

# What is Regex?

## Brief Overview

Regular expressions (AKA regex) allow you to manipulate Dynamic Text that appears in WalkMe content or that is part of automated processes. Using [regular expression syntax](#) is ideal for situations where Dynamic Text values you are using contain both the values you want and other characters that aren't relevant.

## Use Cases

**Regex use cases include the following:**

- When Dynamic Text includes a user's first and last name, use regex to select and display only their first name in a ShoutOut or balloon to make it more personal (visual display);
- Auto-populate your users' email address (when it's part of a larger object with additional data) into a text field upon page load (automation);
- Select only part of a URL when using Dynamic Text to automatically take the user to a different page.

## Example Use Case

A customer wants to have a SmartTip display when an on screen element (illustrating a date) is before or equal to Jan 31, 2019.

How to tackle the problem:

1. Begin by outlining possible values you will want to match and values you do not want to match.
2. Break the problem down into multiple groups of rules which, when combined, encompass all values you want to match.

1. In the above case we have four different patterns we want to match:

- i. Any dates in Jan 2019
- ii. Any date between Jan 1, 2010 and Dec, 31, 2018
- iii. Any date between Jan 1, 2000 and Dec 31, 2009
- iv. Any date between Jan 1, 0000 and Dec 31, 1999

3. Form the rules to handle each of the above cases and test them:

- i. `^Jan[\s]+[0123]?[\d]+,[\s]+2019$`
- ii. `^[w]+[\s]+[0123]?[\d]+,[\s]+201[012345678]$`
- iii. `^[w]+[\s]+[0123]?[\d]+,[\s]+200[0123456789]$`
- iv. `^[w]+[\s]+[0123]?[\d]+,[\s]+[01][\d]{3}$`

4. After testing each group, combine into one string using the pipe (remove excess ^ and \$s)

- i. `^Jan[\s]+[0123]?[\d]+,[\s]+2019|[\w]+[\s]+[0123]?[\d]+,[\s]+201[012345678]|[\w]+[\s]+[0123]?[\d]+,[\s]+200[0123456789]|[\w]+[\s]+[0123]?[\d]+,[\s]+[01][\d]{3}$`

5. Implement the above RegEx in the SmartTip's display segment ruleset using:

OSE/jQuery > Text is Per Regular Expression >

```
^Jan[\s]+[0123]?[\d]+,[\s]+2019|[\w]+[\s]+[0123]?[\d]+,[\s]+201[012345678]|[\w]+[\s]+[0123]?[\d]+,[\s]+200[0123456789]|[\w]+[\s]+[0123]?[\d]+,[\s]+[01][\d]{3}$
```

6. Result: SmartTip will display when the element's text value is Jan 1, 0000 all the way to Jan 31, 2019!

## What Regex Format To Use

Begin with an opening regex tag containing the regular expression itself, followed by the statement you wish to apply the regular expression to, and end with a closing regex tag:

**[regex="{the regex itself}"]{text to perform the regex on}[/regex]**

**Note:** {text to perform the regex on} can be a mix of multiple Dynamic Text values (i.e., variables, cookies, WalkMe Data, and jQuery) and static values. Most, if not all uses of text in regex will be dynamic, with minimal use of static text values.

### Structure

**1. Starting and ending demarcators:** When outside of brackets, ^ will denote the start of a RegEx and is used with the \$ to denote the end of the RegEx

Examples:

- In an extension package:  
`^https://ecolab\my\.salesforce\.com.*|^https://ecolab.*na.*\.visual\.force\.com.*`

- If using regex in the rule engine: User Data > Variable > department > Is Per Regular Expression > ^sales\$

## 2. Single character match: Most basic form of RegEx will do a direct character match

Examples:

- ^walkme\$
  - Will match "walkme"

**3. Lists and Ranges:** [] can be used to specify a list of values and when combined with - can dictate a range of values a corresponding character in the target string can take

Examples:

- ^[dcr][ao][gt]\$
  - Will match "dog" "cat" "rat" "rag"
  - First character must be d, c, or r; Second character must be a or o, and the third character must be g or t
- ^[a-z][0-9]\$
  - First character must be a through z (lowercase) and second character must be 0 through 9

**4. Repetitions:** {3} and {#, #} can be used to avoid having to type the same patterns repeatedly in succession within a RegEx

Examples:

- ^[a-z]{3}\$
  - Will match three lowercase letters in sequence such as "adf" or "kzd"
- ^[a-z]{1,3}\$
  - Will match no less than one and no more than three of the preceding characters such as "a" "am" and "abc" but not "duct"

**5. Wildcards:** These are useful if text in the target string is dynamic or if the contents do not matter. . will match any single character, \* will match zero or more of the preceding pattern, and + will match one or more of the preceding pattern.

Examples:

- ^.\*\$
  - Will match zero or more of any characters and is useful to add to the beginning and ending of a RegEx (particularly within Extensions) because we are only interested in matching (sub)domains
- ^a+b\*c+\$

- Will match because the + specifies that at least one of the preceding character must be present

**6. Escape character:** The backslash \ denotes that the characters following it have special meaning

Examples:

- \.
  - Represents a period and not a wildcard
  - \/, \\$, etc.
    - Will match a / and \$
  - \d and \D
    - Matches digit\non-digit character
  - \s and \S
    - Matches whitespace\non-whitespace character
  - \w and \W
    - Matches alphanumeric\non-alphanumeric character

**7. Exclusion character:** When contained within [] ^ can denote that the character should not be present in the target string

Examples:

- ^[^abc]\$
  - Will match any single character that is not a, b, or c
  - ^[^n-p]\$
    - Will match any single character that is not n, o, or p
  - ^\w[^og]{2}\$
    - Will match bee and sea but not log, bog, fog, etc.

**8. Negative Lookahead:** (?! ) will specify that the following String should not be present in the match and is useful to, for example, prevent a WalkMe extension from injecting into the wrong iFrame.

Examples:

- ^.\*eu1\.visual\.force\.com\/(?!Console).\*\$
  - Will match the [eu1.visualforce.com](https://eu1.visualforce.com) domain but not [eu1.visualforce.com/Console](https://eu1.visualforce.com/Console)

**9. The Pipe:** | is the logical “OR” operator in RegEx and specifies a target string can match multiple patterns.

Examples:

- `^.*ctmstech\.roche\.com/eclinical.*|.*ctms-performance\.roche\.com.*|.*siebelctmsqa\.kau\.roche\.com.*$`

## Where Regex Can Be Used

**Regex can be inserted into the BBCode of the following step-types and WalkMe items:**

- Visual display;
  - Smart Walk-Thru Step/Popup;
  - ShoutOut;
  - SmartTip (Guidance-type);
  - Survey.
- Automation.
  - Smart Walk-Thru Auto-Step (Fill Text & Select List Value step-types);
  - Smart Walk-Thru Redirect Step.

## Regex in WalkMe

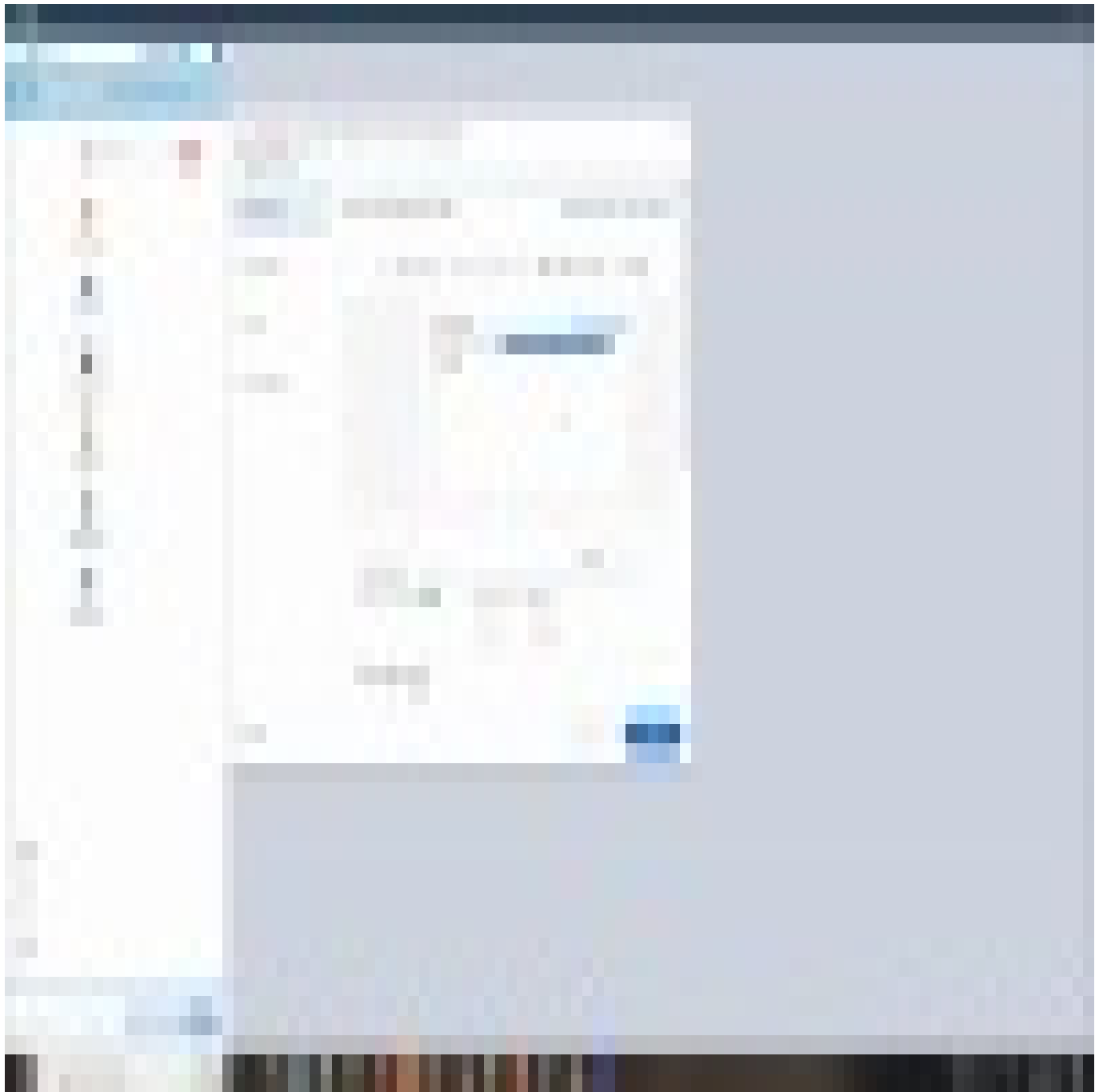
Regex can be used in the following WalkMe instances:

1. Rule Engine for the following rule types:
  - Selected element
  - URL
  - User attributes
  - jQuery
  - On screen element
  - Variable (user data)
2. BBCode

## Tip Tuesday Videos

Using ChatGPT for Regex in WalkMe







Join the conversation

Don't forget to verify all of ChatGPT's suggestions! [Let us know how you've been leveraging OpenAI for your WalkMe content in the Community!](#)