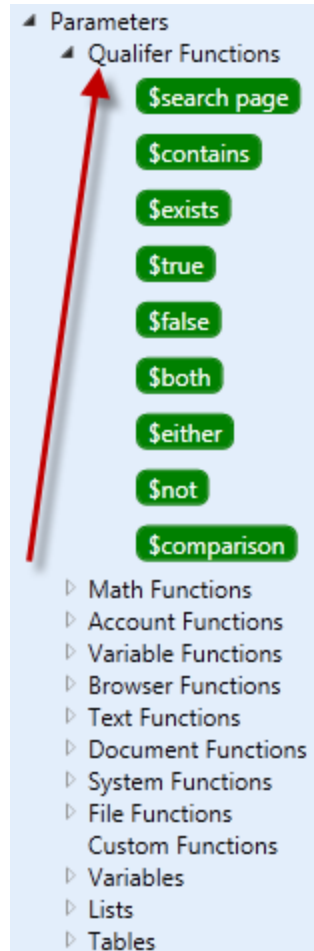


# **The UBot Studio**

## **SCRIPT REFERENCE**

### **The Qualifier Functions**

This section of Functions has not changed much from the earlier v3.5 qualifier functions, however, there are a couple of notable changes that will be detailed here. The Qualifier Functions can be located in the sub-menu “Qualifier Functions” directly under the main “Parameters” menu:



# SEARCH PAGE

This function returns whether or not the specified text is present on the current page. This function is used inside a conditional statement such as if/then. The following is a simple example. The script will look for the text "Sign in" on the Google page. If it locates it, it will click the Sign in link:

Untitled Script

navigate

URL

http://www.google.com/

Advanced

if

Condition To Run Commands:

Search page

Query

Sign in

then

click

Element

<class=gbgt>

Advanced

Drag Commands Here

else

Drag Commands Here

Cancel OK

http://www.google.com/

+You Web Images Videos Maps News Gmail more - Sign in

Google

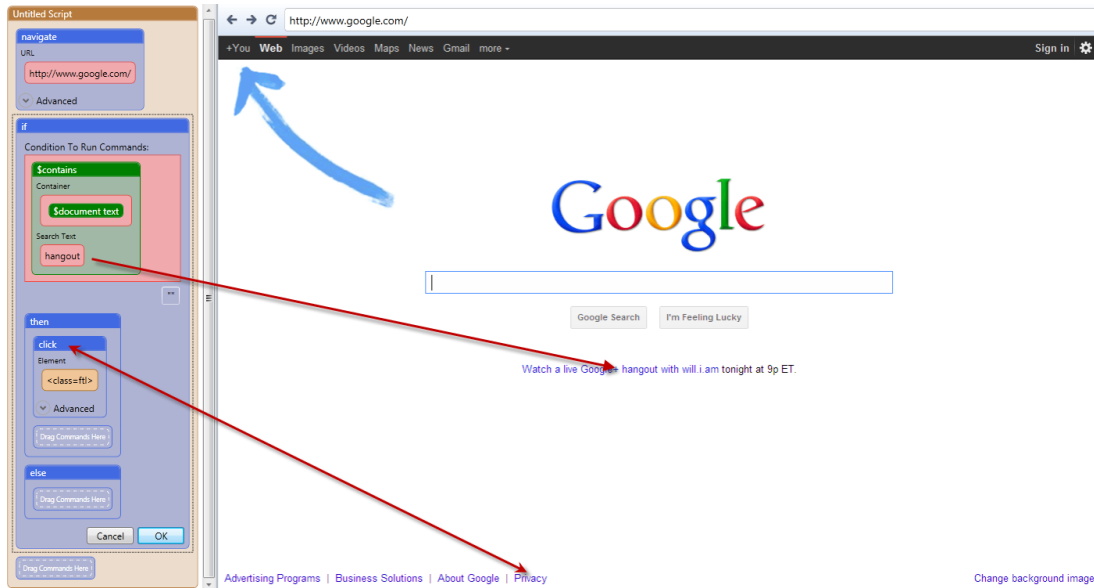
Google Search I'm Feeling Lucky

Watch a live Google+ hangout with will.i.am tonight at 9p ET.

If the text "Sign in" is located, it will then click the link to sign in.

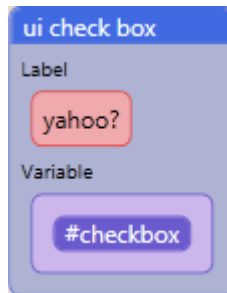
# CONTAINS

This functions returns whether or not the text contains the search text. The following example is similar to the previous example (search page). This will determine whether the document text contains the search text and then click the privacy link if it does:

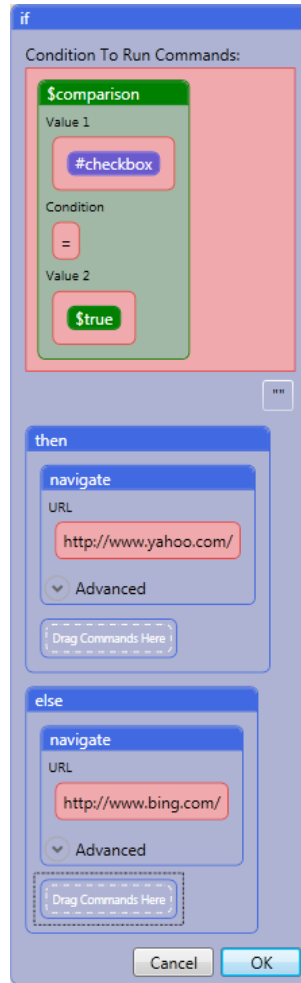


## TRUE/ FALSE

These functions return true and false respectively. They are useful in several situations, but we will demonstrate just one here. We will add a checkbox to the UI and call it #checkbox:



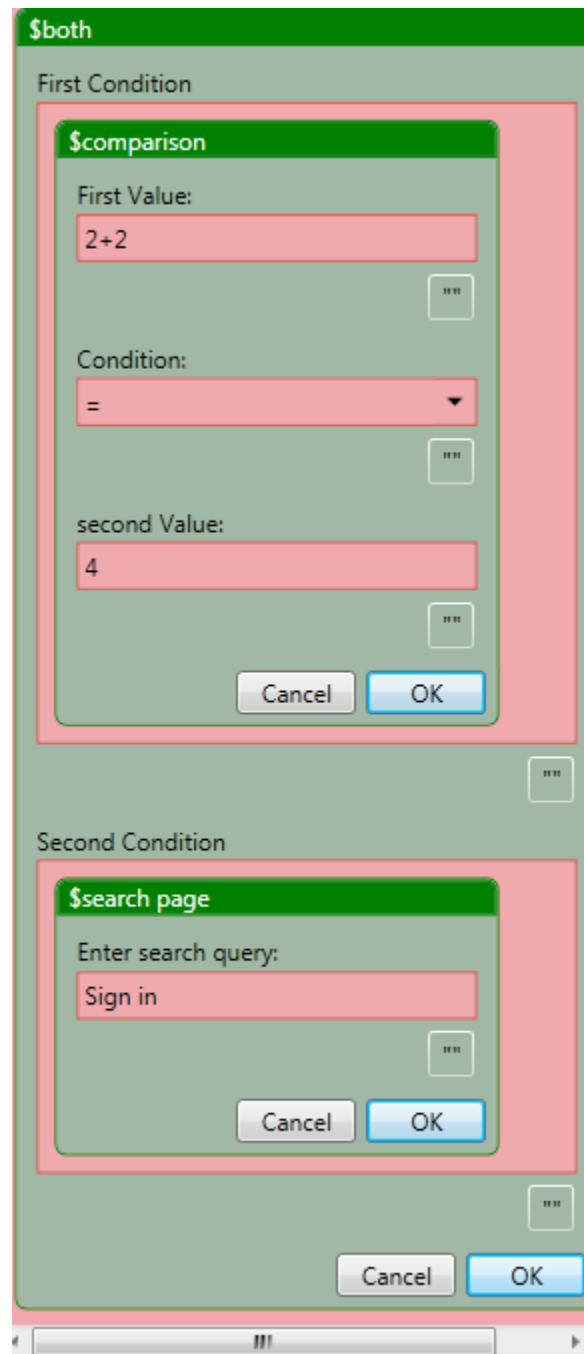
The conditional statement will look as follows:



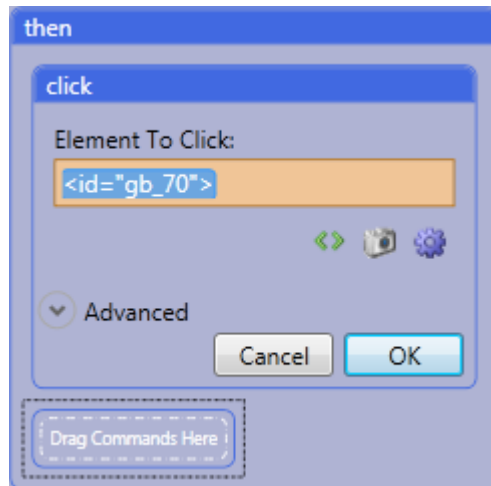
If the checkbox is checked the script will navigate to yahoo.com. If it's not checked it will navigate to bing.com. This example used true, but you can replace true with false and do the same thing. If #checkbox = false, then navigate to bing.com, etc.

## BOTH

This function returns true if both conditions return true. In this case you are setting two conditions that must both return true in order to execute the code inside the Then node. If they do not, the Else node will execute. In this example we are stating that if  $2+2=4$  AND "sign in" is located on the web page, Then navigate to the log in page:



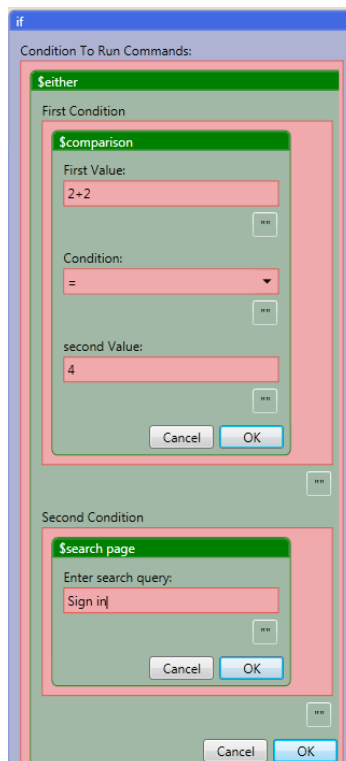
Click OK to close the nodes and add your navigate command to the Then node:



That's it, your command is finished.

## EITHER

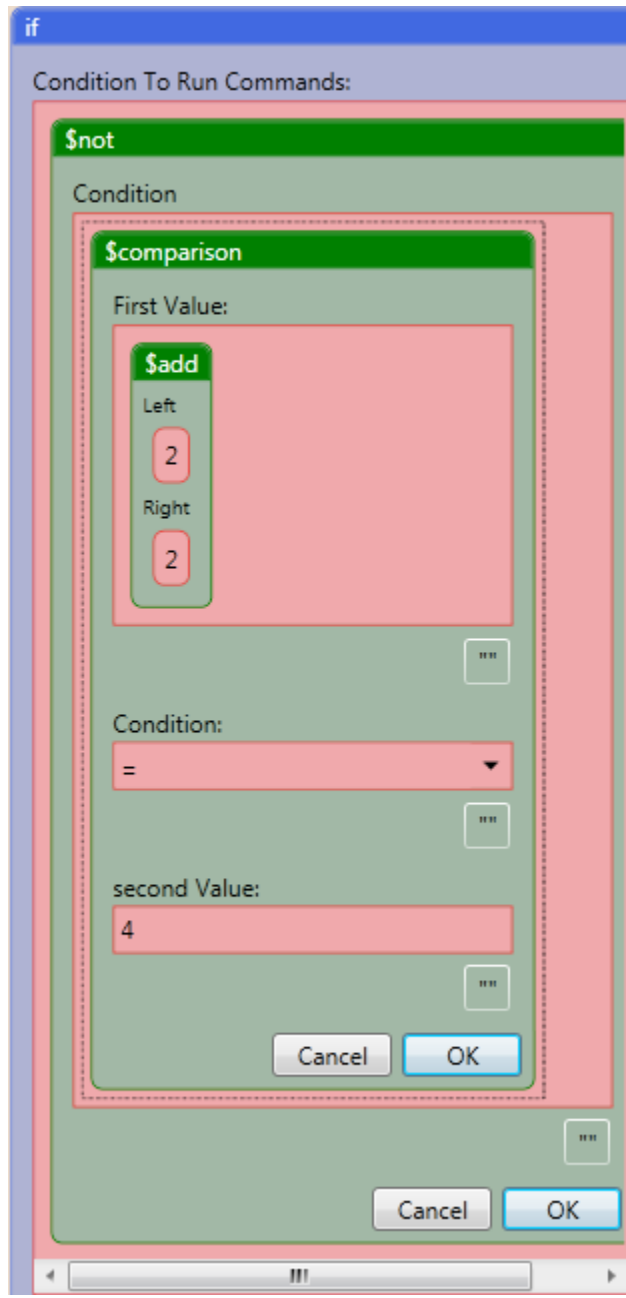
This function returns true if either of the two conditions are true. So using the example above, we can see that instead of both conditions being true, we are only looking for one of them to be true:



## **NOT**

This function returns true if the condition is not true. This condition states, "If  $2+2$  does not  $= 4$  then do something:





## EXISTS

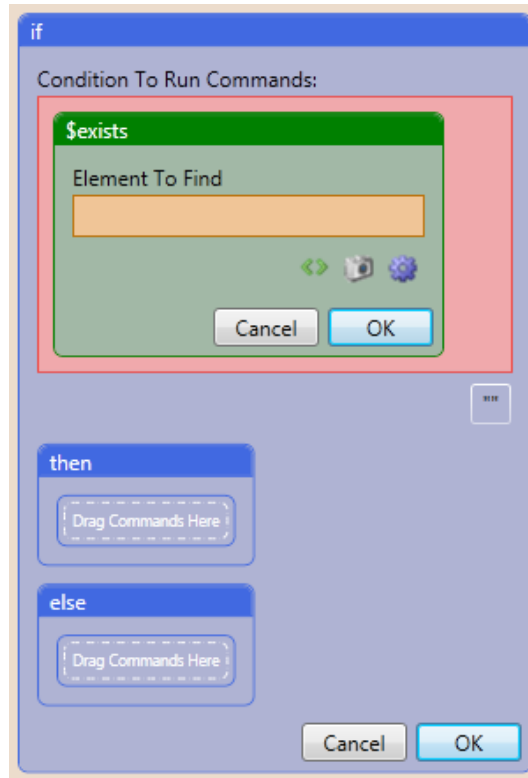
This function returns whether or not an element exists on the current page. A common use for this function is to use it within a conditional statement to help


determine the next set of commands that will execute. The verbal application would be something like: “If something exists, then do something. Else do something different”.

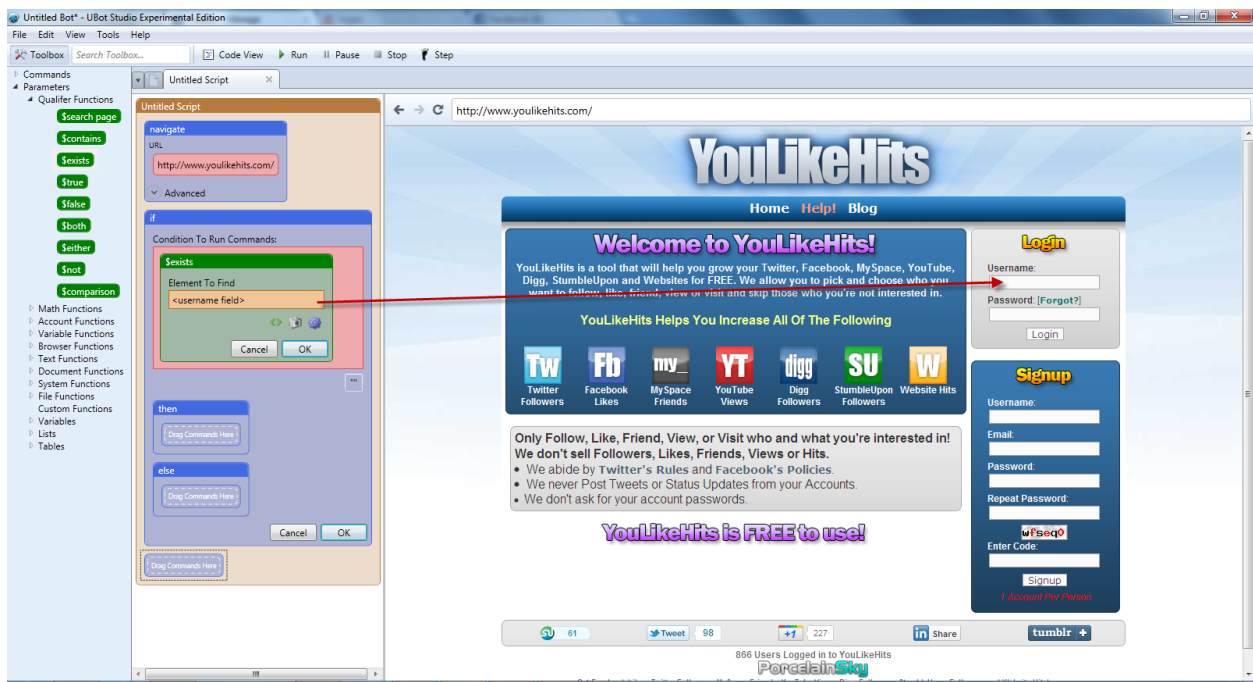
The following example will show the practical application within a conditional statement. We will begin by navigating to youlikehits.com



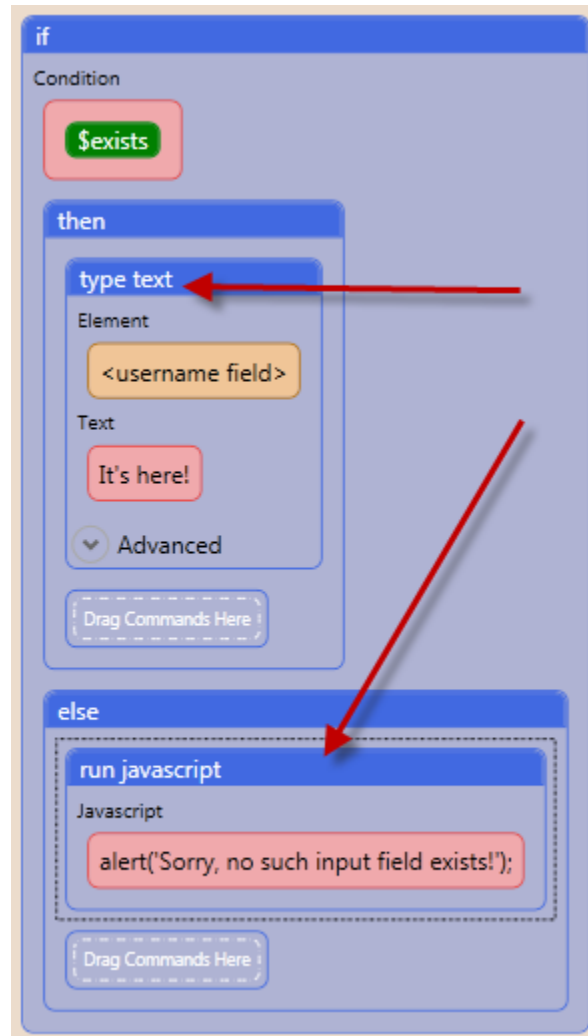
The first step will be to drag an if statement into the scripting area (located in the “Flow Commands” menu), and then dragging an \$exists function into the Condition box of the if statement:



From here you will use the element selector (  ) to choose an element on the page to satisfy the condition. In this example I will choose the “Username” input box in the upper right corner of the page:



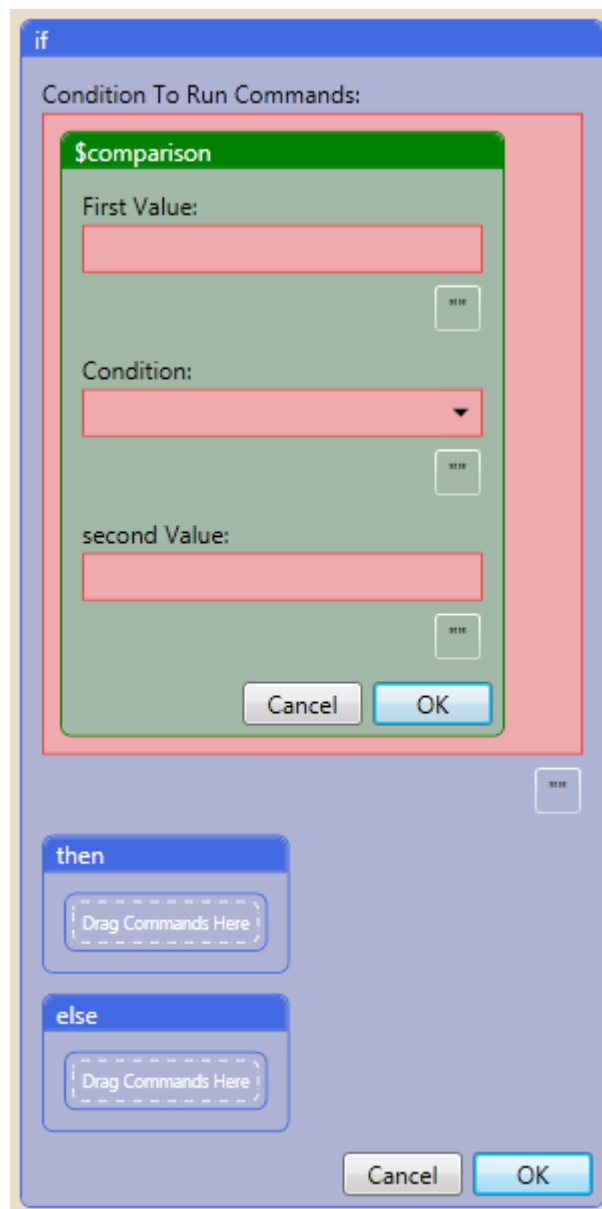
We will then finish the conditional with some actions to take given the results of the condition. So, in this case, if the condition returns true (The element exists) we will tell it to fill it with some text. If it is not true (there is no element present), we will have it display an alert advising us there is no such element. The commands will look like this:



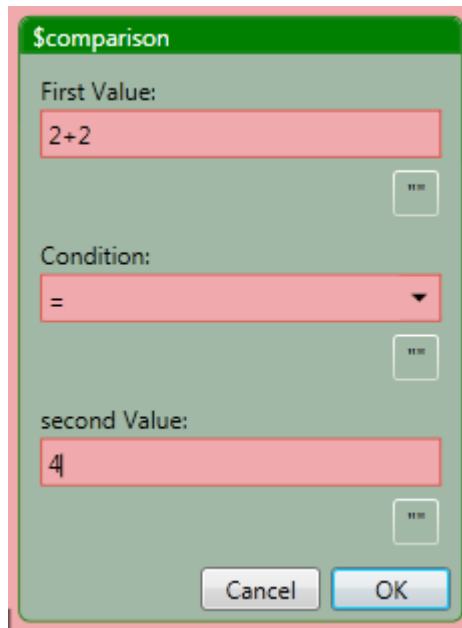
Make sure you close all nodes (click "OK") to ensure they are saved as desired and press play. Your element will fill with the text you provided. Try the same code on a different page to see how it will not work, and consequently display the alert box.

# Comparison

This function compares two values and returns the results as either being true or false. This function is also commonly used in a conditional statement to help determine which commands will follow the condition. As in the previous example we will drag an if statement from the flow commands menu into the scripting area followed by a \$comparison function in the conditional box:

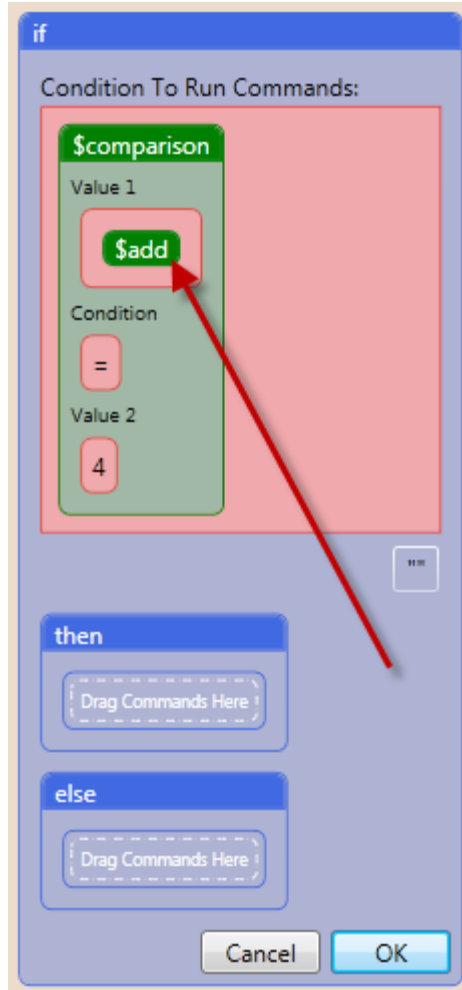


You are asked to provide a first value, a condition, and a second value. For the purpose of this example we will use a simple math statement to compare. In the first value we will enter “2+2”. In the condition we will choose “=” from the dropdown. And the second value will be “4”:



The image shows a dialog box titled "\$comparison" with a green header. It contains three input fields, each with a small "..." button to its right. The first field is labeled "First Value:" and contains the text "2+2". The second field is labeled "Condition:" and is a dropdown menu showing the equals sign "=" with a downward arrow. The third field is labeled "second Value:" and contains the text "4". At the bottom of the dialog are two buttons: "Cancel" and "OK".

When you press ok, the node will look like this:



To finish this example we will just add a navigate to a new page (google.com). So in effect, you are coding the following statement: “If  $2+2=4$ , then navigate to google.com”



To test your code, change the second value of the \$comparison function (i.e.  $2+2=7$ ) and you will see it will not navigate to the desired site.