View on GitHub

jQuery-animations

A CSS3 animations development framework based on jQuery provides a easy way to develop cross browser CSS3 animations and use.

Download this project as a .zip file Download this project as a tar.gz file

jQuery-animations

A CSS3 animation framework based on jQuery providing an easy way to develop cross browser CSS3 animations.

For the user, it could be easy to use jQuery method to perform and combine the animations.

For the developer, it could be easy to develop cross browser CSS3 animations by using JavaScript.

Download

Compress
Uncompress

Demo

Effects
Tile
Integrate with CSS Library

Browser Support

jQuery-animations currently supports IE10+, Chrome, Firefox, Safari and Opera.

Usage

Methods

animate(id, [options])

Perform CSS3 animations. This method extends from jQuery.animate(), so the options naming follows its.

id: `String`

Sets the animation ID(s) or name(s) you want to perform. It could be a mutiple animations by including each one separated by a space.

Available animations please refer to Effects.

ID means that predined in JavaScript plugins, and name means that declares in CSS.

options: `Object`

Sets the animations options.

duration: `Number` *(default: animation define or* `400`*)*

Sets the number determining how long the animation will run(ms).

delay: `Number` *(default: animation define or* `0`*)*

Sets the number determining when the animation will start.(ms).

repeat: `Number` *or* `String` *(default: animation define or* `1`*)*

Sets the number determining the number of times an animation is played.

Available values please refer to animation-iteration-count

easing: `String` *(default: animation define or "*`ease`*")*

Sets the easing function to use for the transition.

Available values please refer to animation-timing-function

direction: `String` *(default: animation define or "*`normal`*")*

Sets the property whether or not the animation should play in reverse on alternate cycles.

Available values please refer to animation-direction

fillMode: `String` (default: animation define or `"none"`)

Sets the property specifies what styles will apply for the element when the animation is not playing. You can use `reset()` method to clear the state after `forwards` or `both`.

Available values please refer to [animation-fill-mode]

combinable: `Boolean` (default: `false`)

Sets the flag determining the animation combines with other animations. It will stop running animation when sets false.

wrap: `Boolean` (default: `false`)

Sets the flag determining the animations applies to a new wrapper.

autoWrap: `Boolean` (default: `true`)

Sets the flag determining the animations applies to a new wrapper when combine other animations automatically. You can disable this flag if the animations are simple without conflicts for better performace. eg. fadeOut and shake can apply in the same element easily, but shake and bounce use the same css property and conflict.

prepare: `Function(options)`

Sets the callback function to call once animations are ready to begin.

start: `Function(options)`

Sets the callback function to call once the animation begins..

complete: `Function(options)`

Sets the callback function to call once the animation is complete.

fail: `Function(options)`

Sets the callback function to call when the animation fails to complete.

always: `Function(options)`

Sets the callback function to call when the animation completes or stops without completing.

end: `Function(options)`

Sets the callback function to call when the animation completes or stops without completing. But, if there is any animation running inside target element, it won't trigger untill all done.

clear: `Function(options)`

Sets the callback function to call when the animation clear or reset triggered.

resize: `Function(options)`

Sets the callback function to call when the window resize and you can handle your customized animations.

custom: `Object`

Set customized options for each animation. Defines the same key as animation ID or name in this object. Customized options structure is the same with options of global.

animation defined options

There could be some plugin defined options, eg. `strength` option in `shake` animation.

animate(animation)

Perform CSS3 inlince animations.

animation: `Object`

Extends options structure previously, but there must be a option `keyframes` in this object.

keyframes: `Object`

Specifies the keyframes of animation, just like css structure.

finish()

Stop CSS3 animations and trigger complete event. This method extends from [jQuery.finish()]

stop()

Stop CSS3 animations and trigger fail event. This method extends from jQuery.stop()

reset()

Clear CSS3 keyframe stopped after `forwards` and `both`.

Example

Basic usage

```
$('#want-to-animate').animate('shake');
```

You can also combine multiple animations once

```
$('#want-to-animate').animate('flyToUp flyToRight fadeOut', {
  complete: function(options) {
    $('#want-to-animate').remove();
  }
});
```

Or combine multiple animations in different time

```
$('#want-to-animate').animate('shake', {combinable: true});
$('#want-to-animate').animate('fadeOut', {combinable: true});
```

Inline animation

```
$('#want-to-animate').animate({
  keyframes: {
    to: {
      transform: 'rotate(360deg)'
    }
  }
});
```

Sometimes you could combine mutiple animations frequent use to a new one.

```
$.animations['hit'] = { shortcut: 'fadeOut shake bounce' };
// now you can call the new animation 'hit'
$('#want-to-animate').animate('hit');
```

With options

```
$('#want-to-animate').animate('animation1 animation2', {
  // Sets the number determining how long the animation will run(ms).
  duration: 400,

  // Sets the number determining when the animation will start.(ms).
  delay: 0,

  // Sets the number determining the number of times an animation is played.
  repeat: 1,

  // Sets the easing function to use for the transition.
  easing: 'ease',

  // Sets the property whether or not the animation should play in reverse on alternate cycles.
  direction: 'normal',

  // Sets the property specifies what styles will apply for the element when the animation is not playing.
  fillMode: 'none',

  // Sets the flag determining the animation combines with other animations. It will stop running animation when sets false.
  combinable: false,

  // Sets the flag determining the animations applies to a new wrapper.
  wrap: false,

  // Sets the flag determining the animations applies to a new wrapper when combine other animations automatically. You can disable this f
  autoWrap: true,

  // Sets the callback function to call once animations are ready to begin.
  prepare: function(options) { },

  // Sets the callback function to call once the animation begins..
  start: function(options) { },

  // Sets the callback function to call once the animation is complete.
  complete: function(options) { },

  // Sets the callback function to call when the animation fails to complete.
  fail: function(options) { },

  // Sets the callback function to call when the animation completes or stops without completing.
  always: function(options) { },

  // Sets the callback function to call when the animation clear or reset triggered.
  clear: function(options) { },

  // Sets the callback function to call when the window resize and you can handle your customized animations.
  resize: function(options) {},

  // Sets the callback function to call when the animation completes or stops without completing. But, if there is any animation running i
  end: function() { },

  // Set customized options for each animation. Defines the same key as animation ID or name in this object. Customized options structure
  custom: {
    animation1: {
      // specify options for this animation...

      duration: 1000

      // ...
    }
  },

  // There could be some plugin defined options.
  var1: 'something'

  // ...
});
```

Animations and Options

fadeIn

No options

fadeOut

No options

bounce

strength: `Number` (default: `20`)

Sets the strength of bounce.

shake

strength: `Number` (default: `20`)

Sets the strength of shake.

slideFromUp

distance: `Number` (default: `null` as element height)

Sets the distance of top position.

slideFromDown

distance: `Number` (default: `null` as element height)

Sets the distance of bottom position.

slideFromRight

distance: `Number` (default: `null` as element width)

Sets the distance of right position.

slideFromLeft

distance: `Number` (default: `null` as element width)

Sets the distance of left position.

slideToUp

distance: `Number` (default: `null` as element height)

Sets the distance of top position.

slideToDown

distance: `Number` (default: `null` as element height)

Sets the distance of bottom position.

slideToRight

distance: `Number` (default: `null` as element width)

Sets the distance of right position.

slideToLeft

distance: `Number` (default: `null` as element width)

Sets the distance of left position.

flyFromUp

distance: `Number` (default: `null` as document height)

Sets the distance of top position.

flyFromDown

distance: `Number` (default: `null` as document height)

Sets the distance of bottom position.

flyFromRight

distance: `Number` (default: `null` as document width)

Sets the distance of right position.

flyFromLeft

distance: `Number` (default: `null` as document width)

Sets the distance of left position.

flyToUp

distance: `Number` (default: `null` as document height)

Sets the distance of top position.

flyToDown

distance: `Number` (default: `null` as document height)

Sets the distance of bottom position.

flyToRight

distance: `Number` (default: `null` as document width)

Sets the distance of right position.

flyToLeft

distance: `Number` (default: `null` as document width)

Sets the distance of left position.

flyFrom

x: `Number` (default: `0`)

Sets the position x.

y: `Number` (default: `0`)

Sets the position y.

relative: `Boolean` (default: `false`)

Specifies position is relative or absolute.

flyTo

x: `Number` (default: `0`)

Sets the position x.

y: `Number` (default: `0`)

Sets the position y.

relative: `Boolean` (default: `false`)

Specifies position is relative or absolute.

flyIn

degree: `Number` (default: `null` as Random)

Sets the degree and fly from outside, 0 is right, 90 is top, 180 is left, 270 is bottom.

flyOut

degree: `Number` (default: `null` as Random)

Sets the degree and fly to outside, 0 is right, 90 is top, 180 is left, 270 is bottom.

rotate

startDeg: `Number` (default: `0`)

Sets the start degree.

endDeg: `Number` (default: `360`)

Sets the end degree.

startOrigin: `String` (default: `"50% 50% 0"`)

Sets the start origin.

endOrigin: `String` (default: `"50% 50% 0"`)

Sets the end origin.

zoomAway

startOrigin: `String` (default: `"50% 50% 0"`)

Sets the start origin.

endOrigin: `String` (default: `"50% 50% 0"`)

Sets the end origin.

zoomNear

startOrigin: `String` (default: `"50% 50% 0"`)

Sets the start origin.

endOrigin: `String` (default: `"50% 50% 0"`)

Sets the end origin.

zoomIn

scale: `Number` (default: `1.2`)

Sets the scale to zoom.

startOrigin: `String` (default: `"50% 50% 0"`)

Sets the start origin.

endOrigin: `String` (default: `"50% 50% 0"`)

Sets the end origin.

zoomOut

scale: `Number` (default: `1.2`)

Sets the scale to zoom.

startOrigin: `String` (default: `"50% 50% 0"`)

Sets the start origin.

endOrigin: `String` (default: `"50% 50% 0"`)

Sets the end origin.

scaleTo

x: `Number` (default: `1`)

Sets the x scale to zoom.

y: `Number` (default: `1`)

Sets the y scale to zoom.

startOrigin: `String` (default: `"50% 50% 0"`)

Sets the start origin.

endOrigin: `String` (default: `"50% 50% 0"`)

Sets the end origin.

scaleFrom

x: `Number` (default: `1`)

Sets the x scale to zoom.

y: `Number` (default: `1`)

Sets the y scale to zoom.

startOrigin: `String` (default: `"50% 50% 0"`)

Sets the start origin.

endOrigin: `String` (default: `"50% 50% 0"`)

Sets the end origin.

flipX

startDeg: `Number` (default: `0`)

Sets the start degree.

endDeg: `Number` (default: `360`)

Sets the end degree.

startOrigin: `String` (default: `"50% 50% 0"`)

Sets the start origin.

endOrigin: `String` (default: `"50% 50% 0"`)

Sets the end origin.

perspective: `Number` (default: `100`)

Sets the perspective.

perspectiveOrigin: `String` (default: `"50% 50%"`)

Sets the perspective origin.

flipY

startDeg: `Number` (default: `0`)

Sets the start degree.

endDeg: `Number` (default: `360`)

Sets the end degree.

startOrigin: `String` (default: `"50% 50% 0"`)

Sets the start origin.

endOrigin: `String` (default: `"50% 50% 0"`)

Sets the end origin.

perspective: `Number` (default: `100`)

Sets the perspective.

perspectiveOrigin: `String` *(default:* `"50% 50%"`*)*

Sets the perspective origin.

Developer Documentation

Coming soon.

License

The project is released under the MIT license.

Contact

The project's website is located at https://github.com/emn178/jquery-animations
Author: emn178@gmail.com

jQuery-animations maintained by emn178

Published with GitHub Pages