

Javascript OVERVIEW

- 1.High level
- 2.prototype-based Object Oriented.
- 3.multi-paradigm: 1.Procedural 2.OOP 3.Functional Programming
- 4.interpreted or just in time compiled
- 5.Dynamic: We don't assign any type to the variables.
- 6.Single threaded
- 7.Garbage collected
- 8.It has first-class functions:functions are treated as variables.
- 9.and a non-blocking event loop concurrency model

Event loop

: if a task is running longer then js will execute it in background and then put it back in main thread once it's complete.

JavaScript Engine

: Program that executes javascript code. e.g. V8 engine (used by chrome and Nodejs) Any Javascript contains :

1. Call Stack
2. Heap

Compilation

: Entire code is converted into machine code at once and make a portable file.

Interpretation

: Interpreter runs through the source code and execute it line by line.

Js - Not an interpreted(completely) language. It uses something called JIT.

Just in Time compilation

: Entire code is converted into machine code at once and then executed immediately.

Js Runtime

: The JavaScript Engine analyzes and prepares the code for execution, while the Runtime executes the code and manages the queues that are associated with code execution. Most common Runtimes are /browsers and Node.js(but no web APIs).

Execution Context

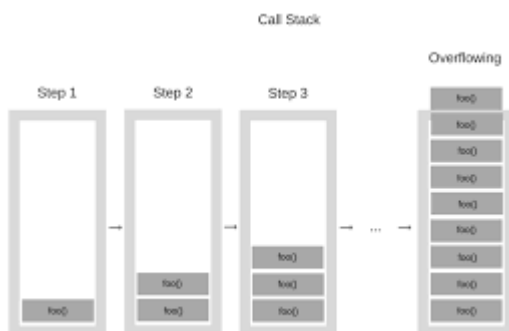
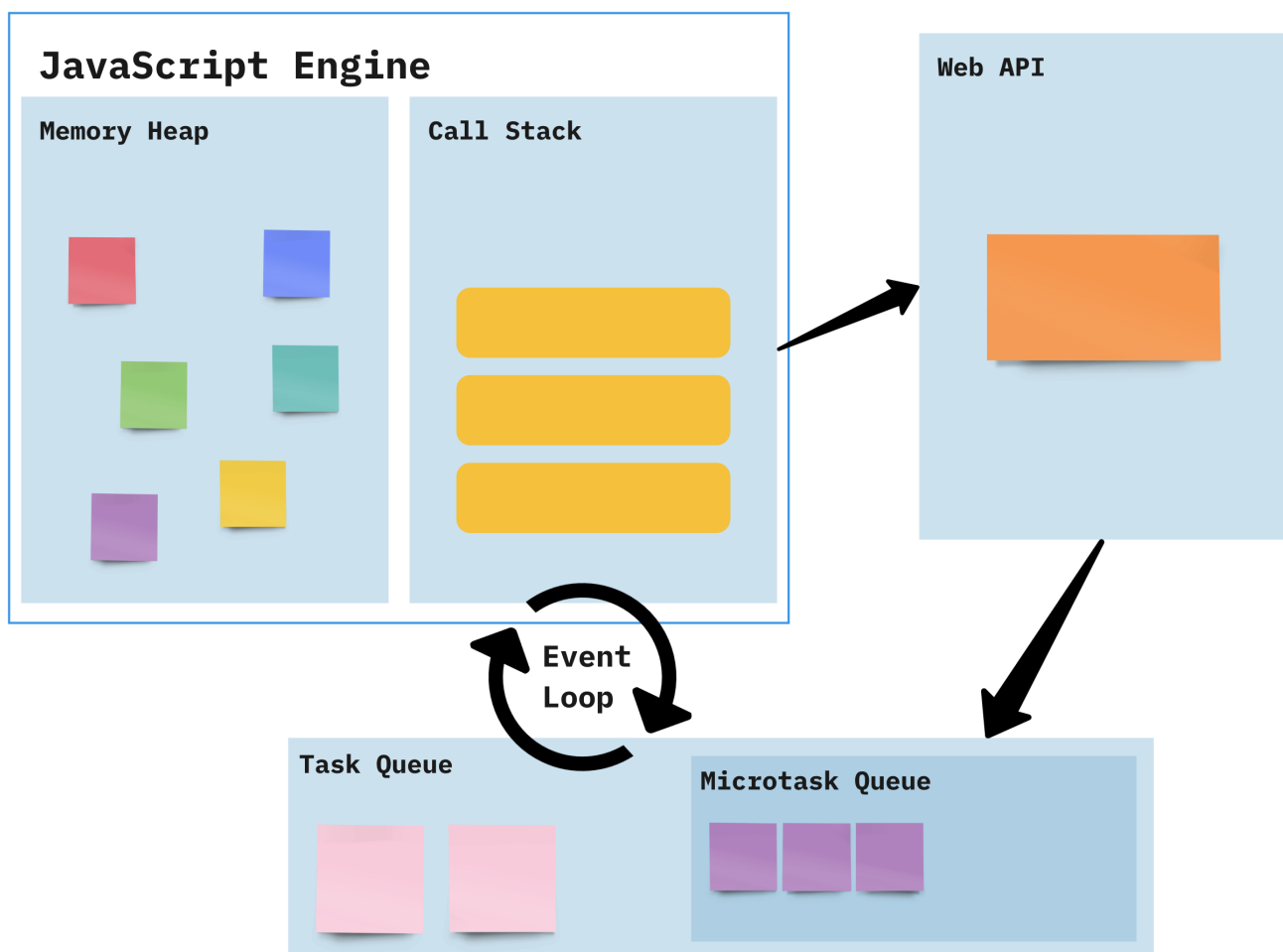
: Environment in which a piece of js is executed. Js code is always executed in an execution context. Only one global execution context : created for code that is not inside any function (top-level).

One execution context per function.

What is inside execution context - 1. Variable Environment All variable declarations , function and arguments object. 2. Scope Chain 3. This keyword Arrow functions do not get access to this keyword and argument object.

![[JS Engine]](https://storage.googleapis.com/algodailyrandomassets/curriculum/frontend/js-engine-runtime/EngineRuntime.png)

JavaScript Runtime Environment

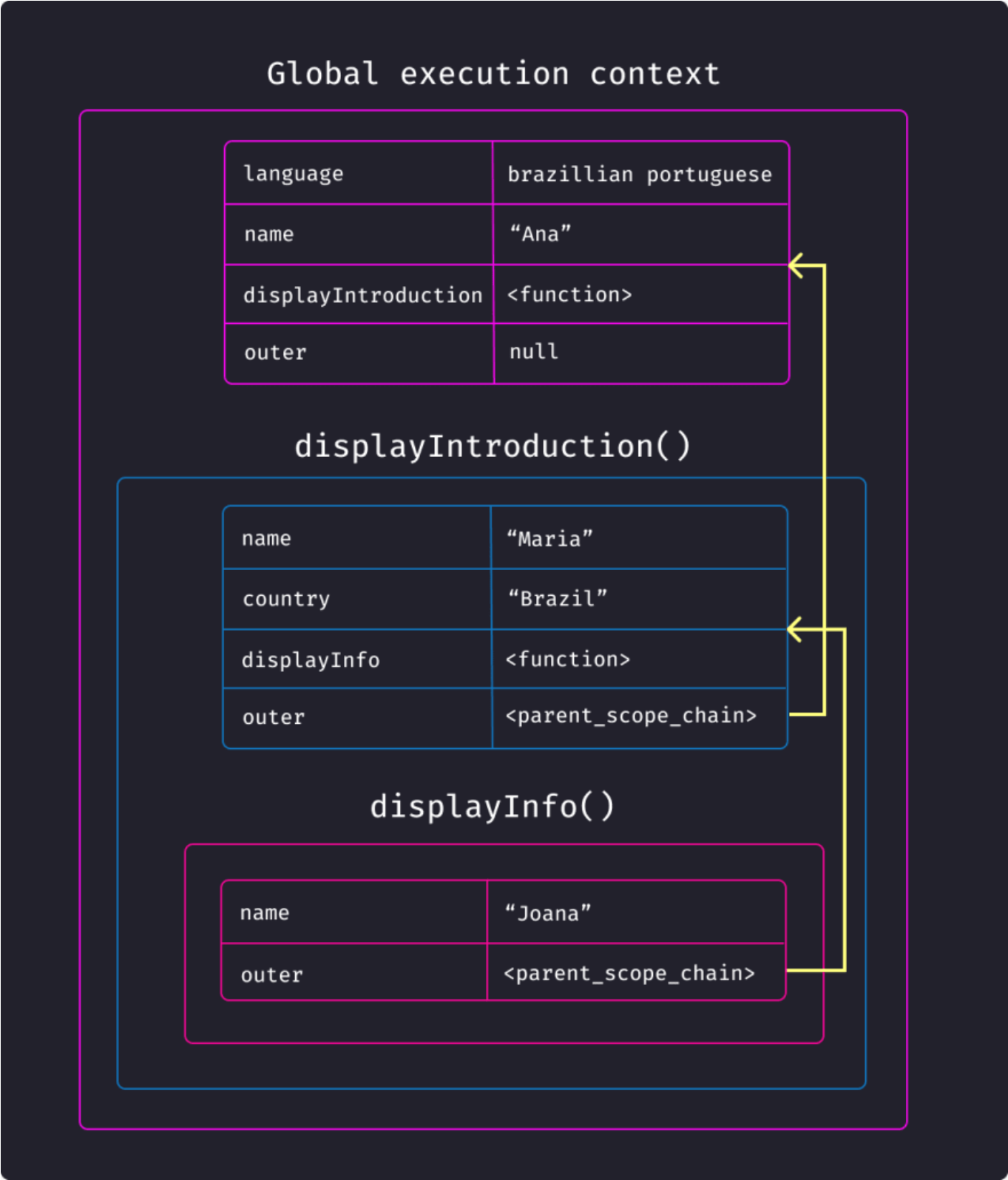


Global execution context only popped when we close the program.

Scoping and Scope Chain

Scope

How our program is organized and accessed. Basically from where we can access our variables. Js uses 'lexical scoping' .



Global Scope

Function Scope

Block Scope (Use let not var and use strict)