

[Features](#)[Business](#)[Explore](#)[Marketplace](#)[Pricing](#)[This repository](#)[Sign in](#) or [Sign up](#)[jlmakes](#) / [scrollreveal](#)[Watch](#)

441

[★ Star](#)

15,169

[Fork](#)

1,881

[Code](#)[Issues](#) 12[Pull requests](#) 2[Wiki](#)[Insights](#)

Join GitHub today

GitHub is home to over 20 million developers working together to host and review code, manage projects, and build software together.

[Sign up](#)[Dismiss](#)

Easy scroll animations for web and mobile browsers. <https://scrollrevealjs.org/>

[reveal](#)[animation](#)[scroll](#)[javascript](#)[330 commits](#)[2 branches](#)[68 releases](#)[7 contributors](#)Branch: **master** ▾[New pull request](#)[Find file](#)[Clone or download ▾](#)**jlmakes** ScrollReveal v3.4.0

Latest commit 63abb40 on Mar 11

[.github](#)

feature(repo): add issue template

2 years ago

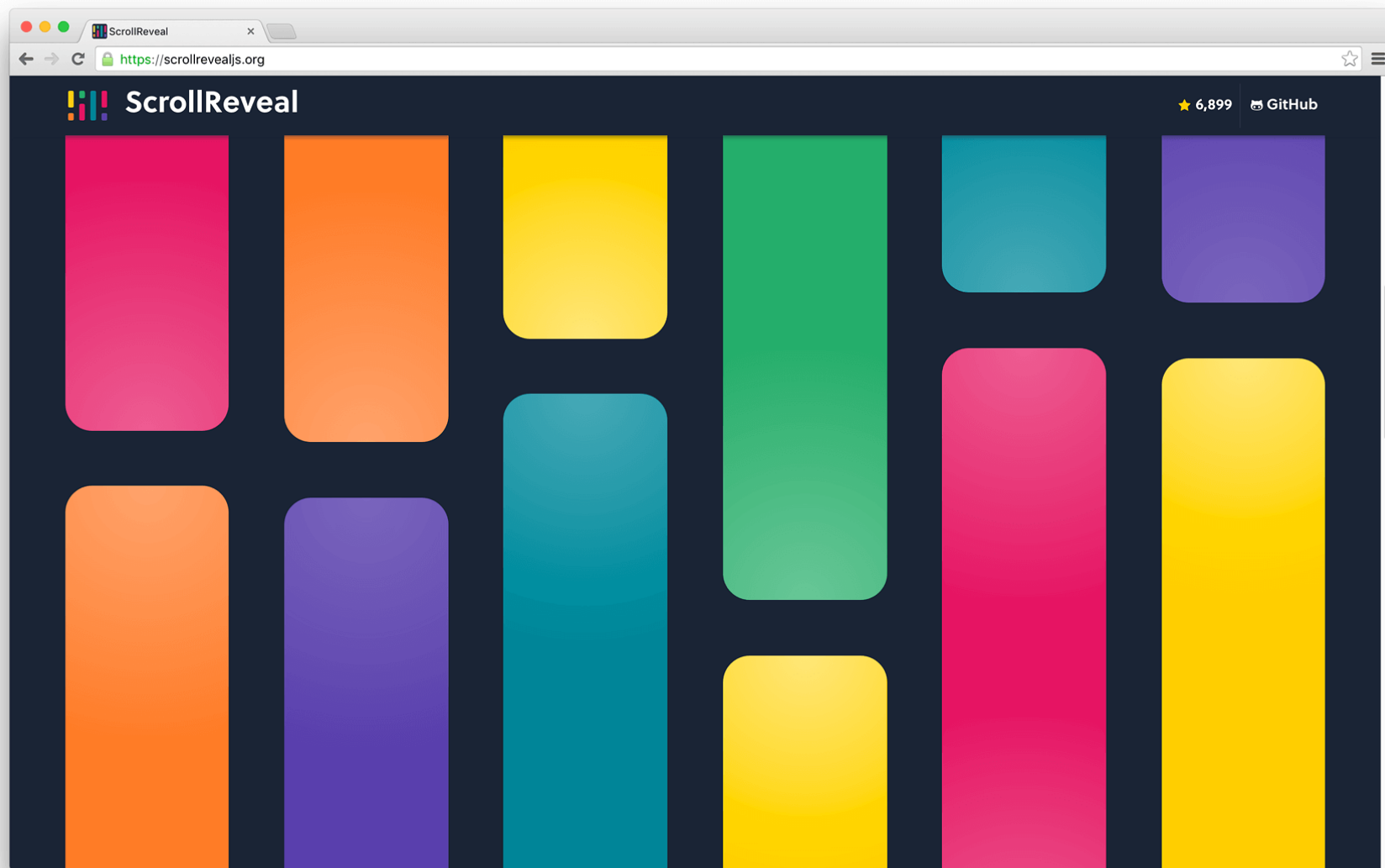
📁 dist	ScrollReveal v3.4.0	2 months ago
📁 lib/gulp	refactor(gulp)	2 years ago
📁 src	ScrollReveal v3.4.0	2 months ago
📄 .gitattributes	cleanup(gitattributes)	2 years ago
📄 .gitignore	update git ignore	2 months ago
📄 CHANGELOG.md	ScrollReveal v3.4.0	2 months ago
📄 LICENSE.md	update copyright	2 months ago
📄 README.md	ScrollReveal v3.4.0	2 months ago
📄 bower.json	bower(homepage): fix homepage typo	2 years ago
📄 gulpfile.js	refactor(gulp)	2 years ago
📄 package.json	ScrollReveal v3.4.0	2 months ago

📖 README.md



ScrollReveal

Easy scroll animations for web and mobile browsers



license MIT npm v3.4.0 downloads 31k/month

- 3.3KB minified and Gzipped
- No dependencies
- From the ❤️ of [@jlmakes](#)

1. Getting Started

1.1. Installation

The simplest method is to copy paste this snippet just before your closing `</body>` tag.

```
<script src="https://unpkg.com/scrollreveal/dist/scrollreveal.min.js"></script>
```

But you can also:

- [Download ZIP](#)
- `npm install scrollreveal`
- `bower install scrollreveal`

1.2. The Basics

The `reveal()` method is the primary API, and makes it easy to create and manage various types of animations.

```
<!-- HTML -->
<div class="foo"> Foo </div>
<div class="bar"> Bar </div>
```

```
// JavaScript
window.sr = ScrollReveal();
sr.reveal('.foo');
sr.reveal('.bar');
```

2. Configuration

Passing a configuration object to `ScrollReveal()` changes the defaults for all reveals, and passing `reveal()` a configuration object customizes that reveal set further.

2.1. Practical Example

```
// Changing the defaults
window.sr = ScrollReveal({ reset: true });

// Customizing a reveal set
sr.reveal('.foo', { duration: 200 });
```

2.2. The Starting Defaults

```
// 'bottom', 'left', 'top', 'right'
origin: 'bottom',

// Can be any valid CSS distance, e.g. '5rem', '10%', '20vw', etc.
distance: '20px',

// Time in milliseconds.
duration: 500,
delay: 0,
```

```
// Starting angles in degrees, will transition from these values to 0 in all axes.
rotate: { x: 0, y: 0, z: 0 },

// Starting opacity value, before transitioning to the computed opacity.
opacity: 0,

// Starting scale value, will transition from this value to 1
scale: 0.9,

// Accepts any valid CSS easing, e.g. 'ease', 'ease-in-out', 'linear', etc.
easing: 'cubic-bezier(0.6, 0.2, 0.1, 1)',

// `` is the default reveal container. You can pass either:
// DOM Node, e.g. document.querySelector('.fooContainer')
// Selector, e.g. '.fooContainer'
container: window.document.documentElement,

// true/false to control reveal animations on mobile.
mobile: true,

// true: reveals occur every time elements become visible
// false: reveals occur once as elements become visible
reset: false,

// 'always' - delay for all reveal animations
// 'once' - delay only the first time reveals occur
// 'onload' - delay only for animations triggered by first load
useDelay: 'always',

// Change when an element is considered in the viewport. The default value
// of 0.20 means 20% of an element must be visible for its reveal to occur.
viewFactor: 0.2,

// Pixel values that alter the container boundaries.
```

```
// e.g. Set `{ top: 48 }`, if you have a 48px tall fixed toolbar.  
// --  
// Visual Aid: https://scrollrevealjs.org/assets/viewoffset.png  
viewOffset: { top: 0, right: 0, bottom: 0, left: 0 },  
  
// Callbacks that fire for each triggered element reveal, and reset.  
beforeReveal: function (domEl) {},  
beforeReset: function (domEl) {},  
  
// Callbacks that fire for each completed element reveal, and reset.  
afterReveal: function (domEl) {},  
afterReset: function (domEl) {}
```

3. Advanced

3.1. Sequenced Animations

You can pass a sequence interval (in milliseconds) to the `reveal()` method, making sequenced animations a breeze.

Note: The interval is the time until the next element in the sequence begins its reveal, which is separate from the time until the element's animation completes. In this example, the animation duration is 2 seconds, but the sequence interval is 50 milliseconds.

```
// interval passed to reveal  
window.sr = ScrollReveal({ duration: 2000 });  
sr.reveal('.box', 50);  
  
// or...  
  
// interval and custom config passed to reveal
```



```
window.sr = ScrollReveal();  
sr.reveal('.box', { duration: 2000 }, 50);
```

3.2. Working With DOM Nodes

You may also pass `Node`, `NodeList` and `Array<Node>` as the reveal target.

```
var node = document.querySelector('.foo');
var nodeList = document.querySelectorAll('.bar');
var array = Array.prototype.slice.call(nodeList);

sr.reveal(node);
sr.reveal(nodeList);
sr.reveal(array);
```

3.3. Custom/Multiple Containers

The default container is the viewport, but you can assign any container to any reveal set.

Tip: ScrollReveal works just as well with horizontally scrolling containers too!

```
<div id="fooContainer">
  <div class="foo"> Foo 1 </div>
  <div class="foo"> Foo 2 </div>
  <div class="foo"> Foo 3 </div>
</div>

<div id="barContainer">
  <div class="bar"> Bar 1 </div>
  <div class="bar"> Bar 2 </div>
  <div class="bar"> Bar 3 </div>
</div>
```

```
window.sr = ScrollReveal();

// as a DOM node...
var fooContainer = document.getElementById('fooContainer');
sr.reveal('.foo', { container: fooContainer });

// as a selector...
sr.reveal('.bar', { container: '#barContainer' });
```

3.4. Asynchronous Content

The `sync()` method updates asynchronously loaded content with any existing reveal sets.

Example:

```
<!-- index.html -->
<div id="fooContainer">
  <div class="foo">foo</div>
  <div class="foo">foo</div>
  <div class="foo">foo</div>
</div>

<!-- ajax.html -->
<div class="foo">foo async</div>
<div class="foo">foo async</div>
<div class="foo">foo async</div>
```

```
var fooContainer, content, sr, xmlhttp;

fooContainer = document.getElementById('fooContainer');
```

```
sr = ScrollReveal();
sr.reveal('.foo', { container: fooContainer });

// Setup a new asynchronous request...
xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
    if (xmlhttp.readyState == XMLHttpRequest.DONE) {
        if (xmlhttp.status == 200) {

            // Turn our response into HTML...
            var content = document.createElement('div');
            content.innerHTML = xmlhttp.responseText;
            content = content.childNodes;

            // Add each element to the DOM...
            for (var i = 0; i < content.length; i++) {
                fooContainer.appendChild(content[ i ]);
            };

            // Finally!
            sr.sync();
        }
    }
}

xmlhttp.open('GET', 'ajax.html', true);
xmlhttp.send();
```

4. Tips

4.1. Order Matters

It's important that `reveal()` calls be made as close to last in your page as possible, so that:

- Elements on the page have loaded
- Any other 3rd party libraries have had a chance to run
- Any other styles added to your elements wont be overwritten

Example:

```
<!DOCTYPE html>
<html>
  <head>
    <!-- load and instantiate ScrollReveal first -->
    <script src="js/scrollreveal.min.js"></script>
    <script>
      window.sr = ScrollReveal();
    </script>
  </head>
  <body>

    <div class="fooContainer">
      <div class="fooReveal"> Foo </div>
      <div class="fooReveal"> Foo </div>
      <div class="fooReveal"> Foo </div>
    </div>

    <!-- make reveal calls last -->
    <script>
      sr.reveal('.fooReveal', { container: '.fooContainer' });
    </script>

  </body>
</html>
```

4.2. Improve User Experience

In most cases, your elements will start at `opacity: 0` so they can fade in. However, since JavaScript loads after the page begins rendering, you might see your elements flickering as they begin rendering before being hidden by ScrollReveal's JavaScript.

The ideal solution is to **set your reveal elements visibility to hidden** in the `<head>` of your page, to ensure they render hidden while your JavaScript loads:

Continuing our example from 4.1.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- load and instantiate ScrollReveal first -->
    <script src="js/scrollreveal.min.js"></script>
    <script>
      window.sr = ScrollReveal();

      // Add class to <html> if ScrollReveal is supported
      // Note: this method is deprecated, and only works in version 3
      if (sr.isSupported()) {
        document.documentElement.classList.add('sr');
      }
    </script>

    <style>

      /* Ensure elements load hidden before ScrollReveal runs */
      .sr .fooReveal { visibility: hidden; }

    </style>
  </head>
```

```

<body>

  <div class="fooContainer">
    <div class="fooReveal"> Foo </div>
    <div class="fooReveal"> Foo </div>
    <div class="fooReveal"> Foo </div>
  </div>

  <!-- make reveal calls last -->
  <script>
    sr.reveal('.fooReveal', { container: '.fooContainer' });
  </script>

</body>
</html>

```

Note: If you prefer not to put styles in the `<head>` of your page, including this style in your primary stylesheet will still help with element flickering since your CSS will likely load before your JavaScript.

4.3. Add Perspective to 3D Rotation

ScrollReveal supports 3d rotation out of the box, but you may want to emphasize the effect by specifying a perspective property on your container.

Continuing our example from 4.2.

```

<!DOCTYPE html>
<html>
  <head>
    <!-- load and instantiate ScrollReveal first -->
    <script src="js/scrollreveal.min.js"></script>
    <script>
      window.sr = ScrollReveal();

```

```

    // Add class to <html> if ScrollReveal is supported
    // Note: this method is deprecated, and only works in version 3
    if (sr.isSupported()) {
        document.documentElement.classList.add('sr');
    }

</script>

<style>

    /* Ensure elements load hidden before ScrollReveal runs */
    .sr .fooReveal { visibility: hidden; }

    /* add perspective to your container */
    .fooContainer { perspective: 800px; }

</style>

</head>
<body>

    <div class="fooContainer">
        <div class="fooReveal"> Foo </div>
        <div class="fooReveal"> Foo </div>
        <div class="fooReveal"> Foo </div>
    </div>

<!-- make reveal calls last -->
<script>
    // use rotation in reveal configuration
    sr.reveal('.fooReveal', { container: '.fooContainer', rotate: {x: 65} });
</script>

```



```
</body>
</html>
```

5. Appendix

Open source under the [MIT License](#). ©2014–2018 Julian Lloyd.

5.1. Browser Compatibility

ScrollReveal works on any JavaScript enabled browser that supports both [CSS Transform](#) and [CSS Transition](#). This includes Internet Explorer 10+, and most modern desktop and mobile browsers.

5.2. Issues and Reporting Bugs

Please search existing issues, before creating a new one; every issue is labeled and attended carefully. If you open a duplicate issue, it will be closed immediately.

If you cannot find your issue/bug in a previous ticket, please include details such as your browser, any other 3rd party JavaScript libraries you are using, and ideally a code sample demonstrating the problem. (Try [JSBin](#))

5.3. Pull Requests

Feeling inspired? Please contribute! Optimizations, compatibility and bug fixes are greatly preferred over new features, but don't be shy. One thing sorely missing from ScrollReveal right now is a test suite.

5.4. Showcase

Here are some cool sites using ScrollReveal:

- [Sequoia Capital](#)

- [Twist App](#)
- [Udacity](#)

Want to see your page here? Please send me your work (or of others) using ScrollReveal on Twitter ([@jlmakes](#))

5.5. Special Thanks

ScrollReveal was inspired by the talented [Manoela Ilic](#) and her [cbpScroller.js](#).

