## **DROP.JS**

THE FAST AND CAPABLE DROPDOWN LIBRARY

Drop is a JavaScript and CSS library for creating dropdowns and other popups attached to elements on the page. Drop uses [Tether.js](http://github.hubspot.com/tether) to efficiently position its elements.

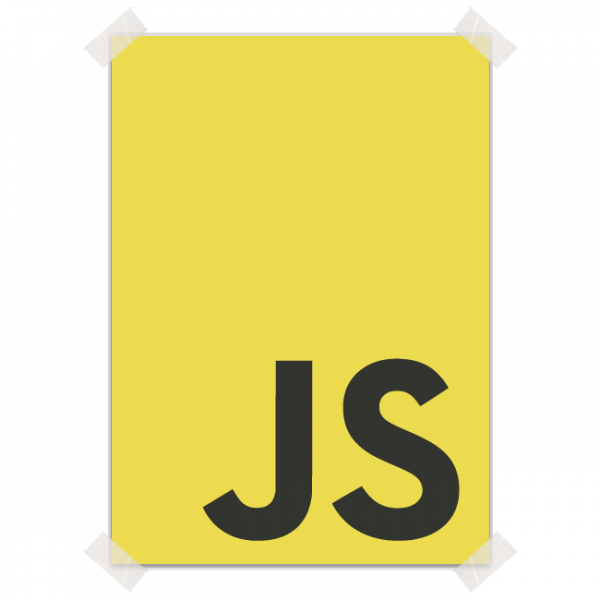


Because DROP.JS depends on Tether.Js It brings all its features:

* Drops automatically reposition on page resizes and scrolls, reorienting to stay in view.
* Drop uses GPU accelerated positioning to maintain 60fps scrolling, even with dozens or hundreds of drops on screen and complex animation
* Drops can be nested within other drops
* Drops can be attached to any of 12 attachment points on the target, or you can leverage the full power of Tether to position your drop anywhere.
* Drops can be configured to open when the user clicks, hovers, or focuses an element.

Drop is maintained by developers at [HubSpot](http://github.hubspot.com/) who care about making it do everything you need.

**A client-side library to make absolutely positioned elements attach to elements in the page efficiently.**

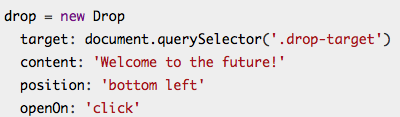


**SUPORTS**



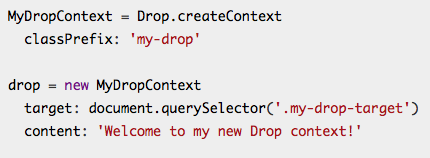
## **How To Start**

Create a Drop instance:



Class Prefix: drop; example: drop-open, data-drop, drop-content;

OR, create Drops from a custom ‘context’:



Class Prefix: drop; example: my-drop-open, data-my-drop, my-drop-content;

Methods:

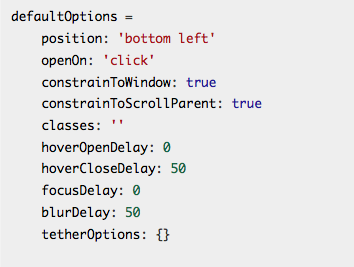
* open();
* close();
* remove();
* toggle();
* isOpened();
* position();
* destroy();

Events:

* on(eventName, handler, [ctx])
* off(eventName, handler)
* once(eventName, handler, [ctx])

Options:

Default Options:



* target:

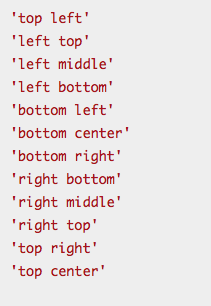
Where Drop.Js should act upon.

* content:

A DOM element, a HTML string a function that returns the two preview mentioned options; content() is called on each open with the drop instance as 1st argument.

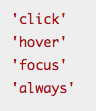
* position:

Defines the attachment point on the target.



* openOn:

Defines the event that trigger the drop to happen. ‘Always’ will open the drop as the page gets rendered and leave it opened. ‘undefined’ or ‘null’ set will require calling open() or close();



* constrainToWindow:

This will make sure the drop does not get outside the viewport. It can’t be set to true or false.

* constrainToScrollParent:

Similar to contrainToWindow but the target element is constrained to the parent element with overflow set to auto or scroll or the body element, whichever comes first.

* classes:

This will set additional class to the drop and can also be applied to a theme or a simple styling custom class.

* remove:

This option will remove the drop when is closed if set to true then be recreated when it is opened.

* beforeClose:

This option is used to use a function before the drop is closed. If the function returns false drop closing will be prevented.

* hoverOpenDelay:

Set delay time (in milliseconds) of drop open on hover.

* hoverCloseDelay:

Set delay time (in milliseconds) of drop close on hover.

* focusDelay:

Set delay time (in milliseconds) of drop open on focus.

* blurDelay:

Set delay time (in milliseconds) of drop close on blur.

* openDelay:

Sets both hoverOpenDelay and focusDelay.

* closeDelay:

Sets both hoverCloseDelay and blurDelay.

* thetherOptions:

See the the [Tether documentation](http://tether.io/) for more information about how to customize drop even further.