

Plugins (<https://plugins.jquery.com/>)

Contribute (<https://contribute.jquery.org/>)

Events (<https://events.jquery.org/>)

Support (<https://jquery.com/support/>)

OpenJS Foundation (<https://openjsf.org/>)

**Download** (<https://jquery.com/download/>)

Search



**API Documentation** (<https://api.jquery.com/>)

**Blog** (<https://blog.jquery.com/>)

**Plugins** (<https://plugins.jquery.com/>)

**Browser Support** (<https://jquery.com/browser-support/>)

**Version Support** (<https://jquery.com/support/>)

# jQuery Core 3.5 Upgrade Guide

## jQuery.htmlPrefilter changes

### A workaround

If you want to upgrade to jQuery 3.5.0 or newer and don't have time to deal with breaking changes at the moment, and you use jQuery Migrate 3.4.0 or newer, you can revert to the previous behavior by invoking:

```
1 jQuery.migrateEnablePatches( "self-closed-tags" );
```

As long as this method has been called, jQuery Migrate 3.4.0 or newer logs warnings for all input that's affected by this change. **Note:** if you overwrite `jQuery.htmlPrefilter` manually, you'll lose those warnings!

If you don't use jQuery Migrate, don't add it just for this one workaround. Instead, you can revert to the previous behavior by redefining `jQuery.htmlPrefilter` after loading jQuery:

```
1 var rxhtmlTag = /<(?!area|br|col|embed|hr|img|input|link|me
2 jQuery.htmlPrefilter = function( html ) {
3     return html.replace( rxhtmlTag, "<$1></$2>" );
4 };
```

Note that if you do this, you lose the jQuery 3.5.0 security fix, and you have to be more careful with what HTML you pass to jQuery manipulation methods; regular HTML sanitizing will not be enough. Some security libraries have special sanitization settings for jQuery. For example, [DOMPurify](https://github.com/cure53/DOMPurify/tree/2.0.17#can-i-configure-dompurify) (<https://github.com/cure53/DOMPurify/tree/2.0.17#can-i-configure-dompurify>) used to support the `SAFE_FOR_JQUERY` flag in versions 2.0.17 or older:

```
1 var sanitizedHtml = DOMPurify.sanitize( unsafeHtml, { SAFE_
2 elem.html( sanitizedHtml );
```

## Description of the change

`jQuery.htmlPrefilter` modifies and filters HTML strings passed through [jQuery manipulation methods](https://api.jquery.com/category/manipulation/) (<https://api.jquery.com/category/manipulation/>). The default value of this function in jQueryes older than 3.5.0 used to replace XHTML-like tags with versions that work in HTML. For example, previously the following:

```
1 jQuery( "<div/><span/>" );
```

would create the following structure:

```
1 <div></div>
2 <span></span>
```

because `<div/>` was replaced with `<div></div>` & `<span/>` with `<span></span>`.

In jQuery 3.5.0, the `jQuery.htmlPrefilter` method always returns its argument unchanged. Modern browsers usually parse and render HTML strings in HTML mode. Elements such as `<p>` and `<span>` are never self-closing in HTML, so the browser interprets a string like `<p/><span/>` as `<p><span>` and leaves the tags open. The browser closes the tags at the end of the string or at an element that cannot be contained in the element, so another `<p>` tag closes an existing open `<p>` tag.

```
1 <div>
2   <span></span>
3 </div>
```

To avoid this, don't use self-closing tags for tags that may have content unless your page runs in XHTML mode. Make sure you're sending a correct mime type: `application/xhtml+xml`; otherwise, your page will really run in HTML mode.

If you're writing a library, and you want it to work both in HTML & XHTML modes, remember to use self-closing tags for empty elements, i.e. ones that don't have closing tags in HTML. For example, instead of:

```
1 jQuery( "<div/><img/>" );
```

do this:

```
1 jQuery( "<div></div><img/>" );
```

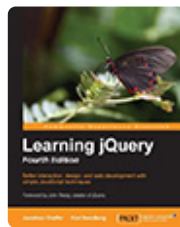
One popular input that still works in jQuery 3.5.0 or newer is the one with a single tag with or without attributes:

```
1 jQuery( "<div class='yellow' />" );
```

This is because it's XHTML-compliant and in HTML the parser first changes it to just the opening tag: `<div class='yellow'>` but then immediately auto-closes it as it reaches the end of input.

---

## BOOKS



Learning jQuery Fourth Edition  
Karl Swedberg and Jonathan Chaffer

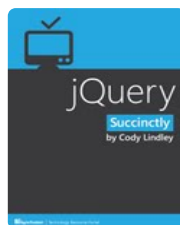
(<https://www.packtpub.com/en-us/product/learning-jquery-fourth-edition-9781782163152>)



jQuery in Action

Bear Bibeault, Yehuda Katz, and Aurelio De Rosa

(<https://www.manning.com/books/jquery-in-action-third-edition>)



jQuery Succinctly  
Cody Lindley

(<https://www.syncfusion.com/ebooks/jquery>)

---

Copyright 2026 OpenJS Foundation (<https://openjsf.org/>) and jQuery contributors. All rights reserved. See jQuery License (<https://jquery.com/license/>) for more information. The OpenJS Foundation (<https://openjsf.org/>) has registered trademarks and uses trademarks. For a list of trademarks of the OpenJS Foundation (<https://openjsf.org/>), please see our Trademark Policy (<https://trademark-policy.openjsf.org/>) and Trademark List (<https://trademark-list.openjsf.org/>). Trademarks and logos not indicated on the list of OpenJS Foundation trademarks (<https://trademark-list.openjsf.org/>) are trademarks™ or registered® trademarks of their respective holders. Use of them does not imply any affiliation with or endorsement by them. OpenJS Foundation Terms of Use (<https://terms-of-use.openjsf.org/>), Privacy (<https://privacy-policy.openjsf.org/>), Code of Conduct (<https://code-of-conduct.openjsf.org/>), and Cookie (<https://www.linuxfoundation.org/cookies>) Policies also apply.

 [Learning Center \(https://learn.jquery.com/\)](https://learn.jquery.com/)

 [Chat \(https://jquery.com/support/\)](https://jquery.com/support/)

 [Twitter \(https://twitter.com/jquery\)](https://twitter.com/jquery)

 [GitHub \(https://github.com/jquery\)](https://github.com/jquery)

Web hosting by Digital Ocean (<https://www.digitalocean.com>) | CDN by Fastly (<https://www.fastly.com/>) | Powered by WordPress (<https://wordpress.org>)