Not quite what you are looking for? You may want to try:

- Create a WCF DataService in Visual Studio 2015
- Story of Equality in .Net - Part 6

highlights off

12,514,752 members (61,394 online) 1 Devendra Katuke 356 Sign out

CODE PROJECT For those who code

articles quick answers discussions community help

Articles » Web Development » Client side scripting » Beginners

Javascript Closure

Md Wahiduzzaman khan, 28 Oct 2014 CPOL

4.91 (9 votes)

Rate:

When an inner function is inside in an outer function that is called closure.

Is your email address OK? You are signed up for our newsletters but your email address is either unconfirmed, or has not been reconfirmed in a long time. Please **click here to have a confirmation email sent** so we can confirm your email address and start sending you newsletters again. Alternatively, you can **update your subscriptions**.

Introduction

Javascript is a fantastic programming language. It has some fantastic feature and probabely one of the importent and best feature is closure.

The technical defination of closure is something like this: "When an inner function is inside in an outer function "

Then inner function has:

1. Access to the variable of the outer function.
2. Access to the argument of the outer function.

Sounds confusing ??? Lets try some code:

Hide Copy Code

```
//
// Any source code blocks look like this
var outerfunc=function(oname)
{
  return function(iname)
  {
    console.log(oname+" "+iname);
  };
};
var one=outerfunc("Wahid");
var two=outerfunc("Faizah");
one("Khan");
two("Malisha");
// output:
//Wahid Khan
//Faizah Malisha
```

So here in the above code we can clearly see there are two functions one function is inside in the another function So its a closure no doubt. The outer function name is "outerfunc" it has one argument and its return type is also a function. Actually in javascript function is a object. Sounds crazy but its true. So we can say outer function "outerfunc " have a return type of a function/object. The second function or we can say the inner function also have one argument and a console log statement.

Outside the outer function we created two functions or objects of type outerfunc with the parameter "Wahid" and "Faizah ". These two functions actually invoked the outer function with "oname" parameter. Next two line one("khan") and two("Malisha") invoked the inner function and produces the output.

In almost every other language when we reach "return" statement all the variables declare in the function will be destroyed and memory is freed up.

But in javascript when we create a closure and return from that function all the local variable will be remain in the memory and be accessible. Its a very powerful tool.

Hide Copy Code

```
//
// Any source code blocks look like this
var counter=function(incnt)
{
  var initcnt=incnt;
  return function(val){
  initcnt=initcnt+val;
    console.log(initcnt);
  };
};
var x=counter(4);
x(10);
x(20);
x(30);
//14
//34
//64
```

In the above code outer fuction will be called once when we instanciate it with :

Hide Copy Code

```
var x=counter(4);
```

Variable "initcnt" initilized only once when the above line execute. So when these lines executes:

Hide Copy Code

```
x(10);
x(20);
x(30);
```

It always invoke the inner function. So when the **x(10)** execute it got the variable initcnt value 14(4+10) when **x(20)** execute it got the **initcnt** value 34(14+20) and when **x(30)** execute it got the **initcnt** value 64(34+30). So we saw that variable **initcnt** initilize only once.

So this is actually the closure concept.

License

This article, along with any associated source code and files, is licensed under The Code Project Open License (CPOL)

Share

EMAIL TWITTER

About the Author

Md Wahiduzzaman khan

Software Developer IPvision Canada Inc
Bangladesh

No Biography provided

Comments and Discussions

Add a Comment or Question

Search Comments Go

Profile popups Spacing Relaxed Layout Open All Per page 25 Update

First Prev Next

"all the local variable will be remain in the memory and be accessible"

Mehuge 30-Oct-14 4:57

This is not strictly correct. Only those variables referenced by the inner function(s) are held in memory (in the closure). You will notice this when you use a debugger, and breakpoint inside the inner function.

For example, in the closure created below, age, toes and fingers do not exist in the closure because they are not referenced.

Hide Copy Code

```
function person(name, age) {
    var height = 66, weight = 130, shoesize = 11;
    var toes = 10, fingers = 10;
    var bmi;
    var bodymass = function() {
        bmi = (weight/(height*height))*703;
        debugger;
    };
    return function(testname) {
        console.log(testname);
        console.log(name);
        bodymass();
        console.log(bmi);
        console.log(shoesize);
        debugger;
    }
}
var P = person('bob',23);
P('test 1');
```

Reply · Email · View Thread · Permalink · Bookmark

My vote of 5

Preeti Burad 30-Oct-14 3:06

Nice explanation...

Reply · Email · View Thread · Permalink · Bookmark

My Vote 5

Shemeemsha (ഷമീംഷ) 29-Oct-14 4:42

Nice Tip

Reply · Email · View Thread · Permalink · Bookmark

nice — **saxenaabhi6** — **28-Oct-14 19:10**

best example of closure so far.

Reply · View Thread · Permalink · Bookmark

Last Visit: 3-Oct-16 6:00 Last Update: 3-Oct-16 8:29 Refresh 1

General News Suggestion Question Bug Answer Joke Praise Rant Admin

Permalink | Advertise | Privacy | Terms of Use | Mobile
Web01 | 2.8.160929.1 | Last Updated 28 Oct 2014

Select Language

Layout: fixed | fluid

Article Copyright 2014 by Md Wahiduzzaman khan
Everything else Copyright © CodeProject, 1999-2016