

Whole javaScript is run

Execution context created(Global execution context)

Two component of

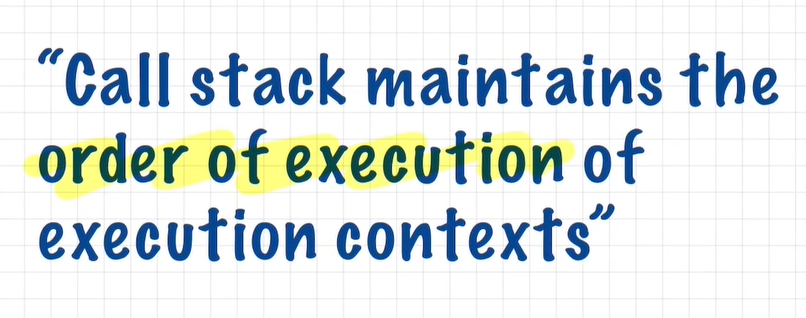
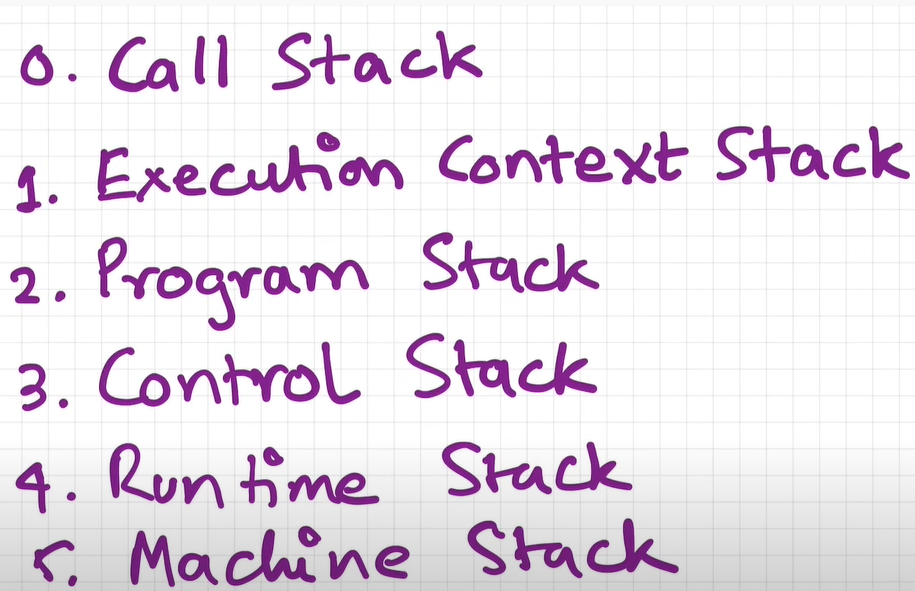
Execution

context

memory component code component

***summary:***

* As soon as the whole javaScript is run => execution context is created (Global execution context is created)
* Execution context has two components:
* First: memory component
* Second: code component
* Two phases by which javaScript code is executed:
* First phase: ***memory creation phase***
* We were allocating memory to all the variables and fuctions inside the global scope/inside the whole program
* We are allocating the memory
* We allocate variable with undefined
* In case of functions , we just copied , literally it just stores the whole function
* The code of whole function inside the memory space
* Second phase: ***code execution phase***
* Noe the js program is executed line by line
* Replace undefined in variable with value
* Invoke the function => execution context is created => whole process repeat ones again
* ***Call stack:(****managing the execution context****)***



* It is a stack
* Everytime => bottom of the stack we have global execution context
* Whenever js program is run => call stack is populated with global Execution context => execution context is pushed inside the stack
* Function invoked => new execution context is created => put inside the stack
* Done with executing the function => pop out of the stack
* Whole thing executed => call stack get empty
* Global execution context => gone
* Done with javaScript program