A Detailed Explanation Of Selectors – UiPath

Posted on [September 23, 2020](https://excelcult.com/a-detailed-explanation-of-selectors-uipath/) by [Kishori Afzulpurkar](https://excelcult.com/author/kishori-afzul/)



What is a selector?

A selector is a string of characters used to identify elements on the screen.

Selectors with wildcards can be used when the object under identification (attribute value) keeps changing.

* Asterisk (\*) – replaces zero or more characters
* Question mark (?) – replaces a single character

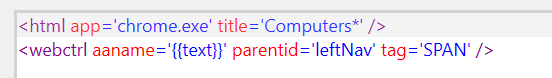
Full Vs Partial Selector:

|  |  |
| --- | --- |
| Generated by Basic Selector | Generated by Desktop Selector |
| Contains all the elements needed to identify a UI element, including top level window | Do not contain information about the top-level window |
| Recommended when switching between multiple windows | Recommended when performing multiple actions in the same window |
| Every activity contains full selector | Activities containing partial selectors are enclosed in a container(Attach Browser or Attach Window) that contains a full selector of the top-level window |
|  | Cannot edit the top-level window in the activities inside the container, must edit at the top-level container |

Example for Full Selector:(Contains top level window which is editable)

https://i0.wp.com/excelcult.com/wp-content/uploads/2020/09/image-157.png?resize=542%2C57&ssl=1

Example for Partial Selector:(Top level window grayed out and read only in below selector)

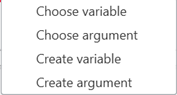


Dynamic Selector:

A dynamic selector can be used when the value of an element keeps changing.

<tag attribute=’{{Value}}’/>

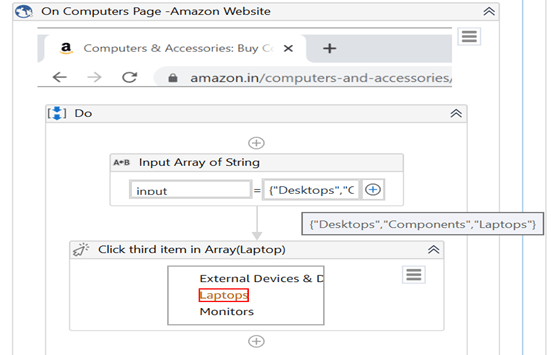
You can choose an existing variable/argument or create a variable/argument and pass as a value to the attribute by pointing and right clicking on the attribute value between single quotes to get the options. ‘<Point here and Right Click to get the options>’



Note: Dynamic variable supports only string and integer values.

Example1:

Input:

https://i0.wp.com/excelcult.com/wp-content/uploads/2020/09/image-127.png?resize=550%2C50&ssl=1

Top Level window selector using wildcard (\*):

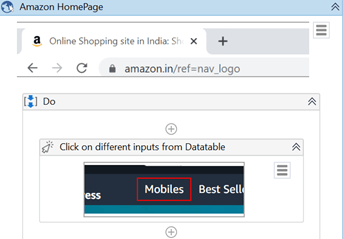
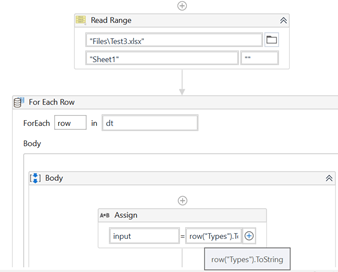
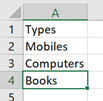
https://i0.wp.com/excelcult.com/wp-content/uploads/2020/09/image-129.png?resize=410%2C46&ssl=1

Dynamic selector text to which we are passing value from the variable List:

https://i0.wp.com/excelcult.com/wp-content/uploads/2020/09/image-130.png?resize=438%2C63&ssl=1

Example2:

Input:



Using Wildcard  \* in selector below:

https://i0.wp.com/excelcult.com/wp-content/uploads/2020/09/image-134.png?resize=438%2C34&ssl=1

Using dynamic selector variable which is coming from  the excel:

https://i0.wp.com/excelcult.com/wp-content/uploads/2020/09/image-135.png?resize=527%2C50&ssl=1

Regex Search:

Used to get a specific match/value from an element. For searching, text processing and data validation we can go with regex search.

Syntax:

matching:<tag\_name>=’regex’

<tag\_name>='<regex\_command>’

Example:

matching:aaname=’regex’

<aaname=’\d{3}’>

RegEx search is supported by the following selector types:

<wnd> ,<html>, <webctrl>, <java>, <ctrl>, <sap>

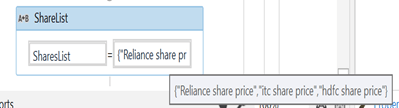
Does not support below selector types:

Tags which identify a process name. Ex: <html app=”chrome.exe”>

Example1:

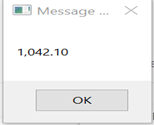
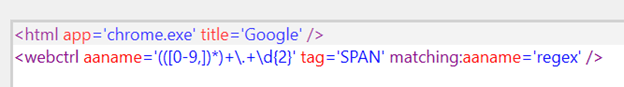
Input:

List containing three values

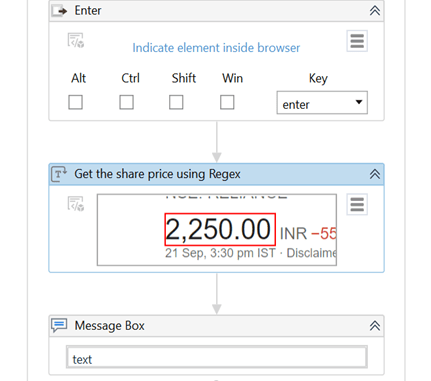
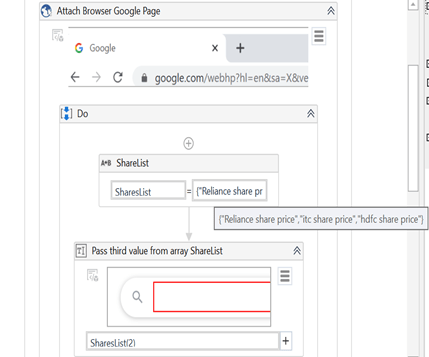


Output:

Regex pattern: ([0-9,])\*+\.+\d{2} to get the text



Process:



Fuzzy Search:

Fuzzy search capabilities enable you locate strings based on a pattern, rather than on an exact match to your input.

Syntax:

matching:<tag\_name>=fuzzy

fuzzyLevel:<tag\_name>=<numerical\_value> , here numerical value is between 0 to 1

* Closer to 0 – returns results with lower similarity to the <tag\_name>
* Closer to 1 – returns results with higher similarity to the <tag\_name>

Example:

If mistake in attribute is more, use a fuzziness level closer to 0

ex:matching:aaname=’fuzzy’ fuzzylevel:aaname=’0.3′

If mistake in attribute is small, use a fuzziness level closer to 1

ex:matching:aaname=’fuzzy’ fuzzylevel:aaname=’0.7′

Fuzzy search is supported by the following selector types:

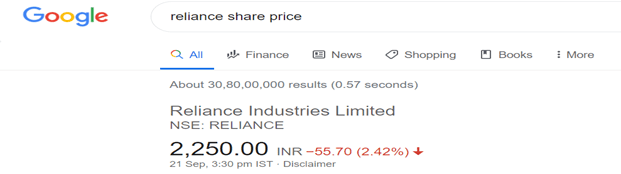
<wnd> ,<html>, <webctrl>, <java>, <ctrl>, <sap>, <rdp>

Doesn’t support : <silverlight>, Tags which identify a process name. Ex: <html app=”firefox.exe”>

Example1:

Validate by using Ui Explorer and modify the attribute value to check how it works.

Input:



Low Fuzziness:

https://i0.wp.com/excelcult.com/wp-content/uploads/2020/09/image-142.png?resize=624%2C62&ssl=1

Example2:

Input:

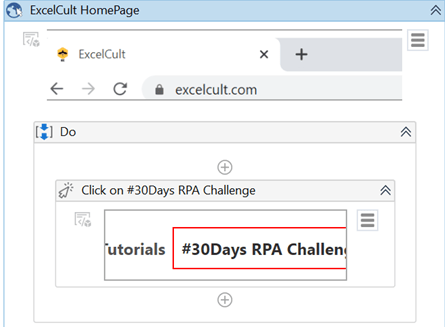
https://i0.wp.com/excelcult.com/wp-content/uploads/2020/09/image-143.png?resize=619%2C63&ssl=1

High Fuzziness:

https://i0.wp.com/excelcult.com/wp-content/uploads/2020/09/image-144.png?resize=624%2C46&ssl=1

Example3:

Input:



Fuzzy search with low fuzziness: To understand this, have changed aaname value and fuzzylevel close to 0.

https://i0.wp.com/excelcult.com/wp-content/uploads/2020/09/image-146.png?resize=624%2C54&ssl=1

Non-Greedy Search:

The Non-Greedy search makes it possible to identify the target element from a pool of similar applications. It needs to be included in the top-level tag of a selector.

If a generated selector contains the idx attribute, its value is set by default to \*.

Searches through all active window instances for the element matching the top-level tag attributes.

<idx=’\*’>

Non-Greedy search is supported for the following selector types:

<wnd>,<html>

Does not support below selector types:

<webctrl>, <java>, <ctrl>, <sap>, <silverlight>, Universal Windows Platform applications and selectors inside Open Application and Browsers, Attach Window and Browser

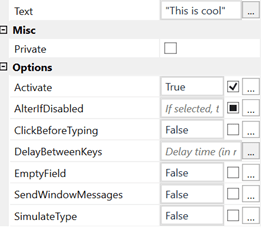
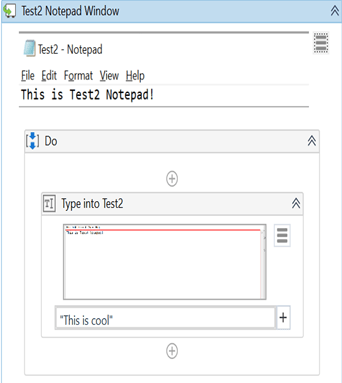
Example1:

Consider there are two notepads open. Test2 is in the background and Test4 notepad in the foreground.

Add idx tag with value \*

ex: idx=\* to your selector to type the text to the desired notepad

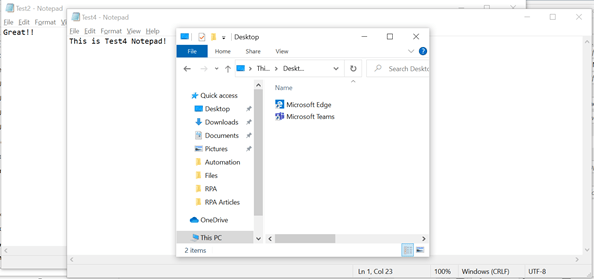
Input:



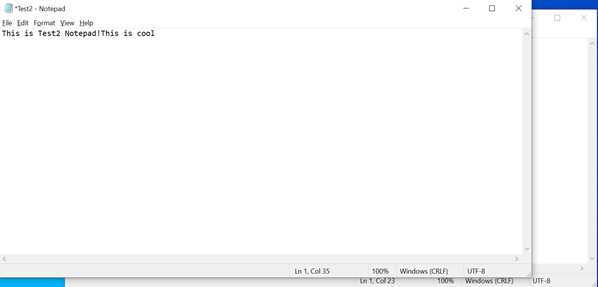
Add attribute idx=\* for the text to be typed into desired notepad:

https://i0.wp.com/excelcult.com/wp-content/uploads/2020/09/image-149.png?resize=485%2C41&ssl=1

We can see two notepads in background and one file explorer in foreground opened. After you add the idx attribute to the attach window and run the flow, it will type into the expected notepad i.e Test2.



Output:



Case-Sensitive Selectors:

You can enable casing for selectors by using the casesensitive attribute. It does not consider lower/upper case and performs the action needed when casing is set to false. By default, the casing is set to true.

Syntax:

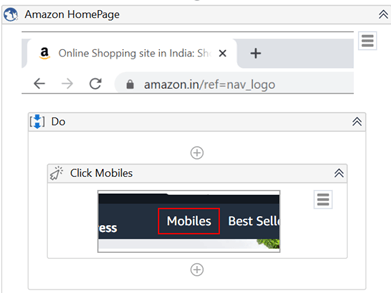
casesensitive:<attribute-name>=’true’

Example:

casesensitive:aaname=’false’

Example1:

Input:



Applied casing to title and aaname attribute and replaced Shopping with “shopping” and Mobiles with mobile.

https://i0.wp.com/excelcult.com/wp-content/uploads/2020/09/image-153.png?resize=456%2C45&ssl=1https://i0.wp.com/excelcult.com/wp-content/uploads/2020/09/image-154.png?resize=458%2C48&ssl=1

Output:

It will click on the desired input i.e Mobile in this example.

We hope you enjoyed the article!!

You can connect with [Kishori Afzulpurka](https://www.linkedin.com/in/kishori-afzulpurkar-57a13b89/)r, she is always happy to help.

Thank you!!