var const let and variables scopes

By CODEMIND Technology

Contact us 966 5044 698
966 5044 598

var const let and Variable Scopes

- var, let and const
- Variable Scopes:
  - Global scope
  - Local scope
  - Block scope

var const and let

- We can use these reserved words to declare a variable in JS
- Let and const introduced in ES6
- var is available since from JS inception
- Variable declared within a JS function are known as local variable

var let const

Quick overview for

- Variable declaration
- variable initialization
- Modification or update
- Variable re-declaration

Variable declaration

```
var score;
let college;
```

Variable decl.& init.

```
var score = 85;
let college = "COEP";
const PIN = 431202;
```

Variable Modification or update

```
score = 90;         --> Allowed
college = "REP";    --> Allowed
PIN = 431202;       --> Not allowed
```

Variable Re-declaration

```
var score = 90;         --> Allowed
let college = "REP";    --> Not Allowed
const PIN = 431202;     --> Not allowed
```

var and let

We can use var and let keywords in order to define variables.

- ‘let’ is introduced in ES6
- ‘let’ has block scope
- For ‘let ’variable re-declaration is not allowed
- For var ’variable re-declaration is allowed
- Modification or updation is allowed

const

- const keyword used to define constants
- It cannot be declared w/t initialization.
- Ex. const PI = 3.14;
- ‘const’ has the block scope
- For ‘const’ re-declaration is not allowed
- Modification is not allowed

What is block ?

Anything written inside open and close curly brackets { } is called as block

Example:

- if block
- else block
- for loop block

Variable Scopes.

Scope in JS determines the accessibility where a variable can be used.

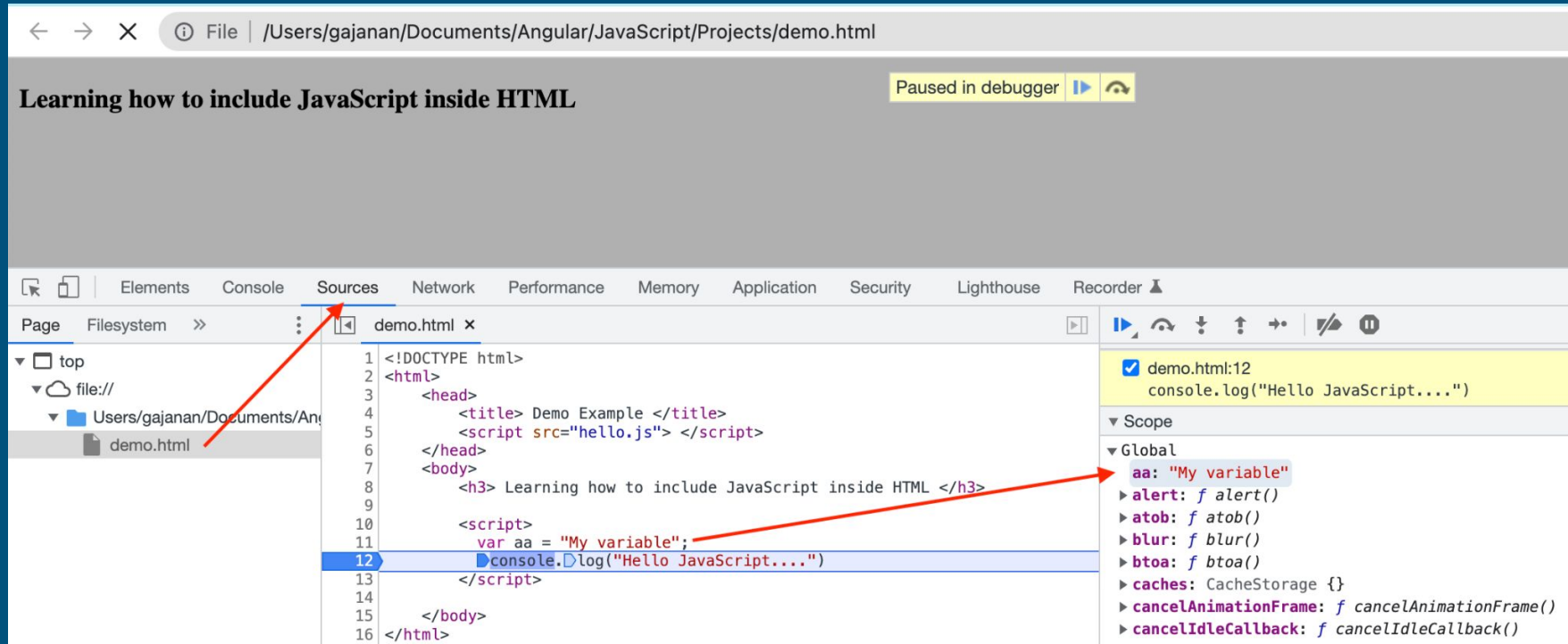
There are three types of scopes in JS

- Global Scope
- Function Scope
- Block Scope

Global Scope

Global scope means a variable declared outside any curly brackets { } that variable has a global scope and can be used anywhere in the file

Example. → var aa =”My Variable”;



Function Scope

Function scope means a variable defined inside a function are not accessible from outside the function

→ ‘message’ variable defined inside function display() and It has the function scope hence, It is not allowed to access outside of function.

In case, will try to access will get the error : ReferenceError: message is not defined

Note: Instead of ‘var’ keyword in case will use ‘let’ to define message variable (line no. 14) Will get the same error message.

```
13  function display() {
14      var message = "Hello";
15  }
16  console.log(message); // 'message' variable not allowed to access outside function
17  display();
```

Block Scope

- Block scope means declaring a variable inside the block or curly brackets can not be accessible from outside the block.
- Variable declared using let and const has the block scope

```
if(true) {
    let message = "Hello";
    const PI = 3.1413;
}
console.log(message); // Not allowed as variable 'message' and 'PI' defined
                      // using let and const keyword hence it has block scope

if(true) {
    var greet = "Good Morning";
}
console.log(greet); // Allowed as variable 'greet' defined
                    // using var keyword hence it has function scope
```

Block Scope

ES6 provides the let and const keywords that allow you to declare variables in block scope.

It can be the area within the if, else,

switch conditions or

for, do while, and while loops.

```
function say(message) {
    if(!message) {
        let greeting = 'Hello'; // block scope
        console.log(greeting);
    }
    // say it again ?
    console.log(greeting); // ReferenceError
}

say();
```

Diff. between

var, let, const