The following columns of a script can take a “formula” which is evaluated at runtime:

Rowswitch

Action

input value

Locator Type

Locator Text

Innertext

The script identifies that a formula is present by looking for a “$”. If a $ exists it will try and evaluate what is entered in that column as a formula.

The formula should be in javascript notation and must be valid syntax. Most examples you may need are documented in this document but you can always google it if you cannot find what you need below.

Remember any text strings should be contained within quotes. Make sure you use straight quotes not curly quotes. (The Front-End editor will use straight quotes automatically so just something to be aware of.)

So use " instead of “

So use ' instead of ‘

One of the main uses would be the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Action** | **Input Value** | **Result Name** | **What happens** |
| Navigate | FWDELUKURL | $ukurl | FWDELUKURL url is looked up in the properties file and that url is navigated to. Also that url is returned as output and the Script Variable $ukurl is populated with this output value. In this example let’s say this is:  http://uk.d100del.dev.practicallaw.com/ |
| . . . |  |  | More lines are run |
| Navigate | $ukurl + "2-500-2291" |  | Because input value contains a $, the expression entered in there is evaluated as a formula. In this example it will evaluate to:  http://uk.d100del.dev.practicallaw.com/2-500-2291.  This is because the “+” is a concatenation operator which concatenates the value of $ukurl with the literal value 2-500-2291 which has to be entered in quotes as any text values in a formula must be entered in quotes. |

# Javascript notation Examples

|  |  |
| --- | --- |
| What you might want to do: | How to do it: |
| Concatenate text strings together | Use the + operator to concatenate strings together.  If you have $myvalue1 and you want to concatenate the text “05/07/2013” onto the end of it and set the resulting value to a new script value $myvalue2, you would do the following:  action=setscriptvalue  input value=$myvalue1 + "05/07/2013"  result name = $myvalue2  So if $myvalue1 is “Tree climbing extravaganza” after the above step is performed, the value of $myvalue2 would be:  Tree climbing extravaganza 05/07/2013  You can concatenate as may text strings together as you like:  E.g. $myvalue1 + " is the " $myvalue2 + " value in the world"  Another example might be:  "//a[contains(href, '" + $myvalue + "')]"  Which if $myvalue has value “rank=2” would evaluate to:  //a[contains(href, 'rank=2')]  Which you may well need to enter into a Locator Text column so you would use the formula entered ("//a[contains(href, '" + $myvalue + "')]" ) |
| Check two text values are equal | Formula: value1.equals(value2)  This returns true if the two are equal or false if not.  Suppose you have a script variable $caseheading and you want to check it equals “Law to do with Trees”.  To do this you would do:  action=checkexpressiontrue  input value=$caseheading.equals("Law to do with Trees ") |
| Check text is contained within another text field | Formula: myvalue.search(expectedcontainedvalue)  This returns the position within the text string where the expected text string first appears (positioning starts at zero) – so will essentially return a number greater than or equal to zero if the text does contain the expected text. Otherwise returns -1.  So to check that the text of script value $myexpectedtext is contained within text of script value $myvalue do:  Action=checkexpressiontrue  Input value=$myvalue.search($myexpectedtext) >= 0  So to check that the text “trees are great” is contained within text of script value $myvalue do:  Action=checkexpressiontrue  Input value=$myvalue.search("trees are great") >= 0  Note you can also use the formula  Myvalue.indexOf(expectedcontainedvalue) |
| Find the position of particular text within another text string | Can use the search method in above example or indexOf method which has formula:  myvalue.indexOf(expectedtext)  or  myvalue.indexOf(expectedtext, position to search from)  For example if you have text string $myvalue which has value “Cats, Rabbits, Cats, Trees, Rabbits”.  If you want to get the position where Rabbits first appears and store this in $rabbitstart do:  Action=setscriptvalue  Input value= $myvalue.indexOf("Rabbits")  Result Name=$rabbitstart  $rabbitstart would be populated with “6” as this is the “position” at which rabbits first appears. Remember the numbering of positioning starts at zero instead of 1;  Cats, Rabbits  0123456  Note that using input value = $myvalue.search("Rabbits") would have the same result.  Suppose you then wanted to check to see if after the first "Rabbits" there was another Rabbits and find its position. With the “indexOf” method you can also pass in the position to search from so you could do the following:  Action=evaluatenumber  Input value=$myvalue.indexOf("Rabbits",$rabbitstart + 1)  This would output 28.  Remember you need to use “evaulatenumber” instead of “setscriptvalue” as $rabbitstart needs to be treated as a number field. |
| Check two text values match (i.e. using regular expressions) | Formula is value1.match(value2)  This returns a null object if the two do not match.  Suppose you have a script variable $caseheading and you want to check it starts with the following text “Law to do with Trees” and has text “no subscription required” somewhere after this in the text. In this case you are checking it matches the following regular expression:  "Law to do with Trees.\*no subscription required.\* "  To do this you would do:  action=checkexpressiontrue  input value=  $caseheading.match("Law to do with Trees.\*no subscription required.\* ") != null) |
| Substring – extract text from another text string. | Formula: myvalue.substring(start position, one after end position you want)  Or  Myvalue.substring(startposition)  (The latter form assumes the end position is the last position of the string.)  The “position” numbering starts at 0.  So say you have $myvalue and you want to extract the first three characters and set $myvalue to this new value you would do:  Action=setscriptvalue  Input value=$myvalue.substring(0,3)  Result Name=$myvalue  Note in the above example $myvalue would get overwritten with the new shorter value.  E.g. if at the start of the step $myvalue had value GFR200121. After the step is executed it will now have value GFR. |
| Find the length of a text string | Formula: mystring.length  This returns the length of the text string.  E.g if you have $myvalue with value “Hello”  If you entered $myvalue.length in any of the script columns which allow formulae to be entered would return the output 5. You may need this to get the last characters in a text string. For example if you wanted to get the last three characters of text string $mytextstring you would do:  Action=setscriptvalue  Inputvalue=$mytextstring($mytextstring.length – 3) |