WalkMe jQuery Cheatsheet

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.

Helpful Tips

- **Console:** This is generally the word used to describe the Developer Tools, specifically the Elements section that shows all the HTML that exists on a page.
    - This can be confusing since there is also a Console section where you enter your wmjQuery to test your selectors. These two sections are usually referred to interchangeably, but if someone is asking you to enter something into the Console, they are referring to the Console tab in the Developer Tools.
    - To access use F12 or go up to your browser menu and select Developer Tools.
- **Capitalization Matters:** All jQuery is case sensitive. If you are typing in the wrong upper or lower case letter, you will not be able to match the desired selector.
- **Always Test:** You should always test out your jQuery via the Developer Tools Console Section before using it.
    - Use the command: wmjQuery("insert_your_selector_here")

jQuery Selectors

- **Elements:** Usually shown in the Developer Tools Elements Section as Pink/Purple writing (as well as on the page upon hovering in the Elements Section). They do not have any special characters to denote them and should be written as a word.
    - Common Examples: div, span, script, head, body, input, li, a, img
    - Example Lookup Using WalkMe jQuery:
        - wmjQuery("span")
- **IDs:** Upon hovering over this selector in the Developer Tools Elements Section, the writing will appear next to the inspected item on screen in Orange led by a "#"
    - IDs are amongst the strongest selectors that we can use since they are usually made by developers to be unique identifiers. Due to this, you will often see IDs with a random

number after them. This situation is known as a Dynamic ID. WalkMe selection, we tend to avoid these as they may change from login to login and are not reliable selectors.
    - There are no common examples since these will be unique site to site. You may see them appear like this: #header, #logo, #searchbar
    - Example Lookup Using WalkMe jQuery:
        - wmjQuery("#logo")
- **Class:** Upon hovering over this selector in the Developer Tools Elements Section, the writing will appear next to the inspected item on screen in Blue led by a "."
    - Classes are very common and are reused throughout a site. You will want to ensure that the class you've chosen only shows once on the page you're intending to select on or that you've told WalkMe specifically which class of the multiple on the page that you're referring to.
    - There are no common examples since these tend to be unique site to site. You may see them appear like this: .btn, .dropdown, .navlink
    - Example Lookup Using WalkMe jQuery:
        - wmjQuery(".btn")
- **Attributes** – you will not see these by hovering and will have to look for them in the Developer Tools Elements Section.
    - They are used to describe a selector and thus should be used in conjunction with an Element, ID, or Class for performance reasons. If used by themselves, WalkMe will have to search through every selector on the page to see if it matches.
    - Common Exampes: value, name, type
    - Example Lookup Using WalkMe jQuery:
        - wmjQuery("#example[name='Website User']")
        - Remember: Always put an Element/ID/Class if front of the attribute as shown in this example.

Special Modifiers

- **Wildcard Usage:** You can use a wildcard to make your selection a little less specific.
    - This is useful in areas where you want to select multiple buttons like a save button but they are a little different from each other. It's also useful for dynamic elements where a part of them stays consistent.
    - Example Lookup Using WalkMe jQuery:
        - wmjQuery("#example[name*='Website']")
        - This will still match the 'Website User' that was used in the attribute example.
    - This can also be used for ID/Class selectors. HTML selectors in the Developer Tools Elements Section show as [class='btn'] as compared to the .btn that you will see onscreen when you hover over the HTML. This means our wildcard can apply here as well.
        - Example Lookup Using WalkMe jQuery:
            - wmjQuery("[class*='btn']")
            - This will still match all the .btn selectors on the page even if it's .btn1234 or

.btntab

- **Relation Selectors:** You can use how selectors are related to one another to make your jQuery more specific.
    - **Descendant:** Denoted using a space
        - Example Lookup Using WalkMe jQuery:
            - wmjQuery("Input .btn")
            - This will return the all btn classes that are descended from an Input element. In the Developer Tools Elements Section, you will see your desired selector indented to the right of the progenitor selector that is above it.
    - **Sibling:** Denoted using a plus sign
        - Example Lookup Using WalkMe jQuery:
            - wmjQuery("Input+.btn")
            - This will return the all btn classes that are siblings with an Input element. In the Developer Tools Elements Section, you will see your desired selector below the sibling and share the same indent level. Always have the sibling element before your desired element.
- **Commands:** You can add commands at the end of your jQuery statement to make your search more precise. These are added by adding a colon followed by the command.
    - **First Selector:** *yourselector***:first**
        - Example Lookup Using WalkMe jQuery:
            - wmjQuery(".btn:first")
            - This will return the first btn class on the current page.
    - **Last Selector:** *yourselector***:last**
        - Example Lookup Using WalkMe jQuery:
            - wmjQuery(".btn:last")
            - This will return the last btn class on the current page.
    - **Selecting by Selector Number:** *yourselector***:eq(***number_of_element***)**
        - Example Lookup Using WalkMe jQuery:
            - wmjQuery(".btn:eq(2)")
            - This will return the 3rd btn class on the current page. It's 3rd because the first selector is labeled as 0. So, in this case, you would have classes btn 0, btn 1, btn 2.
    - **Selecting by Text:** *yourselector***:contains(***"desired selector's text"***)**
        - Example Lookup Using WalkMe jQuery:
            - wmjQuery(".btn:contains('Save')")
            - This will return all btn classes on the current page that have the text Save contained within them. Be sure to open the selector in the Developer Tools Elements Section using the carrot symbol to the left of a line. You'll find the text on the inner most line of
              selectors.
    - **Selecting by Selectors:** *yourselector***:has(***"selectors inside the desired selector"***)**
        - Example Lookup Using WalkMe jQuery:
            - wmjQuery("Input:has('.btn')")

- This will return all Input Elements on the current page that have btn class as a descendant. Be sure to open the selector in the Developer Tools Elements Section using the carrot symbol to the left of a line. You'll find the selectors that are descending from your desired selector.

- **Visible Selector:** *yourselector*:**visible(***number of element***)**
  - Example Lookup Using WalkMe jQuery:
    - wmjQuery(".btn:visible")
    - This is used when there are hidden versions of your desired selector on a page. An example would be on Single Page Applications where you can switch between tabs. This means that the element is still technically on the page even if it's hidden from view.

jQuery Selectors for WalkMe Player Components

This table is helpful for anyone wanting to create WalkMe content that interacts with the WalkMe Player (most frequently SC, PS & CS).

*Replace the XXXXXX's with component's relevant WalkMe ID

Component	**jQuery**
WalkMe Widget	div#walkme-player
WalkMe Menu	div#walkme-menu
WalkMe Help Tab (**Selected**)	div[class='walkme-**XXXXXX**-tab-button walkme-css-reset walkme-tab-button walkme-first-tab walkme-override walkme-css-reset walkme-tab-button-selected']
WalkMe Help Tab (**Not Selected**)	div[class='walkme-**XXXXXX**-tab-button walkme-css-reset walkme-tab-button walkme-first-tab walkme-override walkme-css-reset walkme-tab-button-not-selected'
WalkMe Tasks Tab (**Selected**)	div[class='walkme-**XXXXXX**-tab-button walkme-css-reset walkme-tab-button walkme-override walkme-css-reset walkme-tab-button-selected']
WalkMe Tasks Tab (**Not Selected**)	div[class='walkme-**XXXXXX**-tab-button walkme-css-reset walkme-tab-button walkme-tab-button-not-selected walkme-override walkme-css-reset']
WalkMe Search Bar	div[class='walkme-search-box-container.walkme-override.walkme-css-reset']
WalkMe Progress Bar	div[class='walkme-progress-bar-wrapper walkme-loading-hide walkme-override walkme-css-reset'] **OR** div#walkme-progress-bar

WalkMe Search Bar Search Icon	div[class='walkme-search-box-button walkme-loading-hide walkme-override walkme-css-reset']
WalkMe Onboarding Task	div.walkme-deployable-name.walkme-name.walkme-override.walkme-css-reset:contains('**TASK NAME**')
WalkMe Player Category	div[class='walkme-deployable-row walkme-override walkme-css-reset']
WalkMe Languages	div[class='walkme-current-language walkme-override walkme-css-reset'] **OR** div#walkme-languages

iFrame jQuery

The jQuery selector template below allows WalkMe to target an element in an iFrame. Update the bold items as seen in the example with the respective selectors:

```
{"element": "#yourElement" ,"context": "iframe#iframeselector"}
```

Example: {"element": "**h2.mainTitle**" ,"context": "**iframe#Main**"}

To evaluate the selector in the Dev Tools Console you would need to use the wmjQuery template below:

```
wmjQuery("#yourElement",wmjQuery("iframe#iframeselector").contents())
```

Digital Adoption Institute

- View the ***jQuery* course.**
- Don't have a DAI account yet? Sign up here

jQuery Webinar

Tip: View more helpful webinars at the **WalkMe World Community**.