**Execution Context in JS:**

Khi bạn chạy bất kỳ câu lệnh nào trong JS thì sẽ có một special environment (môi trường đặt biệt) được tạo ra để handle (xử lí) quá trình chuyển đổi (the transformation) và execution of code đó.

Khi thực hiện execution context thì nó sẽ tạo ra 1 memory lưu các biến để lưu dữ liệu của các biến đấy. Rồi sau đó sẽ thực thi những dòng code (line by line) theo từng hàng.

A close-up of a memory

Description automatically generated

Từ đó ta sẽ thấy rằng có 2 giai đoạn:

* Đầu tiên nó sẽ init những biến
* Sau đó sẽ thực thi những hàm, những câu lệnh

**Execution Context sẽ có 2 phase:**

**1/ Memory Creations Phase:**

* Tạo ra global obj (Browser = windown, Node.js = global).
* Sử dụng this obj để tự động gán giá trị vào (cơ chế tự động map những data vào với nhau).
* Setup những memory heap để lưu trữ variables và funtions references.
* Sau khi đã lưu variables và funtions references thì sẽ lưu những cái đó vào global execution context và set chúng là “undefined”.

**2/ Execution Phase**

* Thực hiện code theo từng hàng (line by line)
* Rồi tạo ra một new execution context cho mỗi lần gọi function

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**The Call Stack:**

Stack là 1 cấu trúc dữ liệu (LIFO: last in first out) không phụ thuộc vào bất kỳ ngôn ngữ lập trình nào, mà nó là nguyên lý căn bản của tất cả ngôn ngữ lập trình đều phải tuân thủ theo.

A screenshot of a computer

Description automatically generated

**Thread of Execution:**

Trong JS, thread of execution thường được đề cập đến là thứ tự các biểu thức (Expressions) và mệnh đề (Statements) được thực thi bới JS engine. JS là một single-theared lanuage (*JavaScript engine* runs in a single thread**)**, nghĩa là nó chỉ có thể thực thi 1 task tại 1 thời điểm.

**How does the thread of execution affect the way JavaScript programs run?**

**( Làm cách nào mà thread of execution ảnh hưởng đến JS chạy trương trình ? )**

Thread of execution đóng một vai trò quan trọng trong cách mà JS chạy trương trình. Nó chịu trách nhiệm xác định thứ tự các mệnh đề (Statements) và biểu thức (Expressions) được thực thi bới JS engine.

**What is the call stack and how does it relate to the thread of execution?**

**( Call stack và nó có liên quan gì đến Thread of execution ? )**

Khi một chương trình được thực thi thì JS engine sẽ tạo ra 1 global execution context và push nó vào stack.

