





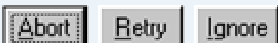





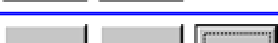


## makeMsgBox()

Our first function will be used to create a VBScript Message Box, and will be called by JavaScript in place of `alert()` and `confirm()`:

```
Function makeMsgBox(title,mess,icon,buts,defbut,mods)
    butVal = buts + (icon*16) + (defbut*256) + (mods*4096)
    makeMsgBox = MsgBox(mess,butVal,title)
End Function
```

Argument	Value	Display	The <code>makeMsgBox()</code> function takes six arguments:	
icon	0	No icon displayed	1.	<code>title</code> - a string to be displayed in the dialog title bar
	1		2.	<code>mess</code> - a string to be displayed as the message to the user
	2		3.	<code>icon</code> - an integer between 0 and 4, inclusive, denoting the icon to be displayed (refer to table)
	3		4.	<code>buts</code> - an integer between 0 and 5, inclusive, denoting the button group to be used (refer to table)
	4		5.	<code>defbut</code> - an integer between 0 and 2, inclusive, denoting the default button (refer to table)
buts	0		6.	<code>mods</code> - an integer, either 0 or 1, denoting the modality of the dialog (refer to table)
	1		When <code>makeMsgBox()</code> is called, it must first calculate the true value of the third to sixth arguments ( <code>icon,buts,defbut,mods</code> ) and then add them up:	
	2		<code>butVal = buts + (icon*16) + (defbut*256) + (mods*4096)</code>	
	3		The <code>butVal</code> variable should now store the correct integer for use as the <code>buttons</code> (second) argument of <code>MsgBox()</code> . So, we call <code>MsgBox()</code> , and account for a return value:	
	4		<code>makeMsgBox = MsgBox(mess,butVal,title)</code>	
	5		Since JavaScript can call <code>makeMsgBox()</code> , we have created a function to produce VBScript message dialogs from JavaScript.	
defbut	0			
	1			
	2			
mods	0	Application Modal		
	1	System Modal		