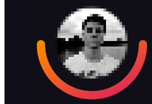
Home / Articles /

AOS: CSS-Driven “On Scroll” Animation Library



Author
Michał Sajnóg

Last Updated
Apr 10, 2017

ANIMATION SCROLLING

The following is a guest post by Michał Sajnóg, a front end developer at Netguru. Michał has created one of those “when you scroll to here, trigger this animation” libraries. One of the things I like about it is that it leaves as much as it can to CSS for creating and controlling the animation themselves. Not to mention it’s proved itself by working well on a number of production sites. I’ll let Michał walk you through it.

Have you ever seen those long web pages where different animations are being applied as you scroll down? I’d like to share with you a plugin I wrote that makes it really easy to handle all kinds of animations with full CSS control and no pain.

If you’d like to get right into it, the code is on GitHub.

The Problem With Other Libraries

In my previous company we were using WOW.js (or other similar libraries) to animate elements on scroll. For simple projects it was quite nice, but on larger sites we wanted to have more control over what's actually happening. In all of popular libraries, animations were completely handled by JavaScript by inserting inline CSS. Arghgh! Inline styles are evil. They are hard to control and override. Though, in some cases it's ok to set them using JavaScript, it's still much cleaner to just leave them where they belong and handle all CSS related things inside CSS file.

I decided to create a library that has a pure goal – detect position of elements and then add appropriate classes when they appear in viewport.

Controlling Animations Entirely in CSS

I wanted to split the responsibilities with my new library:

- Have all the logic inside the JavaScript
- Have all the animations in the CSS

This allows you to add your own animations easily, and do things like change them according to the viewport.

How AOS Works

The idea behind AOS is straightforward: watch all elements and their positions based on settings you provide them. Then add/remove the class `aos-animate`. Of course, in practice, it’s not always that easy, but the idea behind AOS is as simple as that. Every aspect of animation is handled by CSS.

Example Animations in CSS

The are lots of different animations ready to use out of the box, but creating new ones is simple also. Here’s an example:

```css
[aos="fade"] {
  opacity: 0;
  transition-property: opacity;
}

[aos="fade"].aos-animate {
  opacity: 1;
}
```

You don’t have to worry about duration or delay. In the CSS, you only:

- add styles for the attribute `aos` with the name of your animation
- set the `transition-property` (by default this is `all`, so it’s more performant and more predictable if you narrow the transition to the intended properties)
- add the post-transition properties on `.aos-animate`

Things like duration/delay/easing are set independently of the animation.

Example HTML

```html
<div class="some-item" aos="fade">Example Element</div>
```

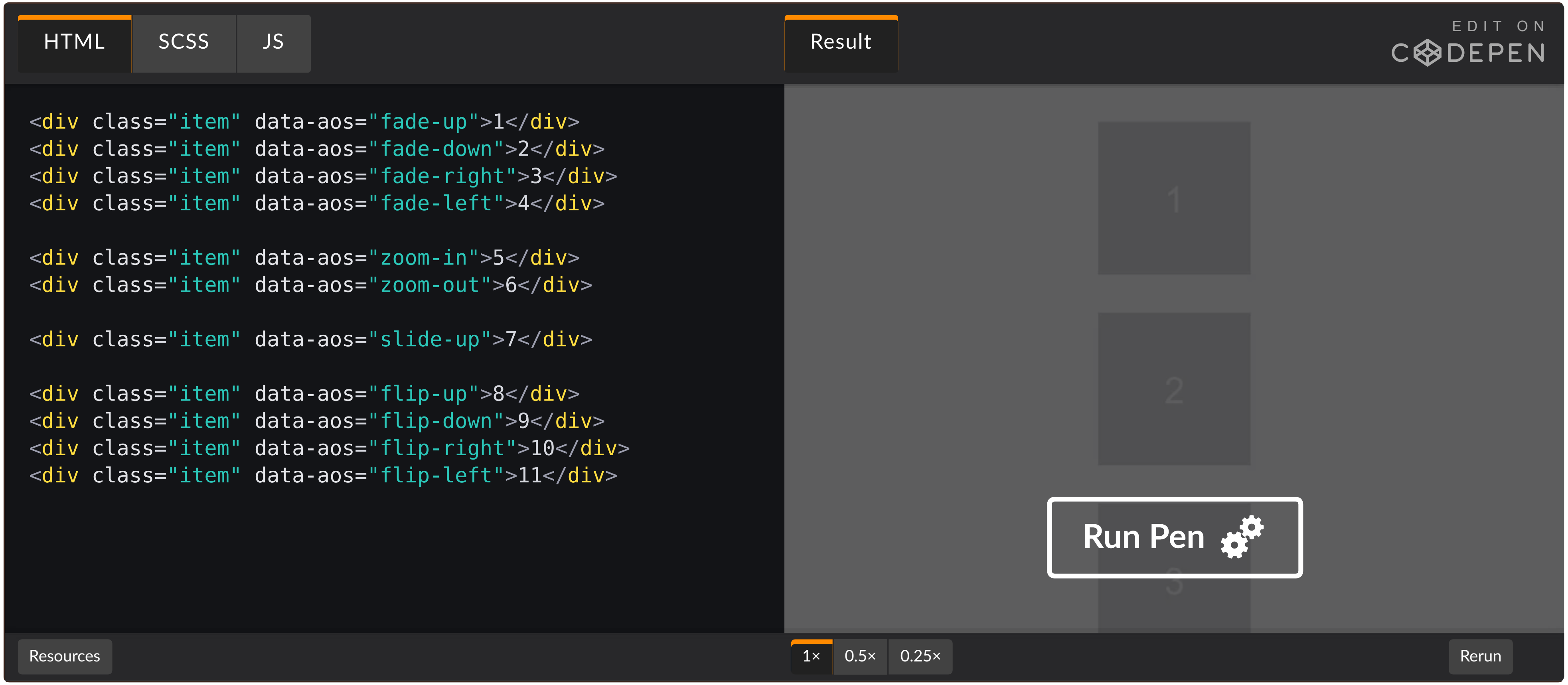
or with a different transition duration:

```html
<div class="some-item" aos="fade" aos-duration="500">Example Element</div>
```

Hey! **Tip:** You can use `data-aos` instead of `aos` to make your HTML validate properly.

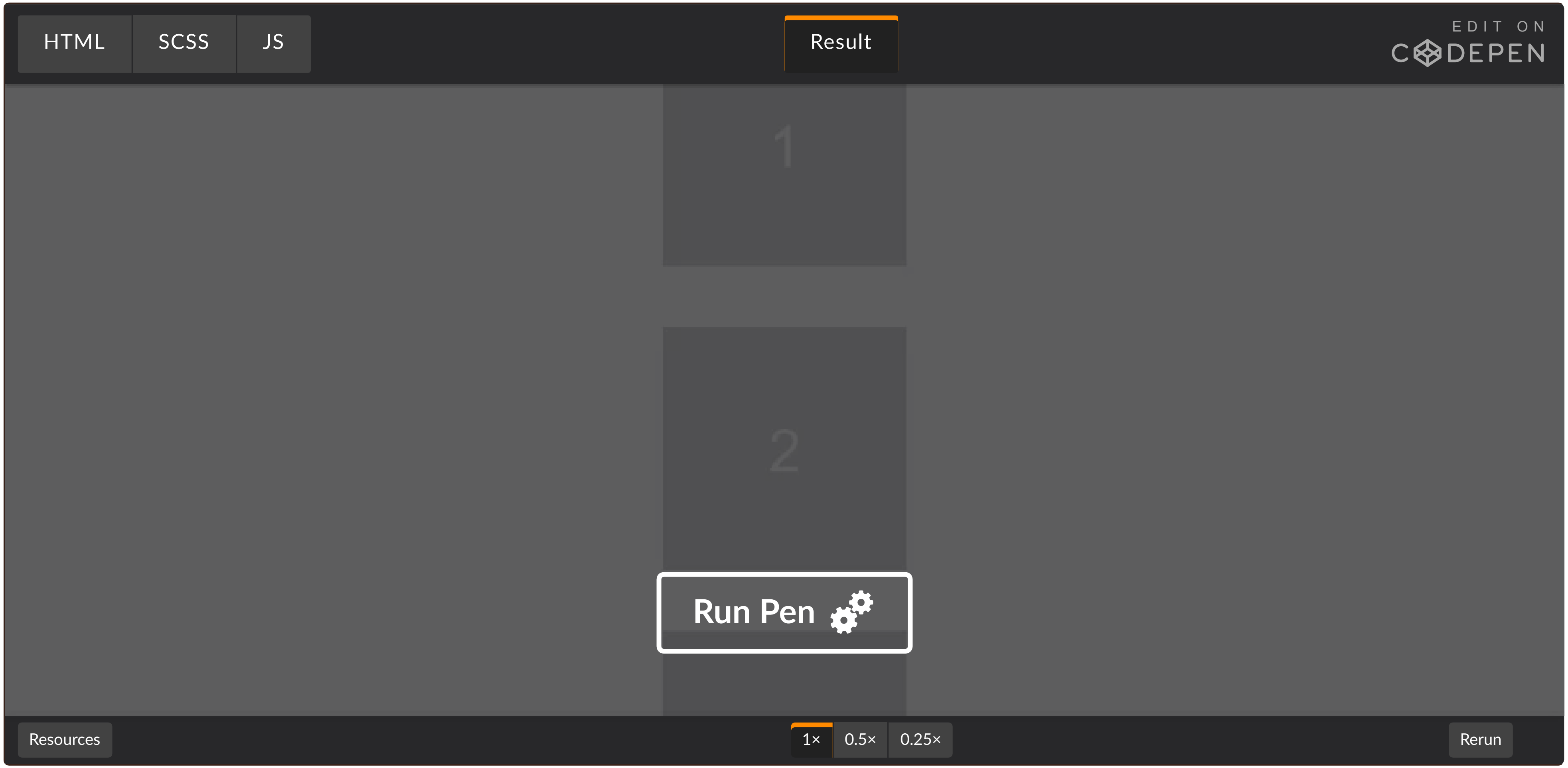
Live Demos

With different animations:

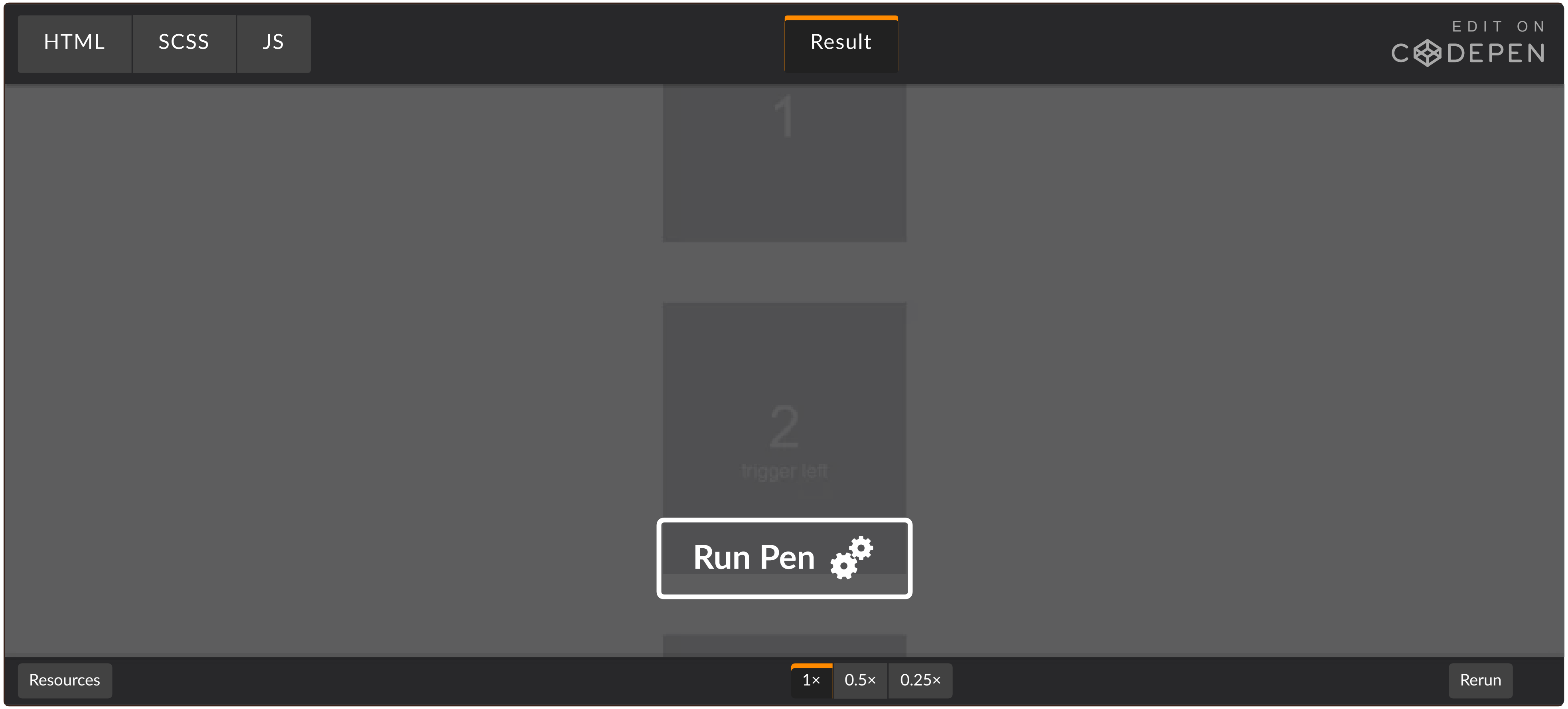


```
<div class="item" data-aos="fade-up">1</div>
<div class="item" data-aos="fade-down">2</div>
<div class="item" data-aos="fade-right">3</div>
<div class="item" data-aos="fade-left">4</div>

<div class="item" data-aos="zoom-in">5</div>
<div class="item" data-aos="zoom-out">6</div>

<div class="item" data-aos="slide-up">7</div>

<div class="item" data-aos="flip-up">8</div>
<div class="item" data-aos="flip-down">9</div>
<div class="item" data-aos="flip-right">10</div>
<div class="item" data-aos="flip-left">11</div>
```

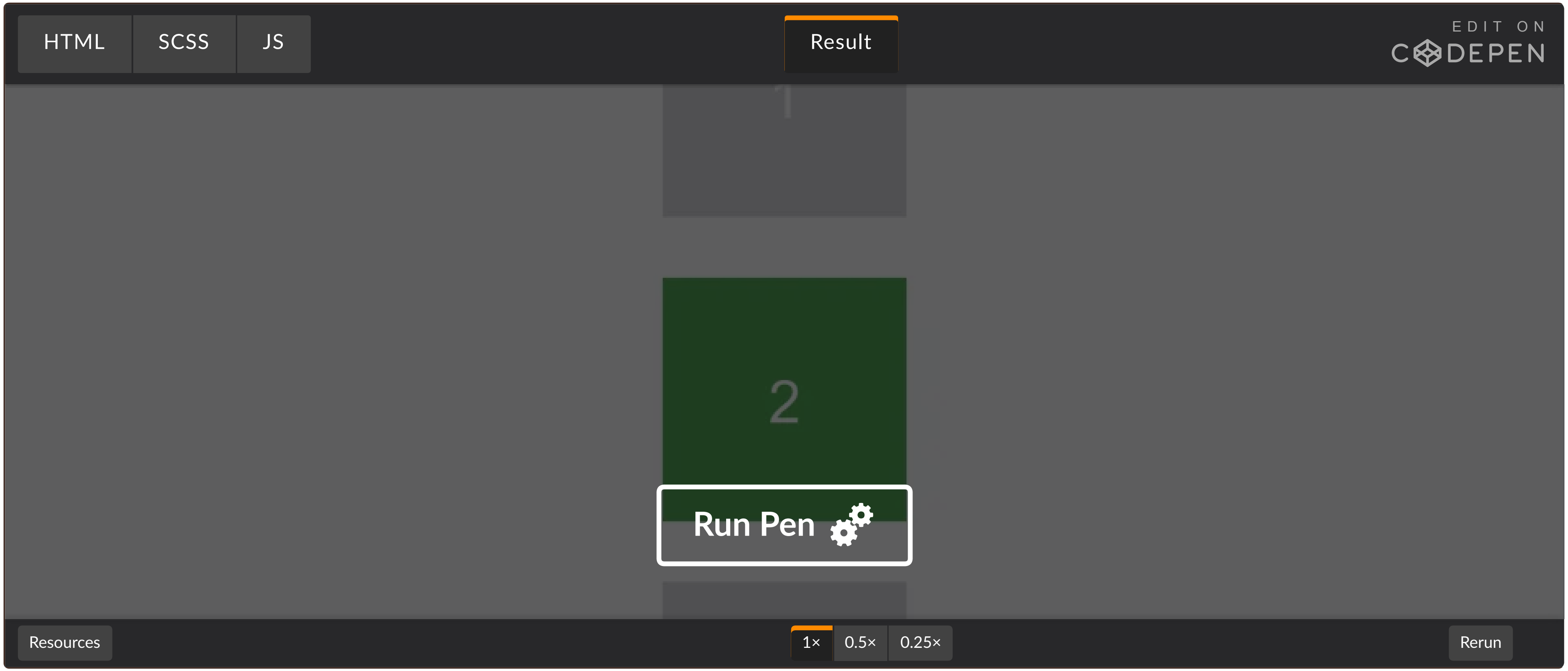
With anchor setting in use:



With anchor placement and different easing:



With simple custom animations:



Additional Features

- **anchor** – Animate one element based on position of other element
- **anchor placement** – Animate an element based on it’s position on the screen. It doesn’t have to animate only when it appears in viewport, but for example when bottom part of

it hits middle of the screen

- **both way animations** – By default elements animate while you scroll up and down, but you can force it to animate only once
- **easy disabling** – Disable animations on mobile devices using predefined options like `mobile`, `phone`, `tablet` or with a custom condition/function
- **async aware** – Recalculates positions of elements on DOM changes, so after loading something using Ajax, you don’t have to worry about anything (unless you support IE9, because it needs mutation observer)
- **no dependencies** – This library is written in pure JS and doesn’t rely on any dependencies

AOS is fully open source, so if you have an interesting idea or something is not working how you’d expect open issue and see you on GitHub! Any contribution is highly appreciated.

Wanna learn SVG & Animation deeply?

Frontend Masters has **an incredible course on all things CSS and SVG animation** from CSS-Tricks own Sarah Drasner. Sarah comprehensively covers the possibilty of animation, the tools, and does it all in a very practical way.

Comments

Guyom

# June 6, 2016

This plugin seems so easy to use, I am very happy to discover it!

Michał Sajnóg

# June 6, 2016

Glad to hear that :)

Niall

# June 28, 2016

I can’t seem to get this working for some reason. I’m working of the CDN versions and following the git set up instructions, and nothing works for me. Any ideas where I could be going wrong?

Flo

# June 6, 2016

I like all the default animations it ships with, but i still prefer jlmakes' 3,3 KiB scrollreveal.js: **https://github.com/jlmakes/scrollreveal.js**

Michał Sajnóg

# June 6, 2016

Scroll Reveal is also a great library – no doubts. AOS provides similar features, but in a slightly different way and gives you more control over animations.

Show All Comments

This comment thread is closed. If you have important information to share, please **contact us**.

June 22, 2020

An Overview of Scroll Technologies

Scroll-related animations have been used on the web for years. In recent years, they've started to become more common, perhaps in part due to devices being higher-performing and thus able to handle more animation. There are a number of scroll related technologies out there, so this article's aim is to…

April 10, 2020

Tips for Writing Animation Code Efficiently

Tips for Writing Animation Code Efficiently

I’ve been coding web animations and helping others do the same for years now. However, I have yet to see a concise list of tips focused on how to efficiently build animations, so here you go! I will be using the GreenSock Animation Platform (GSAP). It provides a simple, readable…

July 22, 2019

CSS Animation Libraries

There are an awful lot of libraries that want to help you animate things on the web. These aren’t really libraries that help you with the syntax or the technology of animations, but rather are grab-and-use as-is libraries. Want to apply a class like ”animate-flip-up” and watch an element, uhhh,…

The related posts above were algorithmically generated and displayed here without any load on my server at all, **thanks to Jetpack**.

The Newsletter

lovetricks@internet.c

Subscribe

CSS-Tricks* is created, written by, and maintained by Chris Coyier and a team of swell people. The tech stack for this site is fairly boring. That's a good thing! I've used WordPress since day one all the way up to v17, a decision I'm very happy with. I also leverage Jetpack for extra functionality and Local for local development.

*May or may not contain any actual "CSS" or "Tricks".

Other Nav

Contact
Writing
Advertise
Subscribe
About
Shop
License
Authors

CODEPEN

CodePen is a place to experiment, debug, and show off your HTML, CSS, and JavaScript creations.

FLYWHEEL

CSS-Tricks is hosted by **Flywheel**, the best WordPress hosting in the business, with a local development tool to match.

SHOPTALK SHOW

ShopTalk is a podcast all about front-end web design and development.