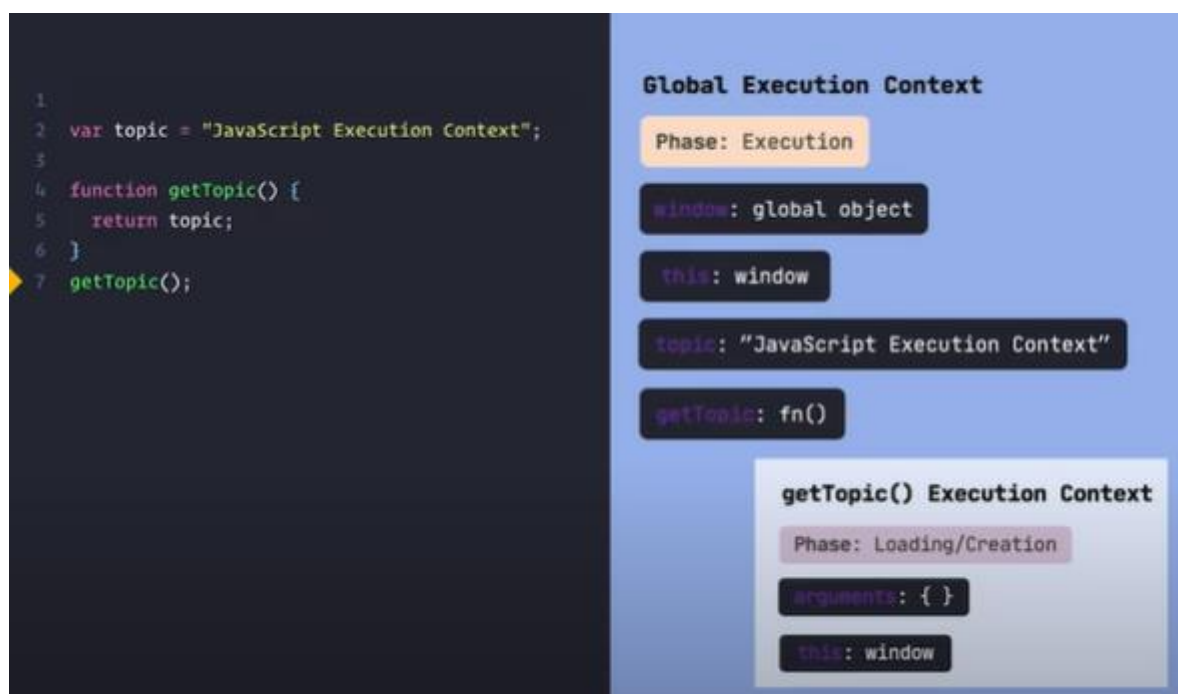
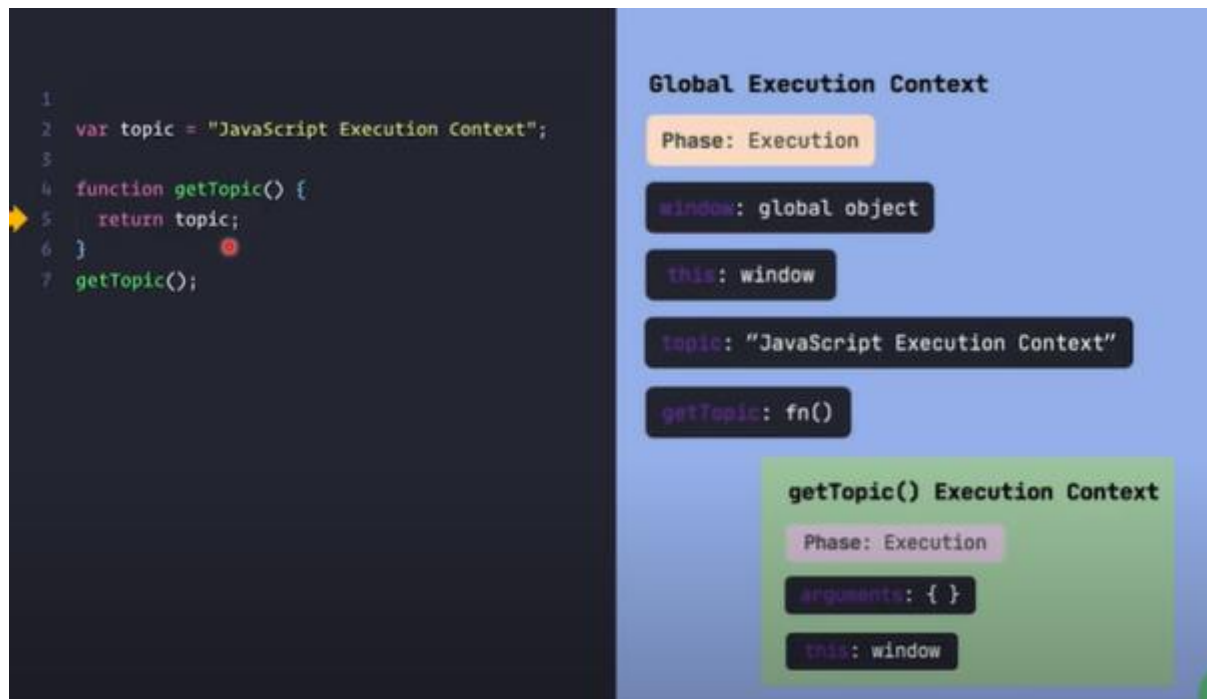


Fig: Function Execution Context loading mode



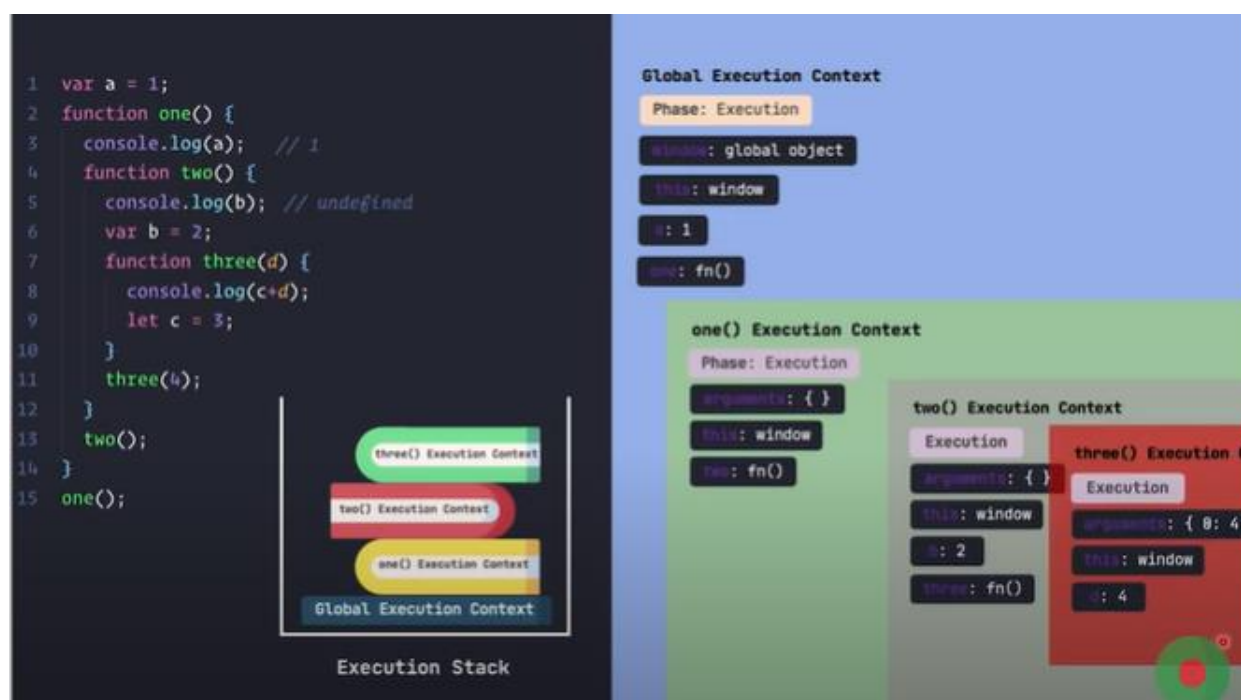
Function Execution Context execution mode

What is the difference between then?

Function Execution context a global object toiri korte hoi na. Function tar parameter er jonno parameter/arguments ekta object toiri kore. Pry same e duita. Ai duitai object.

Execution stack: (Global execution object function Execution object ai je ato object toiri korse so ai gulak to rakte hobe...JS single thread object so take to rakte hobe. take kono na kono data structure flow korte hosse. rakhar jaiga to ekta thread e rakte hobe ektar uppor ekta ba kisu ekta shei ta execution stack/call stack. So 1st global executon rakes then function er upor function er modda Jodi er eka function thake tahole ektar upor er ekta ai vabe rakte thake.)

Stack hosse LFO.



Lexical scope:



Fig: Lexical scope example

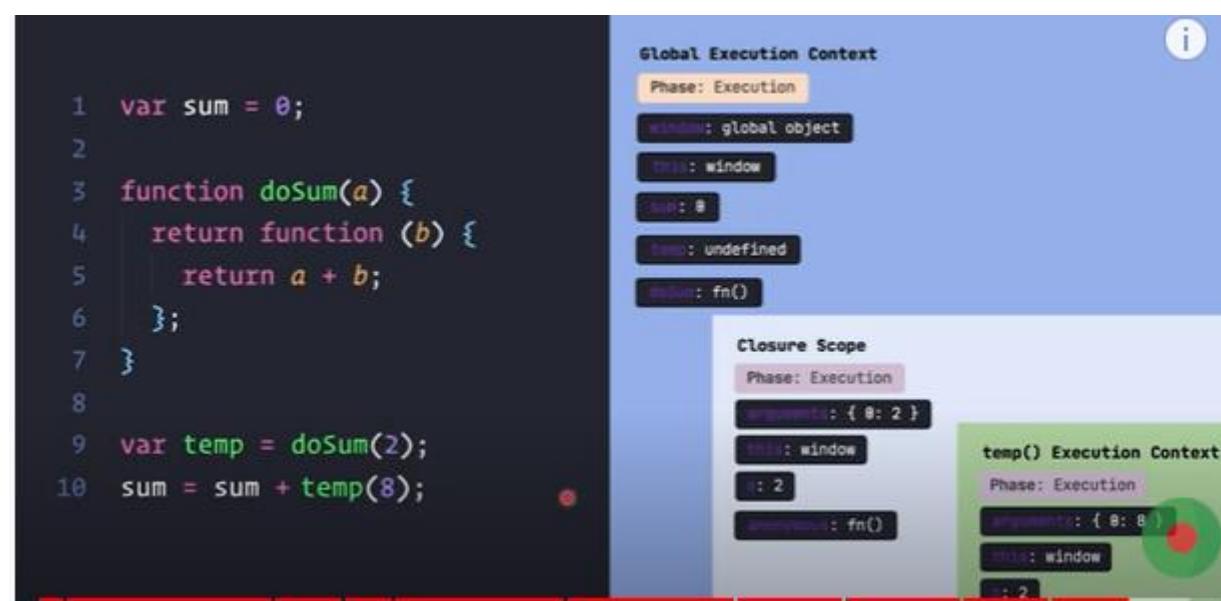
Function call hole notun ekta execution context toiri hoi. Function er kaj sesh hoile function theke bariya ashbey. Console korer age e function pop out hoiya gese jar karone er reference korte parsi na(msg). Context tai nai msg tai refer korte pari na. (ai hose parent child er kahani)

```

function hello() {
  var msg = 'Hello world';
}
hello();
console.log(msg); //reference error output pabo na //aita undefine oboshtai o nai ektom gaieb.

```

Closer:



Jokhon e ekta function er modda r ekta function return hoi/ function er modda function hoi tokhon she Jodi tar outer scope er kono variable use kore tahole take niya closer toiri kore rakhe. Karon she jane aita pore use hote pare. Use korer shomoi jeno refer korte pare.tai closer context a dukiya rakhe.