JS is a single-threaded, synchronous and loosely bounded programming language

Even **before** code starts **executing**, **memory** is **created** for all the variables, functions in JS.

For **variable** 🡪 ***undefined*** is stored in phase 1.

For **function** 🡪 ***whole function*** is stored in phase 2. ie., like a copy

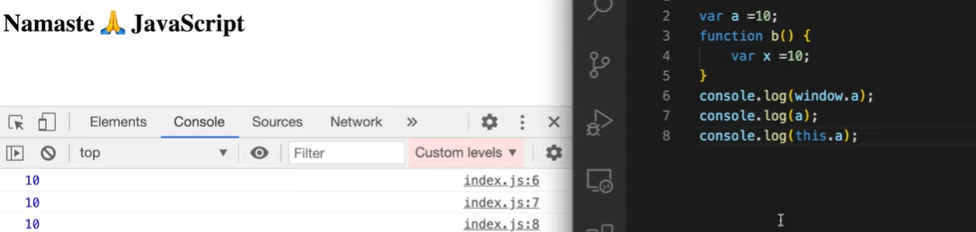
Whenever **Global Execution Context** is **created**, ***window*** is also created

***This***points to **window** object in global scope.

**This** will be **created for every execution context**, even functions

It is responsible for the **JS engines** to **create** a **window** object whenever JS file is being executed.

By default window is created 🡪 creation of any variable inside the file, it is added to that window



Let cannot be redeclared again

Var can be redeclared. So that value is overwritten.

**let name = "Alice";**

**let name = "Bob"; 🡪 error as let is block-scoped**

**console.log(name);**

**var name = "Alice";**

**var name = "Bob";**

**console.log(name); 🡪 Bob**