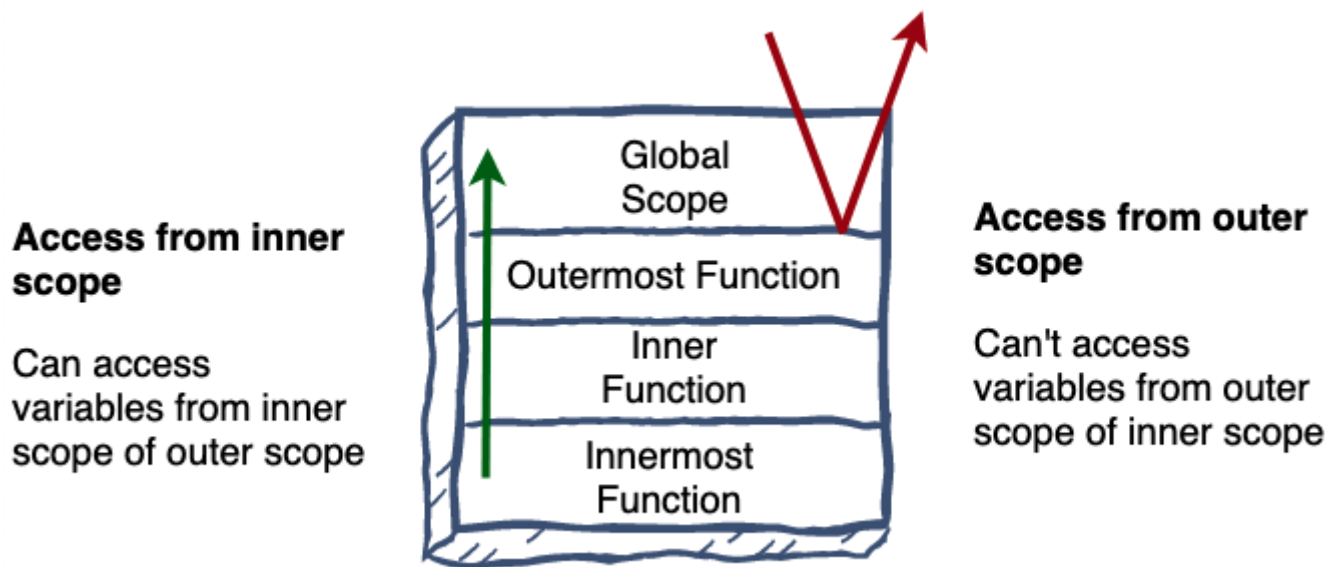


# Scope in Javascript

---



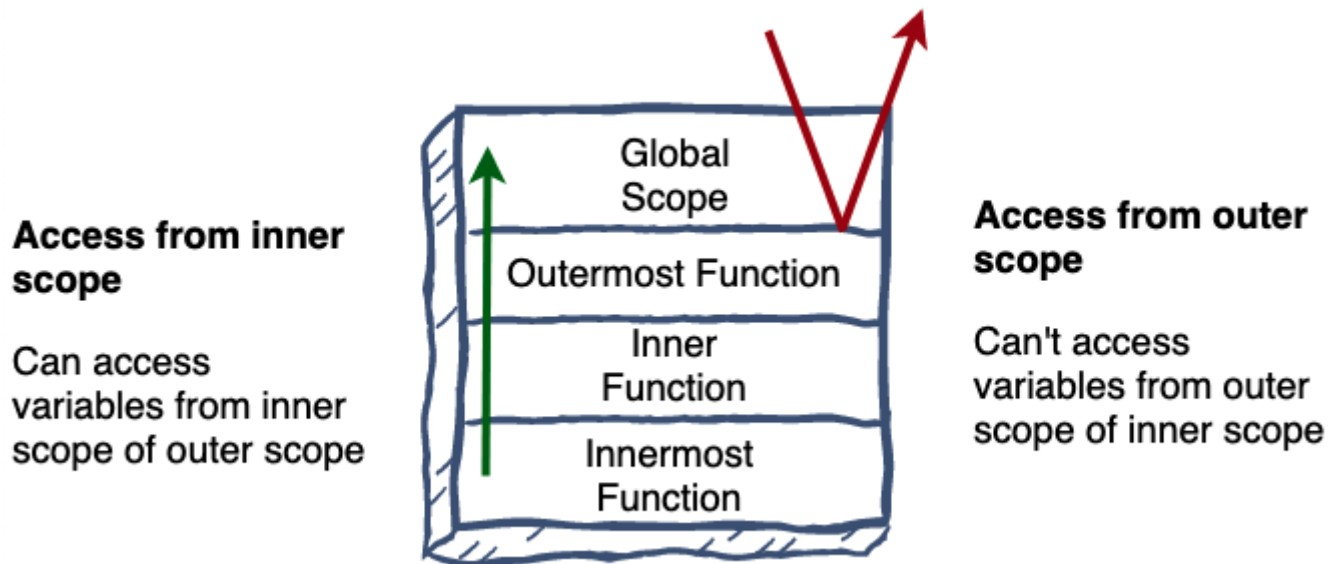
Source: <https://www.educative.io/edpresso/lexical-scope-in-javascript>

## AGENDA

1. Scope overview
2. Block scope
3. Function scope
4. Lexical scope
5. Global scope
6. Scope chain

## 1. Scope overview

- **What:** the limits in which a variable exists
- **Why:** to understand when and where we can access to a variable
- **When:** declare a variable
- **Types of Scope:** block scope, function scope, lexical scope, global scope.



Source: <https://www.educative.io/edpresso/lexical-scope-in-javascript>

## 2. Block scope

- block scope = limit in the pair of curly brackets (braces) {}
- **const/let** has block scope (ES6)

```
{  
  const name = 'Easy Frontend';  
}  
  
console.log(name); // runtime error: name is not defined
```

```
// nested block scope  
{  
  {  
    const name = 'Easy Frontend';  
  }  
  
  console.log(name); // runtime error: name is not defined  
}
```

```
if (n > 10) {  
  const name = 'Easy Frontend';  
}  
  
console.log(name); // runtime error: name is not defined
```

```
for (let i = 0; i < 10; i++) {  
  const name = 'Easy Frontend';  
}  
  
console.log(name); // runtime error: name is not defined
```

```
// how about this case?  
for (let i = 0; i < 10; i++) {  
  if (i % 5 === 0) {  
    const isDivisibleToFive = true;  
  }  
  
  console.log(isDivisibleToFive); // ???  
}  
console.log(isDivisibleToFive); // ???
```

### 3. Function scope

- function scope = limit into current function only.

```
function sayHello() {  
  const name = 'Easy Frontend';  
  console.log(name); // 'Easy Frontend'  
  
  function print() {  
    const language = 'en';  
    console.log(language); // 'en'  
  }  
  
  print();  
  console.log(language);  
  // ReferenceError: language is not defined  
}  
  
sayHello();  
  
console.log(name);  
// ReferenceError: name is not defined
```

### 4. Lexical scope

- Lexical scope = ability for a function scope to access variables from the parent scope. [source](#)
- lexical scope ~ where is this scope defined?
- lexical scope ~ the definition area of an expression
- lexical scope has another name static scope
- lexical scope is the place in which the item got created

```
const name = 'Easy Frontend';  
  
function sayHello() {  
  const language = 'en';  
  console.log(name);  
  // the lexical scope of name is global scope  
  
  function printLanguage() {  
    console.log(language);  
    // the lexical scope of language is a function scope (sayHello)  
  }  
}
```

## 5. Global scope

- global scope can be accessed anywhere
- global scope = declare variable not inside any function

```
let count = 1;
let channelName = 'Easy Frontend'
```

#	Browser	NodeJS	Worker
global object	<a href="#">window</a>	<a href="#">global</a>	<a href="#">WorkerGlobalScope / self</a>

- In JavaScript, there's always a global object defined
- The global object's interface depends on the execution context in which the script is running
- global object is an object that always exists in the global scope. [source](#)
- The global **globalThis** property contains the global **this** value, which is asking to the global object. [source](#)

```
var getGlobal = function () {
  if (typeof self !== 'undefined') { return self; }
  if (typeof window !== 'undefined') { return window; }
  if (typeof global !== 'undefined') { return global; }
  throw new Error('unable to locate global object');
};
```

Source: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/globalThis#search\\_for\\_the\\_global\\_across\\_environments](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/globalThis#search_for_the_global_across_environments)

## 6. Scope chain

A scope chain refers to the unique spaces that exist from the scope where a variable got called to the global scope. [source](#)

```
// First fullName variable defined in the global scope:
const fullName = "Easy";

// Nested functions containing two more fullName variables:
function profile() {
  const fullName = "Frontend";

  function sayName() {
    const fullName = "Boooooom";

    function writeName() {
      return fullName;
    }
    return writeName();
  }

  return sayName();
}
```

## Tham khảo


- <https://www.educative.io/edpresso/lexical-scope-in-javascript>
- <https://www.freecodecamp.org/news/javascript-lexical-scope-tutorial/>

---

## 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>