



# Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

michalsnik / aos

Animate on scroll library

259 commits

5 branches

0 packages

41 releases

18 contributors

MIT

Branch: next

New pull request

Create new file

Upload files

Find file

Clone or download

michalsnik

3.0.0-beta.6

Latest commit 329fb34 on 4 Oct 2018

cypress	Add disableMutationObserver setting	2 years ago
demo	Add disableMutationObserver setting	2 years ago
scripts	Fix code style	2 years ago
src	fix: disable mutation observing when not supported (#405)	2 years ago
.babelrc	Clean repository and configuration, replace webpack with rollup	2 years ago
.editorconfig	replace gulp with webpack, use ES6	4 years ago
.eslintrc.json	Add eslint & prettier	2 years ago
.gitattributes	[TR-3] - adding travis CI, refactoring gulp	5 years ago
.gitignore	Add cypress and live-server, update readme & contribution guide	2 years ago
.npmignore	Remove src from npmignore	2 years ago
.travis.yml	Update travis config	2 years ago
CONTRIBUTING.md	Add cypress and live-server, update readme & contribution guide	2 years ago
ISSUE_TEMPLATE.md	Add Issue and PR templates	2 years ago
LICENSE	initial commit	5 years ago
PULL_REQUEST_TEMPLATE.md	Add Issue and PR templates	2 years ago
README.md	Add throttleDelay and debounceDelay settings	2 years ago
cypress.json	Add cypress and live-server, update readme & contribution guide	2 years ago
package.json	3.0.0-beta.6	2 years ago
rollup.config.js	Update sourcemap setting in rollup	2 years ago
yarn.lock	Add global anchorPlacement setting	2 years ago

README.md

AOS - Animate on scroll library

npm@next

v3.0.0-beta.6

downloads 178k/month

build passing

chat on github



# !!! This is README for aos@next !!!

For last stable release (v2) go [here](#)



## ★ Codepen Examples

- [Different built-in animations](#)
- [With anchor setting in use](#)
- [With anchor-placement and different easings](#)
- [With simple custom animations](#)

👉 To get a better understanding how this actually works, I encourage you to check [my post on CSS-tricks](#).

## ⚙️ Installation

### Basic

Add styles in `<head>` :

```
<link rel="stylesheet" href="https://unpkg.com/aos@next/dist/aos.css" />
```

Add script right before closing `</body>` tag, and initialize AOS:

```
<script src="https://unpkg.com/aos@next/dist/aos.js"></script>
<script>
  AOS.init();
</script>
```

### Using package managers

Install aos package:

- `yarn add aos@next`
- Or `npm install --save aos@next`

Import script, styles and initialize AOS:

```
import AOS from 'aos';
import 'aos/dist/aos.css'; // You can also use <link> for styles
// ..
AOS.init();
```

In order to make it work you'll have to make sure your build process has configured styles loader, and bundles it all correctly. If you're using [Parcel](#) however, it will work out of the box as provided.

## 🤔 How to use it?

### 1. Initialize AOS:

```
AOS.init();

// You can also pass an optional settings object
// below listed default settings
```

```
AOS.init({
  // Global settings:
  disable: false, // accepts following values: 'phone', 'tablet', 'mobile', boolean, expression or function
  startEvent: 'DOMContentLoaded', // name of the event dispatched on the document, that AOS should initialize
  initClassName: 'aos-init', // class applied after initialization
  animatedClassName: 'aos-animate', // class applied on animation
  useClassNames: false, // if true, will add content of `data-aos` as classes on scroll
  disableMutationObserver: false, // disables automatic mutations' detections (advanced)
  debounceDelay: 50, // the delay on debounce used while resizing window (advanced)
  throttleDelay: 99, // the delay on throttle used while scrolling the page (advanced)

  // Settings that can be overridden on per-element basis, by `data-aos-*` attributes:
  offset: 120, // offset (in px) from the original trigger point
  delay: 0, // values from 0 to 3000, with step 50ms
  duration: 400, // values from 0 to 3000, with step 50ms
  easing: 'ease', // default easing for AOS animations
  once: false, // whether animation should happen only once - while scrolling down
  mirror: false, // whether elements should animate out while scrolling past them
  anchorPlacement: 'top-bottom', // defines which position of the element regarding to window should trigger t
});
```

## 2. Set animation using `data-aos` attribute:

```
<div data-aos="fade-in"></div>
```

And adjust behaviour by using `data-aos-*` attributes:

```
<div
  data-aos="fade-up"
  data-aos-offset="200"
  data-aos-delay="50"
  data-aos-duration="1000"
  data-aos-easing="ease-in-out"
  data-aos-mirror="true"
  data-aos-once="false"
  data-aos-anchor-placement="top-center"
>
</div>
```

[See full list of all animations, easings and anchor placements](#)

### Anchor

There is also a setting that can be used only on per-element basis:

- `data-aos-anchor` - element whose offset will be used to trigger animation instead of an actual one.

Examples:

```
<div data-aos="fade-up" data-aos-anchor=".other-element"></div>
```

This way you can trigger animation on one element, while you scroll to another - useful in animating fixed elements.

## API

AOS object is exposed as a global variable, for now there are three methods available:

- `init` - initialize AOS
- `refresh` - recalculate all offsets and positions of elements (called on window resize)
- `refreshHard` - reinit array with AOS elements and trigger `refresh` (called on DOM changes that are related to aos elements)

Example execution:

```
AOS.refresh();
```

By default AOS is watching for DOM changes and if there are any new elements loaded asynchronously or when something is removed from DOM it calls `refreshHard` automatically. In browsers that don't support `MutationObserver` like IE you might need to call `AOS.refreshHard()` by yourself.

`refresh` method is called on window resize and so on, as it doesn't require to build new store with AOS elements and should be as light as possible.

## JS Events

---

AOS dispatches two events on document: `aos:in` and `aos:out` whenever any element animates in or out, so that you can do extra stuff in JS:

```
document.addEventListener('aos:in', ({ detail }) => {
  console.log('animated in', detail);
});

document.addEventListener('aos:out', ({ detail }) => {
  console.log('animated out', detail);
});
```

You can also tell AOS to trigger custom event on specific element, by setting `data-aos-id` attribute:

```
<div data-aos="fade-in" data-aos-id="super-duper"></div>
```

Then you'll be able to listen for two custom events then:

- `aos:in:super-duper`
- `aos:out:super-duper`

## Recipes:

---

### Adding custom animations:

Sometimes built-in animations are just not enough. Let's say you need one box to have different animation depending on resolution. Here's how you could do it:

```
[data-aos="new-animation"] {
  opacity: 0;
  transition-property: transform, opacity;

  &.aos-animate {
    opacity: 1;
  }

  @media screen and (min-width: 768px) {
    transform: translateX(100px);

    &.aos-animate {
      transform: translateX(0);
    }
  }
}
```

Then use it in HTML:

```
<div data-aos="new-animation"></div>
```

The element will only animate opacity on mobile devices, but from 768px width it'll also slide from right to left.

**Adding custom easing:**

Similar to animations you can add custom easings:

```
[data-aos] {
  body[data-aos-easing="new-easing"] &,
  &[data-aos][data-aos-easing="new-easing"] {
    transition-timing-function: cubic-bezier(.250, .250, .750, .750);
  }
}
```

**Customizing default animations distance**

Default distance for built-in animations is 100px. As long as you're using SCSS though, you can override it:

```
$aos-distance: 200px; // It has to be above import
@import 'node_modules/aos/src/sass/aos.scss';
```

You have to however configure your build process to allow it to import styles from `node_modules` beforehand.

**Integrating external CSS animation library (e.g. Animate.css):**

Use `animatedClassName` to change default behaviour of AOS, to apply class names placed inside `data-aos` on scroll.

```
<div data-aos="fadeInUp"></div>
```

```
AOS.init({
  useClassNames: true,
  initClassName: false,
  animatedClassName: 'animated',
});
```

The above element will get two classes: `animated` and `fadeInUp`. Using different combinations of the three above settings, you should be able to integrate any external CSS animation library.

External libraries however don't care too much about animation state before the actual animation. So if you want those elements to be not visible before scrolling, you might need to add similar styles:

```
[data-aos] {
  visibility: hidden;
}
[data-aos].animated {
  visibility: visible;
}
```

**Caveats:****setting: duration , delay**

Duration and delay accept values from 50 to 3000, with step 50ms, it's because those are handled by css, and to not make css longer than it is already I implemented only a subset. I believe those should cover most cases.

If not, you can write simple CSS that will add another duration, for example:

```
body[data-aos-duration='4000'] [data-aos],
[data-aos][data-aos][data-aos-duration='4000'] {
  transition-duration: 4000ms;
}
```

This code will add 4000ms duration available for you to set on AOS elements, or to set as global duration while initializing AOS script. Notice that double `[data-aos][data-aos]` - it's not a mistake, it is a trick, to make individual settings more important than global, without need to write ugly "important" there :)

Example usage:

```
<div data-aos="fade-in" data-aos-duration="4000"></div>
```

## Predefined options

---

### Animations

- Fade animations:
  - fade
  - fade-up
  - fade-down
  - fade-left
  - fade-right
  - fade-up-right
  - fade-up-left
  - fade-down-right
  - fade-down-left
- Flip animations:
  - flip-up
  - flip-down
  - flip-left
  - flip-right
- Slide animations:
  - slide-up
  - slide-down
  - slide-left
  - slide-right
- Zoom animations:
  - zoom-in
  - zoom-in-up
  - zoom-in-down
  - zoom-in-left
  - zoom-in-right
  - zoom-out
  - zoom-out-up
  - zoom-out-down
  - zoom-out-left
  - zoom-out-right

### Anchor placements:

- top-bottom
- top-center
- top-top
- center-bottom
- center-center
- center-top
- bottom-bottom

- bottom-center
- bottom-top

### Easing functions:

- linear
- ease
- ease-in
- ease-out
- ease-in-out
- ease-in-back
- ease-out-back
- ease-in-out-back
- ease-in-sine
- ease-out-sine
- ease-in-out-sine
- ease-in-quad
- ease-out-quad
- ease-in-out-quad
- ease-in-cubic
- ease-out-cubic
- ease-in-out-cubic
- ease-in-quart
- ease-out-quart
- ease-in-out-quart

### ? Questions

---

If you found a bug, have a question or an idea, please check [AOS contribution guide](#) and don't hesitate to create new issues.