Understand JavaScript Closures With Ease

feb. 2 2013 [251](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comments)

Closures allow JavaScript programmers to write better code. Creative, expressive, and concise. We frequently use closures in JavaScript, and, no matter your JavaScript experience, you will undoubtedly encounter them time and again. Sure, closures might appear complex and beyond your scope, but after you read this article, closures will be much more easily understood and thus more appealing for your everyday JavaScript programming tasks.

[[](https://bovacademy.com/)](https://bovacademy.com/" \t "_blank)

[Become an Elite/Highly Paid Specialized Software Engineer (Frontend, Fullstack, etc.)](https://bovacademy.com/" \t "_blank)

[Within 8–10 Months, Earn MORE than the Avg. New CS Graduate Earns.](https://bovacademy.com/" \t "_blank)

**[New Prices (Monthly Subscription):](https://bovacademy.com/" \t "_blank)** [Basic: $250/month, Pro: $350/month, Premium: $600/month](https://bovacademy.com/" \t "_blank)

[By the founder of JavaScriptIsSexy](https://bovacademy.com/" \t "_blank)

This is a relatively short (and sweet) post on the details of closures in JavaScript. You should be familiar with [JavaScript variable scope](http://javascriptissexy.com/javascript-variable-scope-and-hoisting-explained/) before you read further, because to understand closures you must understand JavaScript’s variable scope.

* Receive Updates

Top of Form



Bottom of Form

**What is a closure?**  
A closure is an inner function that has access to the outer (enclosing) function’s variables—scope chain. The closure has three scope chains: it has access to its own scope (variables defined between its curly brackets), it has access to the outer function’s variables, and it has access to the global variables.

The inner function has access not only to the outer function’s variables, but also to the outer function’s parameters. Note that the inner function cannot call the outer function’s *arguments*object, however, even though it can call the outer function’s parameters directly.

You create a closure by adding a function inside another function.  
**A Basic Example of Closures in JavaScript:**

|  |  |
| --- | --- |
|  | function showName (firstName, lastName) { |
|  | ​var nameIntro = "Your name is "; |
|  | // this inner function has access to the outer function's variables, including the parameter​ |
|  | ​function makeFullName () { |
|  | ​return nameIntro + firstName + " " + lastName; |
|  | } |
|  | ​ |
|  | ​return makeFullName (); |
|  | } |
|  | ​ |
|  | showName ("Michael", "Jackson"); // Your name is Michael Jackson |

Closures are used extensively in Node.js; they are workhorses in Node.js’ asynchronous, non-blocking architecture. Closures are also frequently used in jQuery and just about every piece of JavaScript code you read.  
**A Classic jQuery Example of Closures:**

|  |  |
| --- | --- |
|  | $(function() { |
|  | ​ |
|  | ​var selections = []; |
|  | $(".niners").click(function() { // this closure has access to the selections variable​ |
|  | selections.push (this.prop("name")); // update the selections variable in the outer function's scope​ |
|  | }); |
|  | ​ |
|  | }); |

**Closures’ Rules and Side Effects**

1. **Closures have access to the outer function’s variable even after the outer function returns:**  
   One of the most important and ticklish features with closures is that the inner function still has access to the outer function’s variables even after the outer function has returned. Yep, you read that correctly. When functions in JavaScript execute, they use the same scope chain that was in effect when they were created. This means that even after the outer function has returned, the inner function still has access to the outer function’s variables. Therefore, you can call the inner function later in your program. This example demonstrates:

|  |  |
| --- | --- |
|  | function celebrityName (firstName) { |
|  | var nameIntro = "This celebrity is "; |
|  | // this inner function has access to the outer function's variables, including the parameter​ |
|  | function lastName (theLastName) { |
|  | return nameIntro + firstName + " " + theLastName; |
|  | } |
|  | return lastName; |
|  | } |
|  | ​ |
|  | ​var mjName = celebrityName ("Michael"); // At this juncture, the celebrityName outer function has returned.​ |
|  | ​ |
|  | ​// The closure (lastName) is called here after the outer function has returned above​ |
|  | ​// Yet, the closure still has access to the outer function's variables and parameter​ |
|  | mjName ("Jackson"); // This celebrity is Michael Jackson |

1. **Closures store references to the outer function’s variables**; they do not store the actual value.  Closures get more interesting when the value of the outer function’s variable changes before the closure is called. And this powerful feature can be harnessed in creative ways, such as this private variables example first demonstrated by Douglas Crockford:

|  |  |
| --- | --- |
|  | function celebrityID () { |
|  | var celebrityID = 999; |
|  | // We are returning an object with some inner functions​ |
|  | // All the inner functions have access to the outer function's variables​ |
|  | return { |
|  | getID: function () { |
|  | // This inner function will return the UPDATED celebrityID variable​ |
|  | // It will return the current value of celebrityID, even after the changeTheID function changes it​ |
|  | return celebrityID; |
|  | }, |
|  | setID: function (theNewID) { |
|  | // This inner function will change the outer function's variable anytime​ |
|  | celebrityID = theNewID; |
|  | } |
|  | } |
|  | ​ |
|  | } |
|  | ​ |
|  | ​var mjID = celebrityID (); // At this juncture, the celebrityID outer function has returned.​ |
|  | mjID.getID(); // 999​ |
|  | mjID.setID(567); // Changes the outer function's variable​ |
|  | mjID.getID(); // 567: It returns the updated celebrityId variable |

1. **Closures Gone Awry**  
    Because closures have access to the updated values of the outer function’s variables, they can also lead to bugs when the outer function’s variable changes with a for loop. Thus:

|  |  |
| --- | --- |
|  | // This example is explained in detail below (just after this code box).​ |
|  | ​function celebrityIDCreator (theCelebrities) { |
|  | var i; |
|  | var uniqueID = 100; |
|  | for (i = 0; i < theCelebrities.length; i++) { |
|  | theCelebrities[i]["id"] = function () { |
|  | return uniqueID + i; |
|  | } |
|  | } |
|  |  |
|  | return theCelebrities; |
|  | } |
|  | ​ |
|  | ​var actionCelebs = [{name:"Stallone", id:0}, {name:"Cruise", id:0}, {name:"Willis", id:0}]; |
|  | ​ |
|  | ​var createIdForActionCelebs = celebrityIDCreator (actionCelebs); |
|  | ​ |
|  | ​var stalloneID = createIdForActionCelebs [0];  console.log(stalloneID.id()); // 103 |

1. In the preceding example, by the time the anonymous functions are called, the value of i is 3 (the length of the array and then it increments). The number 3 was added to the uniqueID to create 103 for ALL the celebritiesID. So every position in the returned array get id = 103, instead of the intended 100, 101, 102.
2. The reason this happened was because, as we have discussed in the previous example, the closure (the anonymous function in this example) has access to the outer function’s variables by reference, not by value. So just as the previous example showed that we can access the updated variable with the closure, this example similarly accessed the i variable when it was changed, since the outer function runs the entire for loop and returns the last value of i, which is 103.
3. To fix this side effect (bug) in closures, you can use an **Immediately Invoked Function Expression** (IIFE), such as the following:

|  |  |
| --- | --- |
|  | function celebrityIDCreator (theCelebrities) { |
|  | var i; |
|  | var uniqueID = 100; |
|  | for (i = 0; i < theCelebrities.length; i++) { |
|  | theCelebrities[i]["id"] = function (j) { // the j parametric variable is the i passed in on invocation of this IIFE​ |
|  | return function () { |
|  | return uniqueID + j; // each iteration of the for loop passes the current value of i into this IIFE and it saves the correct value to the array​ |
|  | } () // BY adding () at the end of this function, we are executing it immediately and returning just the value of uniqueID + j, instead of returning a function.​ |
|  | } (i); // immediately invoke the function passing the i variable as a parameter​ |
|  | } |
|  | ​ |
|  | return theCelebrities; |
|  | } |
|  | ​ |
|  | ​var actionCelebs = [{name:"Stallone", id:0}, {name:"Cruise", id:0}, {name:"Willis", id:0}]; |
|  | ​ |
|  | ​var createIdForActionCelebs = celebrityIDCreator (actionCelebs); |
|  | ​ |
|  | ​var stalloneID = createIdForActionCelebs [0]; |
|  | console.log(stalloneID.id); // 100​ |
|  | ​ |
|  | ​var cruiseID = createIdForActionCelebs [1]; console.log(cruiseID.id); // 101 |

[[](https://bovacademy.com/)](https://bovacademy.com/" \t "_blank)

[Become an Elite/Highly Paid Specialized Software Engineer (Frontend, Fullstack, etc.)](https://bovacademy.com/" \t "_blank)

[Within 8–10 Months, Earn MORE than the Avg. New CS Graduate Earns.](https://bovacademy.com/" \t "_blank)

**[New Prices (Monthly Subscription):](https://bovacademy.com/" \t "_blank)** [Basic: $250/month, Pro: $350/month, Premium: $600/month](https://bovacademy.com/" \t "_blank)

[By the founder of JavaScriptIsSexy](https://bovacademy.com/" \t "_blank)

Posted in: [16 Important JavaScript Concepts](http://javascriptissexy.com/category/16-important-javascript-concepts/), [JavaScript](http://javascriptissexy.com/category/javascript/) / Tagged: [Closures](http://javascriptissexy.com/tag/closures/), [JavaScript](http://javascriptissexy.com/tag/javascript/), [Learn JavaScript](http://javascriptissexy.com/tag/learn-javascript/)

[Richard](http://javascriptissexy.com/author/richardb/)Thanks for your time; please come back soon. Email me here: javascriptissexy at gmail email, or use the [contact](http://javascriptissexy.com/contact/) form.

251 Comments

1. http://2.gravatar.com/avatar/e7b2db5863294dd5b9cfc0c15504cbd7?s=60&d=mm&r=g

Nimesh

[February 27, 2013 at 4:12 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-3155) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=3155#respond)

Thanks for the great post. I am bookmarking your website. A sexy collection of javascript tutorials.

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[February 27, 2013 at 11:11 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-3166) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=3166#respond)

Thanks much, Nimesh. I am happy to hear the website is helpful and…sexy, too. 

* + - http://1.gravatar.com/avatar/7e3c06237f516617523bbc50e299aca0?s=60&d=mm&r=g

vedran

[August 12, 2013 at 10:33 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-27775) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=27775#respond)

Very good site !!!!

* + - http://1.gravatar.com/avatar/dac79902c081b007ff2840a5225e76f1?s=60&d=mm&r=g

DeVontae Moore

[October 17, 2014 at 8:41 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-305772) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=305772#respond)

theCelebrities[i][“id”] = function (j) {

Im not familiar with the above syntax. Could you please explain?

* + - * http://1.gravatar.com/avatar/1a25442d662669d0c846fe2aaa0b8a50?s=60&d=mm&r=g

MS

[November 7, 2014 at 8:17 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-305854) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=305854#respond)

This statement assigns a new value to a variable inside an array of objects.

Recall that theCelebrities now has the value of [{name:”Stallone”, id:0}, {name:”Cruise”, id:0}, {name:”Willis”, id:0}]  
The first set of square brackets, theCelebrities[i], calls an object by its array position. If i = 0 then theCelebreties[i] is equal to the first object in the array, in this case, {name:”Stallone”, id:0}.  
The second set of brackets, [“id”], call that object’s property by its name “id” .  
The rest of the statement “= function(j){….}(i)” simply reassigns the value of the property “id” from “0” to the return of the function.  
On the macro level {name:”Stallone”, id:0} becomes {name:”Stallone”, id:100}.

This type of index referencing works with any type of nested array:  
if a = [4, [apple, 7, triangle, [88, 69], 6, 6], tomorrow] then:  
a[1][3][0] is equal to 88; and  
a[2] is equal to tomorrow;  
Hope that helps 😉

* + - * + http://0.gravatar.com/avatar/f560b402ce0674f732b20b4b8cf96741?s=60&d=mm&r=g

\_redrobot

[February 13, 2015 at 6:13 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-306101) /

That was an amazing explanation. Thanks for taking the time to write that up!

* + - http://2.gravatar.com/avatar/e46f47c49613e6ad5bb3c6a260924266?s=60&d=mm&r=g

akhil

[February 7, 2018 at 5:44 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-309023) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=309023#respond)

Thanks!…

* + - http://2.gravatar.com/avatar/e46f47c49613e6ad5bb3c6a260924266?s=60&d=mm&r=g

akhil

[February 7, 2018 at 5:57 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-309024) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=309024#respond)

This works fine?!  
function celebid(celebrities){  
var i;  
var id=100;  
for(i=0;i<celebrities.length;i++){  
celebrities[i].id= id+i;  
}  
return celebrities;  
}  
var celebs=[{name:"stallone",id:"0"},{name:"arnold",id:"0"}];  
var createceleb=celebid(celebs);  
var stallone=createceleb[0];  
var arnold=createceleb[1];  
console.log(stallone.id);  
console.log(arnold.id);

1. http://1.gravatar.com/avatar/768aa2e0f451b665241460b13de0ba60?s=60&d=mm&r=g

pradeep

[March 25, 2013 at 3:01 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-4658) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=4658#respond)

very helpful

1. http://2.gravatar.com/avatar/b12ed0f5cbcb4fe4a3feabc831d5a8c7?s=60&d=mm&r=g

Lixiang

[March 28, 2013 at 11:07 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-4897) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=4897#respond)

Good post, It makes me feel Javascript is so sexy and interesting.

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[April 1, 2013 at 8:46 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-5119) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=5119#respond)

Haha, you made my day. Great Comment, Lixiang.

1. http://0.gravatar.com/avatar/954b772a89aff03f6cae6698eb3adc9f?s=60&d=mm&r=g

Roberto

[April 4, 2013 at 4:39 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-5234) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=5234#respond)

Sorry man, I really don’t get the example on the first rule,  
when you say:

mjName (“Jackson”); // This celebrity is Michael Jackson

how are you calling the inner function there?

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[April 4, 2013 at 9:00 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-5238) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=5238#respond)

No need to be sorry, Roberto, I understand how this part is confusing.

Here is the explanation.  
The *celebrityName* function returns a **function**; the function it returns is the *lastName* inner function.  
So when we do:  
var mjName = celebrityName (“Michael”);  
The *mjName* variable now contains the *lastName* inner function that was returned when we called celebrityName (). And the inner function has all the variables, including the name “Michael” that was passed into the celebrityName function.

To prove that the mjName variable contains the lasName function, we can do this:

|  |  |
| --- | --- |
|  | console.log (mjName); |
|  | ​ |
|  | ​// The output is this:​ |
|  | ​function lastName(theLastName) { |
|  | return nameIntro + firstName + " " + theLastName; |
|  | } |
|  | ​ |

So *mjName* is a function expression: it is a variable with a value of a function.  
Now, all we have to do to call the function stored in mjName is add the parentheses () to execute the function, like this:  
mjName (); //When it executes, the following line from the lastName inner function runs;  
return nameIntro + firstName + ” ” + theLastName;  
And this is why the output is: “This celebrity is Michael Jackson”  
When we do mjName (“Jackson”);

* + - http://0.gravatar.com/avatar/954b772a89aff03f6cae6698eb3adc9f?s=60&d=mm&r=g

Roberto

[April 4, 2013 at 9:40 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-5240) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=5240#respond)

Getting there!! but WOW! thanks for taking the time to write such a long and clear answer! I’m wrapping my head around the last one now.

It’s weird though being a JavaScript Blog you dont have client side validation on the comments form. 😉

* + - * http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[April 4, 2013 at 10:02 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-5242) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=5242#respond)

You are welcome.

I haven’t thought about the client-side form validation until you mentioned it, so I just added it. Thanks for the tip. We helped each other out 

* + - * http://1.gravatar.com/avatar/13f98da23e2b0fab1aff45a67165ed34?s=60&d=mm&r=g

Vineet Sajwan

[August 21, 2014 at 2:35 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-274345) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=274345#respond)

Hey Roberto,  
Its quite amazing that I have same two doubts, one is mjName thing and other absence of javascript form validation on a such a SEXY javascript tutorial site.

Thanks Richard Bovell for sharing your valuable knowledge.

* + - http://2.gravatar.com/avatar/252c7694bd8243a515b91e4e8dbb8a97?s=60&d=mm&r=g

Christopher

[June 11, 2013 at 3:46 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-12688) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=12688#respond)

This was \*profoundly\* helpful to me. Thank you so much for taking the time to reply so thoroughly.

* + - http://2.gravatar.com/avatar/23ddb66f2f71b25bdfe7a5814606af70?s=60&d=mm&r=g

[Brett Warner](http://www.brettkwarner.com/)

[August 10, 2013 at 3:14 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-27153) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=27153#respond)

Thank you so much for this comment. I’ve been trying to understand clojures for a few days now and the fact that it was returning the function was the missing piece for me, I feel like all the code I’ve pored over just suddenly started to make sense.

* + - * http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[August 15, 2013 at 2:08 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-28991) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=28991#respond)

Great to hear that you have the moment of absolute clarity with JavaScript, Brett. All of us experience the same thing from time to time.

* + - http://1.gravatar.com/avatar/1aa42ddf36b53a15f1ae82a4e582d28c?s=60&d=mm&r=g

Pravin Waychal

[January 24, 2014 at 8:03 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-181049) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=181049#respond)

Very Helpful and clear explanation. Thanks so much!

* + - http://1.gravatar.com/avatar/aaa69ad5eeecc14c65b70547cb0307b4?s=60&d=mm&r=g

andy

[April 14, 2014 at 2:07 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-198385) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=198385#respond)

everyone trying to understand closures should read this question and answer.

* + - * http://1.gravatar.com/avatar/a20d4f5d185e6e3764e95fa916c57973?s=60&d=mm&r=g

Bryson

[November 21, 2014 at 8:13 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-305903) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=305903#respond)

Yep, agreed. I’ve been trying to understand closures on and off for two years… This might have just finally solved it for me!

* + - http://2.gravatar.com/avatar/ba59dd1efc824e8277be0940293d9ba5?s=60&d=mm&r=g

jackson

[May 9, 2014 at 7:16 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-223751) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=223751#respond)

Thanks for this really helpful reply. Pointing out that the container function returns the inner function was key for me.

Although after typing out the example, I noticed that printing out mjName doesn’t actually show the function.  
console.log(mjName); //prints [Function: lastname]

To show the actual function I had to use the prototype method of the function object:

console.log( mjName.toString() ); //prints function source

* + - http://2.gravatar.com/avatar/b591350a198f2f87b875c5b07257a368?s=60&d=mm&r=g

Thbo

[February 24, 2016 at 3:40 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307905) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307905#respond)

But how does the lastName callback function receive the ‘thelastName'(Jackson) parameter when calling the mjName variable?

1. http://2.gravatar.com/avatar/845035a138a614f29254944d74527685?s=60&d=mm&r=g

Juan

[April 9, 2013 at 11:31 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-5500) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=5500#respond)

This site rocks man!! Thanks for sharing your knowledge in an easy way. Saludos desde Argentina

1. http://1.gravatar.com/avatar/d979737cc03e78a92944765e96b4c9b8?s=60&d=mm&r=g

ali

[April 18, 2013 at 5:28 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-6034) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=6034#respond)

i have one question about the following.  
theCelebrities[i][“id”] = function (j) { // the j parametric variable is the i passed in on invocation of this IIFE

how does it here know the j should take the value of the current i ,maybe i am missing something obvious in here:)

many thanks

* + http://1.gravatar.com/avatar/d979737cc03e78a92944765e96b4c9b8?s=60&d=mm&r=g

ali

[April 18, 2013 at 5:31 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-6035) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=6035#respond)

oops i missed this little devil in here  
(i); // immediately invoke the function passing the i variable as a parameter

sorry

* + - http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[April 25, 2013 at 8:15 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-6405) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=6405#respond)

Ali,  
Are you good now or do you still have a question?

* + - * http://1.gravatar.com/avatar/d979737cc03e78a92944765e96b4c9b8?s=60&d=mm&r=g

ali

[April 26, 2013 at 5:13 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-6429) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=6429#respond)

Yeah its cool now ,thanks alot.

1. http://0.gravatar.com/avatar/34ce4f61511d71f20c1a0e1b24277134?s=60&d=mm&r=g

[Jason Kotenko](http://jasonkotenko.com/)

[May 26, 2013 at 7:37 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-9364) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=9364#respond)

For the CelebrityIdCreator example, could you also use:

theCelebrities[i][“id”] = i;

So that the value is set right then, rather than have “id” be a function?

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[May 27, 2013 at 10:34 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-9479) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=9479#respond)

Very good question, Jason.

To do that, we can make one change:  
Instead of returning a function, we can go ahead and call the function (execute it immediately) and return the uniqueID + J number.

I just updated the code in the article to store numbers in the returned array, instead of functions.

1. http://0.gravatar.com/avatar/9837447b4a883a09649cba853866d2e0?s=60&d=mm&r=g

Piyush

[June 23, 2013 at 4:15 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-14883) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=14883#respond)

Hi Richard,  
Awesome blog….seriously …going through your blog has made some of the difficult concepts so easy. Thanks a lot.

Is it possible to refactor the code for removing the bug in Closure with the below way :

theCelebrities[i][“id”] = (function () {  
return uniqueID + i;  
})();

The one mentioned above somewht confused me at the start , so i guess this would too solve the purpose. In effect, both the code are doing the same things  

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[June 25, 2013 at 1:11 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-15228) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=15228#respond)

HI Piyush,

I am glad you were able to refactor the code to a style that works better for you. That is an important aspect of programming that is underrated. In my opinion, you should feel very comfortable with the way your code looks and reads back in your head.

When I have some time, I will look at the code and see if it is worth refactoring.

Thanks.

1. http://2.gravatar.com/avatar/837b623057aca4f97e8c151f5e780713?s=60&d=mm&r=g

[soncu](http://notyet/)

[June 26, 2013 at 7:33 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-15559) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=15559#respond)

Very good article. I loved your explanation really but when i saw:

theCelebrities[i][“id”] = function (j) {  
return function () {  
return uniqueID + j;  
} ()  
} (i);

i got uncomfortable. Because you introduce two IIFE here. And one can offer in a more complicated way as:

theCelebrities[i][“id”] = function (j) {  
return function () {  
return function() {  
return function() {  
return function() {  
return uniqueID + j;  
}()  
}()  
}()  
} ()  
} (i);

but these all make things more harder. I suggest to use just IIFE in a simple way as:

theCelebrities[i][“id”] = function (j) {  
return uniqueID + j;  
} (i);

and all these versions just use:

theCelebrities[i][“id”] = uniqueID + j;

——————————————  
Besides I want to add one more thing: your first example under “Closures Gone Awry” section all ids assigned to a function but the other example(bug fixing as your phrase) just assigns numeric variable using IIFE.I think it is impossible to solve this problem with assigning function to ids.

* + http://2.gravatar.com/avatar/ec68dbad639bc6e25d1b98f86594a163?s=60&d=mm&r=g

Hadar

[March 11, 2014 at 3:51 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-188550) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=188550#respond)

Exactly what I thought!  
Too bad you didn’t get an answer for that until today.. this could be \*very\* helpful.

1. http://0.gravatar.com/avatar/f62ccf6da5df712e36f0a7ff896e657c?s=60&d=mm&r=g

Nirmal

[July 2, 2013 at 7:21 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-17284) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=17284#respond)

Richard, this is the coolest blog on JavaScript that I ever saw! It helped me in understanding the fundamentals thoroughly.  Thanks!

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[July 5, 2013 at 6:35 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-18205) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=18205#respond)

Thank you very much, Nirmal. I am glad the blog has been helpful.

1. http://1.gravatar.com/avatar/722802d56dd2f3205bbe81db0557adb1?s=60&d=mm&r=g

Evox

[July 12, 2013 at 12:59 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-19416) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=19416#respond)

– In example 2, why we have to add return before get and set function.

– why it have to include return before function and value, can I write it with out return

– Is the inner function excute and return separately from outer function?

Thanks you

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[July 17, 2013 at 3:01 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-20294) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=20294#respond)

You can reformat the code to not return an object with the methods, but then you will have to change the celebrityID function and make it an object. And is you do, there will be no reason to assign it to another variable to use the closure with references to the outer function’s variables.

So the reason we used “return” to return an object in the celebrityID function is precisely because of the closure and accessing the outer function’s variables later.

1. http://1.gravatar.com/avatar/1f0509aa1b6a82b97f22de29907b95ef?s=60&d=mm&r=g

WebGyver

[July 19, 2013 at 5:48 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-20726) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=20726#respond)

First of all, thanks for this article. I greatly appreciate the time you took to work on the article itself but also on answers to questions.

I have a problem with the celebrityName example that returns a function. No matter how hard I try, I can only get the first name back. Instead of a lastName, I only get a 0.

I have set this up at <http://jsfiddle.net/MmDvz/> if anybody else wants to help me see the errors of my ways.

Thank you for any help anyone could offer.

* + http://1.gravatar.com/avatar/722802d56dd2f3205bbe81db0557adb1?s=60&d=mm&r=g

Ecovox

[July 20, 2013 at 1:17 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-20776) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=20776#respond)

I think the mjName(“Miller”); was return and destroyed before you call it. try this: $(‘#name’).html(mjName(“Miller”));.

I’m not sure why it be like this, maybe a concat method from primitive wrapper was call to produce it, after it is call, it will destroy.

sorry for my English.

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[July 31, 2013 at 1:04 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-23168) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=23168#respond)

You have everything correct except you need to call mjName as a function, as **Ecovox** correctly pointed out below.

|  |  |
| --- | --- |
|  | ​ |
|  | ​/\* The mjName variable is being assigned a function, which is the function returned by your celebrityName function. |
|  | \*/​ |
|  | ​var mjName = celebrityName("Mike"); |
|  | ​// So you must call mjName like this: |
|  | $('#name').html(mjName("Miller")); |
|  | ​ |
|  | ​// Instead of like this |
|  | $('#name').html(mjName); |
|  | ​ |

This line below by itself (which you have in your code) is not displaying anything to the screen or on the page because you are not logging (console.log) it to the console or displaying the results to the page:

|  |  |
| --- | --- |
|  | ​ |
|  | ​mjName("Miller"); |
|  | ​ |

So the bottom line is that you can do either this:

|  |  |
| --- | --- |
|  | ​ |
|  | console.log(mjName("Miller)); |
|  | ​ |

Or this:

|  |  |
| --- | --- |
|  |  |
|  | $('#name').html(mjName("Miller")); |
|  | ​ |

Ecovox, thanks for helping out. You answer is correct.

1. http://2.gravatar.com/avatar/5da4dc73f8e2d4824c0454954cef8b18?s=60&d=mm&r=g

Noah Jerreel Guillen

[August 1, 2013 at 8:47 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-23907) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=23907#respond)

Your blog is really amazing. Keep it up.

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[August 15, 2013 at 2:06 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-28989) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=28989#respond)

Thank you much, Noah.

And thanks, Dineshswamy—this is the coolest name, btw 

1. http://2.gravatar.com/avatar/56d9a826d21794a38ca057012a2fac84?s=60&d=mm&r=g

dineshswamy

[August 9, 2013 at 4:19 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-26538) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=26538#respond)

I like most of your articles !! Your articles stands because of one reason .. you dont give space for ads !! Nice.

1. http://2.gravatar.com/avatar/598f6c7aca743120ee9aa84be7b13bbc?s=60&d=mm&r=g

tanyaka

[August 12, 2013 at 3:46 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-27862) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=27862#respond)

Dear Richard, your blog is awsome and I do enjoy reading it, thank very much! Still, I have the same questinon as soncu on June 26 (may by I missed something essential). Why not to use:

theCelebrities[i][“id”] = function (j) {  
return uniqueID + j;  
} (i);

instead of

theCelebrities[i][“id”] = function (j) {  
return function () {  
return uniqueID + j;  
} ()  
} (i);

?

Thank you so much for your time!

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[August 15, 2013 at 2:21 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-28999) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=28999#respond)

The reason the code returns a function is because the point of closures is that the inner function has access to the outer function’s variables even after the outer function has returned, so you can continue to execute code like this: stalloneID.id(). The demonstration was about closures, which are inner functions, hence the example.

Indeed, you can choose to not return a function and simply return the variable, but in some scenarios, you do need to return a function that you want to call later. The code shows the example of returning a function and calling that function at a later time.

* + - http://2.gravatar.com/avatar/598f6c7aca743120ee9aa84be7b13bbc?s=60&d=mm&r=g

tanyaka

[August 15, 2013 at 5:58 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-29068) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=29068#respond)

i see. thank you very much!

* + - http://0.gravatar.com/avatar/9de9ad61f48b277a17060df4978dc7be?s=60&d=mm&r=g

Pearce

[September 28, 2013 at 2:12 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-53228) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=53228#respond)

Hey Richard,  
I believe you are missing soncu and tanyaka’s critique. The second block of code assigns numbers to the id property of elements in the “theCelebrities” array, not functions. This trivializes the problem and makes the use of IIFE’s unnecessary. This is confusing your readers. I think the most instructive solution would be this:

function celebrityIDCreator (theCelebrities) {  
var i;  
var uniqueID = 100;  
for (i = 0; i < theCelebrities.length; i++) {  
theCelebrities[i]["id"] = function (j) {  
return function() { // Return a function, not a number  
return uniqueID + j;  
}  
}(i)  
}

return theCelebrities;  
}

var actionCelebs = [{name:"Stallone", id:0}, {name:"Cruise", id:0}, {name:"Willis", id:0}];

var createIdForActionCelebs = celebrityIDCreator (actionCelebs);

var stalloneID = createIdForActionCelebs [0];  console.log(stalloneID.id()); // 100

* + - * http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Of Stanley](http://javascriptissexy.com/) (Author)

[October 1, 2013 at 4:06 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-61637) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=61637#respond)

I must be missing something because the same code you are suggesting is what I provided in the article above: the first part of section “Closures Gone Awry.”

* + - * + http://0.gravatar.com/avatar/9de9ad61f48b277a17060df4978dc7be?s=60&d=mm&r=g

Pearce

[October 1, 2013 at 10:43 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-62466) /

The code I’m suggesting has 1 IIFE as opposed to 2 (or 0). Sorry for the lack of formatting, I couldn’t figure out how other commenters did the formatted code blocks.

* + - http://2.gravatar.com/avatar/e83e5243cabbf1ee65c983b878c01f24?s=60&d=mm&r=g

San

[May 7, 2014 at 7:01 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-221319) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=221319#respond)

Hello,  
I have also the same confusion.  
In the code if I give as follows the result as follows.  
 console.log(cruiseID.id); // 101  
 console.log(cruiseID.id()); // TypeError: stalloneID.id is not a function

why it shows this error?  
//San

* + - http://1.gravatar.com/avatar/dac79902c081b007ff2840a5225e76f1?s=60&d=mm&r=g

DeVontae Moore

[December 11, 2014 at 1:13 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-305976) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=305976#respond)

It could you point out your fact about the code returning a function and calling that function later.  
I saw it clearly in your IIFE tutorial:  
showName = function (name) {console.log(name || “No Name”)}) (); // No Name​  
​  
showName (“Rich”); // Rich​  
showName (); // No Name

but dont see those same principles here.Thank you for your time

1. http://0.gravatar.com/avatar/c814353cf09c1da09771cc178ace978a?s=60&d=mm&r=g

sameer

[August 20, 2013 at 3:01 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-30917) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=30917#respond)

Thanks for the sexy website

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[August 21, 2013 at 1:09 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-31230) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=31230#respond)

Thanks, Sameer.

I am hopeful the site is sexy and informative at the same time. The saying in English is “brain and brawn” :), which means sexy (or strong for male) and smart at the same time.

I added the explanation for “brain and brawn” in case English is not your first language.

1. http://1.gravatar.com/avatar/43a68ff9076f9688fd586f53511fd064?s=60&d=mm&r=g

[Cosmo Scrivanich](http://www.priceline.com/)

[August 21, 2013 at 10:07 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-31183) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=31183#respond)

For the last solution won’t just be alot easier to use bind?

function celebrityIDCreator (theCelebrities) {  
var i;  
var uniqueID = 100;

for (i = 0; i < theCelebrities.length; i++) {  
theCelebrities[i]["id"] = function (j) { // the j parametric variable is the i passed in on invocation of this IIFE  
return uniqueID + j; // each iteration of the for loop passes the current value of i into this IIFE and it saves the correct value to the array  
}.bind(this, i) // BY adding () at the end of this function, we are executing it immediately and returning just the value of uniqueID + j, instead of returning a function.  
}

return theCelebrities;  
}

var actionCelebs = [{name:"Stallone", id:0}, {name:"Cruise", id:0}, {name:"Willis", id:0}];

var createIdForActionCelebs = celebrityIDCreator (actionCelebs);

var stalloneID = createIdForActionCelebs [0];  
 console.log(stalloneID.id()); // 100

var cruiseID = createIdForActionCelebs [1]; console.log(cruiseID.id()); // 101

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[August 21, 2013 at 1:29 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-31234) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=31234#respond)

Thanks for the bind () alternative, Cosmo.

This is why JavaScript is cool: you can always find an expressive way to execute your code, and the bind example you illustrated is perfectly fine, if you are interested in just returning the value and not a function.

Note that in some scenarios, as I have noted above in an earlier comment, you may want to return a function, which is more versatile than returning just the value. But your solution is a wonderful addition nonetheless.

1. http://2.gravatar.com/avatar/5f22bd37dfb251ca21ef8024b9640ff8?s=60&d=mm&r=g

[Pierre](http://monkinteractive.com/)

[August 23, 2013 at 3:32 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-31990) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=31990#respond)

Hi,

Thanks for this great page. I’ve been pouring over it and absorbing every line. I think I have it all except the following:

for (i = 0; i < theCelebrities.length; i++) {  
theCelebrities[i]["id"] = function (j)

How does the value of i get passed into j if it's never explicitly told they are the same value?

Thanks!  
Pierre

* + http://2.gravatar.com/avatar/5f22bd37dfb251ca21ef8024b9640ff8?s=60&d=mm&r=g

[Pierre](http://monkinteractive.com/)

[August 23, 2013 at 3:53 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-31993) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=31993#respond)

NM just had a facepalm moment. I was so fixated on that part of the code, I neglected to notice it was being passed in below!

* + - http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[August 26, 2013 at 5:35 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-32736) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=32736#respond)

I have never heard the term “facepalm” before, but I take it you have figured it out. Glad to hear 

1. http://0.gravatar.com/avatar/c600075e5202f44a19e62f3e8646e7f0?s=60&d=mm&r=g

hiren

[September 3, 2013 at 11:55 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-34753) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=34753#respond)

Hi Richard,

Great site. Found really helpful.  
Thank you so much.

Hiren

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[September 4, 2013 at 9:37 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-35027) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=35027#respond)

You are welcome, Hiren. And thank you.

1. http://0.gravatar.com/avatar/0fdb8118b1dcd858424bb236a23a9f91?s=60&d=mm&r=g

olivier nguyen

[September 10, 2013 at 6:29 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-36687) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=36687#respond)

Hi there, this is a great website. The suggestion I would make is that the code examples are hard to read because of the narrow width of the main column. And the indentation doesn’t help either.

Looking forward to more posts!

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Bovell](http://javascriptissexy.com/) (Author)

[September 11, 2013 at 8:15 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-37044) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=37044#respond)

Thanks, Olivier. I will definitely redo the layout within a few weeks.

1. http://1.gravatar.com/avatar/75003a667972e54c6b5d21a3d1c9c333?s=60&d=mm&r=g

Ahmed

[September 22, 2013 at 1:07 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-41630) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=41630#respond)

Thanks for the post , its the best I read on Closures

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Of Stanley](http://javascriptissexy.com/) (Author)

[September 23, 2013 at 1:02 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-42691) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=42691#respond)

Thank you very much for the comment, Ahmed. I am happy to hear that the article has been helpful to you.

* + http://0.gravatar.com/avatar/0d6ebf5e71010d2fd2e22c690f6c06ee?s=60&d=mm&r=g

Roshan

[September 27, 2013 at 11:55 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-49843) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=49843#respond)

Hello Richard,

This is one of the best explanations that I have read. It seems like this explanation will be understood even by a 6 year old. Remembering the famous Einstein quote.

I have subscribed and will be waiting to see more from this site.

One quick question  
console.log(cruiseID.id);  
what is this cruiseID?

* + - http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Of Stanley](http://javascriptissexy.com/) (Author)

[October 1, 2013 at 3:55 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-61618) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=61618#respond)

Thanks very much for wonderful compliment, Roshan.

The cruiseID is the variable that will hold the value of Tom Cruise’s data. First, here is the object with all the celebrities:

|  |  |
| --- | --- |
|  | ​ |
|  | ​var actionCelebs = [{name:"Stallone", id:0}, {name:"Cruise", id:0}, {name:"Willis", id:0}]; |
|  | ​ |

Then we set up the array that will hold all the celebrities; we do this by executing the celebrityIDCreator function with the object we created in the line above:

|  |  |
| --- | --- |
|  | ​ |
|  | ​var createIdForActionCelebs = celebrityIDCreator (actionCelebs); |
|  | ​ |

And with those, we then created variables for each celebrity, so the Tom Cruise variable is this one:

|  |  |
| --- | --- |
|  | ​ |
|  | ​var cruiseID = createIdForActionCelebs [1]; console.log(cruiseID.id); // 10​ |
|  | ​ |

1. http://1.gravatar.com/avatar/ac0861e72bbf7b3d9cd5526945106924?s=60&d=mm&r=g

Sneha

[September 27, 2013 at 2:35 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-50182) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=50182#respond)

Hey Richard,  
I am learning Javascript and I really appreciate the efforts u took to explain one of the readers “Roberto” because that really helped me understand clearly.  
Thanks.  
Sneha.

1. http://0.gravatar.com/avatar/928256ec0666983be5ee8e5db08d3cd3?s=60&d=mm&r=g

[Danijel](http://primetimetable.com/)

[September 29, 2013 at 4:50 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-54435) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=54435#respond)

Your blog helps me a lot, and reduce time needed to read all other blogs. I find the information on your blog and the way you are presenting the most informative. Just wanted to say thank you!!!

1. http://1.gravatar.com/avatar/40fc0acae1d23d81d942ea4734314912?s=60&d=mm&r=g

Foysal Mamun

[September 30, 2013 at 9:41 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-57973) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=57973#respond)

really sweet post. Thanks.

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Of Stanley](http://javascriptissexy.com/) (Author)

[October 1, 2013 at 4:15 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-61652) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=61652#respond)

Thanks to Sneha, Danijel (nice name, btw), and Foysal Mamun for the wonderful compliments. Your comments are very encouraging to me and I really appreciate all of your feedback.

1. http://1.gravatar.com/avatar/d5fbc92e114f1c14aed8ce160f94e189?s=60&d=mm&r=g

Vish

[October 1, 2013 at 6:38 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-61940) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=61940#respond)

Hi Richard,  
Sorry for the long post. I wanted to explain my scenario as detailed as possible. I was experimenting with your example to see how it would work if I tweak things a bit. I changed it as follows

function celebrityIndicator(theCelebrities) {  
var i;  
var uniqueId = 100;  
for (i = 0;i < theCelebrities.length; i++) {  
theCelebrities[i]["id"] = function(j) {  
return function() {  
return uniqueId + j;  
}();  
}  
} // did not pass i as a parameter here

return theCelebrities;  
}

var actionCelebs = [{name:"Stallone", id:0}, {name:"Cruise", id:0}, {name:"Willis", id:0}];

var createIdForActionCelebs = celebrityIndicator(actionCelebs);  
var stallone = createIdForActionCelebs[0];  
console.log(stallone.id());

Instead of passing i as a parameter, I assigned the variable theCelebrities[i]["id"] to the function reference and tried printing the id.

I got NaN as the output. Why would this happen ?

1. http://1.gravatar.com/avatar/40e929eaded7a03384849ec1a9ac9f35?s=60&d=mm&r=g

Sriram

[October 10, 2013 at 2:27 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-84199) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=84199#respond)

Hi Richard

This site is very flexible to learn javascript

good going.

1. http://2.gravatar.com/avatar/b56ddd96eb5e0db8be8335f01c5ff091?s=60&d=mm&r=g

Ankit Agarwal

[October 10, 2013 at 6:43 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-84678) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=84678#respond)

This is really an awesome site to learn javascript.  
I have traversed many other sites for js but i haven’t found such a simple and detailed description.

Bookmarked this site !!  
Great work.

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Of Stanley](http://javascriptissexy.com/) (Author)

[October 11, 2013 at 1:58 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-86372) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=86372#respond)

Thank you very much, Ankit.

* + - http://2.gravatar.com/avatar/b56ddd96eb5e0db8be8335f01c5ff091?s=60&d=mm&r=g

Ankit Agarwal

[October 11, 2013 at 2:08 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-86385) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=86385#respond)

Hi Richard,

If you have any js based project then please share i with me, i want to contribute so that i can enhance my skills and can learn.

Thanks  
Ankit

* + - * http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Of Stanley](http://javascriptissexy.com/) (Author)

[October 15, 2013 at 8:29 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-95964) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=95964#respond)

Ankit, you can check on Github. There are many small, medium, and large projects you can contribute to.

* + - http://2.gravatar.com/avatar/b56ddd96eb5e0db8be8335f01c5ff091?s=60&d=mm&r=g

Ankit Agarwal

[October 11, 2013 at 2:09 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-86390) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=86390#respond)

Hi Richard,

If you have any javascript based project then please share i with me, i want to contribute so that i can enhance my skills and can learn.

Thanks  
Ankit

1. http://2.gravatar.com/avatar/bf7e5a966f9d5db874ee1d17c346fc65?s=60&d=mm&r=g

Alisa

[October 11, 2013 at 12:46 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-85174) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=85174#respond)

Hello,  
I’m still struggling with closures but getting um, closer. Where can I find the quiz you mentioned for Feb 7? Muchas gracias, Alisa

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Of Stanley](http://javascriptissexy.com/) (Author)

[October 11, 2013 at 1:57 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-86371) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=86371#respond)

Alisa, you caught me: I was supposed to post that quiz a while back and I completely forgot. I will think about doing it sometime soon; just for you 

Which part of closures are you struggling with? Post your questions here and I will answer them.

1. http://1.gravatar.com/avatar/42ef422ca336360ef3dcfec13cafb29d?s=60&d=mm&r=g

Tyler

[November 1, 2013 at 9:51 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-132755) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=132755#respond)

Immediately invoking function expressions felt like a greased pig until I read your reply about testing a returned inner function. Once assigned to a variable, logging that variable without parenthesis dumps the function declaration. Logging the variable followed by () invokes the function. And so, assigning a function to a variable and following it with () assigns the function to the variable and runs it.

Ok, I just re-read what I wrote and I’m confused again… Maybe.

Curious how a passed parameter doesn’t persist. Perhaps the assignment of the variable using ‘var’ gives the anonymous function a clean slate.

1. http://0.gravatar.com/avatar/cf20d0b549724addaa796f06d6964351?s=60&d=mm&r=g

jeff

[November 19, 2013 at 1:48 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-162345) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=162345#respond)

Enjoying your site…

I was playing with a couple of the examples. Now I’m confused about the syntax difference of two of the examples. The 1st example (function showName) returns the closure function as follows:

return makeFullName (); // the parens are included

The 3rd example (function celebrityName) returns its closure function WITHOUT the trailing parens as follows:

return lastName; // no parens

The 3rd example’s closure function has parameters, the 1st example’s does not, otherwise things seem equal. Dropping parens from the 1st or adding them to the 3rd breaks things.

What’s going on?

Thanks,  
Jeff

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Of Stanley](http://javascriptissexy.com/) (Author)

[November 22, 2013 at 3:30 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-163273) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=163273#respond)

I see that you are paying attention to each line carefully there, Jeff. 

In the first example, we are returning the result of executing the makeFullName function. We actually didn’t need the makeFullName function at all; we could have just add the code right inside the showName function (not inside the makeFullName closure). This was example was just for demonstration of what a closure is hence the reason we added the makeFullName inner function.

But in the second example, we are returning the lastName function BEFORE it is executed, so that we can execute it later. If we return it like this, *return lastName()*, then it would have executed at that point in the code (where it is defined) and it would not have had access to the outer function’s variables and parameters later because we couldn’t call it later.

So the reason we return only lastName (without the params) is that we want to execute the function later and so that it can access the outer functions’s variables.

* + - http://0.gravatar.com/avatar/92e91d9d369e066477b98d730de0b521?s=60&d=mm&r=g

Bart Kleijngeld

[November 28, 2014 at 10:02 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-305917) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=305917#respond)

I had the exact same question as Jeff, and would like to make sure I understand your explanation. Can I summarize it as follows?

In the first example a closure is defined in the most basic way, the purpose to show what a closure is basically, \_but\_ it is a useless example and in practice one will always encounter the returning of closures for them to be useful?

Thanks for the great article!

1. http://1.gravatar.com/avatar/4d061d66779e8d4890fc705b6b4264d8?s=60&d=mm&r=g

Rakesh Sivan

[December 3, 2013 at 12:11 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-167385) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=167385#respond)

I was running between several websites to get a clear picture about closures and also about the java script’s most powerful features in general.Your website has the right stuff with the needed clarity. Thanks 

1. http://1.gravatar.com/avatar/74a87c52725baae5d1fb9c751fbcde6a?s=60&d=mm&r=g

[vimal kumar](http://javascriptissexy/)

[December 10, 2013 at 9:39 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-170322) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=170322#respond)

nice explanation….I liked very much:)

1. http://1.gravatar.com/avatar/7f9a6f23b77b4dbdb06a665a35dc9ba7?s=60&d=mm&r=g

Petr

[December 12, 2013 at 7:07 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-171004) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=171004#respond)

Hi,

firstly thanks for the articles, they are absolutely amazing. I’m reading through all of them and I will recommend them to other people every chance I get!

My question – in the last code block example – why does the inner function return another functions? Isn’t that redundant to create a function whose only aim is to return another function? Or am I missing something?

P.S.: It seems that your comment section has the ability to highlight code, but users don’t use it and I can’t seem to find how to do it either. It’d make the comments section a lot more readable (sexy) though 

Once again, thank you for the great articles!

1. http://2.gravatar.com/avatar/2668c1e79dab427dfa1556fd773710cf?s=60&d=mm&r=g

hausFrauFromHades

[December 17, 2013 at 10:24 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-172459) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=172459#respond)

I am really enjoying your articles, particularly this one and the one on callbacks. Thanks for your work.  
JavaScript IS sexy– who knew?

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Of Stanley](http://javascriptissexy.com/) (Author)

[January 7, 2014 at 1:01 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-177477) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=177477#respond)

It sounds like you have experienced JavaScript enlightenment, hausFrauFromHades. That’s cool.

hausFrauFromHades is an interesting name, BTW.

1. http://2.gravatar.com/avatar/e5f63c47fc5eaac6b319534fcb22717f?s=60&d=mm&r=g

Sarvesh Kesharwani

[December 17, 2013 at 1:35 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-172510) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=172510#respond)

You are just awesome.

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Of Stanley](http://javascriptissexy.com/) (Author)

[January 7, 2014 at 12:58 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-177474) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=177474#respond)

 I wish. But thanks, Sarvesh.

1. http://2.gravatar.com/avatar/8b2de0beab9ffb11a0fa846cad017944?s=60&d=mm&r=g

Shadab

[January 3, 2014 at 5:36 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-176303) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=176303#respond)

function lastName (theLastName) {  
return nameIntro + firstName + ” ” + theLastName;  
}  
return lastName;

in the case above you have written lastname without (),how does js treat this.Is it actually a function call?

* + http://1.gravatar.com/avatar/ad71bdefdcb4247c58f9cc3e27c0dd4d?s=60&d=mm&r=g

[Richard Of Stanley](http://javascriptissexy.com/) (Author)

[January 7, 2014 at 12:13 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-177452) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=177452#respond)

In JavaScript, functions are object, so you can treat them like an object or a variable. When you return a function without the (), you are returning the function definition. So that you can call the function later. For example:

function showMe (){  
console.log (“This is Rich”);  
}

function showRich () {  
return showMe;  
}

If we run this line:  
showRich ();

We get this result:  
function showMe() {  
console.log (“This is Rich”);  
}

Yep, the actual definition of the showMe function.  
But of course it is not very useful just return the showMe function, which is not executing or returning anything itself. The showmen function should execute some action or return something.

1. http://2.gravatar.com/avatar/293da18bbcfcb4f1190186474a4a83f3?s=60&d=mm&r=g

Axel

[January 4, 2014 at 2:24 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-176526) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=176526#respond)

Hello Richard,

I think the last example with the nested IIFE´s does not match your intentions (as I suppose them :-).  
The problem is, that all IIFE’s are executed immediately, so that ‘theCelebrities[i][“id”]’ is a value and not a function anymore. The IIFE’s are equal to assigning the value directly: theCelebrities[i][“id”] = uniqueID + j; (as joncu correctly stated)

What I suppose you wanted is the following:

…  
for (i = 0; i < theCelebrities.length; i++) {  
theCelebrities[i]["id"] = function (j) {  
return function () {  
return uniqueID + j;  
}  
} (i);  
}  
…

This will create a new scope for the inner IIFE at "construction time" containing the current value of i.

Of course the id then must be called (or better – can be calleg) with parnthesises:

…  
 console.log(stalloneID.id()); // 100  
…  
console.log(cruiseID.id()); // 101  
…

By the way, this is the best site I saw up to now explaining the (many) JS-obstacles in an easy and understandable way. Great work!

1. http://0.gravatar.com/avatar/cb0135ecf2a19f48470a8817adaca83a?s=60&d=mm&r=g

Binh Thanh Nguyen

[January 16, 2014 at 11:03 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-179922) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=179922#respond)

Thanks, nice post

1. http://1.gravatar.com/avatar/ae6069a2ca8bf8e0ceefa094dfa3e778?s=60&d=mm&r=g

Jeremy

[January 16, 2014 at 11:48 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-179931) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=179931#respond)

Richard,

This article is very good up until the end. Your example of IIFE is pretty bad, since the result, as others have pointed out, is that every celebrity is assigned an ID when celebrityIDCreator is called and you might have just as well written:

theCelebrities[i]["id"] = uniqueID + i;

Here is a version that uses and IIFE so that theCelebrities.id is still a function but it is not assigned until the id() function is first called. (Then the function is replaced with a new function that always returns the same number.)

function celebrityIDCreator (theCelebrities) {  
var i;  
var uniqueID = 100;  
for (i = 0; i < theCelebrities.length; i++) {  
(function() {  
var thisCelebrity = theCelebrities[i];  
thisCelebrity["id"] = function () {  
var thisID = uniqueID++;  
thisCelebrity["id"] = function() {  
return thisID;  
}  
return thisID;  
}  
})();  
}  
return theCelebrities;  
}

var actionCelebs = [{name:"Stallone", id:0}, {name:"Cruise", id:0}, {name:"Willis", id:0}];

var createIdForActionCelebs = celebrityIDCreator (actionCelebs);

createIdForActionCelebs[2].id(); // 100  
createIdForActionCelebs[0].id(); // 101  
createIdForActionCelebs[2].id(); // 100

1. http://1.gravatar.com/avatar/a506fca29402e0b4f1dc338685647e8c?s=60&d=mm&r=g

Jack

[January 21, 2014 at 2:56 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-180574) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=180574#respond)

Hey man, your site is awesome! I have a question though, for your last example about immediately invoked functions, how come you nested functions instead of a simpler:

function celebrityIDCreator (celebs) {  
var i;  
var uniqueID = 100;  
for (i = 0; i < celebs.length; i++) {  
celebs[i]["id"] = function() {  
return uniqueID + i;  
} ( ); //INVOKE HERE IMMEDIATE  
}  
return celebs;  
}

I'm not just providing a simpler solution because that's not the point of this lesson. But the point was a bit lost on me when I had to deal with more confusing IIFEs.

Could you expound on the point you were making with the last example? Thank you.

1. http://0.gravatar.com/avatar/600d711f4c500569f67f348d88d5ef34?s=60&d=mm&r=g

shams

[March 5, 2014 at 6:35 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-186885) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=186885#respond)

I don’t understand the need of inner function.

theCelebrities[i][“id”] = function (j) { // the j parametric variable is the i passed in on invocation of this IIFE  
return function () {  
return uniqueID + j; // each iteration of the for loop passes the current value of i into this IIFE and it saves the correct value to the array  
} () // BY adding () at the end of this function, we are executing it immediately and returning just the value of uniqueID + j, instead of returning a function.  
} (i); // immediately invoke the function passing the i variable as a parameter  
}

Cant it be simply ?  
theCelebrities[i][“id”] = function (j) { // the j parametric variable is the i passed in on invocation of this IIFE  
return uniqueID + j;  
} (i); // immediately invoke the function passing the i variable as a parameter  
}

1. http://0.gravatar.com/avatar/fb8172f4837c7a4ae8b2be5a76432929?s=60&d=mm&r=g

Yui

[March 12, 2014 at 2:11 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-188628) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=188628#respond)

Greatest post!! I learn a lot!  
I think this might be a little shorter and it still works:  
<http://jsfiddle.net/GGGfW/>

1. http://0.gravatar.com/avatar/9b6b1ef9d92de1784d74f0d74941a99a?s=60&d=mm&r=g

Ibrahim Islam

[March 17, 2014 at 5:31 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-189828) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=189828#respond)

So, in the celebrity id where the id is being assigned a function that returns modified id example when is the anonymous function is executed actually?

1. http://0.gravatar.com/avatar/9b6b1ef9d92de1784d74f0d74941a99a?s=60&d=mm&r=g

Ibrahim Islam

[March 17, 2014 at 7:08 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-189842) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=189842#respond)

In the last example, why are we returning a function and then the modified Id again inside? Why can’t we just return the modified id?

1. http://0.gravatar.com/avatar/9b6b1ef9d92de1784d74f0d74941a99a?s=60&d=mm&r=g

Ibrahim Islam

[March 17, 2014 at 9:15 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-189860) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=189860#respond)

Regarding the last code, this is making more sense to me: <http://jsfiddle.net/L39jM/>

1. http://0.gravatar.com/avatar/f5e893e5e6c751467881864008bb0e93?s=60&d=mm&r=g

ashish

[April 4, 2014 at 6:45 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-194579) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=194579#respond)

You really made Javascript sexy n awsome. !!! Thanks

1. http://2.gravatar.com/avatar/b7c4235432d9f77c946d0f20189014f6?s=60&d=mm&r=g

Kedarnath G

[April 9, 2014 at 6:10 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-196222) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=196222#respond)

Awesome Post :

Excellent Post on Closure. I didn’t think Java script is this much sexy before

Thanks  
Kedarnath

1. http://2.gravatar.com/avatar/28dd1ad9073149acfc57ce8aaa3bbde6?s=60&d=mm&r=g

islam al khaldi

[April 19, 2014 at 5:11 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-202265) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=202265#respond)

First of all, thank you very much for your this nice and constructive blog.  
Refering to your last example in “Understand JavaScript Closures with Ease”, i think the inner IIFE could be removed, because it works fine without it.  
I tried the example in Firebug and it works and gives the same result.  
function celebrityIDCreator (theCelebrities)  
{  
var i;  
var uniqueID = 100;  
for (i = 0; i < theCelebrities.length; i++)  
{

theCelebrities[i]["id"] = function (j)  
{  
return uniqueID + j;  
} (i);  
}  
return theCelebrities;  
}  
var actionCelebs = [{name:"Stallone", id:0}, {name:"Cruise", id:0}, {name:"Willis", id:0}];  
var createIdForActionCelebs = celebrityIDCreator (actionCelebs);  
var stalloneID = createIdForActionCelebs [0];  
console.log(stalloneID.id); // 100  
var cruiseID = createIdForActionCelebs [1]; console.log(cruiseID.id); // 101  
Thanks 

1. http://1.gravatar.com/avatar/a2d7a7167325bb2bc637b2363558ee85?s=60&d=mm&r=g

bhanu

[April 24, 2014 at 8:25 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-207731) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=207731#respond)

Thank you  
i had got a clear picture on closures

1. http://0.gravatar.com/avatar/0c2887c19932ba204e4abe9336971ade?s=60&d=mm&r=g

wdc

[June 15, 2014 at 11:38 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-245747) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=245747#respond)

Call me old school (go ahead, I don’t mind, I’ve been programming since 1974 :), but I honestly don’t see why closure is a good thing. IMO it only allows for obscure, unreliable code. Even in what the first couple of side effect examples, #1 and #2, shows us, you’re telling me that (in #1) the value of ‘Michael’ hangs around in memory, from a previous call to the celebrityName function, so that the subsequent call to the lastName function via the mjName returned function utilizes ‘Michael’ at that time? Why is something like this a good thing? Doesn’t example #2 show why this isn’t a good thing? I would rather have/write software that doesn’t allow stuff like this to happen. Are you telling me that there is a case in which closure HAS to be used, because there is no way to provide a certain functionality by writing functions (most likely by not nesting functions) that does NOT utilize closure in that kind of way? If it IS true that I can write code that provides a certain functionality by not nesting functions and not utilizing closure, in every case where I might see closure being utilized, then I would contend that code that performs that functionality without closure will always be more understandable and reliable than code that utilizes closure, even if the code not using closure takes more lines of code and isn’t regarded as being as “cool” or concise. I would gladly trade more lines of code for better readability/understandability and better reliability/maintainability. If you could provide an example of code that shows the utilization of closure in a way in which the same functionality of that code CANNOT be performed with code NOT utilizing closure, then maybe I would be more open to the technique. It seems to me that sometimes languages working in such a way as this is more because of accident than by original intentional design, and then people start using it just because they can write supposedly “elegant” code that no one else can figure out how it works (at least not without a lot of effort).

* + http://2.gravatar.com/avatar/b169bb187529a040c09e5d2744cb710e?s=60&d=mm&r=g

jwalker

[July 25, 2014 at 3:14 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-262803) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=262803#respond)

Awesome man! Exactly what I was thinking. Writing code like this is just confusing and totally unnecessary.

1. http://2.gravatar.com/avatar/84d14a84b8786a6d58ee535d0b15f09c?s=60&d=mm&r=g

John

[June 26, 2014 at 2:50 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-250413) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=250413#respond)

Hi Richard,

Thanks for writing such a helpful JS blog! I’m having trouble generally understanding when closures should be used. Why would we use a closure to create IDs in the celebrity example? Couldn’t we just do the following with the same effect?

…  
function celebrityIDCreator (theCelebrities) {  
var i;  
var uniqueID = 100;  
for (i = 0; i < theCelebrities.length; i++) {  
theCelebrities[i]["id"] = uniqueID + i;  
}  
return theCelebrities;  
}

1. http://0.gravatar.com/avatar/c9f47374034cb75cfb719926707fcc12?s=60&d=mm&r=g

Kaustubh

[July 21, 2014 at 1:56 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-260988) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=260988#respond)

Thanks Richard for this useful blog.  
function closure is the true heart of javascript and you helped understanding it better

1. http://0.gravatar.com/avatar/0c2887c19932ba204e4abe9336971ade?s=60&d=mm&r=g

wdc

[July 28, 2014 at 12:02 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-264093) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=264093#respond)

I’m still trying to understand why closure is a “good thing”.

Let me take another short stab at justifying it as a language feature and coding technique to use.

In one of your comment responses, you say:

‘So the reason we return only lastName (without the params) is that we want to execute the function later and so that it can access the outer functions’s variables.’

That statement (particularly the word “later”) made me think of an aspect of closure that I can actually accept as being useful. Could one say that the real, underlying usefulness of javascript closure (at least one useful thing, anyway) is that it allows for a kind of data persistence without having to use an actual datastore? IOW, that it is a kind of “in-memory” datastore, with respect to the outer function’s variables and parameter values?

Basically (stated another way), that closure allows “setting up” a function (the inner function) for execution sometime LATER in time, knowing that the values of the outer function’s variables and parameter values will still be there (have been “stored” in memory for use later)? WITHOUT having to store those values in a file on disk, or in an actual database, etc.

This is what I have found lacking in every discussion/explanation of javascript closure – underlying, conceptual REASONS why closure is valuable, not just how the code works in a mechanical sense (and with “mickey mouse” examples that don’t even seem useful in a real-world sense on top of it, which compounds the problem of old-timers like me being able to accept it as a good thing (if you read my previous post)).

ASSUMING that what I am observing in this post is correct – that at least one of the main, beneficial, underlying reasons for closure is that it allows “setting up” a function for later use, and “remembering”, or persisting some data values in memory for use at that later time, without having to use a formal datastore… Can you verify this particular take on it, or am I off-base about it?

If so, then I can grudgingly see how javascript closures might be useful for that particular scenario 

BECAUSE (to expand on the conceptual reasoning) from a software architecture / OOP viewpoint, it allows a function (the inner function) to be polymorphic in the sense that it will have different behavior when it is invoked, based on the values that were passed into the outer function at some previous time.

THAT I can understand and see as being useful.

1. http://2.gravatar.com/avatar/5001a5f64b9f1ffd62f31028dff9af3a?s=60&d=mm&r=g

bayloox

[September 19, 2014 at 10:16 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-292321) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=292321#respond)

Thanks very much for the post.

The “sexy” for JS is catching on man. If it becomes global all the credit will be yours!

1. http://2.gravatar.com/avatar/2789258bb07961a5a9edf64cfe5f3de0?s=60&d=mm&r=g

Kevin

[September 22, 2014 at 8:22 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-294119) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=294119#respond)

Hey Richard,

First of all I would like to thank you for this website because it has really aroused my interest in JavaScript so much so that I really wanna dive more into it.

Now coming to the doubt that I had; in the “Closures Rules and Side Effects” section no. 2 you made a statement that said,

“ Closures get more interesting when the value of the outer function’s variable changes before the closure is called.”

My only doubt in the above scenario is that the example is simple to understand but can you please elaborate more on this topic since the statement is not clear enough from the example that you have given.

1. http://1.gravatar.com/avatar/464724038a353e3c406136f5ed159389?s=60&d=mm&r=g

[iJocuri](http://www.ijocuricopii.ro/)

[October 7, 2014 at 4:15 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-305601) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=305601#respond)

Hey there, thank you once again, you have been a great resource. I have been finding some great resources here.  
Well article.. i just love Javascript!

1. http://2.gravatar.com/avatar/efe8d63c40cc32559e9ce86a809e1adf?s=60&d=mm&r=g

maaz

[October 10, 2014 at 7:12 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-305724) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=305724#respond)

aslamualykum  
i am proud of you  
because you solved lot of my problems.  
thank you so much sir .

1. http://1.gravatar.com/avatar/dac79902c081b007ff2840a5225e76f1?s=60&d=mm&r=g

DeVontae Moore

[October 17, 2014 at 8:42 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-305773) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=305773#respond)

Im not familiar with the following syntax and its purpose:  
theCelebrities[i][“id”]

1. http://1.gravatar.com/avatar/1b7c7a375926b6810e69ae1b00ed5f5a?s=60&d=mm&r=g

ARSHAD

[November 27, 2014 at 1:54 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-305915) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=305915#respond)

WOW!!! SUCH A NICE WAY TO EXPLAIN THE CORE CONCEPTS.. JAVASCRIPT IS REALLY GETTING SEXIER FOR ME NOW… 

1. http://0.gravatar.com/avatar/c519f048d0e98dd15c68762d6b54da3d?s=60&d=mm&r=g

[charles ross](http://chilismaug.github.io/)

[November 28, 2014 at 10:06 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-305918) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=305918#respond)

It’s a great post. But there’s not so much ease. The rule 1 example never could make it into my brain, so I refactored it to make as clear as possible, with descriptive variable names and output harness included. Then the d-oh lamp lit up.

function celebrityNameMunger(firstName) {  
var nameIntro = “This celebrity is “;  
// this inner function has access to the outer function’s variables, including  
// the parameter  
function lastNameAppender(theLastName) {  
return nameIntro + firstName + ” ” + theLastName;  
}  
return lastNameAppender;  
}

console.log (” celebrityNameMunger is global: ” + celebrityNameMunger );  
console.log (” …and now the closure lastNameAppender gets assigned to mjNameBuilder…” );  
var mjNameBuilder = celebrityNameMunger(“Michael”);

console.log (” >>>>>>>>>mjNameBuilder >>>>>>>: ” + mjNameBuilder );  
// At this juncture, the celebrityNameMunger outer function has returned.?  
// The closure (lastNameAppender) is called here after the outer function has returned above  
// Yet, the closure still has access to the outer function’s variables and parameters

console.log (“mjNameBuilder(‘Jackson’): ” + mjNameBuilder(“Jackson”)); // This celebrity is Michael Jackson

We loggin’, bro

1. http://1.gravatar.com/avatar/4c6e8b479d17a8430787806b7de9f0db?s=60&d=mm&r=g

Sakthivel

[December 11, 2014 at 2:55 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-305977) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=305977#respond)

Hi,

Excellent Stuff!! Appreciate your work. Nice to learn.

1. http://1.gravatar.com/avatar/1448d087d72cfb8b32fdd72f881c03ad?s=60&d=mm&r=g

[Anh Tran](http://metabox.io/)

[January 28, 2015 at 1:36 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-306073) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=306073#respond)

Closure is one of the most headache issue in Javascript and I usually find hard to explain that to my developers. This post is very helpful and easy to understand. It helps me a lot for training devs of my team. Thanks for sharing.

1. http://1.gravatar.com/avatar/a4861a4ab4db5f729312673019d9e80d?s=60&d=mm&r=g

Tarak

[January 28, 2015 at 9:21 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-306076) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=306076#respond)

Really sexy stuff i found……

1. http://0.gravatar.com/avatar/38668b6010c6ccef53235d02fc18ab9f?s=60&d=mm&r=g

Mark

[February 24, 2015 at 9:09 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-306135) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=306135#respond)

In your last example, doesn’t

function celebrityIDCreator(theCelebrities) {  
var i; var uniqueID=100;  
for (i = 0; i <theCelebrities.length; i++){  
theCelebrities[i]["id"] = function (){  
return uniqueID + i; }();  
}  
return theCelebrities;  
}

create the same effect with one less level of convolution? although maybe I'm just missing the point you were trying to make.

1. http://2.gravatar.com/avatar/20542811a94198669835eac0276ce6ba?s=60&d=mm&r=g

Samir

[March 17, 2015 at 3:38 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-306275) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=306275#respond)

Dude you are awesome and your website is sexy. Keep doing sexy work 

1. http://0.gravatar.com/avatar/6e836a6b3086fae75dd5f2a4371454d9?s=60&d=mm&r=g

Anthony

[May 9, 2015 at 11:30 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-306716) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=306716#respond)

Hi Richard,

Thanks a lot for this post. It’s great, I’ve learned a lot from your site. FYI, I’m new to Javascript.

But I was wondering if the following paragraph could be made a little clearer, (right after the “Closures gone Awry” section).

“In the preceding example, by the time the anonymous functions are called, the value of i is 3 (the length of the array and then it increments). The number 3 was added to the uniqueID to create 103 for ALL the celebritiesID. So every position in the returned array get id = 103, instead of the intended 100, 101, 102.”

changed to…

“In the preceding example, AFTER THE LAST CALL OF THE ANONYMOUS FUNCTION (singular) IN THE FOR LOOP, the value of i is 3 (the length of the array and then it increments). THE UPDATED VALUE OF i=3 IS THEN ALSO APPLIED TO THE PREVIOUS CALLS OF THIS ANONYMOUS FUNCTION, CREATING 103 FOR ALL celebritiesID. So every position in the returned array get id = 103, instead of the intended 100, 101, 102.”

I read this section like 4-5 times and was still confused, but I think I finally understand.

Thanks again! Cheers!

-Anthony

1. http://2.gravatar.com/avatar/e2be61104d345eb826282d615aa423c9?s=60&d=mm&r=g

Murari

[May 20, 2015 at 4:16 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-306782) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=306782#respond)

Hi,

for (i = 0; i < theCelebrities.length; i++) {  
theCelebrities[i]["id"] = function () {  
return uniqueID + i;  
}  
}

Why do you say that value of i is 3 by the time annonymous functions are called? Isn't it that anonymous functions are being called with every for iteration?

* + http://1.gravatar.com/avatar/48f63d5ea580a6f7c1de7d70963f2c5d?s=60&d=mm&r=g

[Kreg](http://idk/)

[July 22, 2015 at 10:30 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307223) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307223#respond)

I could be wrong (as I am a total noob at js), but I think it’s because the  
anonymous functions are being called \_after\_ the outer function has gone out of  
scope, at which time i has reached the value 3 and terminated the for loop.  
Now, because it was a reference to an anonymous function being stored in the  
array, the function hasn’t been called yet, but when it is, it is referencing  
the value of i at the time of call, which will yeild 3, the value that  
terminated the loop; I hope this helps (and I certianly hope it’s correct lest  
I give bad information).

1. http://2.gravatar.com/avatar/5a43634bcfb2176aa35b0135ee2f415b?s=60&d=mm&r=g

santhosh

[June 5, 2015 at 5:10 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-306961) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=306961#respond)

It’s one of the best explination abt closure thks

1. http://0.gravatar.com/avatar/cb2f01d9336edc21d1558c704c96e72b?s=60&d=mm&r=g

Nikhil Fadnis

[June 6, 2015 at 2:37 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-306976) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=306976#respond)

Dude you are awesome.

1. http://1.gravatar.com/avatar/170b2d6e431e2a1afcc6e2cc9f800f1e?s=60&d=mm&r=g

Chalam CH

[June 7, 2015 at 1:34 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-306980) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=306980#respond)

Hi Richard,  
It is the best explanation on closures. The difficulty in understaning is due to our habitual thinking.  
1. In Closures gone awry section, in each iteration of the loop, what is assigned is NOT a value BUT a function. The same (anonymous) function is assigned(saved) but NOT executed. When we called the saved function later, the function has latest values.  
2. The debate on whether closures are good or bad is another story.  
Thanks a lot for the post.

1. http://1.gravatar.com/avatar/a801781566d4321f3b78be02d95620f1?s=60&d=mm&r=g

Ananth

[June 24, 2015 at 7:24 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307113) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307113#respond)

Example for closure holding reference is not correct. Because Number is immutable data type. If you change the Number then its reference will also be changed.  
Reason because it has 3 is, thats what the value of its outer function’s variable when the closure is called.  
Closure is simply a function defined inside an object anonymously. We all know that Function is an Object, closure will have access to the variables of the object(ie., function) it defined.

1. http://0.gravatar.com/avatar/06781996aed6281cfd73b9924c1921a4?s=60&d=mm&r=g

Jessie

[July 5, 2015 at 2:35 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307162) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307162#respond)

I agree. Tried the same thing without the second inner function and it worked fine.

1. http://0.gravatar.com/avatar/6d2622e44c6a759c5a518b041711a284?s=60&d=mm&r=g

Kartik

[July 6, 2015 at 10:34 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307163) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307163#respond)

Hi Richard,  
In the IIFE i guess this would suffice  
function myClass(myList) {  
var id=100;  
for(var i=0;i<3;i++){  
myList[i]["id"] = function(){  
return id+i;  
}();  
}  
return myList  
}  
var myObj\_1 = [{"name":"kartik", "id":"0"},{"name":"Varun", "id":"0"},{"name":"Sid", "id":"0"}];  
var myObj\_2 = new myClass(myObj\_1);  
console.log(myObj\_2[2].id);

There is no need to write a return function which will return Uniqueid+j, Using just IIFE without paramters (value of i) can be done. Refer the code above.

1. http://0.gravatar.com/avatar/634be31fbd1b1bcb780e41179a84a2f2?s=60&d=mm&r=g

Shaik Amesh

[July 15, 2015 at 4:20 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307207) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307207#respond)

Awesome Tutorial dude

1. http://1.gravatar.com/avatar/75e62b950a50a579e81e54c5b86d8852?s=60&d=mm&r=g

Shabnam

[July 15, 2015 at 2:39 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307210) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307210#respond)

Came across this wonderful site!

JavaScript is sexier now 😉

1. http://1.gravatar.com/avatar/d3888b3b31d26751906be08215cd7bb7?s=60&d=mm&r=g

Jason

[July 26, 2015 at 8:09 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307247) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307247#respond)

Thank you for the great article, Richard.

I’m not sure if you’re aware, but Hack Reactor is referencing this article for potential students who are trying to learn JavaScript, so I figured I would make a few suggestions…

First, in your second point, both the name of your outer function AND the variable inside it is the same (i.e. CelebrityID). This is confusing, and if I’m not mistaken, it produces an error. Additionally, in that same section, you make reference to a “changeTheID function” (which doesn’t exist). I believe you meant to reference the “setID” function.

And finally, regarding section three, I don’t quite understand why an IIFE inside of an IIFE is being used here (along with another parameter). Wouldn’t the following code suffice?

theCelebrities[i][​”id”​] ​=​ ​function​ () { ​return​ uniqueID ​+​ i; } ();

Or maybe I’m missing something? Either way, thanks again!

1. http://1.gravatar.com/avatar/767f291f3e9293476ac5da61f690c0d7?s=60&d=mm&r=g

Jared Hensley

[July 26, 2015 at 8:54 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307248) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307248#respond)

I am a bit confused by this Richard. I know you are busy but if you have a moment, I’d appreciate some clarification. Thank you!

1. http://1.gravatar.com/avatar/767f291f3e9293476ac5da61f690c0d7?s=60&d=mm&r=g

Jared Hensley

[July 26, 2015 at 8:55 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307249) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307249#respond)

I am a bit confused by this Richard. I know you are busy but if you have a moment, I’d appreciate some clarification. Thank you! whoops, double post but trying to get notifications  work around.

1. http://2.gravatar.com/avatar/e6263d96a4208c3de57312b30d5eb875?s=60&d=mm&r=g

kmcguire

[July 29, 2015 at 2:20 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307255) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307255#respond)

You can also create a new variable instance and that will be referenced, not the original memory location.

for (var x = 0; x < 10; ++x) {  
var newvariableinstance = x;  
store(function () {  
alert(newvariableinstance);  
});  
}

1. http://0.gravatar.com/avatar/63fc346385a94ed941d8c690019bd05d?s=60&d=mm&r=g

Yogesh

[August 21, 2015 at 6:46 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307312) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307312#respond)

Am server side developer. can you please tell me what is the best way to debug J query and Java script.

1) console.log  
2) Alert

please suggest.

1. http://0.gravatar.com/avatar/f965a49f2735cd52d6ffbffc6c66b5dd?s=60&d=mm&r=g

Chella

[August 25, 2015 at 6:46 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307319) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307319#respond)

OMG… Finally I understood the Closure… Thanks for make it simple…

1. http://1.gravatar.com/avatar/a4584d47aea909975a99ffc6d7c95c68?s=60&d=mm&r=g

Gonzalo

[August 27, 2015 at 10:57 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307326) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307326#respond)

Thank you for such a great explanation. I had seen some articles explaining closures but I wasn’t able to wrap my head around it until I read this comment. Perhaps you might consider adding it to the blogpost (maybe as a side note or something). It truly is a great explanation  Thanks again!

1. http://1.gravatar.com/avatar/4570bf29e200b3b1ac26f9e195e109e0?s=60&d=mm&r=g

Joe Miller

[September 4, 2015 at 10:25 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307347) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307347#respond)

Thank you so much for this reply! I have been hung up on this exact example since I read it in eloquent JS and the explanation is SIMPLE, I just never got that the outer function returned the inner one. Thank you sexyJS.

1. http://0.gravatar.com/avatar/3e71951c218411d03adb514a4512d3e6?s=60&d=mm&r=g

Raghavendra Karteek

[September 13, 2015 at 3:15 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307374) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307374#respond)

Please provide angularjs examples also

1. http://2.gravatar.com/avatar/2443f7237f10215ac5c5c1f06ef79dd6?s=60&d=mm&r=g

Sree

[September 22, 2015 at 1:26 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307388) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307388#respond)

Really a very good website. Thanks a lot

1. http://1.gravatar.com/avatar/46040c38d1cbe8ffcd3df6c8ba787951?s=60&d=mm&r=g

Ozzy

[September 25, 2015 at 2:42 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307399) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307399#respond)

This is by far the best and simplest explanation of closures i have read. Simple and to the point. Awesome job.

1. http://2.gravatar.com/avatar/866ccdffd1d1bfd1fbf47e10fcf0655e?s=60&d=mm&r=g

Ashwani

[September 27, 2015 at 12:41 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307401) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307401#respond)

This is incredibly helpful.. found about this website couple of days ago and couldn’t get my eyes of it.. Thanks for the simplicity while explaining such complex concepts.

1. http://0.gravatar.com/avatar/357a20e8c56e69d6f9734d23ef9517e8?s=60&d=mm&r=g

deejsis

[September 27, 2015 at 11:33 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307405) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307405#respond)

But then you have changed the call itself.  
Before: var stalloneID = createIdForActionCelebs [0];  console.log(stalloneID.id()); // 103  
After your workaround with the IFFE: var stalloneID = createIdForActionCelebs [0];  console.log(stalloneID.id); // 100​  
Notice you are not longer calling id() – you are simply getting the value back.

Is it possible to still keep the original call ie stalloneID.id() and still get the correct result (100) ?

1. http://1.gravatar.com/avatar/dd63408f2e92f6c17c1e48ea4e1538f7?s=60&d=mm&r=g

Vinod

[October 5, 2015 at 8:23 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307425) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307425#respond)

Great Post!  
I have already used Immediately Invoked Function Expression (IIFE) but I don’t know the actual use of IIFE.  
Also I understood closures first time by the examples u explained.

1. http://0.gravatar.com/avatar/30c79e100a9b8995374ce2f1425b2ba8?s=60&d=mm&r=g

Jatin Bansal

[October 14, 2015 at 7:34 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307440) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307440#respond)

Some days before, I’d read about “Scope Chain” in detail. But now I was not able to find that page on this site. I urgently need to read that. Plz somebody can help me.

1. http://1.gravatar.com/avatar/a61ea94618bd372e51725d200d5ba75e?s=60&d=mm&r=g

[raj](http://www.gms-ts.com/)

[October 19, 2015 at 3:19 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307446) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307446#respond)

First of all, thanks for this article. I greatly appreciate the time you took to work on the article.

1. http://1.gravatar.com/avatar/d3d1106e58060c511ca1a69cf1a3bb4f?s=60&d=mm&r=g

gero

[October 30, 2015 at 4:51 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307492) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307492#respond)

Hi! Many thanks for this!  
gero (from Paris)

1. http://2.gravatar.com/avatar/2da71eb980a71e9b68b525dc6cb28405?s=60&d=mm&r=g

[Johnny Chan](http://mathalope.co.uk/)

[October 30, 2015 at 9:38 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307493) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307493#respond)

I was also wondering how that `x = function(j) {…} (i}` part work. This was a new type of syntax to me.

So I came up with two “equivalent” examples that have helped “convincing” me that this syntax works:

Example 1:

“`  
var i = 1;  
console.log( function (j) {return j+100} (i) ) ;  
// return 101  
“`

Example 2:

“`  
function testIt(j) {return j + 100};  
var i = 1;  
console.log( testIt(i) );  
// returns 101 also  
“`

i.e. the syntax…

`testIt(i)`

is equivalent to…

`function (j) {return j+100} (i)`

I’m still a newbie and I hope this makes sense? 

Johnny  
@jAtlas7

1. http://1.gravatar.com/avatar/d3d1106e58060c511ca1a69cf1a3bb4f?s=60&d=mm&r=g

gero

[October 30, 2015 at 10:13 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307494) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307494#respond)

Hi again,  
Is there any difference between the IIFE solution:  
– theCelebrities[i][“id”] = function (j) { return function () {return uniqueID + j;} () } (i);

and a simpler Immediately Invoqued Function:  
– theCelebrities[i][“id”] = return function () {return uniqueID + j;} () ;

or a basic:  
– theCelebrities[i][“id”] = function (j) { return function () {return uniqueID + j;} () } (i);

Any comments?

1. http://0.gravatar.com/avatar/ffbde0a2bd83dd605b42147cf6059a25?s=60&d=mm&r=g

Imbrod

[November 6, 2015 at 3:40 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307507) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307507#respond)

I agree. And probably DeathShadow (that I agree in most posts) from Digital Point Forums would too.

1. http://2.gravatar.com/avatar/27acbf2ee93e450bcd7e50cef769551a?s=60&d=mm&r=g

[Showket](http://www.goexalt.com/)

[November 13, 2015 at 5:52 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307522) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307522#respond)

Really amazing stuff in a simple way. This is the contribution that makes javascript really Sexy. hats off to the stuff writer.

1. http://0.gravatar.com/avatar/9fe71edf9100e70f1ac1fdad778ed0a7?s=60&d=mm&r=g

gel

[November 13, 2015 at 11:28 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307524) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307524#respond)

My reuseable solution:  
var starIDMaker = function(){  
var uniqueID = 100;  
return function(arr){  
for(var i = 0; i < arr.length; i++){  
arr[i]["id"] = function(id){  
return ++uniqueID;  
}(uniqueID)  
}  
}  
}()

1. http://2.gravatar.com/avatar/bc9dd8995cb3481d6b605eceb5638d0a?s=60&d=mm&r=g

Rob

[November 18, 2015 at 11:33 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307529) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307529#respond)

The fact that a contained function may access a variable in the closure, and that variable may change before the contained function executes is not a bug as you keep saying. It is intended functionality. You just have to know how closures work, which the rest of your article demonstrates very well.

1. http://1.gravatar.com/avatar/db29dea74838047209c89e1275b3331c?s=60&d=mm&r=g

treefish

[November 19, 2015 at 11:49 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307546) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307546#respond)

? Your first and last approaches are identical.

1. http://0.gravatar.com/avatar/f5186cfae9f253b648117177d14c5dd7?s=60&d=mm&r=g

Rahul

[November 22, 2015 at 4:17 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307553) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307553#respond)

Hello, I’ve just started learning Javascript. Your tutorial is so good that it clears my all doubts regarding closures. Thank you so much for this article.

1. http://1.gravatar.com/avatar/1e8dd52c4bdb2c89c47d8a1b9858a910?s=60&d=mm&r=g

Sravani

[December 4, 2015 at 7:43 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307630) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307630#respond)

Hi,

Excellent Stuff.Very helpful in understanding .

I am pretty much confused with the “Closures gone awry” and IIFE .Would you please explain me further or let me know the reference books to understand clearly.

Thanks and Regards,  
Sravani.

1. http://1.gravatar.com/avatar/4b01bbfc1613ed005adb2c20d3b371c6?s=60&d=mm&r=g

firos

[December 14, 2015 at 2:52 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307714) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307714#respond)

Thanks very much. I am trying for a long time to understand closures concept but always end up with some confusion. Best example. Well explained.

1. http://0.gravatar.com/avatar/986187a10ff8ee759a1711f31fb119e2?s=60&d=mm&r=g

Tay

[December 21, 2015 at 12:05 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307748) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307748#respond)

Thanks for that

1. http://0.gravatar.com/avatar/cd634cef2ad851fbce0763cc751ff7cd?s=60&d=mm&r=g

Aurel

[December 26, 2015 at 3:12 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307757) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307757#respond)

In the first points: there is a sintax error  
mjName (“Jackson”);  
Should be:  
mjName.lastName (“Jackson”);

Is this correct?

1. http://1.gravatar.com/avatar/745937040100a3f2aee71cdf6dea6af3?s=60&d=mm&r=g

Hari

[January 4, 2016 at 8:59 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307768) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307768#respond)

Very nice post, Thanks for posting. I bookmarked this link in my favourites.

1. http://0.gravatar.com/avatar/3b161dee739100785c461e7a8184ce94?s=60&d=mm&r=g

Parish

[January 13, 2016 at 1:55 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307788) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307788#respond)

In the last code example –

theCelebrities[i][“id”] = function (j) {  
return function () {  
return uniqueID + j;  
}()  
}(i);

why i need – return function() { return uniqueID + j; } when the code below is working fine.

theCelebrities[i][“id”] = function (j) {  
return uniqueID + j;  
}(i);

1. http://0.gravatar.com/avatar/c02c94fad306f01c3bfb03b840003bca?s=60&d=mm&r=g

[Çağlar ORHAN](http://co-scripts.com/)

[January 16, 2016 at 3:49 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307802) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307802#respond)

You should write a book about js, if you didn’t write one yet. Clearly explained, keeped light to the dark corners which beginners couldn’t see (also afraid of). You are awsome.  
Thank you so much.

1. http://2.gravatar.com/avatar/28d9bd305f4e1b712076fd2df6071ab1?s=60&d=mm&r=g

[Marlo Dela Torre](http://www.facebook.com/marloeleven)

[January 27, 2016 at 7:53 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307830) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307830#respond)

such a great tutorial.. bookmarked your site and reading it when I have spare time.. thanks!

1. http://1.gravatar.com/avatar/d33a3d8bb5857af1c186f801e9125d36?s=60&d=mm&r=g

ugison

[February 2, 2016 at 6:16 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307843) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307843#respond)

Wonderful solution, this IIFE! Love it..

1. http://1.gravatar.com/avatar/db19a5bb3e4ea8f7848fb7d6b00bc7ad?s=60&d=mm&r=g

mwas

[February 6, 2016 at 1:50 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307858) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307858#respond)

please help solve a tabular array to change the cars in sequence ona platform ‘demo’ when this displays make the image hover on the other cells.thanx var cars=[]; cars[0]= ida; cars up to INDEX[11] and their ids. when i use a closer it only shows the last image bt dont loop

1. http://2.gravatar.com/avatar/b86d8d7af00dc940779c29ac2ab999f7?s=60&d=mm&r=g

[Arifur Rahman](http://n/A)

[February 9, 2016 at 11:53 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307861) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307861#respond)

To solve “3. Closures gone awry” we can just change line #6,7, &8. No need to write and return another anonymous function. instead of that we can just do this,

line:6. theCelebrities[i][“id”] = function (j) {  
line:7. return uniqueID + j;  
line:8. }(i)

1. http://0.gravatar.com/avatar/c598f350a7fc3bbe25199f953734cb08?s=60&d=mm&r=g

Charles Robertson

[February 10, 2016 at 6:33 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307866) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307866#respond)

This is quite simply one of the finest JavaScript tutorials out there. In fact, I was asked about JavaScript closures in a phone interview today. I wish I had read this first. Doh! Can I ask what a parametric variable is?

= function (j) {

1. http://2.gravatar.com/avatar/2eb6f17160cc2bcdfaff0c1fadaff487?s=60&d=mm&r=g

Ray

[February 13, 2016 at 7:04 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307877) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307877#respond)

Thanks for a nicely laid out explanation. Years old and still helpful.  
Try this:

function celebrityName (celebObj) {  
var nameIntro = nameIntro?”The other celeb is “:”This celebrity is “;  
var celebObj2 = {  
name:celebObj.name  
};

setTimeout((function lastName () {  
console.log(nameIntro + this.name);  
}).bind(celebObj),1000);

setTimeout((function lastName () {  
console.log(“Again, ” + this.name);  
}).bind(celebObj2),1100);

setTimeout((function lastName (celeb2) {  
console.log(“All 4 objects: ” + this.name + ” ” + celeb2.name +  
” ” + celebObj.name + ” ” + celebObj2.name);  
}).bind(celebObj,celebObj2),1200);  
}  
celebrityName ({name:”Michael”});  
celebrityName ({name:”Peter”});

The result was as expected. The javascript engine could make copies of the functions and objects to maintain the scope. I’m wondering if persistent scope is implemented by referencing the objects or if copies are made containing only properties that have been used in declarations yet to be executed; and if it makes a difference to how callbacks are handled. It seems that it probably doesn’t matter; the scope remains as though the outer function is still running. It starts to get confusing in things like angular.js

1. http://1.gravatar.com/avatar/a71fa1a9d1b2f0cb544f3661c689ed16?s=60&d=mm&r=g

[Bright Nkrumah](http://brightnkrumah.com/)

[February 15, 2016 at 1:35 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307882) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307882#respond)

Very nice tutorial

1. http://0.gravatar.com/avatar/3c13508059c94a7614e50563765ebfe7?s=60&d=mm&r=g

Anmol Dani

[February 16, 2016 at 3:29 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307884) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307884#respond)

Your tutorials are indeed sexy! 😀

1. http://0.gravatar.com/avatar/0f34824bacf56e78dcd8b12958009ad3?s=60&d=mm&r=g

[Jhony Grillet](https://jotaeseymas.wordpress.com/)

[February 26, 2016 at 7:33 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307916) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307916#respond)

Definitively, i must write a (lot of) post(s) about this page… OMG, with this kind of posts, Javascript seems so SEXY!!!

1. http://2.gravatar.com/avatar/5309b7e401d4d4127e3039a5aca89eed?s=60&d=mm&r=g

Anusha

[February 29, 2016 at 7:34 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307918) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307918#respond)

Thanks for the post. I am not clear with this line  
“Note that the inner function cannot call the outer function’s arguments object, however, even though it can call the outer function’s parameters directly.”

1. http://0.gravatar.com/avatar/c7c2afeb3a72dcf91c7147203c305c72?s=60&d=mm&r=g

[Roshni](http://www.roshnidevadiga.net/)

[March 4, 2016 at 5:01 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307927) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307927#respond)

Great post!

1. http://2.gravatar.com/avatar/be428d01d080fb9908a1787704fdf567?s=60&d=mm&r=g

sheik

[March 10, 2016 at 4:35 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307940) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307940#respond)

Its very useful adnd good understandable explanation.

1. http://0.gravatar.com/avatar/0ac7011e5a274161ecac6f9ab6dc66ac?s=60&d=mm&r=g

Bah Djibril

[March 16, 2016 at 12:07 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307955) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307955#respond)

I know namy already replied but here is what I did. (mjName is still a function that has to be executed)

var fullName = mjName(“Miller”);

$(‘#name’).html(fullName);

1. http://1.gravatar.com/avatar/785cb107f94f466326413751413b2f84?s=60&d=mm&r=g

Harish

[March 16, 2016 at 6:41 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307957) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307957#respond)

when i cpoy and paste code after that i have always got illegal character error , why it is ??

1. http://1.gravatar.com/avatar/decd41bd417805a299cb041f88eaea81?s=60&d=mm&r=g

Nagaraju

[March 24, 2016 at 8:44 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307976) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307976#respond)

Very good example explained clearly .. Thanks for the article.

1. http://1.gravatar.com/avatar/decd41bd417805a299cb041f88eaea81?s=60&d=mm&r=g

Nagaraju

[March 24, 2016 at 8:45 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307977) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307977#respond)

Very good examples explained clearly .. Thanks for the article.

1. http://1.gravatar.com/avatar/decd41bd417805a299cb041f88eaea81?s=60&d=mm&r=g

Nagaraju

[March 24, 2016 at 9:14 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307978) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307978#respond)

I really did have confusions on javascript closure .. this is awesome and supreb .. very good and clear explanation …I really fell in love with closure and javascript (sexy script).

But i tried some other approach that also works, please let me know is this correct and also rectify if i am wrong with the approach.

function celebrityIdCreator(celebList){  
var uniqueID =100;  
return function(){// returning inner function  
for(var i=0;i<celebList.length;i++){  
celebList[i]["id"] = uniqueID +i;

}

return celebList; // updated celebList of the inner function returned.  
}

}

var actionCelebList = [{name:'Arnold',id:0},{name:'The Rock',id:0},{name:'Stallon',id:0}];  
var celebIdFn = celebrityIdCreator(actionCelebList); // celebIdFn has inner function (closure)..  
newActionCelebList= celebIdFn(); // By calling celebIdFn .. we get the updated array ..  
var theRock = newActionCelebList[1];  
console.log(theRock.name+" :"+theRock.id);

1. http://0.gravatar.com/avatar/36b2974a6c878fbb92501fa7781c3aee?s=60&d=mm&r=g

Umesh

[March 29, 2016 at 6:48 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-307988) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=307988#respond)

Nice explanation……really java script is sexy 

1. http://2.gravatar.com/avatar/ea527f01fd11d5d4da1574930609fd9a?s=60&d=mm&r=g

Luke

[April 5, 2016 at 9:45 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308006) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308006#respond)

Awesome explanation Richard! When I initially tried to learn closures they seemed so intensely hard to learn. This makes it crystal clear. Thank you!

1. http://2.gravatar.com/avatar/2e41d52b2986c0c63a2f464e34b7b3f3?s=60&d=mm&r=g

[Awesome](http://no/)

[April 7, 2016 at 4:24 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308009) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308009#respond)

Still awesome in 2016

1. http://0.gravatar.com/avatar/9c10371c528bc9ebc0d67e265ebf5537?s=60&d=mm&r=g

Satish

[April 7, 2016 at 6:36 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308012) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308012#respond)

I was always wondering what is this closure, prototype and brackets at the end of functions and they all are explained neatly and in a very simple way. Thanks a lot this great article.

1. http://1.gravatar.com/avatar/434b8fef6d0434f67d1a6baf6445668d?s=60&d=mm&r=g

thanhdong

[April 9, 2016 at 4:32 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308016) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308016#respond)

thank you very much. 😀

1. http://2.gravatar.com/avatar/24c065878f3c862369a1328667023dec?s=60&d=mm&r=g

Sirisha

[April 10, 2016 at 3:27 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308018) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308018#respond)

Some real time JavaScript Closure example  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Practical closures

Let’s consider their practical implications. A closure lets you associate some data (the environment) with a function that operates on that data. This has obvious parallels to object oriented programming, where objects allow us to associate some data (the object’s properties) with one or more methods.

Consequently, you can use a closure anywhere that you might normally use an object with only a single method.

Situations where you might want to do this are particularly common on the web. Much of the code we write in web JavaScript is event-based — we define some behavior, then attach it to an event that is triggered by the user (such as a click or a keypress). Our code is generally attached as a callback: a single function which is executed in response to the event.

Here’s a practical example: suppose we wish to add some buttons to a page that adjust the text size. One way of doing this is to specify the font-size of the body element in pixels, then set the size of the other elements on the page (such as headers) using the relative em unit:  
body {  
font-family: Helvetica, Arial, sans-serif;  
font-size: 12px;  
}

h1 {  
font-size: 1.5em;  
}

h2 {  
font-size: 1.2em;  
}123456789101112

Our interactive text size buttons can change the font-size property of the body element, and the adjustments will be picked up by other elements on the page thanks to the relative units.

Here’s the JavaScript:  
function makeSizer(size) {  
return function() {  
document.body.style.fontSize = size + ‘px';  
};  
}

var size12 = makeSizer(12);  
var size14 = makeSizer(14);  
var size16 = makeSizer(16);123456789

size12, size14, and size16 are now functions which will resize the body text to 12, 14, and 16 pixels, respectively. We can attach them to buttons (in this case links) as follows:  
document.getElementById(‘size-12′).onclick = size12;  
document.getElementById(‘size-14′).onclick = size14;  
document.getElementById(‘size-16′).onclick = size16;123  
[12](http://javascriptissexy.com/understand-javascript-closures-with-ease/)  
[14](http://javascriptissexy.com/understand-javascript-closures-with-ease/)  
[16](http://javascriptissexy.com/understand-javascript-closures-with-ease/)123

1. http://0.gravatar.com/avatar/6e5e45ea9b0fbe6deedcf33cb3f084a6?s=60&d=mm&r=g

sunil

[April 21, 2016 at 3:12 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308043) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308043#respond)

Great Post. I regularly visit your website to sharpen my skill in JS. Keep up the good work!

1. http://0.gravatar.com/avatar/6f3f666d008cbb8fd41ea0d353ba2f0f?s=60&d=mm&r=g

[Vijay](http://applitter.com/)

[April 23, 2016 at 12:43 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308045) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308045#respond)

You can also refer to this blog which explains the concepts of functions and closures <http://www.applitter.com/introduction-to-javascript-functionsvariablesscopes-and-closures/>

1. http://2.gravatar.com/avatar/ecc9a84b755aaae317d9005f5eb7d6d4?s=60&d=mm&r=g

[Luong](http://blogtao.com/)

[April 26, 2016 at 10:33 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308049) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308049#respond)

This is very clear. You help me to dig deeply in JS. Thank you.

1. http://2.gravatar.com/avatar/b8726a0430944eee54739823d243e31d?s=60&d=mm&r=g

Bob Robertson

[April 30, 2016 at 12:43 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308056) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308056#respond)

Great post, I learned a lot. Thanks.

1. http://2.gravatar.com/avatar/55fc274c10abc06b0d488f0650317822?s=60&d=mm&r=g

AJ

[May 2, 2016 at 9:17 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308059) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308059#respond)

Your articles are so easy to understand and make learning advanced JS very very easy. You should start a JavaScript University. I have not seen such great articles anywhere else. Thank you very much.

1. http://2.gravatar.com/avatar/be6bb241b03442fdeffa5758a2beec1d?s=60&d=mm&r=g

Ganesan Natarajan

[October 14, 2016 at 4:49 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308368) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308368#respond)

celebrityName(“Michael”)(“Jackson”) // will return The celebrity is Michael Jackson. right?

1. http://0.gravatar.com/avatar/3ce4bff4a34fd25d2758531a6a68da3a?s=60&d=mm&r=g

[pavithran](https://technuked.wordpress.com/)

[October 27, 2016 at 6:36 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308388) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308388#respond)

The IIFE caught me by surprise. Do you have an article on this alone? I am trying to wrap my head around this concept.

1. http://1.gravatar.com/avatar/1463ff6724c7ade9885434d5d79102f2?s=60&d=mm&r=g

Prabhat

[November 13, 2016 at 10:32 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308408) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308408#respond)

Closures have access to the outer function’s variable even after the outer function returns….  
I don’t find this point correct. The lastname function is called before the the outer function returns.  
I think the sequence is:  
celebrityName function called  
lastName function called  
lastName function returns  
celebrityName function returns.. I am applying the concepts of assembly language programming.

1. http://0.gravatar.com/avatar/651dc43f74e341a0cf76c9e92960e066?s=60&d=mm&r=g

Kim

[November 17, 2017 at 5:55 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308917) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308917#respond)

HI Richard Bovell,

Your post is so great ! And I am new to JS, so could you please help me explain that when and why do we need closure function in reality? Thank you.

1. http://1.gravatar.com/avatar/df24007c4bc31cc127c0120823a8cca8?s=60&d=mm&r=g

Vignesh

[November 24, 2017 at 10:29 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308930) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308930#respond)

The best thing I learned today. I have seen this in our project and thought why they have written like this without knowing that they were Closures. Really felt like I achieved something when I learned and written some examples based on Closures. 😛  
Great work. Applause !!!

1. http://2.gravatar.com/avatar/ebf3ca390fc54d46412b025dee9b8c7e?s=60&d=mm&r=g

Hieu Nguyen

[December 13, 2017 at 2:48 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308959) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308959#respond)

Thank you so much. This is the first time I see a very profound JS knowledge and so clear explanation!

1. http://2.gravatar.com/avatar/e1d740095271137fd9d738370493c706?s=60&d=mm&r=g

Nuoya

[December 19, 2017 at 7:31 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308969) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308969#respond)

Thank you so much for this! I like your website a lot!!

1. http://1.gravatar.com/avatar/4093dee3302769ae64e412853c54ae62?s=60&d=mm&r=g

Himanshu

[December 27, 2017 at 8:54 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308977) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308977#respond)

Hi,

Is the inner function body required to be nested inside the outer function or the reference would also work?

callService(abc,pqr,239,function(data){  
anotherFunction(data,345,998); //data gets available to the function.  
anotherFunction(345,998); //data is un available to the function.  
});  
It would have worked if the anotherFunction body was nested inside instead of the reference?  
Please clarify.

Detailed version:  
<https://stackoverflow.com/questions/47991103/access-outer-function-param-without-passing-in-inner-function>

P.S Thanks for making this great site.Its a lovely easy peasy reference.

1. http://0.gravatar.com/avatar/3c49ffe0fb4ff3bb2341926d8860e736?s=60&d=mm&r=g

Ayushi

[December 27, 2017 at 2:52 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308978) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308978#respond)

thanks Richard or all your posts and this explanation also. I really love these post please write more. These are very helpful

1. http://1.gravatar.com/avatar/4093dee3302769ae64e412853c54ae62?s=60&d=mm&r=g

Hima

[December 28, 2017 at 1:06 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308979) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308979#respond)

The inner function body has to be nested inside the closure,It wont get access to the outer function entities if it is being called by name & its body is written elsewhere.  
is that right?

1. http://0.gravatar.com/avatar/080f95d069837603c5b5cdb7b2052190?s=60&d=mm&r=g

[Mulagura](https://mulagura.github.io/NpnInfra/wall)

[January 2, 2018 at 8:26 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308982) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308982#respond)

Js is sexy and your tutorials are more sexy

1. http://0.gravatar.com/avatar/019942d7ab22a3de0ee8a44836aaa057?s=60&d=mm&r=g

Mohammad

[January 8, 2018 at 2:26 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-308988) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=308988#respond)

Great article, just have a quick question (sorry if already asked)..

In your example:

function celebrityName (firstName) {  
var nameIntro = “This celebrity is “;  
// this inner function has access to the outer function’s variables, including the parameter​  
function lastName (theLastName) {  
return nameIntro + firstName + ” ” + theLastName;  
}  
return lastName;  
}  
​  
​var mjName = celebrityName (“Michael”); // At this juncture, the celebrityName outer function has returned.​  
​  
​// The closure (lastName) is called here after the outer function has returned above​  
​// Yet, the closure still has access to the outer function’s variables and parameter​  
mjName (“Jackson”); // This celebrity is Michael Jackson

I console.logged mjName and just got the function definition of lastName, why is this?

Also after mjName returns after passing “Michael”, and we call it again with “Jackson”, how does the function know which parameter to pass these arguments to? How does it “know” whether to pass each argument to firstName or theLastName? For example, why doesn’t it overwrite “Michael” with “Jackson” or mixup firstname and theLastName?

1. http://2.gravatar.com/avatar/b34ad29935df4a6b4707cd5ae22477b3?s=60&d=mm&r=g

Rahul

[January 19, 2018 at 4:30 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-309001) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=309001#respond)

It’s 2018 and still one of the best page to understand these concepts

1. http://2.gravatar.com/avatar/57975a1bdb170131b1e61cf0d8878fcf?s=60&d=mm&r=g

Cenob

[June 1, 2014 at 6:12 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-239346) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=239346#respond)

The short answer:  
remove the “()” that comes just before the comment starting in “// BY adding () at the end of this…”

By the way, the solution above can be deduced by reading the entirety of that comment, which is:  
// BY adding () at the end of this function, we are executing it immediately and returning just the value of uniqueID + j, instead of returning a function.  
Note that if you copy-pasted the IIFE code and you make the change noted above, you’ll also need to change the console.log lines to reference the ‘id’ attribute as a function “.id()” instead of as a value “.id”.

But wait, why was this question even asked (and the ones by soncu, tanyaka, and Pearce)?  
Given that the article was written to explain closures, it doesn’t make sense to return values instead of functions in the IIFE example code.

I’m just getting started on Javascript, so please check my facts here but I don’t think the code using IIFEs is a closure since no references to celebrityIDCreator’s local variables persist after the function returns, and as such there’s no need for IIFEs.

I’m pretty sure the IIFE example was originally written to return functions, but that it was changed in conjunction with the note on May 27th, 2013 where Richard wrote “I just updated the code in the article to store numbers in the returned array, instead of functions.”

\*\*Please change it back.\*\*

The change reduced the celebrityIDCreator function from a closure to a plain function, and as such removed the need for IIFEs, and also left the “closures gone awry” problem unsolved.

Thank you for writing this up Richard, it was very helpful and I’m sure it will continue to help people after me, which is why I bothered with this big post.

1. http://1.gravatar.com/avatar/4e92e8b5850a9dd38145bfdba501016a?s=60&d=mm&r=g

Shawn K.

[July 19, 2014 at 2:01 am](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-260033) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=260033#respond)

i is passed in at the end of the function like this (i). Make sure you’re reading the entire function. I also had a few minutes of not understanding how j got the value of i, but it does indeed get passed in.

1. http://1.gravatar.com/avatar/48cb4a2446034dc6443a7a40f2b00608?s=60&d=mm&r=g

Deepak

[September 5, 2014 at 11:49 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-283662) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=283662#respond)

Yes , I have also the same doubt. Why to write another IIFE inside an IIFE , as inside the first IIFE we can directly return the value.

1. http://2.gravatar.com/avatar/27fd29728b9425c26a552fc9d84e84a3?s=60&d=mm&r=g

Nick

[September 23, 2014 at 11:18 pm](http://javascriptissexy.com/understand-javascript-closures-with-ease/#comment-295103) / [Reply](http://javascriptissexy.com/understand-javascript-closures-with-ease/?replytocom=295103#respond)

I’ve done the same thing.