**.venScrollbar()**

***Requires jQuery v1.4.3+***

**v2.0**

***Description:*** *Replaces the standard OS scrollbar with a custom one.*

**.venScrollbar( [settings], [ready] )**

***Returns: jQuery***

**settings** A set of key/value pairs that configures the plugin.

|  |  |  |
| --- | --- | --- |
| **Property** | **Default** | **Description** |
| anchor | true | Override the default behavior when navigating the page with anchors. |
| arrows | true | Inject the base HTML for the arrows. |
| autoHide | false | Hide the UI when idle. |
| deltaSmall | 40 | The page delta in pixels when scrolling with the arrows or arrow keys. |
| deltaLarge | 100 | The page delta in pixels when scrolling with the mouse wheel. |
| drag | false | Enable click-and-drag on the body. |
| fadeIn | 200 | The UI fade-in duration in milliseconds. |
| fadeOut | 400 | The UI fade-out duration in milliseconds. |
| idle | 1000 | The milliseconds of inactivity required before becoming idle. |
| inertial | true | Enable inertial scrolling like on iOS devices. |
| initHide | false | Hide the UI on initialization. |
| keyboard | true | Enable keyboard navigation support. |
| lag | 0 | The delay in milliseconds before responding to scrollbar drag input. |
| overlay | false | Overlay the UI on top of the viewport. |
| select | true | Enable text selection. |
| smooth | false | Enable smooth scrolling. |
| stealFocus | false | Always consume mouse wheel and touch events. |
| themeRoller | false | Enable jQuery UI ThemeRoller support. |
| touch | true | Enable touch support for mobile devices. |
| track | true | Injects the base HTML for the track. |
| wheel | true | Enable mouse wheel support. |

**Note**

The fadeIn, fadeOut, and idle settings are meaningless if autoHide is set to false.

**ready( data, ui )** A function to call once the plugin has initialized.

**Styling**

Styling is much simpler now than it was before. Instead of me (the author) trying to guess the necessary markup for a typical theme, I now just provide the base structure that is needed to make everything work. This means fewer HTML strings in the script file and less time spent injecting potentially unused elements.

<div id='#box' class='venscrollbar-root'>

<div class='venscrollbar-viewport'>

<div class='venscrollbar-body'>

<!-- The children of #box are moved here. -->

</div>

</div>

<div class='venscrollbar-ui'>

<div class='venscrollbar-ui-x'>

<div class='venscrollbar-ui-x-track' />

<div class='venscrollbar-ui-x-bar' />

</div>

<div class='venscrollbar-ui-y'>

<div class='venscrollbar-ui-y-track' />

<div class='venscrollbar-ui-y-bar' />

</div>

<div class='venscrollbar-ui-up' />

<div class='venscrollbar-ui-down' />

<div class='venscrollbar-ui-left' />

<div class='venscrollbar-ui-right' />

</div>

</div>

The layout is mostly unchanged. The *body* is the one that gets moved around. The *viewport* allows us to position the body and resize the visible area without interfering with the rest of the UI. What’s different is the *track*. In the old version, the *bar* existed inside the track, and the track size was used to determine the scrolling range. To make styling easier, I turned the track into a sibling and created a separate wrapper element (.ui-x / .ui-y) to take its place. The track is still used to receive mouse clicks, but now the track can be positioned independently of the bar.

There are three UI states:

* .venscrollbar-active
* .venscrollbar-hover
* .venscrollbar-disabled (If overflow is set to scroll, but there’s no need for a scrollbar.)

It is important that we don’t rely on CSS pseudo selectors as they are inadequate and behave differently in different browsers. For example, the bar should have the active state the entire time it is being dragged around even if the mouse wanders outside the element’s bounds. In Firefox, the :active selector does exactly what we want. But in Opera, the active state is removed whenever the mouse hovers off the element even if the mouse button is never let up. On the flip side, the arrow buttons need to behave in the opposite (Opera) way. If the user clicks on an arrow and drags off of it, the active state should be removed.

