

## Why Curly is Not Just For Yucks

Posted At : February 10, 2009 1:41 PM | Posted By : Mark Kruger

Related Categories: ColdFusion

Most Muse readers know I'm a fan of Cfscript. I often use it in the sample code as well as in the code I write and demonstrate. I use it frequently for a couple of reasons. One reason is that I have an easier time of commenting with cfscript because I can just add a couple of slashes, type a comment and hit enter. I also like cfscript for working with CFCs. For me the code ends up being more readable. As you probably know you can save a few lines with Cfscript as well. Consider the following example:

```
<cfif isDefined('url.myvar') AND isNumeric(url.myvar)>
    <cfset myvar2 = url.myvar * 2/>
</cfif>
```

Inside of a cfscript block this same code would look like this:

```
<cfscript>
    if(isDefined('url.myvar') AND isNumeric(url.myvar))
        myvar2 = url.myvar * 2;
</cfscript>
```

So a series of validations or sets can be written to conserve space and bring your code together. You might notice something else about the code above. I'm not using curly braces around my "set" statement. The same block of code *could* be written like this:

```
<cfscript>
    if(isDefined('url.myvar') AND isNumeric(url.myvar))
        {    myvar2 = url.myvar * 2;        }
</cfscript>
```

...but like most scripting languages, if you leave off the curly braces the code will look for the very next statement on it's own line and execute it as if it *were* separated by curly braces.

```
<cfscript>
    if(isDefined('url.myvar') AND isNumeric(url.myvar))
    {    myvar2 = url.myvar * 2;
        myvar3 = 5;
        myvar4 = 0;
    } else
        myvar2 = 0;
</cfscript>
```

Nice and neat right? Well there is a down side to this approach. I still like it for a big group of validations or checks with sets, but I don't like it so much when combined with a multiline IF (as above).

Let's say someone fiddling with my code above has to add additional sets in the "else" block. Let's also presume that this code is nested one level deep (as is sometimes the case). So the code looks like this:

```
<cfscript>
    if(runthiscode()) {
        if(isDefined('url.myvar') AND isNumeric(url.myvar))
            {    myvar2 = url.myvar * 2;
                myvar3 = 5;
            }
```

```

        myvar4 = 0;
    } else
        myvar2 = 0;
}
</cfscript>

```

Even an advanced programmer adding additional sets might do something like this:

```

<cfscript>
    if(runthiscode()) {
        if(isDefined('url.myvar') AND isNumeric(url.myvar))
            {
                myvar2 = url.myvar * 2;
                myvar3 = 5;
                myvar4 = 0;
            } else
                myvar2 = 0;
                myvar3 = 55;
                myvar4 = '';
    }
</cfscript>

```

And you know - it actually looks nearly right. But ask your self what will happen if "url.myvar" exists and is a number? What will be the outcome? It would be:

myvar2 equals 0

myvar3 equals 55

myvar4 equals '' (blank)

You see what happens? The only statement that is "skipped" in the code above would be the statement immediately underneath the "else". The other 2 statements actually exist outside of the else - even though they look pretty cozy. The proper statement should look like this:

```

<cfscript>
    if(runthiscode()) {
        if(isDefined('url.myvar') AND isNumeric(url.myvar))
            {
                myvar2 = url.myvar * 2;
                myvar3 = 5;
                myvar4 = 0;
            } else {
                myvar2 = 0;
                myvar3 = 55;
                myvar4 = '';
            }
    }
</cfscript>

```

If the programmer had included curly braces around the original "else" part of the statement then the modification would have worked right out of the gate. As it is, the programmer modifying the code probably spent 15 minutes banging his forehead against the desk crying "Why is myvar3 55??" So the muse rule of thumb is, if you are in an IF/Else block in CFSCRIPT and either of the blocks (the IF or the ELSE) is more than one line - make sure and use curly braces for both of them. You will save yourself and your fellow programmers some grief.