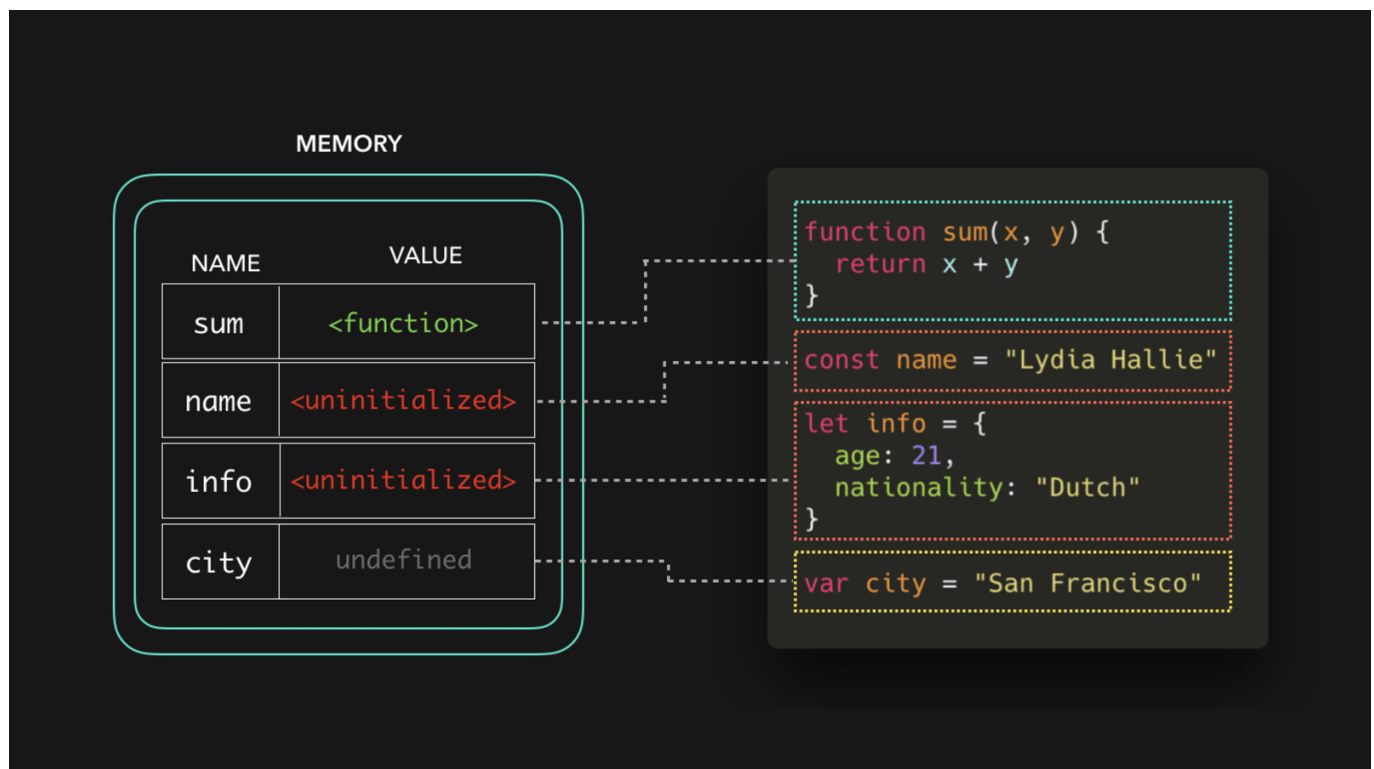


Hoisting in Javascript 🤪



Source <https://dev.to/lydiahallie/javascript-visualized-hoisting-478h>

JavaScript Hoisting refers to the process whereby the compiler allocates memory for **variable** and **function declarations** **prior to** execution of the code. [source](#)

#	Hoisting	Example
var	YES	<code>var count = 1;</code>
const / let	NO	<code>let name = 'Easy';</code>
function declaration	YES	<code>function sayHello() {}</code>
function expression	NO	<code>const sum = function() {}</code>

IMPORTANT: Only declarations are hoisted, not initializations.

```
console.log(num); // Returns 'undefined' from hoisted var declaration
var num; // Declaration
num = 6; // Initialization
```

```
// no var declaration
console.log(num); // ReferenceError: num is not defined
num = 6; // Initialization
```

function declaration hoisting

- Able to use function before the declaration.

```
sum(1, 2); // also works

function sum(a, b) {
  return a + b;
}

sum(1, 2); // works
```

- **BE CAREFUL** with function expression, the **var** is hoisted with **undefined** value, not the function.

```
sum(1, 2); // TypeError: sum is not a function
// because sum is var
// and get hoisted with undefined value

var sum = function (a, b) {
  return a + b;
};

sum(1, 2); // works
```

Hack não tí

```
var tip = 100;

(function () {
  console.log("I have $" + husband());

  function wife() {
    return tip * 2;
  }

  function husband() {
    return wife() / 2;
  }

  var tip = 10;
})();
```

source: <https://github.com/yeungon/In-JavaScript-we-trust#5-whats-the-output>

var, const and let

	var	const	let
initializing	optional	required	optional
re-declare (same scope)	yes	no	no
can be re-assigned	yes	no	yes
scope	function or global	block or global	block or global
hoisting	yes	no	no
when to use	avoid using it	constant, unchanged vars	changable vars
become a member of window in global	yes	no	no

```
var channelName = '';  
var channelName = 'Easy Frontend'; // it works ==)
```

```
const channelName = '';  
const channelName = 'Easy Frontend';  
// Uncaught SyntaxError: Identifier 'channelName' has already been  
declared
```

```
let channelName = '';  
let channelName = 'Easy Frontend';  
// Uncaught SyntaxError: Identifier 'channelName' has already been  
declared
```

```
// function scope vs block scope  
function sayHello() {  
  {  
    const language = 'en'; // block scope  
    var message = 'hello'; // function scope  
  }  
  
  console.log(language); // ReferenceError: language is not defined  
  console.log(message); // 'hello'  
}
```

Summary

- Không dùng **var** nữa --> chỉ dùng **const/let**
- Cần phân biệt được function declaration và function expression
- Hoisting cho function declaration giúp mình có thể gọi hàm được ngay cả trước khi nó được định nghĩa.

Tham khảo

- <https://developer.mozilla.org/en-US/docs/Glossary/Hoisting>
 - <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/const>
 - <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/let>
 - <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/var>
-

Khoá học Javascript cho người mới bắt đầu 2021 🎉

- Tác giả: **Hậu Nguyễn** - Founder Easy Frontend
- Khoá học chỉ được published trên Udemy, không thông qua trung gian.
- Khoá học không bán dạng videos upload trên Google Drive hay bất cứ hình thức nào tương tự.
- Khoá học có nhóm discord để hỗ trợ trong quá trình học tập.

☎️ Liên hệ tác giả để được hỗ trợ:

- ✅ Facebook: <https://www.facebook.com/nvhauesmn/>
- ✅ Fanpage: <https://www.facebook.com/learn.easyfrontend>
- ✅ Youtube Channel: <https://www.youtube.com/easyfrontend>