Removing the IF in the code

Posted At : June 28, 2018 6:51 PM | Posted By : Scott Rogers
Related Categories: Coldfusion, Development

Yesterday I came upon this interesting read on the Anti-If pattern. It has some good tips on how to remove if-statements from your code. As I primarily work with ColdFusion, I thought it would be good to document a trick I learned to eliminate if-statements.

Does this (quite contrived) pattern look familiar?

```
<cfif timeOfDay eq "am">
     <cfset welcomeMessage = "Good morning!"/>
<cfelseif timeOfDay eq "pm">
     <cfset welcomeMessage = "Good afternoon!"/>
 <cfelse timeOfDay eq "night">
     <cfset welcomeMessage = "Good evening!"/>
</cfif>

<cfoutput>#welcomeMessage#</cfoutput>
```

It's pretty straightforward. But the multi-tier if-statement isn't necessary. Look at this:

```
<cfset welcomeMessages = {
    'am' = "Good morning!",
    'pm' = "Good afternoon!",
    'night' = "Good evening!"
}/>

<cfoutput>#welcomeMessages[timeOfDay]#</cfoutput>
```

There are fewer lines of code to maintain and no decisions to be made. Instead, we just output the welcomeMessage value that corresponds to the value in timeOfDay.

There are two "gotchas" to be aware of: missing values and boolean values.

Missing Values

If, in this example, timeOfDay was set to an empty string, we have no option for that and as is, an error would be thrown.

If the situation is such that all possible values can't be represented there is an option using a single if-statement:

Add a new struct key: 'default' = "Good day!"

```
<cfoutput> #structKeyExists( welcomeMessages, timeOfDay ) ? welcomeMessages[timeOfDay] : welcomeMessages.default#</cfoutput>
```

This ternary-condition checks to see if a key value exists that correlates to the value of "timeOfDay". If it does, use it. If not, use the value in the "default" key.

This approach is really handy when you have more than 2 values to compare against.

Boolean values

Have you ever had a form element that should have a class in certain circumstances and a different class otherwise? What did you do? Add a cfif inside the input element? Yuck.

What if you could do this:

```
<cfset useActiveClass= {
  'yes' = "active",
  'no' = ""
}/>

<input type="text" name="name" class="#useActiveClass[local.isActive]#" value=""/>
```

If local.isActive evalutes to "yes", then the class "active" is used. If not, no class is added. This is now super-generic and can be used on any input where you need to make this decision. Pretty cool, huh?

But this exposes the other gotcha I've run into: how ColdFusion evaluates a boolean value. true, yes, and any non-zero number can all be true while false, no and zero are processed as false. What if local.isActive is stored as a true boolean value of "true"? We run into the missing value issue.

Before we just added the missing values and all was well. But it's a bit ridiculous to have to include 3 values for true and 3 for false each time we want to evaluate "truthiness". Fear not, the solution is super simple. Wrap the value in yesNoFormat():

```
<input type="text" name="name" class="#useActiveClass[yesNoFormat(local.isActive)]#" value=""/>
```

This will convert the boolean value,whatever it is, to "yes" or "no" and allows it to play nicely with our simple struct. I think you would agree this pattern is a simple way to remove if-statements through your methods and views reducing decisions and making code easier to read.

Leave a comment if you have other suggestions on how to minimize the use if-statements in your code.