

# CÁCH THỨC HOẠT ĐỘNG CỦA TỪ KHÓA THIS

- 👉 từ khóa/biến `this`: Biến đặc biệt được tạo cho mọi execution context (mọi hàm). Lấy giá trị của (trở đến) “chủ sở hữu” của hàm mà từ khóa `this` được sử dụng.
- 👉 Từ khóa `this` **KHÔNG** tĩnh. Nó phụ thuộc vào **cách** hàm được gọi và giá trị của nó chỉ được gán khi hàm **thực sự** được gọi.

EXECUTION CONTEXT

✓

Môi trường biến

✓

Scope chain

👉

từ khóa `this`

Không sở hữu `this`

Phương thức 👉 `this = <Object gọi phương thức>`

Gọi hàm đơn giản 👉 `this = undefined`

Ở strict mode!  
Nếu không:  
`window` (trong trình duyệt)

Hàm mũi tên 👉 `this = <this của hàm xung quanh (lexical this)>`

Event listener 👉 `this = <thành phần DOM mà trình xử lý gắn vào>`

`new, call, apply, bind` 👉 `<Sau này sẽ học... ⌚>` ⌚

thức

👉 Ví dụ về phương thức:

```
const jonas = {
  name: 'Jonas',
  year: 1989,
  calcAge: function() {
    return 2037 - this.year
  }
};
jonas.calcAge(); // 48
```

calcAge là phương  
jonas  
1989

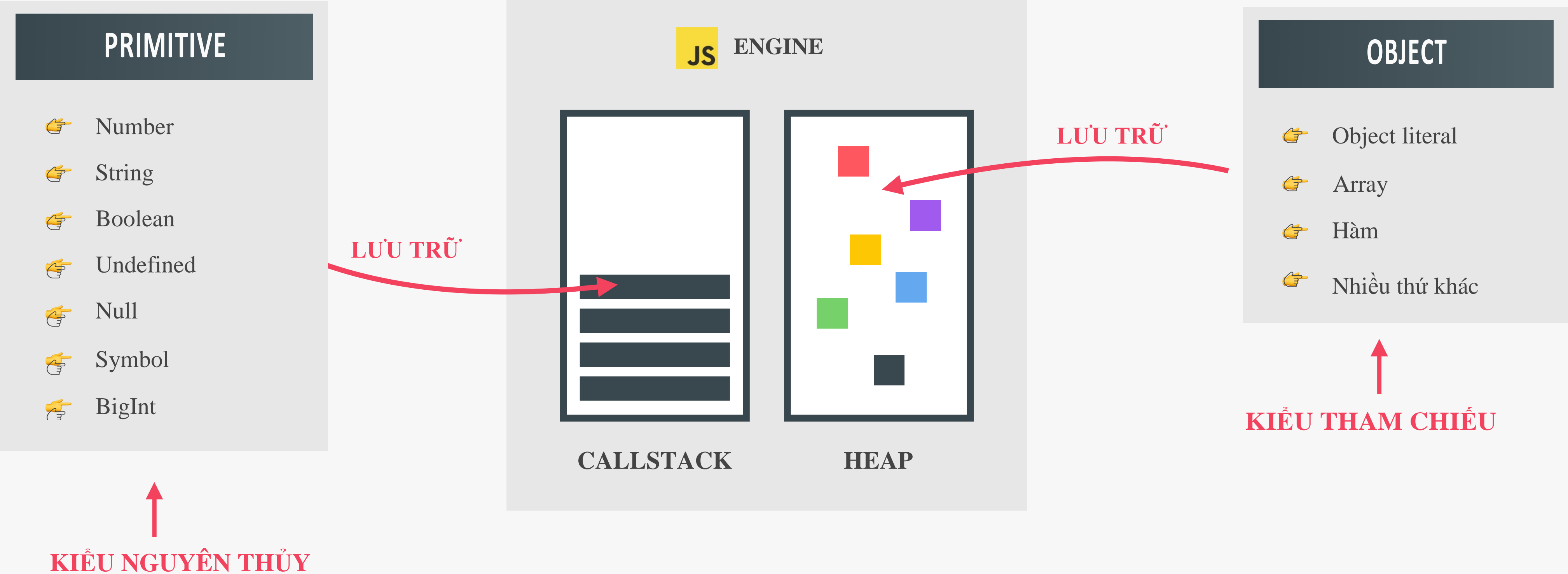


this **KHÔNG** trở vào chính hàm, và cũng **KHÔNG** trở vào môi trường biến của nó!

Tốt hơn nhiều so với dùng

```
jonas.year!
```

# ÔN TẬP: PRIMITIVE, OBJECT VÀ JAVASCRIPT ENGINE



# GIÁ TRỊ NGUYÊN THỦY VỚI GIÁ TRỊ THAM CHIẾU

👉 Ví dụ giá trị nguyên thủy:

```
let age = 30;
let oldAge = age;
age = 31;
console.log(age); // 31
console.log(oldAge); // 30
```

👉 Ví dụ giá trị tham chiếu:

```
const me = {
  name: 'Jonas',
  age: 30
};
const friend = me;
friend.age = 27;

console.log('Friend:', friend);
// { name: 'Jonas', age: 27 }

console.log('Me:', me);
// { name: 'Jonas', age: 27 }
```

Không sao, vì chúng ta KHÔNG thay đổi giá trị ở address 0003!

