

jQuery Selectors

jQuery should be used for edge cases only

We recommend using jQuery as little as possible. Web applications are constantly changing and small updates to the UI and page structure can break any jQuery used.

WalkMe's out of the box element selection adapts to changes in the underlying application without the use of jQuery.

Brief Overview

jQuery is a Javascript library that can be used to manipulate a page of HTML. jQuery selectors allow you to identify elements in the HTML structure based on their characteristics such as ID, class, type, attributes, or values of attributes.

The most common jQuery selectors in the WalkMe Editor are used to:

- **Create Onboarding Goals:** Create Onboarding goals based on an On Screen Element
- **Selected Element Launchers:** Add the jQuery Selector into the Selected Element tab of a Launcher
- **Grab Values:** Grab a value from a user variable or a text on the screen to be incorporated into a dynamic announcement or use it in a rule
- **Check Page Conditions:** Direct WalkMe to analyze the attributes of an element on the screen to use as a Rule Type in the Rule Engine

How It Works

Find the element in the HTML by right-clicking on the element and selecting Inspect Element. The element will be highlighted on the screen and in the developer panel. Identify the element's unique characteristics (such as location, class, ID, or style) in order to build your jQuery selector. Elements can be identified by their location or attributes (class, ID, and style).

□ **Digital Adoption Institute**

- View the [Basic and Advanced jQuery](#) courses
- Don't have a DAI account yet? [Sign up here](#)

Step 1: Build the jQuery selector

The basic syntax for using jQuery elements in WalkMe is: **Element → Selector → Value**

After identifying the element, build a jQuery Selector that will look at specifically what you are interested in.

Here are some examples of jQuery Selectors:

- ul element with id create-account: `ul#create-account`.
- ul elements with class nav: `ul.nav`.
- ul elements with class nav and class fl type: `ul.nav.fl`.

[See a list of useful jQuery Selectors below.](#)

Create jQuery using your WalkMe Data

You can now utilize existing WalkMe Data in favor of a jQuery selector.

This can be accomplished by adding '`${wm-data-key}`' to the desired selector. For example: the selector: `[id="${key}"]`, will have the '`${key}`' replaced by the actual WalkMe data and will look like: `[id="value"]`.

Step 2: Test your selector in the console

1. Type `wmjQuery("")` in the command line in the developer panel with your jQuery selector between the "" and click **Enter**
 - **Note:** You'll only be able to use `wmjQuery` in a browser with the Editor Extension enabled
</nas/content/live/kbwalkme/wp-content/uploads/2017/09/jquery-console-video.mp4>
2. If your query returns the desired value, your selector is working!
 - **Note:** To see the selected elements on the screen, click **More...**
3. Hover over the returned objects to see the highlighted element

Step 3: Use jQuery selectors in WalkMe

</nas/content/live/kbwalkme/wp-content/uploads/2017/09/walkme-editor.mp4>

Depending on how you would like to use the jQuery element, jQuery can be used in WalkMe in the following locations:

- Use the jQuery element to identify an element for a step or Launcher using the Selected Element tab of the Options Menu
- Use the jQuery element to trigger the next step in a Walk-Thru when clicked or hovered in the Additional Step Triggers Menu

Note

Using a non-unique selector to detect click or hover on multiple elements at once might not work as expected.

- Use the jQuery element to grab a value on screen to use as Dynamic Text in the Rich Text Editor in a step balloon or ShoutOut
- Use the jQuery element to build a goal in the Goal tab
- Use the jQuery Selector in the Rule Engine as a Rule Type (similar to jQuery is used similarly to On Screen Element Rule Type)

Tip

Remember that jQuery should be used sparingly in your WalkMe builds, and always in combination with lighter conditions. Please refer to the following article for more information: [WalkMe Performance Best Practices](#)

Available Rule Engine jQuery Operators:

- **True:** Means that at least one element was found
- **False:** Means that no elements were found
- **Visible:** Means that the element is visible
- **Text Is / Is Not:** Means text is equal to text input
 - **Note:** When the selector returns more than one element, all elements' text are combined as one element
- **Greater Than / Less Than:** Means value is above or below a whole number
- **Like / Not Like:** Means a complicated text syntax
- **Length Is / Is Not / Greater Than / Less Than:** Means equal to, above or below a whole number
 - Used when to query how many elements the query returns
 - Discover exact query length in the console: `wmjQuery("element.value").length`

Note

WalkMe's Editor only supports selectors written in jQuery syntax up to that used in jQuery 1.8.3. However, you may use any version of jQuery for other purposes on your website without fear of a conflict with WalkMe's version of jQuery (wmjQuery).

We recommend that you also utilize [WalkMe's jQuery Selector Optimizer](#).

Useful jQuery Selectors

Query	Description
element	Query an element
element[attribute]	Query an attribute of an element
element[attribute][attribute]	Query multiple attributes of an element
element[attribute="attribute value"]	Query an attribute of an element and its value
element[attribute^="text"]	Query text at the beginning of an attribute
element[attribute*="text"]	Query text contained in an attribute
element[attribute!="text"]	Query text excluded from an attribute
element child-element	Query the child of an element (space indicates that the first element is the parent of the second)
element:checked	Query the status of a checkbox (checked or unchecked)
[element=""]:visible	Query if an element is visible
element:contains(Text)	Query element contains text
element:eq(#)	Query specific result in order (count starts from 0)

jQuery Cheatsheet

For more jQuery tips and examples, please refer to our [jQuery Cheatsheet](#)

Testing Your jQuery Selectors

When choosing a jQuery selector, it is wise to compare a few to ensure you are picking the one with the least performance impact on your website/application. To do so, please follow these steps:

1. Open your web browser's dev tools
2. Navigate to the dev tools' console tab
3. Type the following into the console:
 - **`var time1 = window.performance.now(); wmjQuery('<your jQuery selector>');`**
`var time2 = window.performance.now(); console.log(time2 - time1);`
4. Click **Enter**
5. Run this command several times in the console to see the approximate average of the time it takes
6. Run this same command with some alternative jQuery selectors and choose the one that takes the least time to run

Note

- If Shadow DOM support, you should then be able to capture Shadow DOM elements in the Editor and use jQuery to identify them
- However, you won't be able to run `wmjQuery` in the DevTools Console, you'll need to use this syntax instead:

```
_walkmeInternals.ctx.get('jQuerySelectorRunnerShadow').run('enter your selector here')
```