Search...

Tooltips

Documentation and examples for adding custom Bootstrap tooltips with CSS and JavaScript using CSS3 for animations and data-attributes for local title storage.

Overview

Things to know when using the tooltip plugin:

- Tooltips rely on the 3rd party library Popper.js for positioning. You must include popper.min.js before bootstrap.js or use `bootstrap.bundle.min.js` / `bootstrap.bundle.js` which contains Popper.js in order for tooltips to work!
- If you're building our JavaScript from source, it requires `util.js`.
- Tooltips are opt-in for performance reasons, so **you must initialize them yourself**.
- Tooltips with zero-length titles are never displayed.
- Specify `container: 'body'` to avoid rendering problems in more complex components (like our input groups, button groups, etc).
- Triggering tooltips on hidden elements will not work.
- Tooltips for `.disabled` or `disabled` elements must be triggered on a wrapper element.
- When triggered from hyperlinks that span multiple lines, tooltips will be centered. Use `white-space: nowrap;` on your `<a>`s to avoid this behavior.
- Tooltips must be hidden before their corresponding elements have been removed from the DOM.
- Tooltips can be triggered thanks to an element inside a shadow DOM.

The animation effect of this component is dependent on the `prefers-reduced-motion` media query. See the reduced motion section of our accessibility documentation.

Got all that? Great, let's see how they work with some examples.

Example: Enable tooltips everywhere

One way to initialize all tooltips on a page would be to select them by their `data-toggle` attribute:

Copy

```
$(function () {
  $('[data-toggle="tooltip"]').tooltip()
})
```

Examples

Hover over the links below to see tooltips:

Tight pants next level keffiyeh you probably haven't heard of them. Photo booth beard raw denim letterpress vegan messenger bag stumptown. Farm-to-table seitan, mcsweeney's fixie sustainable quinoa 8-bit american apparel have a terry richardson vinyl chambray. Beard stumptown, cardigans banh mi lomo thundercats. Tofu biodiesel williamsburg marfa, four loko mcsweeney's cleanse vegan chambray. A really ironic artisan whatever keytar, scenester farm-to-table banksy Austin twitter handle freegan cred raw denim single-origin coffee viral.

Hover over the buttons below to see the four tooltips directions: top, right, bottom, and left.

Tooltip on top | Tooltip on right | Tooltip on bottom | Tooltip on left | Tooltip with HTML

Copy

```
<button type="button" class="btn btn-secondary" data-toggle="tooltip" data-placement="top" title="Tooltip on top">
  Tooltip on top
</button>
<button type="button" class="btn btn-secondary" data-toggle="tooltip" data-placement="right" title="Tooltip on right">
  Tooltip on right
</button>
<button type="button" class="btn btn-secondary" data-toggle="tooltip" data-placement="bottom" title="Tooltip on bottom">
  Tooltip on bottom
</button>
<button type="button" class="btn btn-secondary" data-toggle="tooltip" data-placement="left" title="Tooltip on left">
  Tooltip on left
</button>
```

And with custom HTML added:

Copy

```
<button type="button" class="btn btn-secondary" data-toggle="tooltip" data-html="true" title="<em>Tooltip</em> <u>with</u> <b>HTML</b>">
  Tooltip with HTML
</button>
```

Usage

The tooltip plugin generates content and markup on demand, and by default places tooltips after their trigger element.

Trigger the tooltip via JavaScript:

Copy

```
$('#example').tooltip(options)
```

Overflow `auto` and `scroll`

Tooltip position attempts to automatically change when a parent container has `overflow: auto` or `overflow: scroll` like our `.table-responsive`, but still keeps the original placement's positioning. To resolve, set the `boundary` option to anything other than default value, `'scrollParent'`, such as `'window'`:

```
$('#example').tooltip({ boundary: 'window' })
```

Markup

The required markup for a tooltip is only a `data` attribute and `title` on the HTML element you wish to have a tooltip. The generated markup of a tooltip is rather simple, though it does require a position (by default, set to `top` by the plugin).

Making tooltips work for keyboard and assistive technology users

You should only add tooltips to HTML elements that are traditionally keyboard-focusable and interactive (such as links or form controls). Although arbitrary HTML elements (such as `<span>`s) can be made focusable by adding the `tabindex="0"` attribute, this will add potentially annoying and confusing tab stops on non-interactive elements for keyboard users, and most assistive technologies currently do not announce the tooltip in this situation. Additionally, do not rely solely on `hover` as the trigger for your tooltip, as this will make your tooltips impossible to trigger for keyboard users.

Copy

```html
<!-- HTML to write -->
<a href="#" data-toggle="tooltip" title="Some tooltip text!">Hover over me</a>

<!-- Generated markup by the plugin -->
<div class="tooltip bs-tooltip-top" role="tooltip">
  <div class="arrow"></div>
  <div class="tooltip-inner">
    Some tooltip text!
  </div>
</div>
```

Disabled elements

Elements with the `disabled` attribute aren't interactive, meaning users cannot focus, hover, or click them to trigger a tooltip (or popover). As a workaround, you'll want to trigger the tooltip from a wrapper `<div>` or `<span>`, ideally made keyboard-focusable using `tabindex="0"`, and override the `pointer-events` on the disabled element.

Disabled button

Copy

```html
<span class="d-inline-block" tabindex="0" data-toggle="tooltip" title="Disabled
tooltip">
  <button class="btn btn-primary" style="pointer-events: none;" type="button"
disabled>Disabled button</button>
</span>
```

Options

Options can be passed via data attributes or JavaScript. For data attributes, append the option name to `data-`, as in `data-animation=""`.

Note that for security reasons the `sanitize`, `sanitizeFn` and `whiteList` options cannot be supplied using data attributes.

Name	Type	Default	Copy Description
animation	boolean	true	Apply a CSS fade transition to the tooltip
container	string | element | false	false	Appends the tooltip to a specific element. Example: `container: 'body'`. This option is particularly useful in that it allows you to position the tooltip in the flow of the document near the triggering element - which will prevent the tooltip from floating away from the triggering element during a window resize.
delay	number | object	0	Delay showing and hiding the tooltip (ms) - does not apply to manual trigger type If a number is supplied, delay is applied to both hide/show Object structure is: `delay: { "show": 500, "hide": 100 }`
html	boolean	false	Allow HTML in the tooltip. If true, HTML tags in the tooltip's `title` will be rendered in the tooltip. If false, jQuery's `text` method will be used to insert content into the DOM. Use text if you're worried about XSS attacks.

Name	Type	Default	Description
placement	string | function	'top'	How to position the tooltip - auto | top | bottom | left | right. When `auto` is specified, it will dynamically reorient the tooltip. When a function is used to determine the placement, it is called with the tooltip DOM node as its first argument and the triggering element DOM node as its second. The `this` context is set to the tooltip instance.
selector	string | false	false	If a selector is provided, tooltip objects will be delegated to the specified targets. In practice, this is used to also apply tooltips to dynamically added DOM elements (`jQuery.on` support). See this and an informative example.
template	string	`'<div class="tooltip" role="tooltip"><div class="arrow"></div><div class="tooltip-inner"></div></div>'`	Base HTML to use when creating the tooltip. The tooltip's `title` will be injected into the `.tooltip-inner`. `.arrow` will become the tooltip's arrow. The outermost wrapper element should have the `.tooltip` class and `role="tooltip"`.

Name	Type	Default	Description
title	string | element | function	''	Default title value if `title` attribute isn't present. If a function is given, it will be called with its `this` reference set to the element that the tooltip is attached to.
trigger	string	'hover focus'	How tooltip is triggered - click | hover | focus | manual. You may pass multiple triggers; separate them with a space. `'manual'` indicates that the tooltip will be triggered programmatically via the `.tooltip('show')`, `.tooltip('hide')` and `.tooltip('toggle')` methods; this value cannot be combined with any other trigger. `'hover'` on its own will result in tooltips that cannot be triggered via the keyboard, and should only be used if alternative methods for conveying the same information for keyboard users is present.

Name	Type	Default	Description
offset	number | string | function	0	Offset of the tooltip relative to its target. When a function is used to determine the offset, it is called with an object containing the offset data as its first argument. The function must return an object with the same structure. The triggering element DOM node is passed as the second argument. For more information refer to Popper.js's offset docs.
fallbackPlacement	string | array	'flip'	Allow to specify which position Popper will use on fallback. For more information refer to Popper.js's behavior docs
boundary	string | element	'scrollParent'	Overflow constraint boundary of the tooltip. Accepts the values of `'viewport'`, `'window'`, `'scrollParent'`, or an HTMLElement reference (JavaScript only). For more information refer to Popper.js's preventOverflow docs.
sanitize	boolean	true	Enable or disable the sanitization. If activated `'template'` and `'title'` options will be sanitized.
whiteList	object	Default value	Object which contains allowed attributes and tags

Name	Type	Default	Description
sanitizeFn	null | function	null	Here you can supply your own sanitize function. This can be useful if you prefer to use a dedicated library to perform sanitization.
popperConfig	null | object	null	To change Bootstrap's default Popper.js config, see [Popper.js's configuration]()

Data attributes for individual tooltips

Options for individual tooltips can alternatively be specified through the use of data attributes, as explained above.

Methods

Asynchronous methods and transitions

All API methods are **asynchronous** and start a **transition**. They return to the caller as soon as the transition is started but **before it ends**. In addition, a method call on a **transitioning component will be ignored**.

[See our JavaScript documentation for more information]().

`$().tooltip(options)`

Attaches a tooltip handler to an element collection.

`.tooltip('show')`

Reveals an element’s tooltip. **Returns to the caller before the tooltip has actually been shown** (i.e. before the `shown.bs.tooltip` event occurs). This is considered a “manual” triggering of the tooltip. Tooltips with zero-length titles are never displayed.

Copy

```
$('#element').tooltip('show')
```

`.tooltip('hide')`

Hides an element’s tooltip. **Returns to the caller before the tooltip has actually been hidden** (i.e. before the `hidden.bs.tooltip` event occurs). This is considered a “manual” triggering of the tooltip.

Copy

```
$('#element').tooltip('hide')
```

`.tooltip('toggle')`

Toggles an element’s tooltip. **Returns to the caller before the tooltip has actually been shown or hidden** (i.e. before the `shown.bs.tooltip` or `hidden.bs.tooltip` event occurs). This is considered a “manual” triggering of the tooltip.

Copy

```
$('#element').tooltip('toggle')
```

`.tooltip('dispose')`

Hides and destroys an element's tooltip. Tooltips that use delegation (which are created using the `selector` option) cannot be individually destroyed on descendant trigger elements.

Copy

```
$('#element').tooltip('dispose')
```

`.tooltip('enable')`

Gives an element's tooltip the ability to be shown. **Tooltips are enabled by default.**

Copy

```
$('#element').tooltip('enable')
```

`.tooltip('disable')`

Removes the ability for an element's tooltip to be shown. The tooltip will only be able to be shown if it is re-enabled.

Copy

```
$('#element').tooltip('disable')
```

`.tooltip('toggleEnabled')`

Toggles the ability for an element's tooltip to be shown or hidden.

Copy

```
$('#element').tooltip('toggleEnabled')
```

`.tooltip('update')`

Updates the position of an element's tooltip.

Copy

```
$('#element').tooltip('update')
```

Events

Event Type	Description
show.bs.tooltip	This event fires immediately when the `show` instance method is called.
shown.bs.tooltip	This event is fired when the tooltip has been made visible to the user (will wait for CSS transitions to complete).
hide.bs.tooltip	This event is fired immediately when the `hide` instance method has been called.
hidden.bs.tooltip	This event is fired when the tooltip has finished being hidden from the user (will wait for CSS transitions to complete).
inserted.bs.tooltip	This event is fired after the `show.bs.tooltip` event when the tooltip template has been added to the DOM.

Copy

```
$('#myTooltip').on('hidden.bs.tooltip', function () {
  // do something...
})
```