Sometimes there are issues that the jQuery core team believes can't be adequately fixed. Typically the reasons fall into one or more of these categories:

- Lack of browser APIs to detect and fix the problem;
- Conflicts--for example, fixing a bug in Browser A creates one in Browser B;
- Severe breakage to jQuery APIs are required to fix the problem;
- Serious performance implications of fixing a relatively rare problem; or
- Significant amounts of code are required to fix the problem.

The issues below are ones that have been repeatedly raised over time that the team has decided not to fix. If you have specific new ideas for a solution that have not already been considered, please open a new ticket.

## Modifying Object.prototype

Modifying JavaScript's `Object.prototype` by adding new properties or methods has long been acknowledged as a bad idea; it'll break a lot of code, and not just in jQuery.

**For further reference**:

- Object.prototype is verboten
- trac-2721: Object.prototype

## SVG or VML

jQuery is primarily intended as a library for the HTML DOM, so most problems related to SVG/VML documents are out of scope. Although XML-based markup such as SVG may *look* like HTML, its behavior is often very different and browsers do not support the same APIs on them. We do try to address problems that "bleed through" to HTML documents, such as events that bubble out of SVG embedded into HTML. As of jQuery 3.0 we also allow class name operations on SVG elements.

**For further reference**:

- trac-4208: Selector inconsistency with XML Namespaces
- trac-7584: SVG support metaticket
- trac-9807: IE7/8 alerts a 'Failed' error on VML objects
- trac-13092: vendorPropName on a namespaced element
- gh-2199: jQuery Class Operations on SVG DOM Attributes
- gh-2889: Chrome is deprecating offsetWidth and offsetHeight

## Console warnings and breakpoint on exceptions

Some versions of Chrome and Firefox will generate warnings on the console when specific properties are accessed. Browsers may also throw exceptions that jQuery catches and handles. Both are part of jQuery's cross-browser feature detection. When using browser dev tools, you may encounter a breakpoint inside jQuery if you have enabled "Break on caught exceptions" but this is normal.

**For further reference**:

- trac-7535: Firefox Console error: Unexpected token in attribute selector
- trac-10531: Consider removing layerX and layerY from $.event.props
- trac-13881: IE10 SCRIPT5022: Caught exception occurred : SyntaxError
- gh-2889: Chrome is deprecating offsetWidth and offsetHeight

## CSS Transition/Animation/Transforms

Modern browsers provide native means of doing CSS animations and transforms just by adding or removing class names. jQuery's `.animate()` method is best saved for more complex animations where the CSS mechanisms are insufficient. If you wish to take advantage of CSS animations through jQuery's `.animate()` method, we recommend a plugin.

**For further reference**:

- trac-4171: Allow animation of CSS Transform Properties

Plugins available:

- jQuery Animate Enhanced
- jquery.transform.js
- jQuery++ Animate

## Native Selector Bugs

Browsers have inconsistencies with their implementations of `querySelectorAll`, which bleed through to jQuery. Thankfully these issues are usually fixed quickly. If they aren't that severe then we just bump them back to the browser vendor and let them handle it. We also are hesitant to "fix" cases that aren't specified by either a formal standard or (at minimum) *de facto* agreement by browsers.

**For further reference**:

- trac-9690: nth-child + class does not work in Firefox 5
- gh-2670: :target doesn't match elements on initial page load in Chrome and Firefox

## Focus and Blur Event Order

The `focus`, `blur`, `focusin`, and `focusout` events occur in different orders in different browsers, and it is not practical for jQuery to impose a consistent order--especially when most browsers do not follow any standard. In addition, IE11 `focus` and `blur` events are asynchronous; the events don't fire on the same thread as where they are triggered.

**For further reference**:

- gh-3123: Shim focusin/focusout in all browsers?
- W3C Focus Event Order

## Inline Event Handlers

In general, inline event handlers are incredibly hard to work with and should not be used. In nearly all cases, issues involving inline event handlers are out of scope. Inline handlers do not behave the same way as event handlers added with the standard DOM `addEventListener` method.

**For further reference**:

- gh-2918: Removing the parent node during event bubbling
- Quirksmode - Bad Practices

## Negative margins on Android 4.0-4.3 browser

In Android 4.0-4.3, the DOM `getComputedStyle()` method sometimes returns percents instead of pixel values. The workaround for this bug in jQuery leads to another issue that causes the return of incorrect values when retrieving negative margins.

**For further reference**:

- trac-1815: .css method return wrong value for negative percent margin

## Quirks Mode

Some browsers provide a "Quirks Mode" that does not conform to the standard W3C CSS box model; jQuery does not support this mode due to the differences it introduces in measuring element dimensions and positions. To ensure that an HTML document does not render in Quirks Mode, always use a doctype at the top of the document.

**For further reference**:

- Quirks Mode on Wikipedia

## `outerHeight()` behavior for tables with captions

The spec is unclear whether `getComputedStyle(table).height` should take table's caption into account or not. Firefox doesn't include it while Chrome does. As long as it's not clear which of those behavior is expected, we're not going to normalize in either direction.

**For further reference**:

- A comment on the jQuery issue #4105 summarizing the situation
- A Firefox issue

## Passing `documentFragment` to the API for manipulation or traversal

jQuery uses document fragments internally and doesn't expect them to be passed as API inputs.

**For further reference**:

- trac-9903 - Slow DOM manipulation because .append() can't use documentFragments
- trac-11566 - node.append et al. does not work when node is a DocumentFragment
- trac-13545 - .find breaks with documentfragments
- #4317 - Is not possible to listen events on ShadowRoot