daneden / **animate.css**

🍿 A cross-browser library of CSS animations. As easy to use as an easy thing. http://daneden.github.io/animate.css

#css-animations #css #animation #stylesheets

428 commits | **4** branches | **24** releases | **91** contributors | MIT

Branch: **master** ▾ | New pull request | Create new file | Upload files | Find file | Clone or download ▾

WarenGonzaga and **daneden** Update devdependencies (#888) … Latest commit 01e83a7 19 days ago

source	Add support for prefers-reduced-motion media query (#832)	5 months ago
.editorconfig	More info in editorconfig	a year ago
.gitignore	Minor Updates and Small Fixes (#790)	5 months ago
.prettierignore	EOL	a year ago
.travis.yml	Add new animation speed & delay classes	5 months ago
LICENSE	Update license year (#775)	a year ago
README.md	Cleaning up the Readme file (#872)	2 months ago
animate-config.json	Fixed typos 😃	5 months ago
animate.css	Add support for prefers-reduced-motion media query (#832)	5 months ago
animate.min.css	Add support for prefers-reduced-motion media query (#832)	5 months ago
bower.json	Format files	a year ago
gulpfile.js	Add support for prefers-reduced-motion media query (#832)	5 months ago
npm-shrinkwrap.json	Update devdependencies (#888)	19 days ago
package.json	Update devdependencies (#888)	19 days ago

README.md

Animate.css

release v3.7.0 | cdnjs v3.7.0 | build passing | devDependencies up to date | chat gitter | npm package 3.7.0

Just-add-water CSS animation

`animate.css` is a bunch of cool, fun, and cross-browser animations for you to use in your projects. Great for emphasis, home pages, sliders, and general just-add-water-awesomeness.

Installation

Install via npm:

```
$ npm install animate.css --save
```

or yarn:

```
$ yarn add animate.css
```

Usage

To use animate.css in your website, simply drop the stylesheet into your document's `<head>`, and add the class `animated` to an element, along with any of the animation names. That's it! You've got a CSS animated element. Super!

```
<head>
  <link rel="stylesheet" href="animate.min.css">
</head>
```

or use a CDN hosted version by CDNJS

```
<head>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/3.7.0/animate.min.css">
</head>
```

Animations

To animate an element, add the class `animated` to an element. You can include the class `infinite` for an infinite loop. Finally you need to add one of the following classes to the element:

Class Name			
`bounce`	`flash`	`pulse`	`rubberBand`
`shake`	`headShake`	`swing`	`tada`
`wobble`	`jello`	`bounceIn`	`bounceInDown`
`bounceInLeft`	`bounceInRight`	`bounceInUp`	`bounceOut`
`bounceOutDown`	`bounceOutLeft`	`bounceOutRight`	`bounceOutUp`
`fadeIn`	`fadeInDown`	`fadeInDownBig`	`fadeInLeft`
`fadeInLeftBig`	`fadeInRight`	`fadeInRightBig`	`fadeInUp`
`fadeInUpBig`	`fadeOut`	`fadeOutDown`	`fadeOutDownBig`
`fadeOutLeft`	`fadeOutLeftBig`	`fadeOutRight`	`fadeOutRightBig`
`fadeOutUp`	`fadeOutUpBig`	`flipInX`	`flipInY`
`flipOutX`	`flipOutY`	`lightSpeedIn`	`lightSpeedOut`
`rotateIn`	`rotateInDownLeft`	`rotateInDownRight`	`rotateInUpLeft`
`rotateInUpRight`	`rotateOut`	`rotateOutDownLeft`	`rotateOutDownRight`
`rotateOutUpLeft`	`rotateOutUpRight`	`hinge`	`jackInTheBox`
`rollIn`	`rollOut`	`zoomIn`	`zoomInDown`
`zoomInLeft`	`zoomInRight`	`zoomInUp`	`zoomOut`
`zoomOutDown`	`zoomOutLeft`	`zoomOutRight`	`zoomOutUp`
`slideInDown`	`slideInLeft`	`slideInRight`	`slideInUp`
`slideOutDown`	`slideOutLeft`	`slideOutRight`	`slideOutUp`
`heartBeat`			

Full example:

```
<h1 class="animated infinite bounce delay-2s">Example</h1>
```

Check out all the animations here!

It's possible to change the duration of your animations, add a delay or change the number of times that it plays:

```
.yourElement {
  animation-duration: 3s;
  animation-delay: 2s;
  animation-iteration-count: infinite;
}
```

Usage with jQuery

You can do a whole bunch of other stuff with animate.css when you combine it with jQuery. A simple example:

```
$('#yourElement').addClass('animated bounceOutLeft');
```

You can also detect when an animation ends:

```
// See https://github.com/daneden/animate.css/issues/644
var animationEnd = (function(el) {
  var animations = {
    animation: 'animationend',
    OAnimation: 'oAnimationEnd',
    MozAnimation: 'mozAnimationEnd',
    WebkitAnimation: 'webkitAnimationEnd',
  };

  for (var t in animations) {
    if (el.style[t] !== undefined) {
      return animations[t];
    }
  }
})(document.createElement('div'));

$('#yourElement').one(animationEnd, doSomething);
```

View a video tutorial on how to use Animate.css with jQuery here.

Note: `jQuery.one()` is used when you want to execute the event handler at most *once*. More information here.

It's possible to extend jQuery and add a function that does it all for you:

```
$.fn.extend({
  animateCss: function(animationName, callback) {
    var animationEnd = (function(el) {
      var animations = {
        animation: 'animationend',
        OAnimation: 'oAnimationEnd',
        MozAnimation: 'mozAnimationEnd',
        WebkitAnimation: 'webkitAnimationEnd',
      };

      for (var t in animations) {
        if (el.style[t] !== undefined) {
          return animations[t];
        }
      }
    })(document.createElement('div'));

    this.addClass('animated ' + animationName).one(animationEnd, function() {
      $(this).removeClass('animated ' + animationName);

      if (typeof callback === 'function') callback();
    });

    return this;
  },
});
```

And use it like this:

```
$('#yourElement').animateCss('bounce');
or;
$('#yourElement').animateCss('bounce', function() {
  // Do something after animation
});
```

Setting *Delay* and *Speed*

Delay Class

It's possible to add delays directly on the element's class attribute, just like this:

```
<div class="animated bounce delay-2s">Example</div>
```

Class Name	Delay Time
`delay-2s`	`2s`
`delay-3s`	`3s`
`delay-4s`	`4s`
`delay-5s`	`5s`

> ***Note**: The default delays are from 1 second to 5 seconds only. If you need custom delays, add it directly to your own CSS code.*

Slow, Slower, Fast, and Faster Class

It's possible to control the speed of the animation by adding these classes, as a sample below:

```
<div class="animated bounce faster">Example</div>
```

Class Name	Speed Time
`slow`	`2s`
`slower`	`3s`
`fast`	`800ms`
`faster`	`500ms`

> ***Note**: The `animated` class has a default speed of `1s`. If you need custom duration, add it directly to your own CSS code.*

Custom Builds

Animate.css is powered by gulp.js, which means you can create custom builds pretty easily. First of all, you'll need Gulp and all other dependencies:

```
$ cd path/to/animate.css/
$ sudo npm install
```

Next, run `gulp` to compile your custom builds. For example, if you want only some of the "attention seekers", simply edit the `animate-config.json` file to select only the animations you want to use.

```
"attention_seekers": {
  "bounce": true,
  "flash": false,
  "pulse": false,
  "shake": true,
  "headShake": true,
  "swing": true,
  "tada": true,
  "wobble": true,
  "jello":true
}
```

Accessibility

Animate.css supports the `prefers-reduced-motion` media query so that users with motion sensitivity can opt out of animations. On supported platforms (currently only OSX Safari and iOS Safari), users can select "reduce motion" on their operating system preferences and it will turn off CSS transitions for them without any further work required.

License

Animate.css is licensed under the MIT license. (http://opensource.org/licenses/MIT)

Contributing

Pull requests are the way to go here. We only have two rules for submitting a pull request: match the naming convention (camelCase, categorised [fades, bounces, etc]) and let us see a demo of submitted animations in a pen. That **last one is important**.