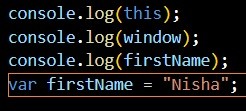


1. Creation phase 2. Code Execution Phase



- Code will be executed line by line

1. **console.log(firstName); // Output: Undefined**
2. **var firstName = "Nisha"; // This line saying print firstName = Nisha**
3. **console.log(firstName); // Nisha**

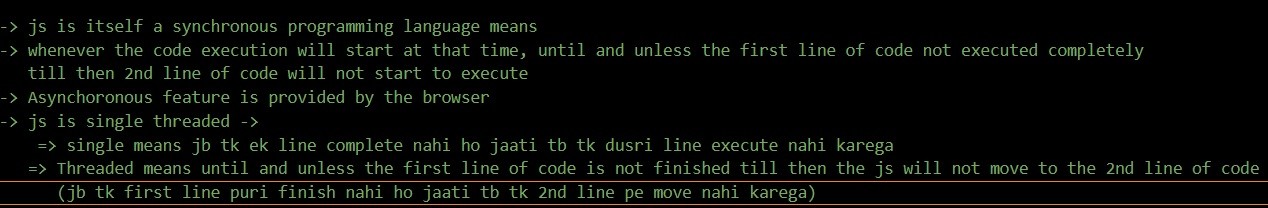
- In above code,

value of this will be equal to window in global memory. window : { } is a global object in js

firstName : undefined in global memory bcz it is a variable & here firstName is in inside global memory means it is in global scope.

- Global memory is created where all the global variable will be stored.

1. **console log this**
2. **console log window**



console log this console.log(window); console.log(myFunction); console.log(firstName);

Global Scope has:

window object will always be there in the Global scope.

variable - firstName, lastName, fullName function - myFunction

this : window

**js store the variable in object means in key-value pair form**

**firstName : undefined lastName : undefined fullName : undefined myFunction : f**

function myFunction(){ console.log("this is my function");

}

var firstName = "Nisha"; var lastName = "Kumari";

var fullName = `${firstName} ${lastName}`; console.log(fullName);

**Code Execution Phase:**

1. **this will print window object**
2. **window object**
3. **print the whole function**
4. **undefined**
5. **myFunction is already present in the global memory**
6. **firstName : Nisha**
7. **lastName : Kumari**
8. **fullName is in global memory**
9. **Nisha Kumari**