USE CASE

- jQuery and CSS selectors can be used to control when Whatfix content displays for any given user based on the presence or absence of other page content (e.g. text, icons, active or inactive buttons, links, dropdown menus, etc.). For example:
- Beacons can be made visible only for users who have more than one document loaded.
    - Smart Tips can be activated for users who have skipped a form field.

NOTES

- The color scheme in this document refers only to the HTML code in the Elements section of the Chrome code inspector box. This color scheme is not present in the Console section or in the selector itself, and it differs slightly across other browsers.
- These instructions apply to jSign but may not apply to other websites or applications whose programming doesn't allow for these types of queries.
- Identifying appropriate selectors for use in Whatfix display rules can be challenging, particularly when you are still learning the process. It is important to experiment and think carefully about how you want your Whatfix content to behave.

REQUIREMENTS

1. You must identify an appropriate selector for the element of interest. Generally, you will want to identify a selector that is only associated with that particular element.
2. You must create a display rule for the Whatfix content/widget using that selector.

PROCEDURE #1: IDENTIFY A SELECTOR

1. Consider whether any element or bit of text on the web page appears or disappears when the Whatfix content/widget should also appear or disappear. For example:
    a. To make a beacon visible only for users with more than one document, find a bit of text indicating how many documents are uploaded or an element that only appears/disappears when more than one document is uploaded.
    b. To activate a Smart Tip when a user skips a form field, recreate that by skipping that form field and completing the following field.
2. Right-click on your element and select *Inspect.* The code inspector box will appear.
3. Examine the highlighted line of code in the code inspector.
    a. The line of code begins with < which is followed by the element's tag in purple font.
    b. The element's attributes appear in red font.
    c. The content of each attribute appears in blue within quotation marks.
    d. Any non-image text or numeric value that displays on the web page appears in black.
4. Construct a jQuery referencing one or more of the element's features listed above.
    a. Double-click on an attribute="content" combination within the element's line of code, select the entire combination with your mouse, then copy it. Avoid copying content that includes numbers, since those may be dynamic resulting in an unreliable selector.
        i. Tags, attributes, and values can include numbers.
    b. Click on *Console* in the top menu of the code inspector box.

c. Clear the console with Ctrl+L or by clicking the button in the top left.
d. Type in the following function:

```
jQuery('[]')
```

e. Paste your copied attribute="content" combination inside the two brackets.

```
jQuery('[attribute="content"]')
```

5. Hit *Enter.*
6. Determine whether the jQuery selector you selected is unique.
    a. The code returned after hitting *Enter* should begin with *w.fn.init*
        i. Red errors indicate a structural issue with the code you entered. Ensure the capitalization, order, punctuation, and spacing of your code match the code.
    b. Click on the small black triangle icon at the left of the returned code. You will see either:
        i. three lines of code with the first reading "length: 0" which means the jQuery could not identify any corresponding elements, or
        ii. a list of elements. Toward the end of that list will be a line reading "length: " followed by a number. That number indicates how many elements were identified by the jQuery selector.
    c. The jQuery selector is unique if there is only one element in the list.
        i. The line of code will begin with "0**:** " and the next line will read "length**:** 1".
7. If the jQuery selector you constructed is not unique, you can either:
    a. Use the jQuery selector in your Whatfix rule
        i. Depending on how your rule should function, a jQuery selector that identify more than one element may be acceptable.
    b. Search for a different unique element
    c. Refine your current jQuery selector in one of three ways:
        i. You can combine the attribute="content" with any number of parent attribute="content" combinations. However, it is best to aim for the fewest number possible.
            1. The code for a child element is nested within its parent element's code. The parent attributes and content will thus come from a line of code above the child element. For example:
                a. If this document were code and this line an element, line 7ci1 would be its parent, 7ci its grandparent, etc.

```
jQuery('[grandparentattribute="content"] [parentattribute="content"] [childattribute="content"]')
```

        ii. You can include a value that is (or is not) in the element's code.

```
jQuery('[attribute="content"]:contains(value)')

jQuery('[attribute="content"]:not(:contains(value))')
```

        iii. You can include a tag associated with the element.

```
jQuery('tag[parentattribute="content"] [attribute="content"]:not(:contains(value))')
```

PROCEDURE #2: CREATING WHATFIX RULES

1. Once you have identified an appropriate jQuery selector, you must create a Whatfix display rule.
2. Depending on the type of Whatfix content/widget and the rule type, you can select the following rule parameters from the display rule dropdown menus:
    a. *Other Element on Page* combined with either *Exists jQuery Selector* or *Not Exists jQuery Selector*
        i. Determines whether the element referenced by the selector is on the page
    b. *Action Element is* combined with *jQuery Selector*
        i. Determines whether a user clicks on the element referenced by the selector
3. When setting up the rule, you should copy and paste the selector from the code inspector. You will only copy what appears within the opening/closing brackets. The parentheses and single quotation marks are not part of the selector.

Do not include in rules: *jQuery*(' ')

For this jQuery selector: *jQuery*('[attribute="content"]')

Type this into Whatfix: [attribute="content"]

For this jQuery selector: *jQuery*('[attribute="content"] [attribute="content"]')

Type this into Whatfix: [attribute="content"] [attribute="content"]

For this jQuery selector: *jQuery*('[attribute="content"]:contains(value)')

Type this into Whatfix: [attribute="content"]:contains(value)

For this jQuery selector: *jQuery*('[attribute="content"]:not(:contains(value))')

Type this into Whatfix: [attribute="content"]:not(:contains(value))

For this jQuery selector: *jQuery*('tag[attribute="content"] tag[attribute="content"]')

Type this into Whatfix: tag[attribute="content"] tag[attribute="content"]