



jquery.qualcomm-reveal.js

Plugin call:

```
$('.content-container').qualcommreveal({ //options here });
```

Basic functionality:

- The DOM element which the plugin is called upon is given a margin-bottom of 100px
 - Best if it is the wrapper for your content
- The plugin then appends `<div id="qreveal-trigger"></div>` as the last child of the container DOM the plugin was initialized on
- `div#qreveal-trigger` is given a height of 90px, and centered
- The height is 90px so the *jquery.inview.js* plugin can check to see if either the bottom or both the top and bottom of the element is visible to the learner before fading in the loader, and then loading the the next section.
- As content is revealed, we just simply push down `div#qreveal-trigger` until the last target section where we remove the 100px margin and `div#qreveal-trigger`

Events Fired:

This plugin uses a pub-sub model. jQuery functions [`.on()`, `.off()`, `.trigger()`] are mapped to the [`$.subscribe()`, `$.unsubscribe()`, and `$.publish()`].

- `$.publish('qualcommreveal/reveal-start')`
 - Fires when the animation for loading the next section starts
- `$.publish('qualcommreveal/reveal-finish')`
 - Fires when the animation for loading the next section starts
- `$.publish('qualcommreveal/last-target')`
 - Fires when the last target has been revealed

`'qualcommreveal/reveal-start'` & `'qualcommreveal/reveal-finish'` both send two parameters to the `$.publish` event, that you can use in the fallback function.

- `elem` -- the element is going to be faded in or was just faded in
- `counter` -- the counter # for the element that is going to be faded or was just faded in

Example:

```
$.subscribe('qualcommreveal/reveal-start', function( event, elem, counter ) { ...do stuff... });
```

Plugin Options:

Below are the plugin's default options. Any option can be overwritten during plugin initialization.

- `targets: '.panel-snap'`,
 - This is the target (class) that will be hidden, and shown throughout the module
 - Do not add this class to the first section you want visible, as anything with this class will be set to `"display:none"`
- `offset: 60`,
 - This is the offset for the scrolling.
 - Offset is added to the current viewport's scroll position, not to the next section's scroll position
- `scrollspeed: 1000`,
 - The duration of the scroll to the new position
- `fadespeed: 800`,
 - The duration of the fade-in animation of the next section and the fade-in/fade-out of the spinner
- `easing: 'swing'`,
 - The easing to be used for the scroll animation
- `waitTime: 1000`,
 - The delay before the next section is loaded in
- `spinOptions: {`
 - `lines: 9,` // The number of lines to draw
 - `length: 1,` // The length of each line
 - `width: 6,` // The line thickness
 - `radius: 15,` // The radius of the inner circle
 - `corners: 1,` // Corner roundness (0..1)
 - `rotate: 0,` // The rotation offset
 - `direction: 1,` // 1: clockwise, -1: counterclockwise
 - `color: '#000',` // #rgb or #rrggbb or array of colors
 - `speed: 1,` // Rounds per second
 - `trail: 60,` // Afterglow percentage
 - `shadow: false,` // Whether to render a shadow
 - `hwaccel: true,` // Whether to use hardware acceleration
 - `className: 'spinner',` // The CSS class to assign to the spinner
 - `zIndex: 2e9,` // The z-index (defaults to 2000000000)
 - `top: 'auto',` // Top position relative to parent in px
 - `left: 'auto'` // Left position relative to parent in px
- `}`
 - The options for the spin.min.js plugin.
 - Options for the spinner can be previewed here: <http://fgnass.github.io/spin.js/>