

Variable Scopes

Friday, March 11, 2022 4:33 PM

ES 5 +

- JavaScript variables have two different scopes in three syntax

1. Explicit Global

- Any variable declared with 'var' in the global scope outside of any function
- It is visible and accessible throughout the application

2. Implicit Global

- A variable created (or used for the first time) without declaring it with 'var'
- They are visible throughout the application

3. explicit local

- These variables are declared inside some block (eg. Function, loop)
- They are visible in the function that is declared.
- They are declared using 'var' inside some function or inner block of the function.

```
var gobj="global";

function f1(){
  console.log("in f1()");
  console.log('gobj',gobj);
  var f1Local="f1Local";
  console.log('f1Local',f1Local);
  f1Local+="modified by f1";
  gobj+="modified by f1";
}

function f2(){
  console.log("in f2()");
  console.log('gobj',gobj);
  gobj+="modified by f2";
  f2Local="f2Local";
  console.log('f2Local',f2Local);
}

function f3(){
  console.log("in f3()");
  console.log('gobj',gobj);
  console.log('f2Local',f2Local);
  console.log('f1Local',f1Local);
}

f1();
f2();
f3();
```

How can f3() access a local variable from f

- f2Local is not really a local variable.
- It is an implicit global variable
 - A variable declared anywhere without using 'var' is considered implicit

```
in f3()
gobj global-modified by f2
f2Local f2Local
d:\MyWorks\Corporate\202203-infogain-react\basic-html-css-js\basic-js\demo15-scopes.js:29
  console.log('f1Local',f1Local);
  ^
ReferenceError: f1Local is not defined
```

- Function f3() can't access f1Local which is variable in f1() and can't be accessed in f3().
 - This error is logical and acceptable

Beaware of Implicit local variable

- They are almost always a potential source of confusion.

- You may not realize that two unrelated variables may end up being same

Main Problem

- We are using same 'i' variable in both function
 - Implicit global
 - Missing 'var'

```

4
3
4 function work(id, times){
5   for(i=0;i<times;i++){
6     console.log('worker #' + id + " working " + i);
7   }
8 }
9
10
11 Step 1.
12   i=0;
13   function schedule(workerCount, workCount)
14   {
15     for(i=0; i<workCount; i++){
16       work(i, workCount)
17     }
18   }
19
20 schedule(10, 4);

```

Step 2:

- "i" increases from 0 to 4 in work loop

Step 3.

- Schedule loop increases "i" to 5

Step 1

- Calls work again with i=5

Step 2.

- Again counts from 0-4

Step 3

- Schedule again increments i to 5

```

worker #0 working 0
worker #0 working 1
worker #0 working 2
worker #0 working 3
worker #5 working 0
worker #5 working 1
worker #5 working 2
worker #5 working 3
worker #5 working 0
worker #5 working 1
worker #5 working 2
worker #5 working 3
worker #5 working 0
worker #5 working 1
worker #5 working 2
worker #5 working 3
worker #5 working 0
worker #5 working 1
worker #5 working 2
worker #5 working 3

```

Solution #1 NEVER USE IMPLICIT GLOBAL VARIABLES

- Always use the var keyword