SCRIPT REFERENCE

UBot Studio Version 4

The Text Functions

Spin

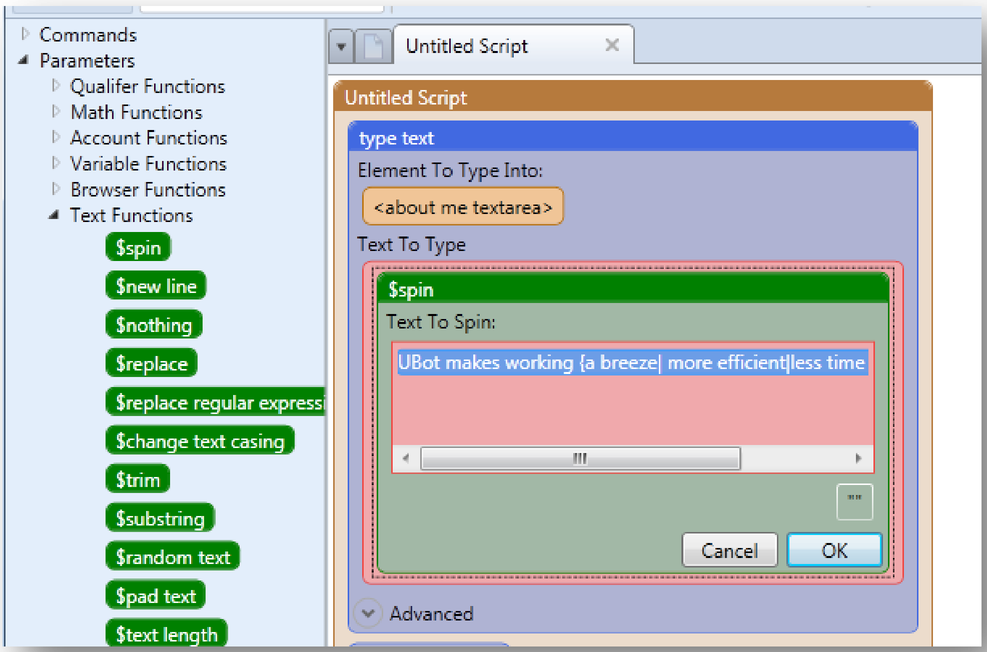
This function will spin whatever spin-ready content you place inside it. The content can be typed directly into the spin function, or the content can be called using another function like the list item functions. You can also insert a variable from a UI text box or a UI block text command to spin whatever you type into the UIs.

Spin ready sentenced look like the following sentence:

UBot makes working {a breeze| more efficient| less time consuming}

This means that we want the sentence to be finished with either “a breeze”, “more efficient”, or “less consuming”.

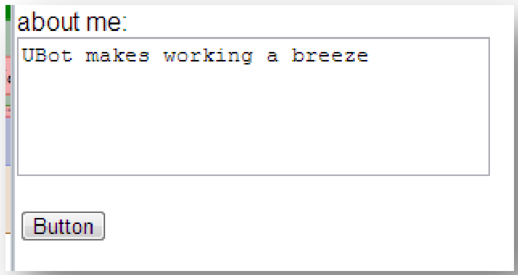
Let’s drag the field we want to fill with our spun sentence into the scripting area. After the type text command appears, go under your text functions and drag the Spin function into the type text command.



UBot makes working {a breeze| more efficient|less time

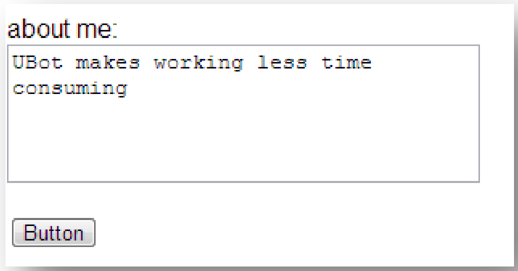
Type your spin ready content into the spin function. Click ok when you're done editing.

Each time you run your script, the field will be filled with a spun version of the sentence with a random ending.



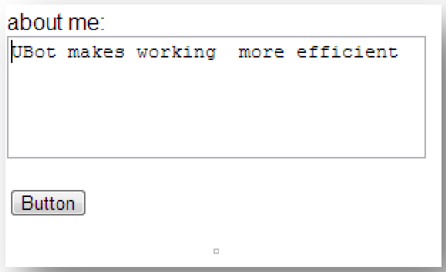
about me:

UBot makes working a breeze



about me:

UBot makes working less time consuming

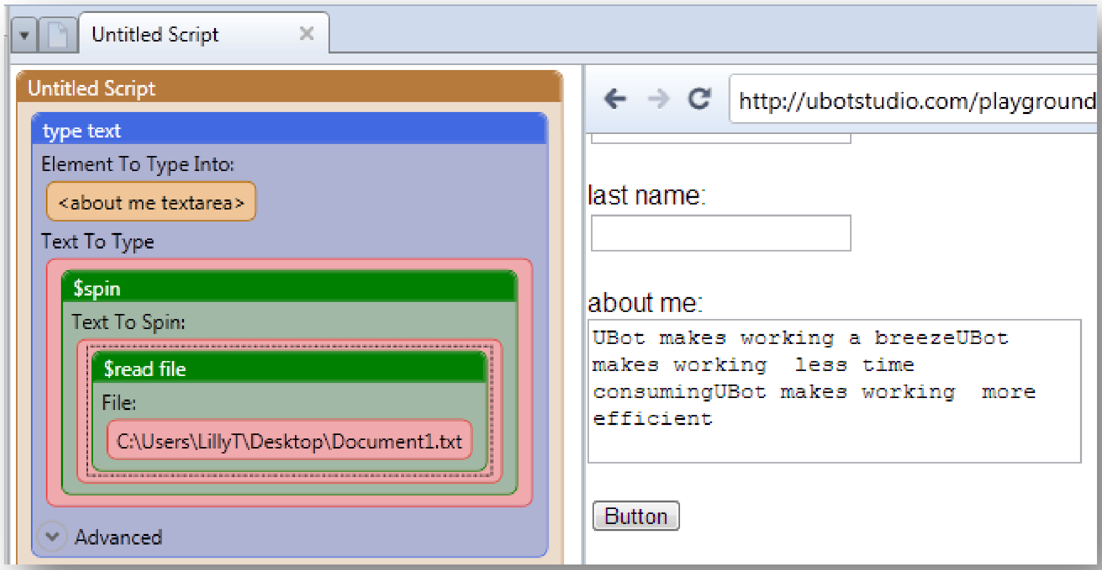


about me:

UBot makes working more efficient

To fill the same field with spin ready content from a file, simply replace the spin ready sentence in the spin function with the read file function, found under the Variable functions. Browse for the file that contains your spin ready content. Once you've found the file, click ok on all commands and functions to exit editing.

Once you run the script, the field will be filled with the spun version of the contents of the file.



```
UBot makes working a breezeUBot
makes working  less time
consumingUBot makes working  more
efficient
```

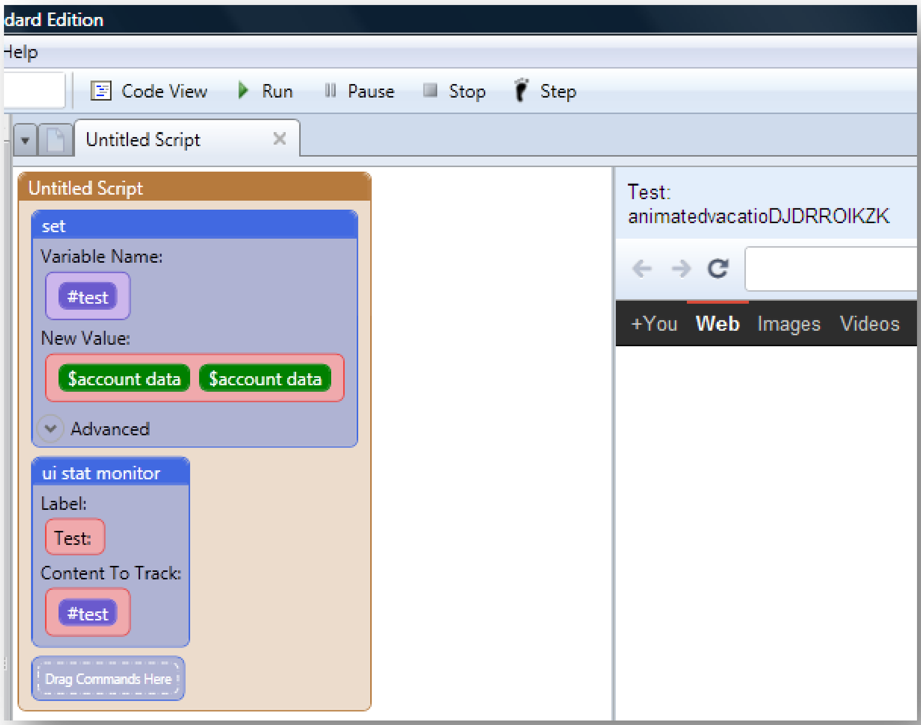
To spin content from a UI text box or a UI block text command, insert the variable for the UI into the spin function.

New Line

This function will insert a new line where ever you place it. It is akin to pressing the enter key on your keyboard to create a new line.

A simple demonstration can be shown through the set and the UI stat monitor commands.

Let’s say we want to set account data to a variable, and then watch that variable with the UI stat monitor command. Notice that when we do that, the credentials are all clutter together.

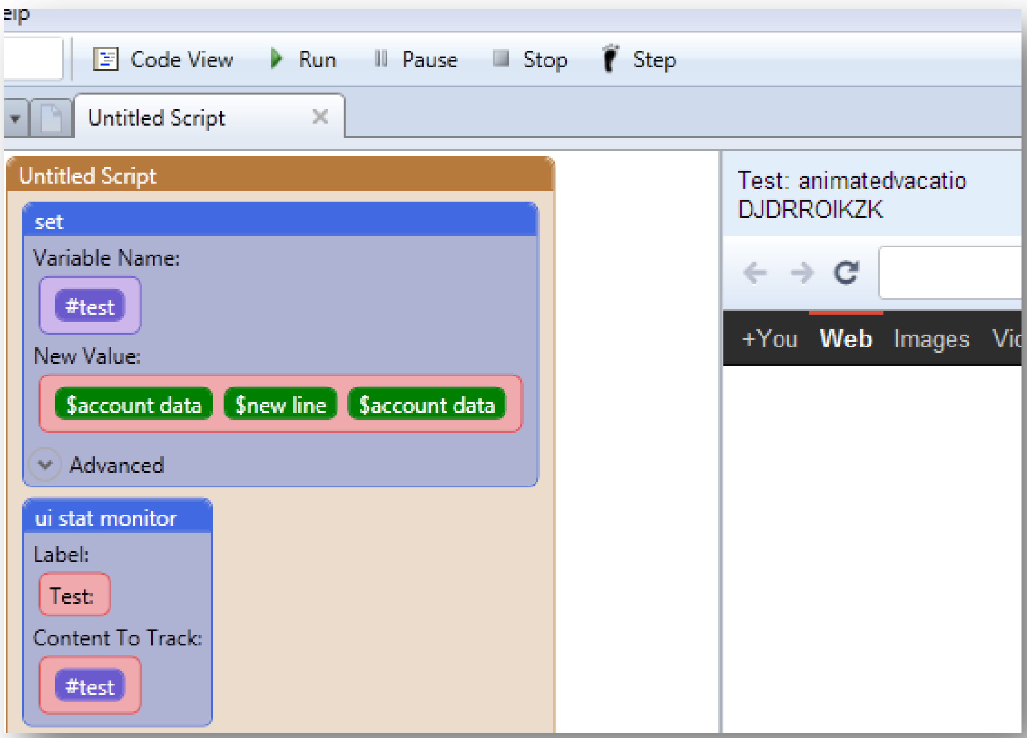


What if we could insert a new line between the username and the password to help organize the information?

Let's do just that. Right click the set command and click "edit". Remove the two account data commands. We are going to start over. Go under the account functions, and select the account data function. Select the username option in the drop down menu within the account data function.

Go to the Text function and select the new line function. Drag it next to the first account data function. Now go back to the Account Functions and select another account data function for the password. Select "password" from the drop down menu within the account data function. Click ok on all functions and commands to exit the editing mode.

When we run the script, notice that a newline has been inserted between the username and password.



Test: animatedvacatio
DJDRROIKZK

This demonstrates that if you need a new line between two pieces of information, then you can use the $newline function to create a new line between those pieces of information.

Nothing

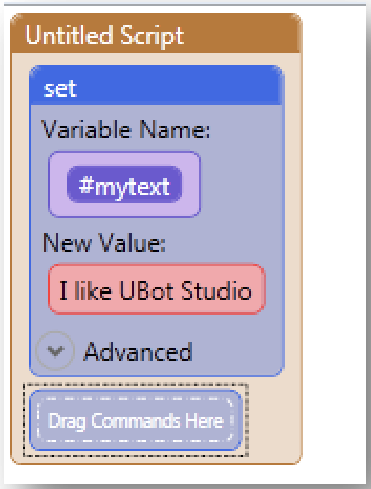
This function Returns an empty value. This is a great function to use when cleaning up scraped data when used in conjunction with the $replace function.

(See the $replace reference below for an example of this).

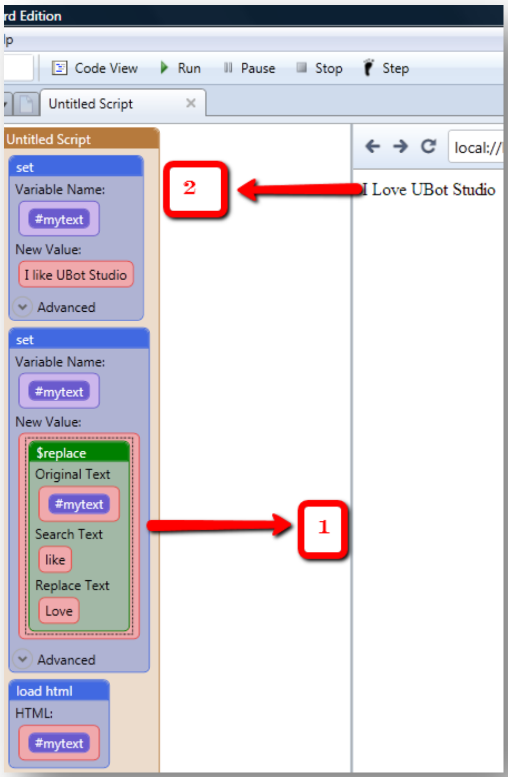
Replace

Returns a piece of text modified from an original by replacing a piece of it with another. For example, if you started with the original text "I like UBot Studio", and you replaced "Like" with "Love", the result would be "I love UBot Studio".

The simplest way to demonstrate this function is to set a variable with the text “I like UBot Studio”.



To replace the text we will re-set the variable using the $replace function:



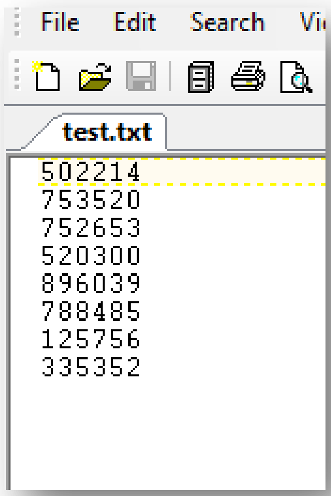
1) Re-set the variable #mytext using $replace as the content. The three labeled fieldss (as seen in the image) are original text = #mytext, search for = “like” and replace with = “Love”. Add a load html command to see the results.

2) The results show the new text after the replace has taken place.

As promised in the $nothing function reference, you can use this same example except instead of using “Love” in the replace labeled field, use $nothing instead and the result will be “I UBot Studio”.

Find regular expression

Find regular expression allows users to conduct specific searches on text strings using regular expressions. This function is used for lists, variables, or whatever piece of information you are trying to scrape. In our examples, we have created a list of random numbers and saved them to a list in a .txt file.



```
502214
753520
752653
520300
896039
788485
125756
335352
```

Next, we are going to write a simple regular expression code to find all the numbers that start with the numbers between the numbers [2-3]. The numbers should also end with the numbers between [0-6].

Here is our regular expression for that criteria:

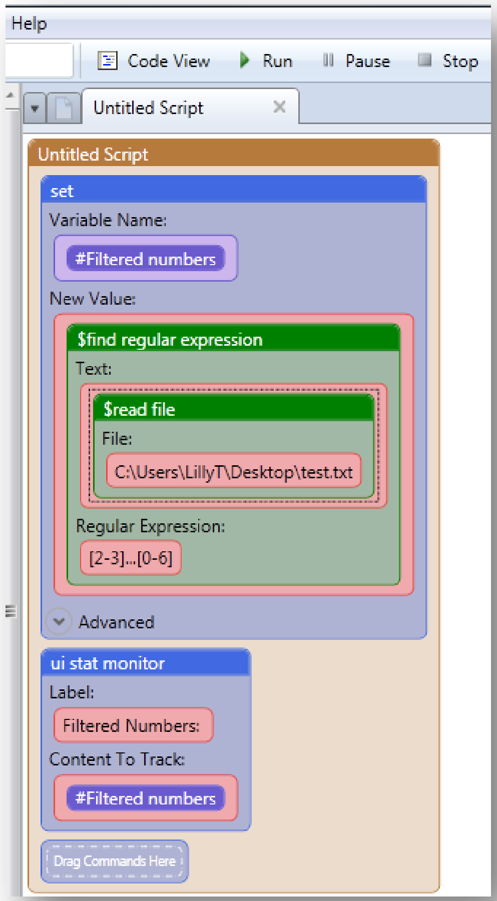
[2-3]...[0-6]

We are going to display the numbers that meet that criteria on the UI of our bot. We have a set command set the find regular expression function, which will analyze the numbers based on our regular expression criteria. We drag in the UI stat monitor command to display the variable we set the regular expression to.

Under "Text" in the Find regular expression function will go the item you are trying to analyze with the regular expression. Under "Expression" will go your regular expression, in this case:

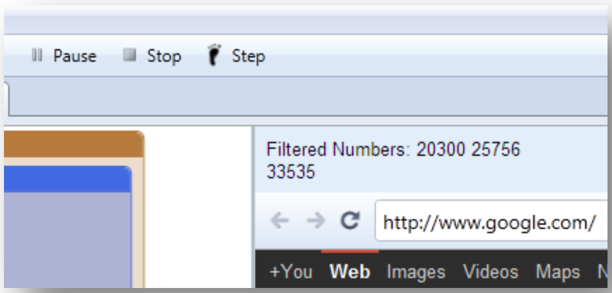
[2-3]...[0-6]

Here is our script set up to do just that:



Untitled Script
set
Variable Name:
#Filtered numbers
New Value:
$find regular expression
Text:
$read file
File:
C:\Users\LillyT\Desktop\test.txt
Regular Expression:
[2-3]...[0-6]
Advanced
ui stat monitor
Label:
Filtered Numbers:
Content To Track:
#Filtered numbers

When we run the script, notice that the numbers that numbers that start with the numbers between 2-3 and ends with the numbers between 0-6 are displayed on our UI.

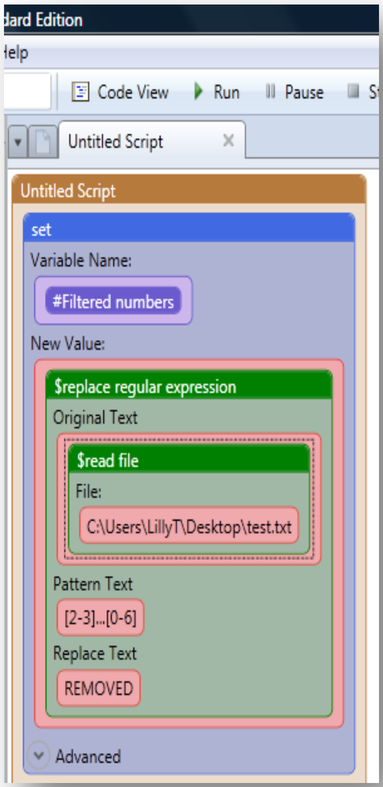


Filtered Numbers: 20300 25756
33535

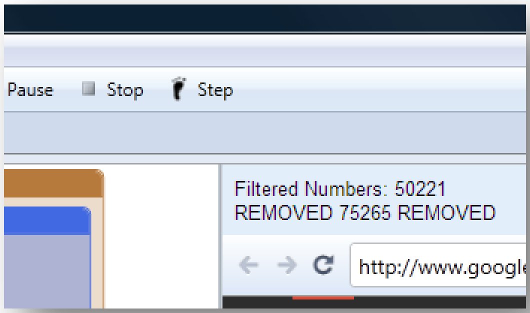
Replace Regular Expression

This allows users to find a specific element within a list or a simple word and replace the item using regular expression. This function works just like the find regular expression function. We can take a look at the last example. The only thing we would need to change in the previous example is the section within the code where the find regular expression function is located within the set command. In the Pattern Text field for the replace regular expression command, we will use the same regular expression we used in the previous example. In the field labeled Replace Text will go whatever you are trying to replace the items that meet the regular expression criteria with. In this case, we will replace the items with the word “REMOVED”. In the field labeled original text will go the read file function with the file that contains the numbers we will be filtering.

So this is the only change we will see in the script:



When the script is run, the numbers that meet the regular expression criteria will be replace with the word "REMOVED":



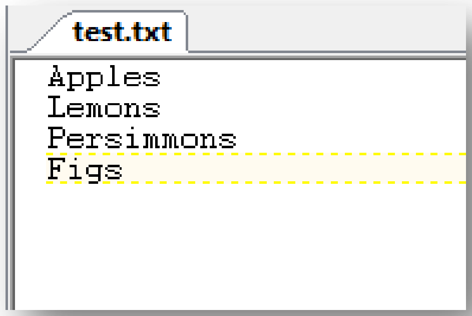
```
Filtered Numbers: 50221
REMOVED 75265 REMOVED
```

Change Text Casing

This function changes all text in a string to uppercase, lowercase, or proper case.

You can turn any text item you place into the field labeled “Text to Change” in the function to uppercase, lowercase or proper case. It can be a list item from a list or simply a string you are trying to fill a field with. Let’s use the set up from our last example for the replace regular expression function to demonstrate this function.

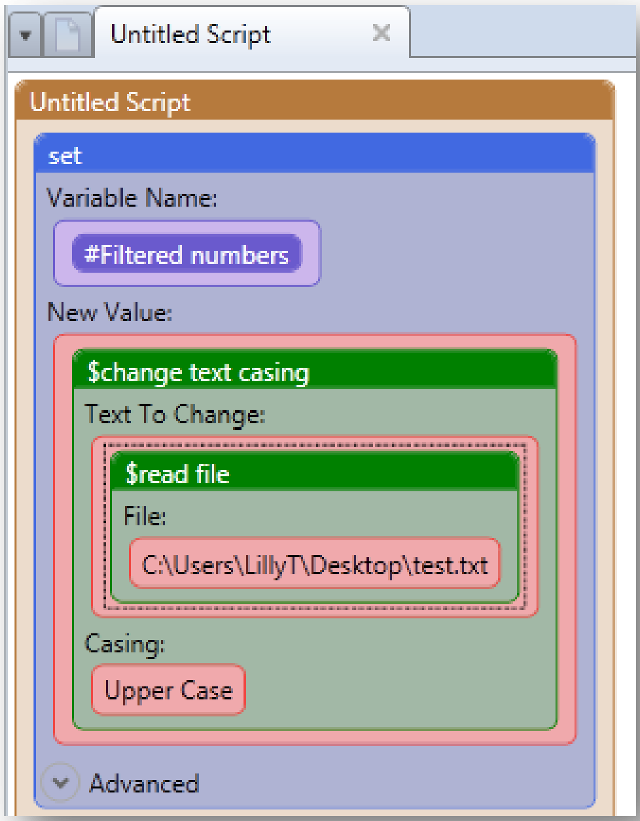
Simply edit the set command, and replace the replace regular expression function with the change text casing function. Insert a read file function from under the variable functions, and choose the file that contains a few simply words, as seen below:



```
Apples
Lemons
Persimmons
Figs
```

Let’s choose the “Upper Case” option from the drop down menu within the function labeled “Casing”.

Click ok on all functions and commands to finish editing. This is the set up we have after we’re done adding the function and browsing for our file:


Untitled Script

Untitled Script

set

Variable Name:

#Filtered numbers

New Value:

$change text casing

Text To Change:

$read file

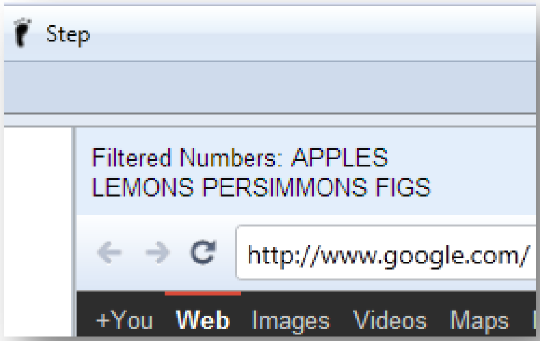
File:

C:\Users\LillyT\Desktop\test.txt

Casing:

Upper Case

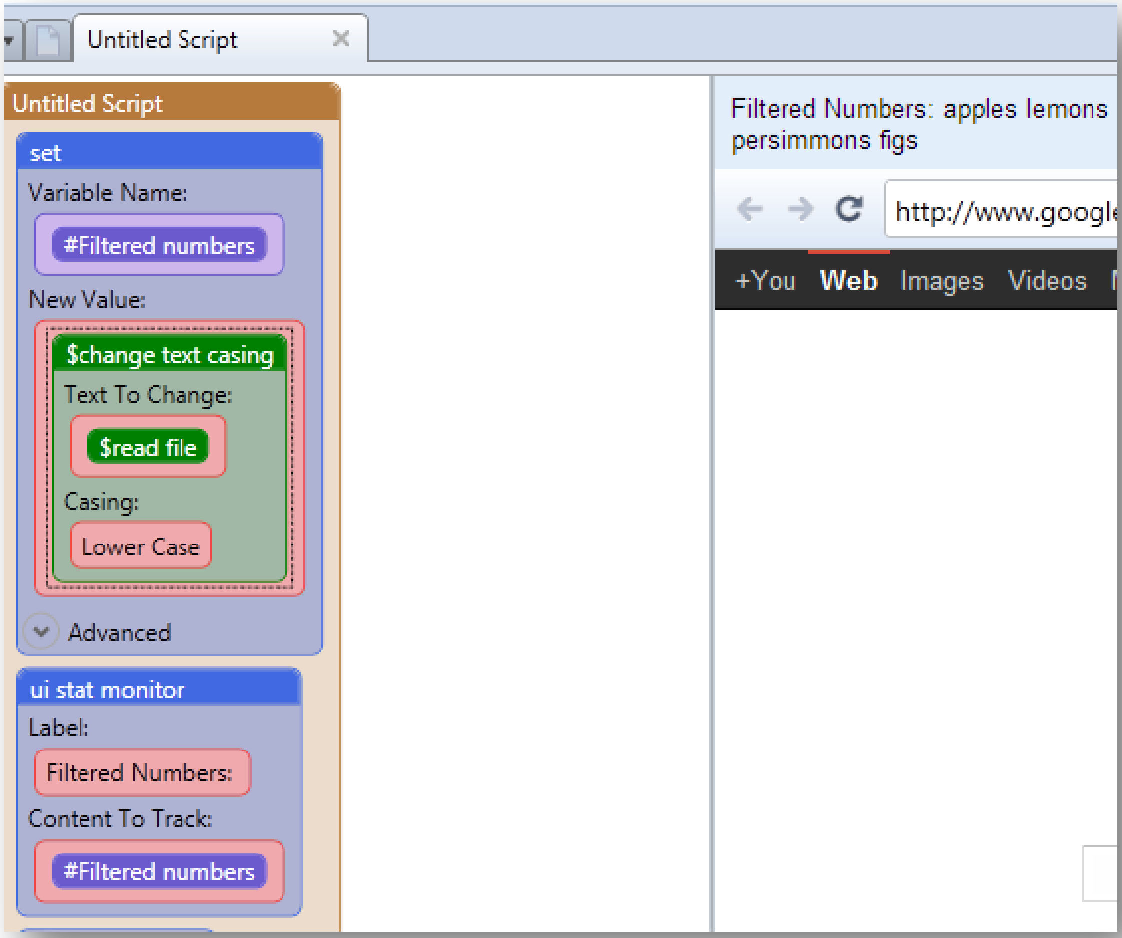
When we run the script, the items in our file are displayed on the UI in Upper case text casing (meaning they are all in CAPS):



Filtered Numbers: APPLES
LEMONS PERSIMMONS FIGS

http://www.google.com/

Now let’s edit the set command, right click to edit the change text casing function. Let’s change the option under the drop down menu labeled “Casing” to Lower Case.


Untitled Script

set

Variable Name: #Filtered numbers

New Value: $change text casing

Text To Change: $read file

Casing: Lower Case

Advanced

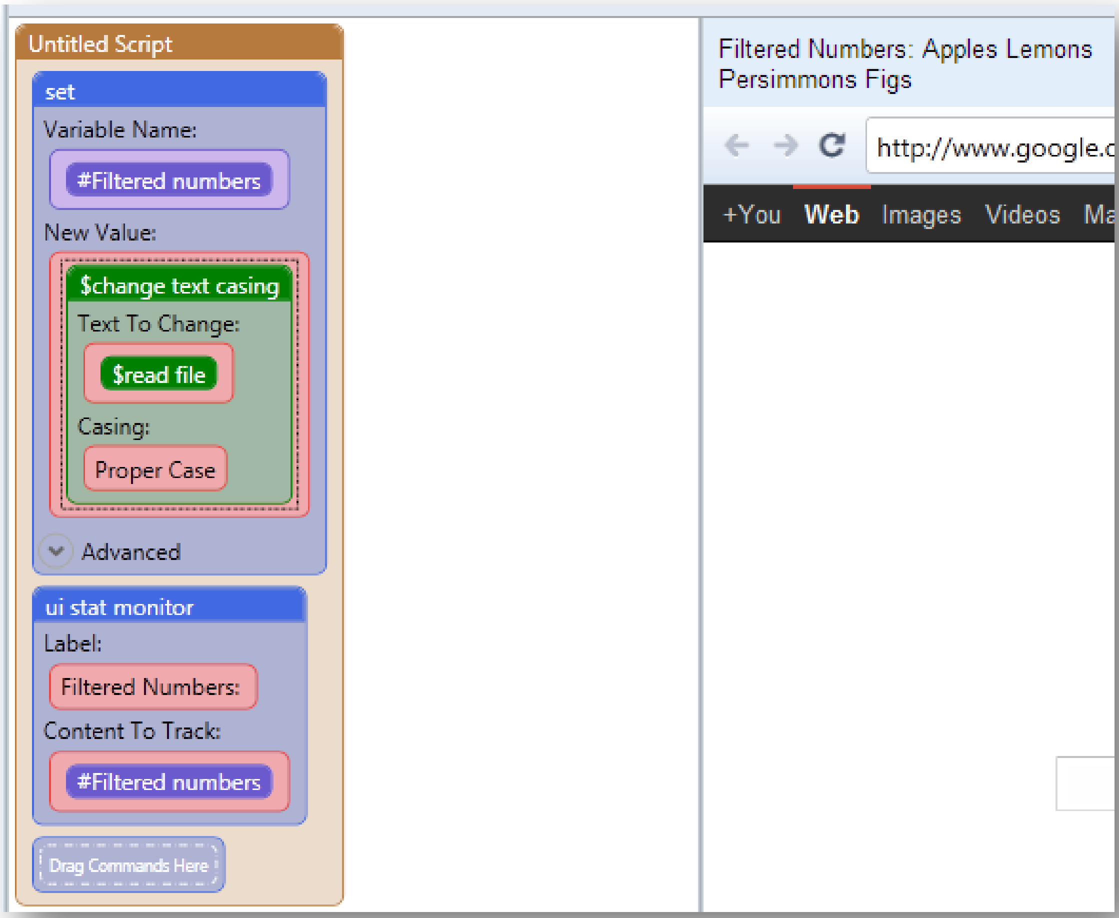
ui stat monitor

Label: Filtered Numbers:

Content To Track: #Filtered numbers

Filtered Numbers: apples lemons persimmons figs

Notice that all the items in the file have been changed to lower case. Now let's change the option to "Proper Case" and see what happens to the items:

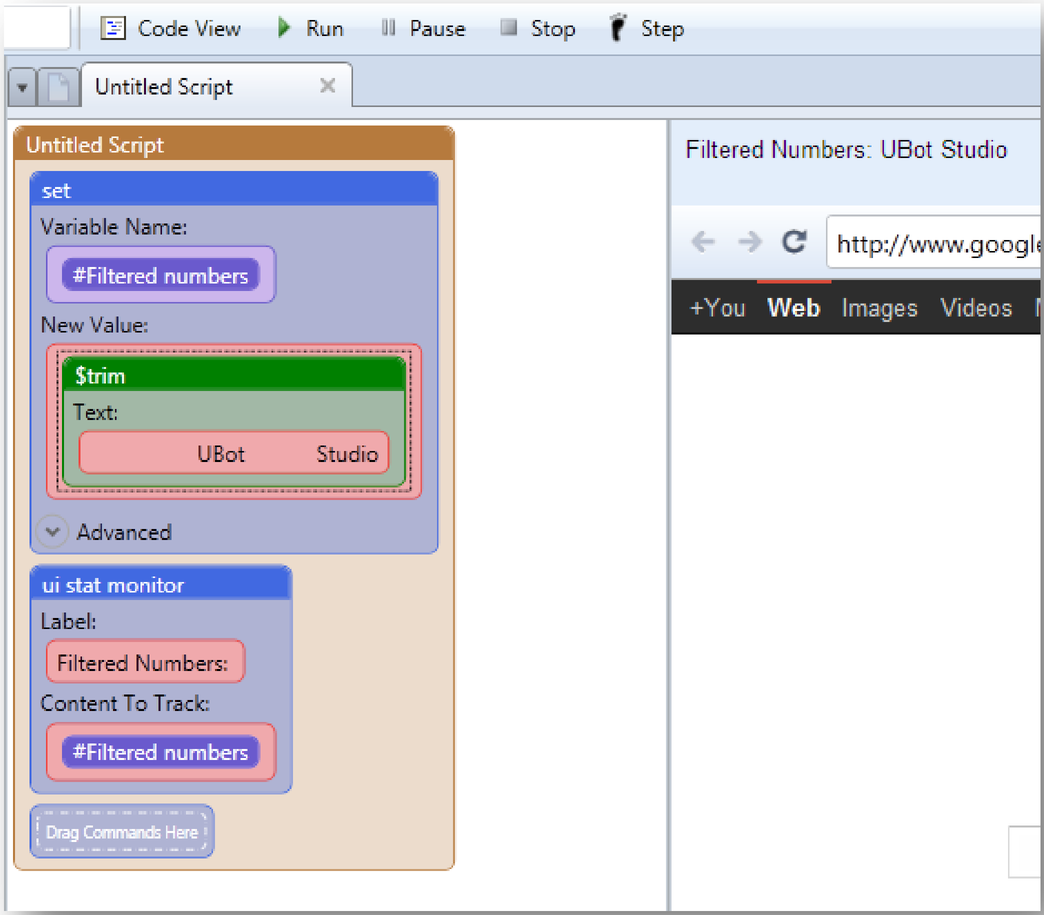


Notice that all the items displayed on the UI are now in Proper Casing, or camel casing.

Trim

This function moves any leading or trailing blank spaces from a string.

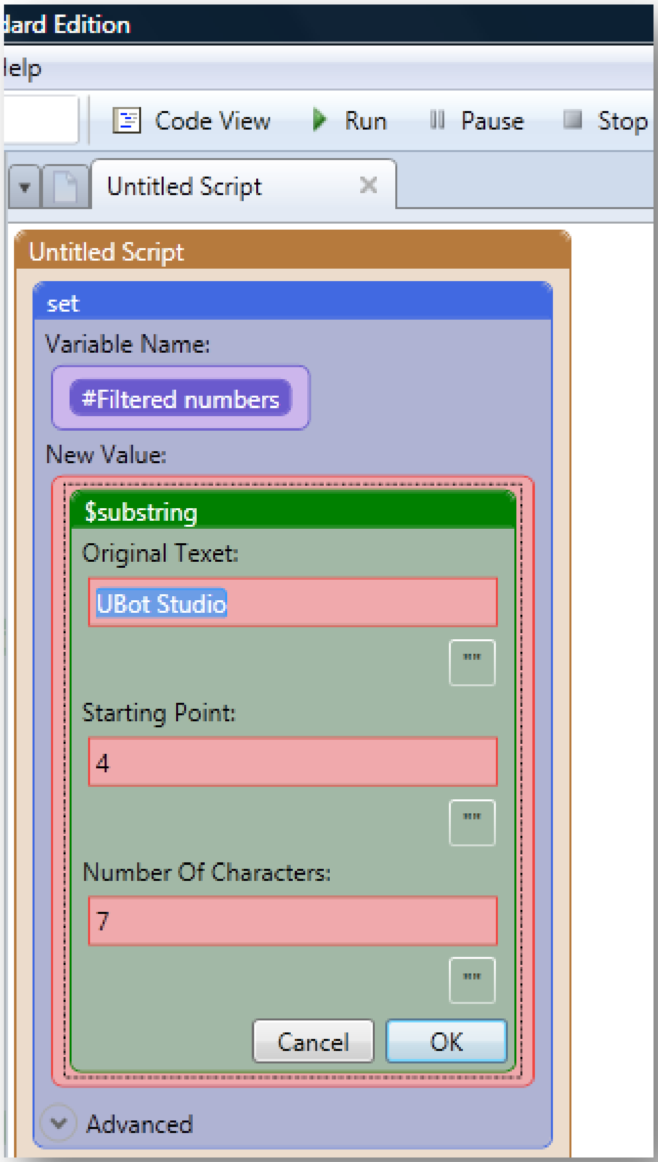
It can be used the same way you would use the other two functions, except it removes extra spaces in a list item. In this example, I am trying to post a string that has a large amount of space preceding one of the words and after the first word. With the trim function, we are able to remove the blank space before the string.



Notice that the spaces before the word "UBot" and before the word "Studio" has all been removed as they were displayed on the UI.

Substring

This returns a new string that is a substring of an original string. The substring begins at a given starting point and extends up to the given length. Again, this function can be used the way that the other functions can be used, except this finds the determined subtext of a string. Let's use the same set command/ui stat monitor set up from before. After you drag the substring function into the set command, you will be asked to determine where the subtext starts and what the total length of the subtext is within the string.



Variable Name: #Filtered numbers

New Value:

$substring

Original Texet: UBot Studio

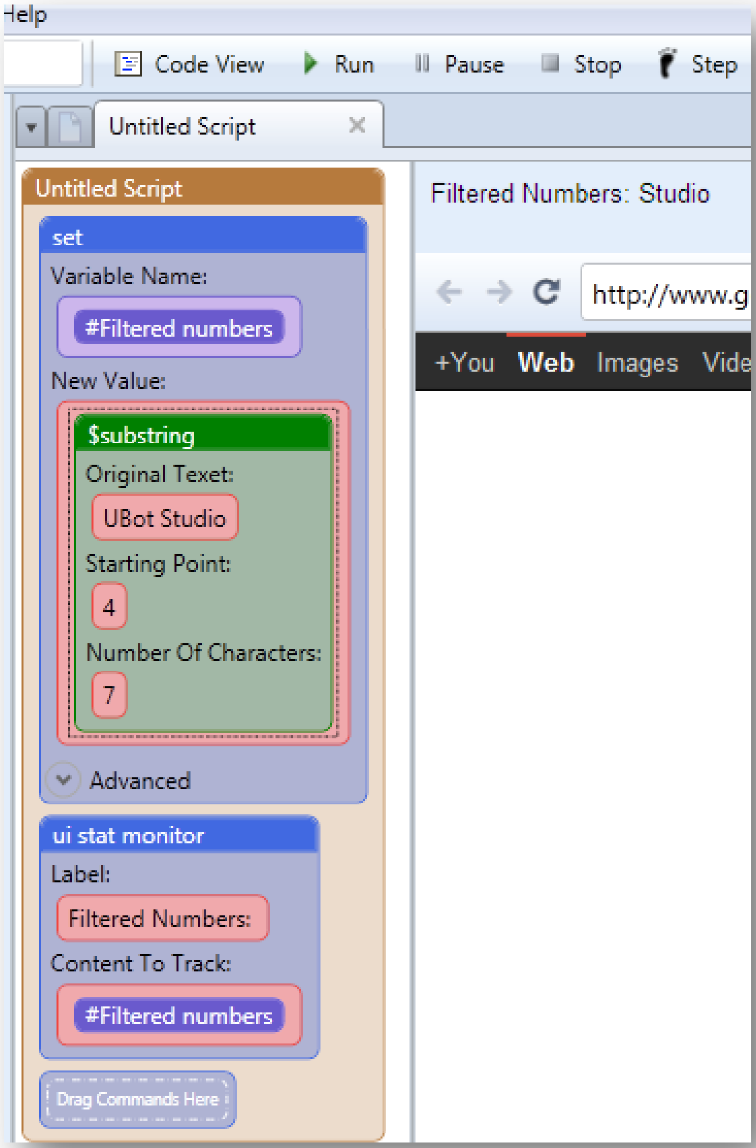
Starting Point: 4

Number Of Characters: 7

Starting point refers to where your substring begins. In this case, our substring, which is the word “Studio”, begins after the word “UBot”. There are 4 letters in UBot, so the starting point is 4.

Number of Characters is referring to how long the substring you are trying to get is. In this case, Studio is 6 letters long. However, there is one more added to it to count the space that comes before the first letter of the word.

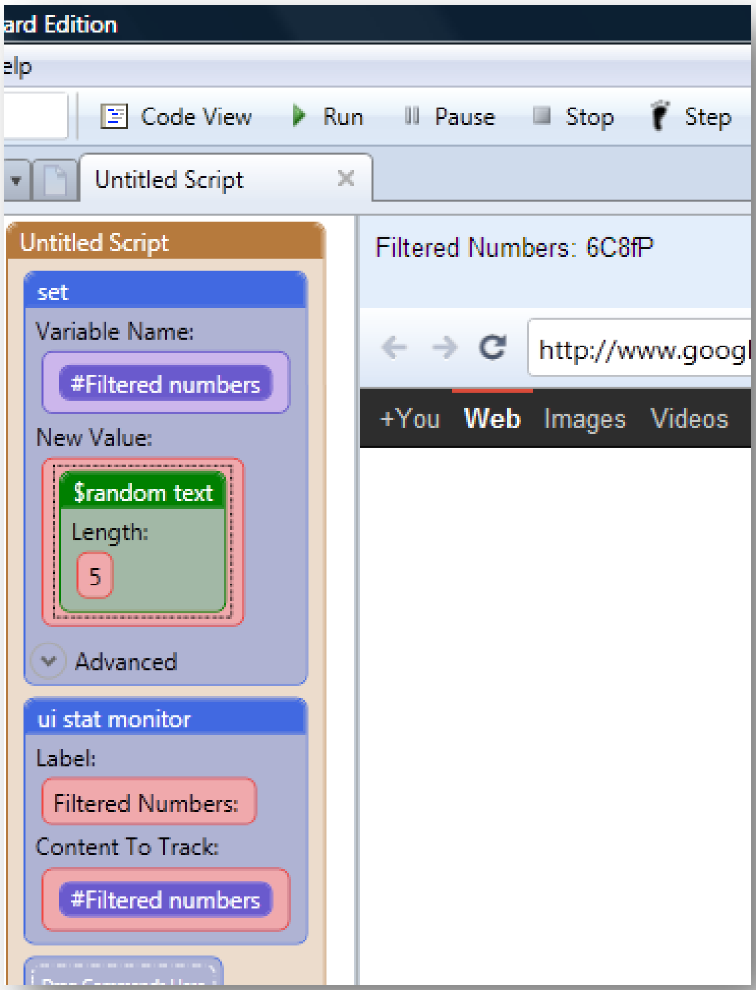
When you run your script, the word “Studio” is displayed on the UI of the script.



Random text

The function generates a random alpha-numeric string of user defined length, including upper and lowercase letters. This function will fill a field with a bunch of random text. It is usually used as space filler.

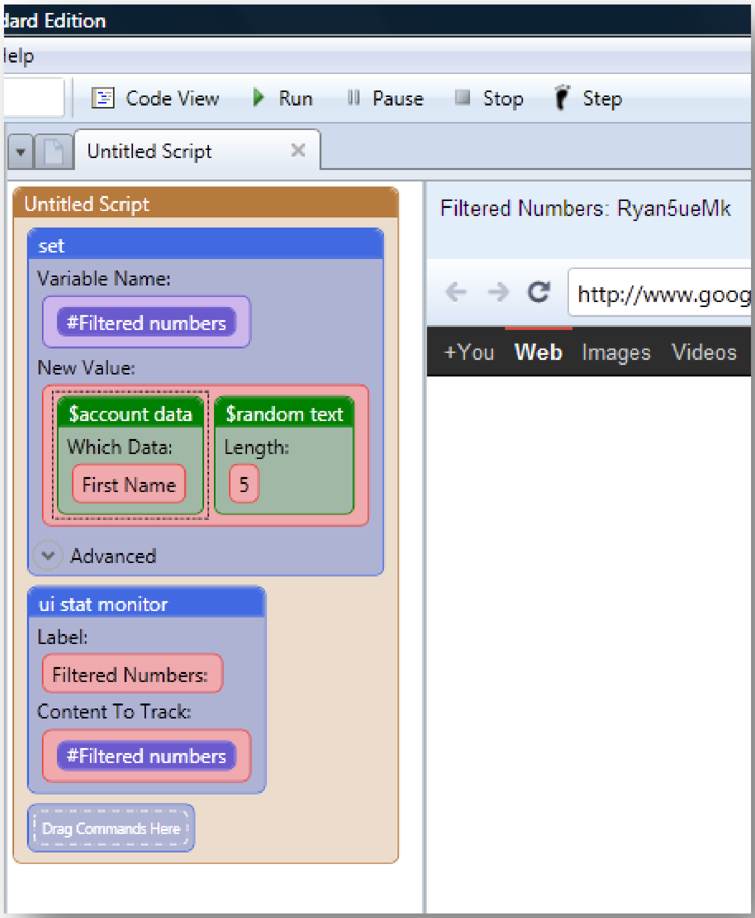
Using our last example with the set command and the UI stat monitor, replace the substring function with the random text function. A field will appear within the function asking how long you want the random text to be. Let's choose 5 in this example. Click ok on all functions and commands. Run the script. Our random text is displayed on the UI.



Our random text is 6C8fP.

So what if we want to combine this function with another function, such as the account data function set to “Firstname”?

Right click and select edit on your set command. Go under account functions and drag the account data function in front of the random text function. Select “Firstname” from the drop down menu that appears after dragging in the account data function. Click ok on all commands and functions to exit editing. Run the script.



The random text function plus the generated account data function creates the item displayed on the UI.

Pad text

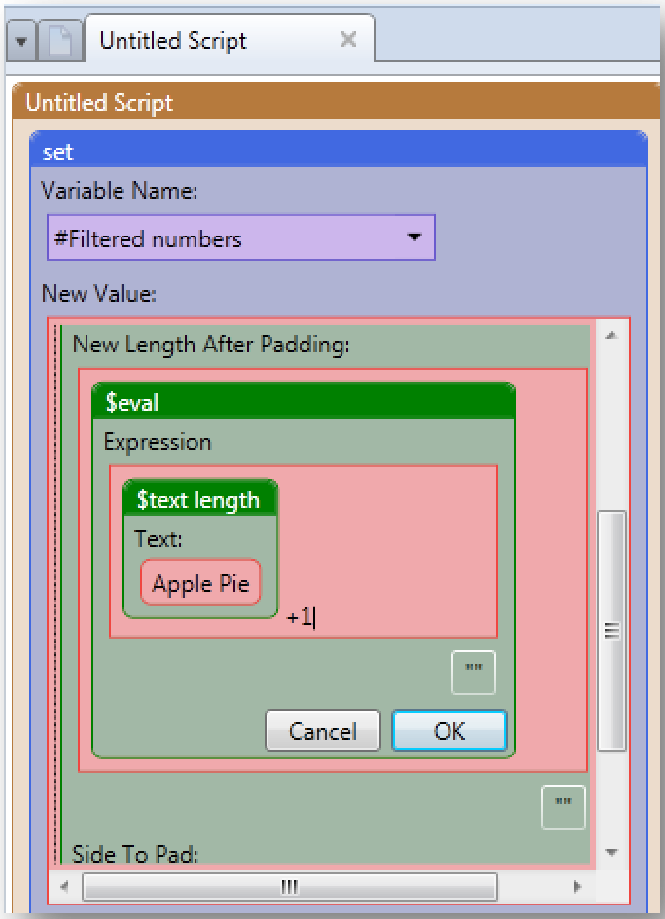
This function adds a fixed number of characters to the left or right side of a string based on the total length of the string selected.

In this example, we are padding the text “Apple Pies”. Let’s use the set command and Ui stat monitor command example. Place the pad text function inside the set command under value. Three fields will appear within the function.

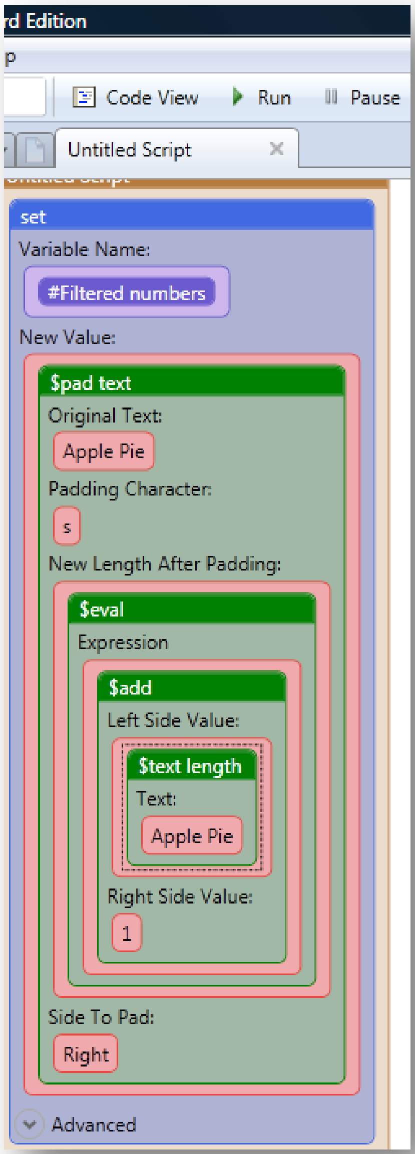
Original Text: The text you want to pad. Can be written directly into the field or be a list item.

Padding Character: The character you want to pad the original text with.

New length after Padding: How long the original text will be after padding. To calculate this, simply drag the eval function into this field. When the field labeled “Expression” appears in the Eval function, drag in the text length function. A field will appear within that function. Type in the original text again, and click ok on the function. After clicking ok, type + 1 next to the text length function. Click ok for the eval function.



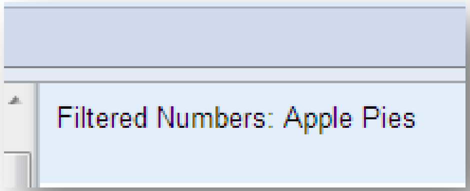
The number 1 determines how many times the original text will be padded with the padding Character. Scroll down in the pad text function. Under drop down menu labeled “Side to Pad”, choose right or left. If you choose left, the Padding Character will pad the left side of your Original text. If you choose right, it will pad the right side of your original text. Click ok on all functions and commands. Let us choose right for this example.



set
Variable Name:
#Filtered numbers
New Value:
$pad text
Original Text:
Apple Pie
Padding Character:
s
New Length After Padding:
$eval
Expression
$add
Left Side Value:
$text length
Text:
Apple Pie
Right Side Value:
1
Side To Pad:
Right

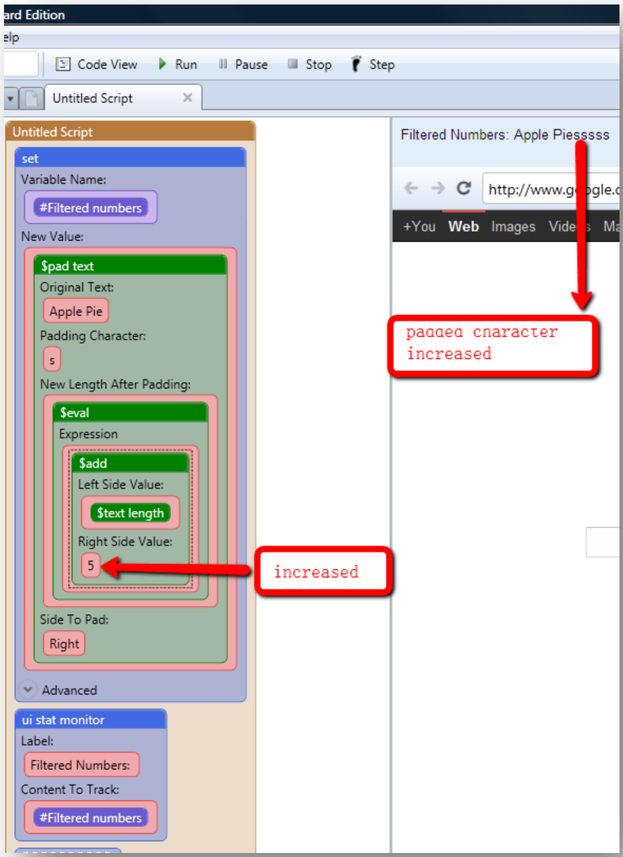
Notice the add function that appeared in the eval function. Since you are adding the text length of the original text plus 1, an add function appears to encase the process. If you were subtracting, a subtract function would appear instead.

Click ok for all commands and functions, and run the script.


Filtered Numbers: Apple Pies

Our original text, “Apple Pie” has been padded once by the character “S” on the right side, to create the phrase “Apples pies”

We can pad the text with many more “s”’s by increasing the number in the right side value for the eval function, under $add function.



Untitled Script
set
Variable Name: #Filtered numbers
New Value:
$pad text
Original Text: Apple Pie
Padding Character: s
New Length After Padding:
$eval
Expression
$add
Left Side Value: $text length
Right Side Value: 5
Side To Pad: Right
Advanced
ui stat monitor
Label: Filtered Numbers:
Content To Track: #Filtered numbers

Filtered Numbers: Apple Piesssss
http://www.google.c

Text length

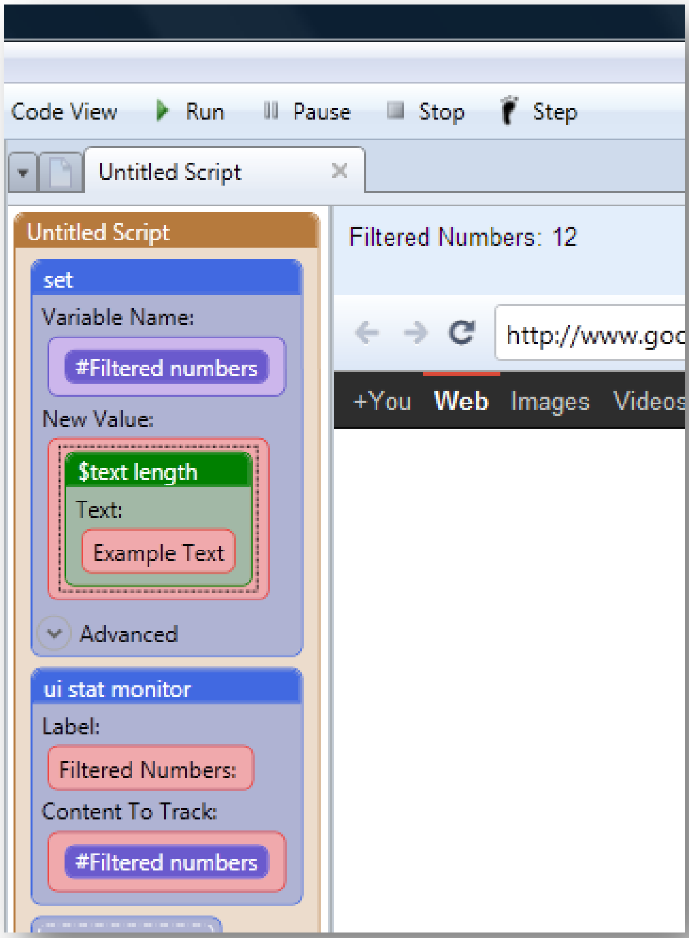
This function returns the total length of some text. Please see the previous example for pad text, where text length is used.

You can fill a field with the function to find the text length of either an account

Function (username, first name, password, etc), a variable, a list item or a piece of text.

In this example, we have set the text length function to a variable, and displayed the amount on the UI.

The text length of the phrase "Example Text" is 12.



Insert text

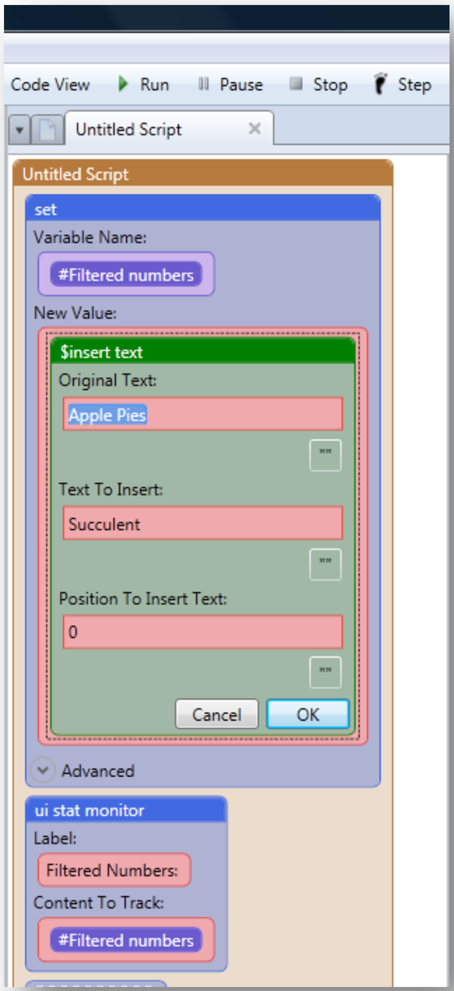
Inserts a substring into a string based on the specified position. In this example, we are trying to insert the word “Succulent” at position 0 on the phrase “Apple Pies”, so we can try to create the phrase: “Succulent Apples Pies”

Original Text: Where our phrase “Apple Pies” will go

Text to Insert: Where the item you are trying to insert will go.

Position: Where exactly you want the item inserted.

A space is added after the word “Succulent” to avoid the phrase coming out like: SucculentApple Pies



Variable Name: #Filtered numbers

New Value:

$insert text

Original Text: Apple Pies

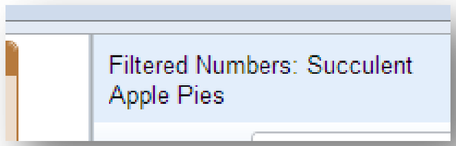
Text To Insert: Succulent

Position To Insert Text: 0

Label: Filtered Numbers:

Content To Track: #Filtered numbers

Click ok on all commands and functions (starting with the functions) and run your script. This is what we see displayed on our UI:



Filtered Numbers: Succulent Apple Pies

Change the position number to see how that affects where the "Text to Insert" goes in the original text.

Find index

This function finds the index of any subtext within a text string. This function is perfect for use within the insert text function. You can find the index of an element and then use that index number to determine where you want a new item inserted. In this example, we would like to insert the name "Manning" into the item "Jayne333" between Jayne and *333. The first thing we did was find the index of where we would like to insert the word and set that to a variable named "username". So we are finding the index of 333 within Jayne 333. The index is 5.

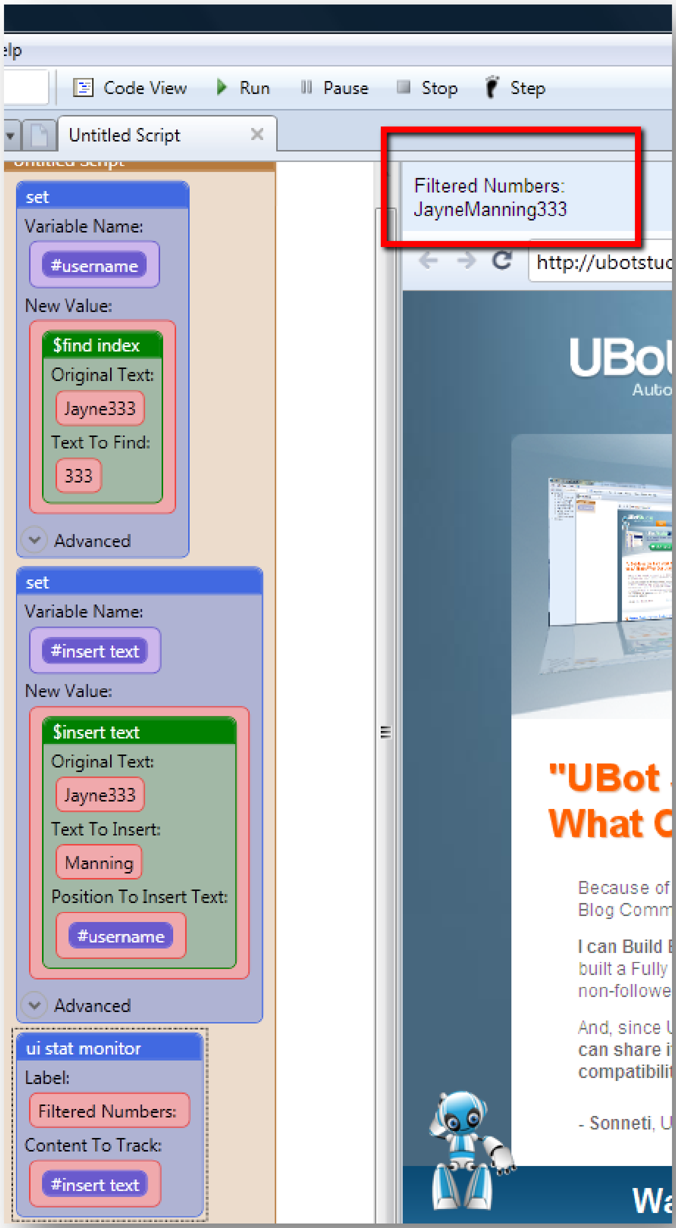
So then we move on to display the result on the UI. You will go under text functions and choose the Insert Text function to insert into another set command.

Original Text: Where Jayne333 will go

Text to Insert: We are trying to insert the word "Manning"

Position: The position will be the variable with the index set to it.

After running the script, "Jayne333" turns into "JayneManning333".



set
Variable Name: #username
New Value:
$find index
Original Text: Jayne333
Text To Find: 333

set
Variable Name: #insert text
New Value:
$insert text
Original Text: Jayne333
Text To Insert: Manning
Position To Insert Text: #username

ui stat monitor
Label: Filtered Numbers:
Content To Track: #insert text

Filtered Numbers:
JayneManning333

Append line to text

This function appends a new line to a piece of text.

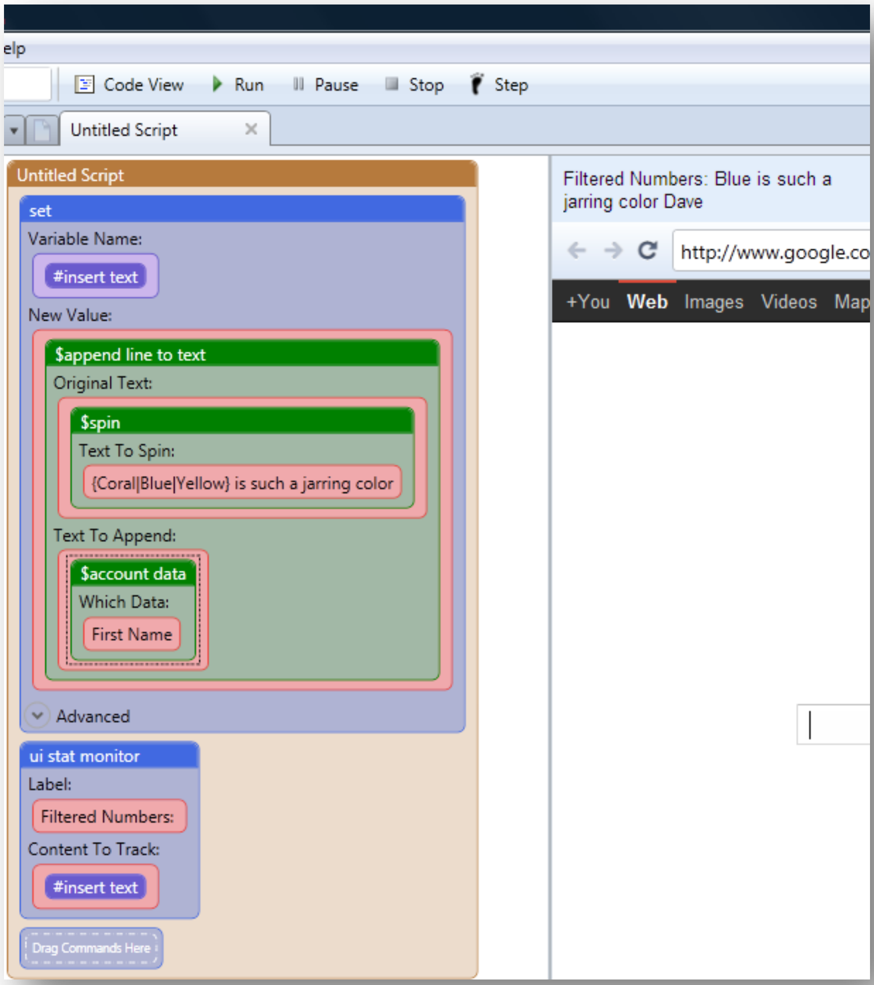
It simply adds one piece of text to another. So, in this example, we have a sentence that is being spun. We are then going to append the spun content to a first name generated with the account data function.

When you drag in the append line to text function into the set command, you will notice there are two columns:

Original Text: The text you are trying to append another text to. This is where your spin function will go.

Text to Append: Which refers to the text you are trying to append into the original text. This is where the username function will go.

Click ok on all functions and commands after setting up them up and inserting the appropriate values. Once you run your script, the UI area of your script will be filled with the resulting appended text.


elp
Code View
Run
Pause
Stop
Step
Untitled Script
Untitled Script
set
Variable Name:
#insert text
New Value:
$append line to text
Original Text:
$spin
Text To Spin:
{Coral|Blue|Yellow} is such a jarring color
Text To Append:
$account data
Which Data:
First Name
Advanced
ui stat monitor
Label:
Filtered Numbers:
Content To Track:
#insert text
Drag Commands Here
Filtered Numbers: Blue is such a jarring color Dave
http://www.google.co
+You
Web
Images
Videos
Map