JAVASCRIPT: ANONYMOUS VS. NAMED FUNCTIONS

[MAY 8, 2015](http://www.whatthewebtech.com/2015/05/08/javascript-anonymous-vs-named-functions/) [ADMIN](http://www.whatthewebtech.com/author/admin/) [LEAVE A COMMENT](http://www.whatthewebtech.com/2015/05/08/javascript-anonymous-vs-named-functions/#respond)

My hunt for green October (internship for the fall time) began this week, though in all reality it should have started last week. In response to this pressing need, I turned to my old text books and began my excursion into the deep cavernous mountains of knowledge known as *scripting* languages!

Obviously it’s not that bad, otherwise I would have switched career paths a long time ago–though the $50k salary is sounding pretty good as I look around my kitchen and see the mess of five roommates who are likely as busy as I am. To draw your attention (and mine) back to the *real* subject matter at hand, as I scoured through my*JavaScript and jQuery* textbook, I discovered a topic I hadn’t ever really thought about before.

**What is the difference between an anonymous function and named function?**

Here’s an example of each:

**Anonymous Function**

var $ = function (id) {

return document.getElementById(id);

}

The above code allows the developer to use

$("[element id]")

instead of the default JavaScript

document.getElementById("[element id]")

This is similar to how developers utilize jQuery. The main difference, however, is that jQuery has other functions inside its library besides just this Quick-E-Mart short cut.

**Named Function**

function calculateTax (subtotal, taxrate) {

var tax = subtotal \* taxrate;

tax = parseFloat(tax.toFixed(2));

return tax;

}

*Murach's JavaScript and jQuery* p.102

Yes, this is a simple function. But guess what? I’m going to guide you along through the code anyways. Who knows, maybe a future employer will enjoy it, eh? \*nudge\* \*nudge\*

The function’s name is **calculateTax**. It can be called by the following:

window.confirm(calculateTax(25, 0.08));

The values 25 and 0.08 are arbitrary. When **calculateTax** gets called, the function assigns its variables *subtotal* and *taxrate* to 25 and 0.08 respectively. These values are multiplied and assigned to *tax* where it is parsed into a floating decimal value with a decimal place of 2. The result is then returned to whatever called the function. In this case, it would be displayed in a confirmation box.

**What is the advantage of using one versus the other?**

The textbook I am going through, *Murach’s JavaScript and jQuery*indicate that it’s more common to use anonymous functions rather than named functions. I would venture to disagree, but then again my experience in JavaScript is limited to three years. Meh.

The advantage of using an anonymous function is that it can be assigned to a variable name (which to me is basically the same as making a named function as well). However, using an anonymous function requires a more procedural approach to the scripting. If the function is defined above the place where the variable is called, then it will call correctly. Beware, however, of calling the anonymous function above where it is defined. It will not work. Really.

Hence, I think it is better design to use named functions wherever possible. Even if the only reason is that the developer no longer has to define the function before it’s called, that’s good enough for me.

If you’re a developer with at least some experience, I’d love to hear your thoughts on this matter.