

# Was ist RegEx?

## Kurzübersicht

Reguläre Ausdrücke (Regex) ermöglichen es Ihnen, dynamischen Text zu manipulieren, der in WalkMe-Inhalten angezeigt wird oder Teil automatisierter Prozesse ist. Die Verwendung von [Regex-Syntax](#) ist ideal für Situationen, bei denen Werte von dynamischem Text verwendet werden, die zugleich die gewünschten Werte und weitere Zeichen enthalten, die nicht relevant sind.

## Anwendungsfälle

### **Anwendungsfälle für Regex umfassen Folgendes:**

- Wenn dynamischer Text den Vor- und Nachnamen von Benutzern enthält, verwenden Sie Regex, um nur den Vornamen in einem ShoutOut oder einer Sprechblase anzuzeigen, um es persönlicher zu gestalten (visuelle Anzeige);
- Setzen Sie die E-Mail-Adresse Ihrer Benutzer (wenn sie Teil eines größeren Objektes mit zusätzlichen Daten ist) beim Laden der Seite (Automation) automatisch in ein Textfeld ein;
- Wählen Sie nur einen Teil einer URL aus, wenn Sie dynamischen Text verwenden, um Benutzer automatisch auf eine andere Seite zu leiten.

## Beispiel für die Verwendung

Ein Kunde möchte, dass ein SmartTip angezeigt wird, wenn ein Bildelement (zur Visualisierung eines Datums) vor oder gleich dem 31. Januar 2019 liegt.

### Problembehandlung:

1. Definieren Sie zunächst mögliche Werte, die Sie abgleichen wollen, und Werte, die Sie nicht abgleichen wollen.
2. Gliedern Sie das Problem in mehrere Regelsätze auf, die zusammengenommen alle Werte umfassen, die Sie abgleichen wollen.

1. Im vorliegenden Fall haben wir vier verschiedene Muster, die wir abgleichen wollen:

- i. Alle Termine im Januar 2019
- ii. Alle Termine zwischen dem 1. Januar 2010 und dem 31. Dezember 2018
- iii. Alle Termine zwischen dem 1. Januar 2000 und dem 31. Dezember 2009

iv. Alle Termine zwischen dem 1. Januar 0000 und dem 31. Dezember 1999

3. Formulieren Sie die Regeln für jeden der oben genannten Fälle und testen Sie sie:

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. Nach der Prüfung der einzelnen Gruppen werden diese mithilfe der Pipe zu einer Zeichenkette zusammengefasst (überflüssige `^` und `$`s entfernen)

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. Integrieren Sie das obige RegEx in den Regelsatz des SmartTip-Anzeigesegments mit:

OSE/jQuery > Text ist pro regulärem Ausdruck >

`^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. Ergebnis: SmartTip wird angezeigt, wenn der Textwert des Elements vom 1. Januar 0000 bis zum 31. Januar 2019 reicht!

## Welches RegEx-Format zu verwenden ist

Beginnen Sie mit einem öffnenden RegEx-Tag, das den regulären Ausdruck selbst enthält, gefolgt von der Anweisung, auf die Sie den regulären Ausdruck anwenden möchten, und beenden Sie mit einem schließenden RegEx-Tag:

**[Regex=" {das RegEx selbst} „] {auf das RegEx anzuwendender Text} [/regex]**

**Anmerkung:** {auf das RegEx anzuwendender Text} kann ein Mix aus mehreren dynamischen Textwerten sein (z. B. Variablen, Cookies, WalkMe-Daten und jQuery) und statischen Werten. Der Großteil, wenn nicht alle Textverwendungen in RegEx werden dynamisch sein, mit minimaler Verwendung von statischen Textwerten.

### Struktur

**1. Beginnende und endende Abgrenzungen:** Wenn `^` außerhalb von Klammern steht, bezeichnet es den Beginn eines RegEx und wird zusammen mit dem `$` verwendet, um das Ende des RegEx zu

bezeichnen

Beispiele:

- In einem Erweiterungspaket:  
^https://\ecolab\.my\.salesforce\.com.\*|^https://\ecolab.\*na.\*\.visual\.force\.com.\*
- Bei Verwendung von RegEx in der Regel-Engine: Benutzerdaten > Variable > Abteilung > Ist pro regulärem Ausdruck > ^sales\$

**2. Abgleich eines einzelnen Zeichens:** In der einfachsten Form von RegEx wird ein direkter Zeichenabgleich durchgeführt

Beispiele:

- ^walkme\$
  - Wird „walkme„ entsprechen

**3. Listen und Bereiche:** [] kann zur Angabe einer Liste von Werten verwendet werden und kann in Verbindung mit - einen Wertebereich vorgeben, den ein entsprechendes Zeichen in der Zielzeichenkette annehmen kann

Beispiele:

- ^[dcr][ao][gt]\$
  - Passt zu „Hund,, „Katze,, „Ratte,, „Lappen“
  - Das erste Zeichen muss d, c oder r sein; das zweite Zeichen muss a oder o sein, und das dritte Zeichen muss g oder t sein
  - ^[a-z][0-9]\$
    - Das erste Zeichen muss a bis z (Kleinbuchstaben) sein und das zweite Zeichen muss 0 bis 9 sein

**4. Wiederholungen:** {3} und {#, #} können so eingesetzt werden, dass die gleichen Muster in einer RegEx nicht wiederholt eingegeben werden müssen

Beispiele:

- ^[a-z]{3}\$
  - Stimmt mit drei Kleinbuchstaben in Folge überein, z. B. „adf,, oder „kzd,,.
  - ^[a-z]{1,3}\$
    - Stimmt mit mindestens einem und höchstens drei der vorangehenden Zeichen überein, z. B. „a,, „am,, und „abc,, , aber nicht „duct,,.

**5. Platzhalter:** Sie sind hilfreich, wenn der Text in der Zielzeichenkette dynamisch ist oder wenn der Inhalt keine Rolle spielt. . passt zu jedem beliebigen einzelnen Zeichen, \* passt zu null oder mehr des vorhergehenden Musters und + passt zu einem oder mehreren des vorhergehenden

Musters.

Beispiele:

- `^.*$`
  - Passt auf keine oder mehrere beliebige Zeichen und kann am Anfang und am Ende einer RegEx eingefügt werden (insbesondere innerhalb von Erweiterungen), da wir nur an passenden (Sub-)Domains interessiert sind
  - `^a+b*c+$`
    - Entspricht, da + angibt, dass mindestens eines der vorangehenden Zeichen vorhanden sein muss

**6. Escape-Zeichen:** Der Backslash `\` bedeutet, dass die nachfolgenden Zeichen eine besondere Bedeutung haben

Beispiele:

- `\.`
  - Steht für einen Punkt und nicht für einen Platzhalter
  - `\V`, `\\$`, etc.
    - Passt zu `a /` und `$`
  - `\\d` und `\\D`
    - Entspricht einem numerischen und einem nicht numerischen Zeichen
  - `\\s` und `\\S`
    - Passt zu einem Leerzeichen\Nicht-Leerzeichen
  - `\\w` und `\\W`
    - Passt zu einem alphanumerischen und nicht alphanumerischen Zeichen

**7. Ausschlusszeichen:** Innerhalb von `[]` kann `^` bedeuten, dass das Zeichen in der Zielzeichenkette nicht vorkommen soll.

Beispiele:

- `^[^abc]$`
  - Passt zu jedem einzelnen Zeichen, das nicht a, b oder c ist
  - `^[^n-p]$`
    - Passt zu jedem einzelnen Zeichen, das nicht n, o oder p ist
  - `^[^og]{2}$`
    - Passt zu `bee` und `sea`, aber nicht zu `log`, `bog`, `fog` usw.

**8. Negative Vorausschau:** `(?!)` gibt an, dass die folgende Zeichenkette im Abgleich nicht vorhanden sein soll, und ist nützlich, um z. B. zu verhindern, dass eine WalkMe-Erweiterung in den falschen iFrame eingespeist wird.

Beispiele:

- `^.*eu1\.visual\.force\.com\(?!\Console).*`
  - Passt zur Domäne [eu1.visualforce.com](https://eu1.visualforce.com), aber nicht zu [eu1.visualforce.com/Console](https://eu1.visualforce.com/Console)

**9. Pipe:** | ist der logische „OR“-Operator in RegEx und gibt an, dass eine Zielzeichenfolge mit mehreren Mustern übereinstimmen kann.

Beispiele:

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

Wo Regex verwendet werden kann

**Regex kann in den BBCode der folgenden Schritttypen und WalkMe-Objekte eingefügt werden:**

- Visuelle Anzeige;
  - Smart Walk-Thru-Schritt/Popup
  - ShoutOut
  - SmartTip (Anleitungstyp);
  - Survey
- Automatisierung.
  - Smart Walk-Thru Auto-Schritt (Text einfügen und Listenwert-Schritttypen auswählen);
  - Umleitungsschritt für Smart Walk-Thrus

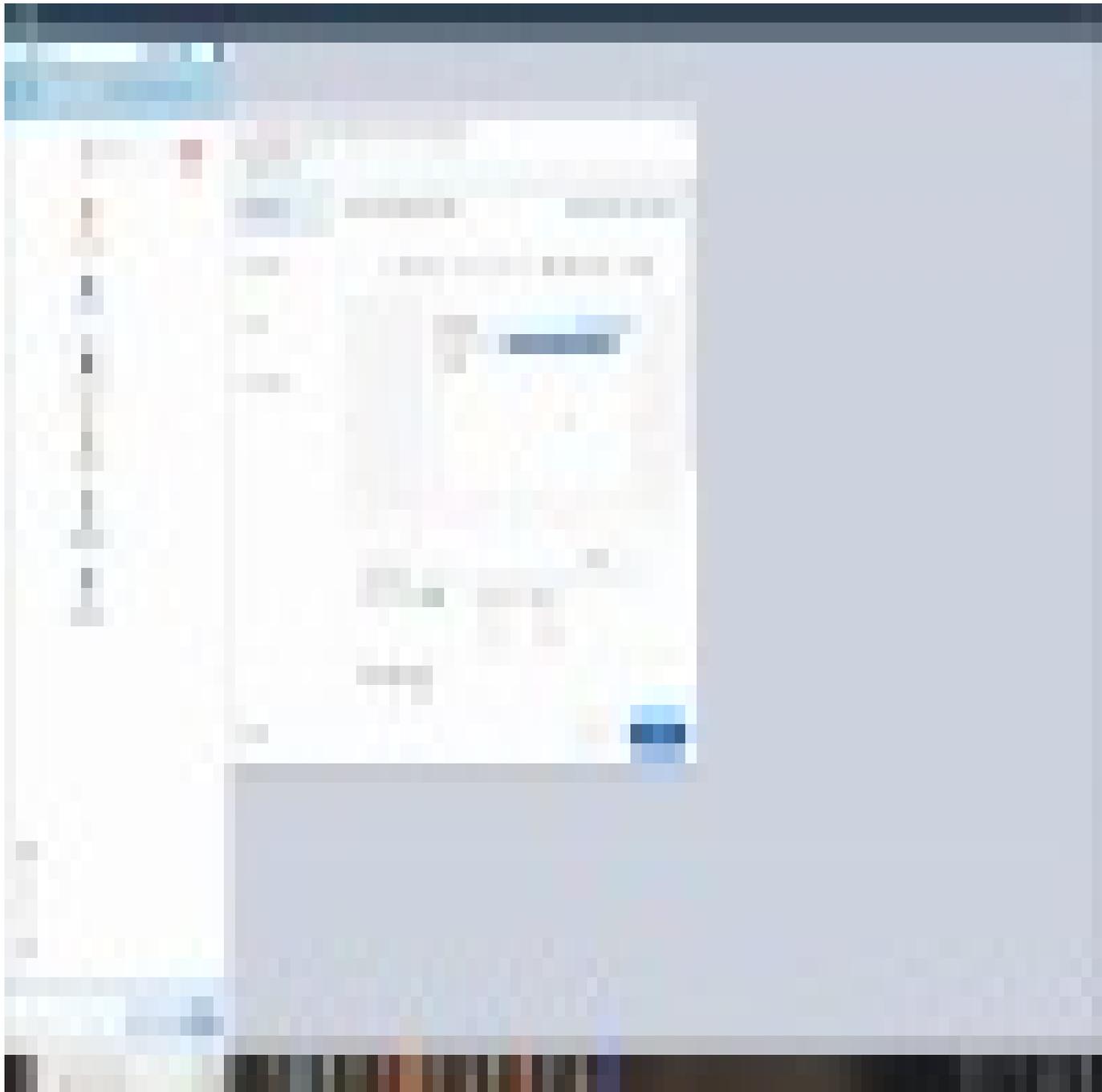
## Regex in WalkMe

Regex kann in den folgenden WalkMe-Instanzen verwendet werden:

1. Rule Engine für die folgenden Regeltypen:
  - Ausgewähltes Element
  - URL
  - Benutzerattribute
  - jQuery
  - Auf dem Bildelement
  - Variable (Benutzerdaten)
2. BBCode

## Tipp - Dienstag-Videos

Verwenden von ChatGPT für Regex in WalkMe



Beteiligen Sie sich am Gespräch

Vergessen Sie nicht, alle Vorschläge von ChatGPT zu überprüfen! [Teilen Sie uns mit, wie Sie OpenAI für Ihre WalkMe-Inhalte in der Community genutzt haben!](#)