jNewsticker Documentation

jNewsticker is a powerful jQuery plugin designed to seamlessly integrate into your website. Pre-packed with three news ticker styles and three skins, you can easily create your own skins.

Developers will find the plugin easily extendible. If you understand Javascript and jQuery, you can create your own news ticker styles to sit on top of jNewsticker.

A Wordpress version is coming soon.

FEATURES

- Three professionally designed skins
- Three in-built news ticker animation styles: fade, scroll and reveal
- Design your own skins
- Extend with your own news ticker animation styles
- Get news items from RSS feeds, Twitter, your Facebook page and Last.fm
- Fine-grained control over how items loaded from RSS, Last.fm, Twitter or Facebook are displayed
- Handles window resizing
- Supports multiple instances on a page
- Full documentation covering how to implement the news ticker, theming guide and developer guide
- Compatible with IE7+, Firefox, Safari, Chrome and Opera

Implementation

BASIC SETUP

To set up jNewsticker, you need to include the following in the head of your HTML:

- jQuery 1.4+
- jNewsticker script
- jNewsticker skin (optional)

For example, to include the jNewsticker with the blue skin:

```
<link href="css/corporate_blue.css" rel="stylesheet" type="text/css" />

<script id="jquery" src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>

<script id="newsticker-pro" src="newsticker.jquery.min.js"></script>
```

jNewsticker requires an unordered list (<ul></ul>) to populate. The list may contain items or can simply be empty if you are fetching news items from an RSS feed.

```
<ul id="newsticker_1" class="newsticker">
    <li>Lorem ipsum dolor sit amet, consectetur adipisicing elit...</li>
    <li>Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...</li>
    <li>Duis aute irure dolor in reprehenderit in voluptate velit esse cillum...</li>
    <li>Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia...</li>
    <li>Bubble bubble ipsum dolor sit amet, consectetur adipisicing elit...</li>
</ul>
```

Note that if you want to use one of the skins provided, it is important that your list has a class of newsticker.

Finally, turn the list into a news ticker by calling the newsticker method on the list:

```
(function($){
    $(document).ready(function() {
            $('#newsticker_1').newsticker();
    });
})(jQuery);
```

All together, an example:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Newsticker Pro</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <script id="jquery" src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
        <script id="newsticker-pro" src="newsticker.jquery.js"></script>
        <link rel="stylesheet" type="text/css" href="css/corporate_blue.css" />
    </head>
    <body>
        <ul id="newsticker_1" class="newsticker">
            <li>Lorem ipsum dolor sit amet, consectetur adipisicing elit...</li>
            <li>Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip...</li>
            <li>Duis aute irure dolor in reprehenderit in voluptate velit esse cillum...</li>
            <li>Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia...</li>
            <li>Bubble bubble ipsum dolor sit amet, consectetur adipisicing elit...</li>
        </ul>
    </body>
    <script>
        (function($){
            $(document).ready(function() {
                $('#newsticker_1').newsticker();
            });
        })(jQuery);
    </script>
</html>
```

Settings

You can customise your news ticker with a great range of settings.

GENERAL SETTINGS

style: String. One of ‘fade’, ‘reveal’ or ‘scroll’. Default is ‘scroll’.

showControls: True or false. Whether to show the pause, resume, next and previous buttons. Default is true.

tickerTitle: String. News ticker title text. Nothing by default.

autoStart: True / false. Whether to start the news ticker animation automatically. Default is true.

feeds: String or array.URL of an RSS feed you want to fetch news items from. If you want to fetch news items from multiple RSS feeds, use an array. Each element of the array can either be a simple string, or an object with two properties, URL and number, defining the URL to fetch from and the number of items to request. Some examples are provided below:

Single RSS feed:

```
feeds: 'http://themeforest.net/feeds/new-themeforest-items.atom'
```

Multiple RSS feeds:

```
feeds: [ 'http://themeforest.net/feeds/new-themeforest-items.atom', 'http://codecanyon.net/feeds/new-codecanyon-items.atom' ]
```

Multiple RSS feeds with number of items to fetch for each:

```
feeds:    [
          {
                'url' : 'http://themeforest.net/feeds/new-themeforest-items.atom',
                'count' : 3
          },
          {
                'url' : 'http://codecanyon.net/feeds/new-codecanyon-items.atom',
                'count' : 5
          }
          ];
```

By default, this is blank.

twitter: Twitter username. Will fetch most recent tweets from given user and format them using **twitterFormat** (or **feedFormat** if **twitterFormat**) is empty. Default is empty.

twitterFormat: String. Customize how Tweets will be formatted. See **feedFormat** for available variables. Leave it blank to use **feedFormat**. Default is `%excerpt% posted on <a href=”%link%”>Twitter</a> %timeago%`

twitterCount: Number of tweets to retrieve. If not set, it will fetch the number of items specified by **feedCount**. Default is empty.

parseTwitterLinks: By default, URLs, user mentions and hashtags within tweets are all converted to links. This can be switched off by setting this to `false`.

openLinksInNewWindow: Set whether parsed Twitter links (see above) are opened in a new window. Default is false.

facebook: Facebook page ID or name. Will fetch most recent posts from given Facebook page and format them using **facebookFormat** (or **feedFormat** if **facebookFormat**) is empty. Note that this is only possible for Facebook pages at this stage – it will not work for personal profiles. Default is empty.

facebookFormat: String. Customize how Tweets will be formatted. See **feedFormat** for available variables. Leave it blank to use feedFormat. Default is `%excerpt%... Posted on <a href="%link%">Facebook</a> %timeago%`

facebookCount: Number of Facebook posts to retrieve. If not set, it will fetch the number of items specified by **feedCount**. Default is empty.

lastfm: Last.fm username. Will fetch most recently listened to tracks for user and format them using **lastfmFormat** (or **feedFormat** if **lastfmFormat**) is empty. Default is empty.

lastfmFormat: String. Customize how last.fm items will be formatted. See **feedFormat** for available variables. Leave it blank to use **feedFormat**. Default: `<a href=”%link%”>%title%</a> listened to %timeago%`

lastfmCount: Number of items to retrieve. If not set, it will fetch the number of items specified by **feedCount**. Default is empty.

pinterest: Your Pinterest username.

pinterestFormat: How your Pinterest news items will be formatted. See **feedFormat** for a list of variables that can be used. Default: `<a href=”%item_link%”>%excerpt%</a> pinned %timeago%`

pinterestCount: Integer. The number of Pinterest pins to display. If left blank, will default to using the value for **feedItems**. Default is blank.

tumblr: Your Tumblr username.

tumblrFormat: How your Tumblr news items will be formatted. See **feedFormat** for a list of variables that can be used. Default: `<a href=”%link%”>%title%</a> posted by %author%, %timeago%`

tumblrCount: Number of items to pull from your Tumblr feed.

noCache: Prevent feed caching by setting `noCache` to `true`. Note that this will have a performance impact. By default, this is set to `false`.

excerptLength: Number of characters to include for excerpt. Leave blank to include entire excerpt (size will vary depending on type of feed and length of post). Set to ‘100’ by default.

sortBy: Options: ‘timestamp’ or ‘random’. Timestamp orders by date posted. Random is self-explanatory. Default is ‘timestamp’.

reverseOrder: True or false. If news items are ordered by timestamp, this can be used to display oldest items first, or vice versa. By default, newest items are shown first. Set to true to show oldest items first.

feedItems: Number of RSS feed items to show.

feedFormat: String. Template for how each RSS feed item should be formatted. a number of variables are available to insert snippets from the feed output:

- %author% – Display news item’s author.
- %excerpt% – Display snippet of news item content.
- %link% – Display link to news item when fetched from a normal feed. For Facebook, this displays the link to the Facebook page (not the Facebook post itself); similarly, for Twitter, this displays the link to the Twitter profile, not the tweet itself. To link to the actual tweet or Facebook post, use %item_link%.
- %item_link% – Display direct link to tweet or Facebook post. For Last.fm feeds, this will link directly to the artist’s last.fm page (not the user’s page). Will return the same value as %link% when used for a normal feed.
- %title% – Display news item’s title. This will return nothing when displaying Twitter news items.
- %timeago% – Display time since news item was posted. Output can be modified by providing a callback function for timeagoFormat.
- %date% – Display date news item was posted. Output can be modified by providing a callback function for dateFormat.

timeAgoFormat: Function. Customise how the %timeago% variable (see feedFormat) is formatted.

dateFormat: Function. Customise how the %date% variable (see feedFormat) is formatted.

pauseOnHover: True or false. Whether the animation should be paused when the mouse hovers over the news ticker. Defaults to true for ‘fade’ and ‘scroll’ animations; defaults to false for ‘reveal’ animation.

FADE SETTINGS & EXAMPLE

The ‘fade’ animation style has the following additional settings: **fadeOutSpeed**, **fadeInSpeed** and **transitionSpeed**.

fadeOutSpeed: Speed in milliseconds at which a news item fades out. May be specified as an integer or as ‘fast’ (200 milliseconds) or ‘slow’ (400 milliseconds). Default is ‘slow’.

fadeInSpeed: Speed in milliseconds at which a news items fades in. May be specified as an integer or as ‘fast’ (200 milliseconds) or ‘slow’ (400 milliseconds). Default is ‘slow’.

transitionSpeed: Length of time a news item appears before the next one fades in. May be specified as an integer or as ‘fast’ (200 milliseconds) or ‘slow’ (400 milliseconds). Default is 4000 (4 seconds).

See below for an example implementation of the fading news ticker. It is configured to pull in 5 latest items from the ThemeForest new item feed.

```
(function($){
    $(document).ready(function() {
        $('#newsticker_1').newsticker({
            'style' : 'fade', // Use fade animation
            'showControls' : true, // Display pause, previous and next buttons
            'tickerTitle' : 'New ThemeForest items', // News ticker title text
            'autoStart' : true, // Start animation automatically
            'feeds' : 'http://themeforest.net/feeds/new-themeforest-items.atom', // News items will be pulled from the RSS feed
            'feedItems' : 5, // 5 news items will be displayed
            'feedFormat' : '%author% posted <a href="%link%?ref=Studio164a">%title%</a> %timeago%.', // Format that each news item will take
            'timeAgoFormat' : function( date ) {
                return $.fn.newsticker.methods.getTimeAgo( date );
            }, // Implements $.fn.newsticker.methods.getTimeAgo()
            'dateFormat' : function( date ) {
                return $.fn.newsticker.methods.getFormattedDate( date );
            }, // Implements $.fn.newsticker.methods.getFormattedDate()
            'fadeOutSpeed' : 'slow', // Old news item fades out over 400 milliseconds
            'fadeInSpeed' : 'slow', // New news item fades in over 400 milliseconds
            'transitionSpeed' : '4000', // News item will display for 4 seconds before fading into next item
            'pauseOnHover' : true  // Animation will be paused when hovering over the news ticker.
        });
    });
})(jQuery);
```

REVEAL SETTINGS

The ‘reveal’ animation style has the following additional settings: **letterRevealSpeed** and **transitionSpeed**.

letterRevealSpeed: Speed in milliseconds at which each letter is revealed. May be specified as an integer or as ‘fast’ (200 milliseconds) or ‘slow’ (400 milliseconds). Default is 70.

transitionSpeed: Length of time a news item appears before the next one begins to be revealed. May be specified as an integer or as ‘fast’ (200 milliseconds) or ‘slow’ (400 milliseconds). Default is 2000 (4 seconds).

It is possible to use HTML (ie. links, <strong> tags, <em> tags) within the news item format in the reveal animation. However, it is not possible to have nested HTML tags, as this breaks the animation.

See below for an example implementation of the reveal news ticker. It is configured to pull in the last 5 news items from the New York Times feed.

```
(function($){
    $(document).ready(function() {
        $('#newsticker_1').newsticker({
            'style' : 'reveal', // Use reveal animation
            'showControls' : true, // Display pause, previous and next buttons
            'tickerTitle' : 'Latest news', // News ticker title text
            'autoStart' : true, // Start animation automatically
            'feeds' : 'http://feeds.nytimes.com/nyt/rss/HomePage', // News items will be pulled from the RSS feed
            'feedItems' : 5, // 5 news items will be displayed
            'feedFormat' : '%title% posted %timeago%.', // Format that each news item will take
            ‘timeAgoFormat’ : function( date ) {
                return $.fn.newsticker.methods.getTimeAgo( date );
            }, // Implements $.fn.newsticker.methods.getTimeAgo()
            ‘dateFormat’ : function( date ) {
                return $.fn.newsticker.methods.getFormattedDate( date );
            }, // Implements $.fn.newsticker.methods.getFormattedDate()
            'letterRevealSpeed' : 70, // Each new letter is revealed after 70 milliseconds
            'transitionSpeed' : '2000', // News item will display for 2 seconds before fading into next item
            'pauseOnHover' : false // Animation will be not be paused when hovering over the news ticker.
        });
    });
})(jQuery);
```

SCROLL SETTINGS

The ‘reveal’ animation style has the following additional settings: **scrollSpeed**, **slideSpeed** and **slideEasing**.

scrollSpeed: Speed in milliseconds at which the scrolling action occurs. May be specified as an integer or as ‘fast’ (200 milliseconds) or ‘slow’ (400 milliseconds). Default is 50.

slideSpeed: Speed in milliseconds at which the slide between news items should occur. May be specified as an integer or as ‘fast’ (200 milliseconds) or ‘slow’ (400 milliseconds). Default is 1000.

slideEasing: String. The easing function used to slide between news items. jQuery comes with two available easing functions: ‘swing’ and ‘linear’. You may have more available (see http://api.jquery.com/animate/#easing). The slideEasing setting is set to ‘swing’ by default.

See below for an example implementation of the reveal news ticker. It is configured to pull in the last 5 news items from the NetTuts feed.

```
(function($){
    $(document).ready(function() {
          $('#newsticker_1').newsticker({
                'style' : 'scroll', // Use scroll animation
                'showControls' : true, // Display pause, previous and next
                buttons
                'tickerTitle' : 'Latest news', // News ticker title text
                'autoStart' : true, // Start animation automatically
                'feeds' : 'http://net.tutsplus.com/feed/', // News items will
                be pulled from the RSS feed
                'feedItems' : 5, // 5 news items will be displayed
                'feedFormat' : '<a href="%link%">%title%</a> posted on
                %date%.', // Format that each news item will take
                ‘timeAgoFormat’ : function( date ) {
                      return $.fn.newsticker.methods.getTimeAgo( date );
                }, // Implements $.fn.newsticker.methods.getTimeAgo()
                ‘dateFormat’ : function( date ) {
                      return $.fn.newsticker.methods.getFormattedDate( date );
                }, // Implements $.fn.newsticker.methods.getFormattedDate()
                'scrollSpeed' : 50, // Scrolling action speed is 150
                milliseconds by default. Reduce it to make it faster; increase
                it to make it slower.
                'slideSpeed' : 1000, // When sliding between items using the
                controls, slide takes 1 second.
                'pauseOnHover' : true // Animation will be be paused when
                hovering over the news ticker.
          });
    });
})(jQuery);
```

Theming guide

Theming the news ticker is not difficult if you understand CSS and are aware of the HTML structure created by News Ticker JS.

As a starting point, we suggest looking at the three pre-packaged CSS files. You are free to modify them as you please.

Shown below is a sample HTML outline:

```
<!--
    The second class given for the opening wrapper depends on the type of
    animation chosen. For a scroll animation, it the wrapper has a class
    of newsticker_style_scroll. For a reveal animation, it has a class of
    newsticker_style_reveal. For a fade animation, the wrapper has a class of
    newsticker_style_fade.
    The example below if for a scroll animation.
-->
<div class="newsticker_wrapper newsticker_style_scroll">
    <!-- If tickerTitle is set (ie. not an empty string, the title is
    displayed as below -->
    <div class="newsticker_title">
          <h4>News</h4>
    </div>
    <div class="newsticker_scroller">
          <ul id="newsticker_1" class="newsticker">
                <li>Lorem ipsum dolor sit amet, consectetur adipisicing
                elit...</li>
                <li>Ut enim ad minim veniam, quis nostrud exercitation ullamco
                laboris nisi ut aliquip...</li>
                <li>Duis aute irure dolor in reprehenderit in voluptate velit
                esse cillum...</li>
                <li>Excepteur sint occaecat cupidatat non proident, sunt in
                culpa qui officia...</li>
                <li>Bubble bubble ipsum dolor sit amet, consectetur adipisicing
                elit...</li>
          </ul>
    </div>
    <!-- If showControls is set to true, controls are displayed -->
    <ul class="newsticker_controls">
          <li class="previous"></li>
          <li class="pause"></li>
          <li class="next"></li>
    </ul>
</div>
```

Note that the plugin also adds inline styles to some elements, which are not shown here. To override any of these, you will have to use the !important CSS tag. Use Firebug or something similar to inspect the actual DOM when creating your custom theme.

Extend

jNewsticker was specifically designed to be extendable. You can create your own animation styles and use these when calling the news ticker.

An animation style is a Javascript object registered under `$.fn.newsticker.styles`.

Let's create an animation called 'flash':

```
$.fn.newsticker.styles.flash = {}
```

The flash animation can now be used, but does nothing.

The news ticker looks for (but will work without) a number of core methods inside the animation object. A skeleton outline of the flash animation object is shown below with all the core methods included:

```
$.fn.newsticker.styles.flash = {
    // Return an object literal containing additional settings (with their 
    defaults) for this animation
    settings : function() {
    },

    // Initialize animation object
    init : function() {
    },

    // Make modifications to the DOM
    manipulateDom : function() {
    },

    // Loop through individual list items making any necessary modifications to 
    the DOM
    itemLoop: function() {
        var self = this;

        // Start loop
        this.dom.items.each( function() {
            self.delegate( 'manipulateItem', [$(this)] );
        }
    },
    // Manipulate list item (called by itemLoop)
    manipulateItem: function( item ) {
    }

    // Start the animation
    start : function() {
    },

    // Called on window resize event
    resize : function() {
    },

    // Called when news ticker animation is paused
    pause : function() {
    },

    // Called when news ticker animation is resumed after being paused
    resume : function() {
    },

    // Called when 'next' button is clicked
    next : function() {
    },

    // Called when 'previous' button is clicked
    previous : function() {
    }
}
```

Please note that the script will not fail if any of these functions are missing – it may just not behave properly. For example, the pause, resume, next and previous are all optional, but if you don’t define them, nothing will happen when someone clicks the control buttons.

Two important functions

There are two other functions that are important to understand when creating your own animation. These are **this.api()** and **this.delegate()**.

this.api('method', 'args')

this.api() is the preferred way to call methods defined within the `$.fn.newsticker.methods` object, because it ensures that the api method is passed the current news ticker object to become this.

It takes a method name parameter to signify the api method to be called, and an args parameter with arguments to send to the api method. If the api method is not found, it returns false. Otherwise it returns whatever the api method returns.

this.delegate('method', 'args')

this.delegate() is the preferred way to call methods defined within the animation object, because it ensures that the animation object method is passed the current news ticker object to become this.

It takes a method name parameter to signify the animation object method to be called, and an args parameter with arguments to send to the animation object method. If the animation object method is not found, it returns false. Otherwise it returns whatever the animation object method returns.