* [Chart.js](http://docs.google.com/)
* [Getting Started](http://docs.google.com/getting-started/)
  + [Installation](http://docs.google.com/getting-started/installation.html)
  + [Integration](http://docs.google.com/getting-started/integration.html)
  + [Usage](http://docs.google.com/getting-started/usage.html)
* [General](http://docs.google.com/general/)
  + [Responsive](http://docs.google.com/general/responsive.html)
  + [Pixel Ratio](http://docs.google.com/general/device-pixel-ratio.html)
  + [Interactions](http://docs.google.com/general/interactions/)
    - [Events](http://docs.google.com/general/interactions/events.html)
    - [Modes](http://docs.google.com/general/interactions/modes.html)
  + [Options](http://docs.google.com/general/options.html)
  + [Colors](http://docs.google.com/general/colors.html)
  + [Fonts](http://docs.google.com/general/fonts.html)
* [Configuration](http://docs.google.com/)
  + [Animations](http://docs.google.com/animations.html)
  + [Layout](http://docs.google.com/layout.html)
  + [Legend](http://docs.google.com/legend.html)
  + [Title](http://docs.google.com/title.html)
  + [Tooltip](http://docs.google.com/tooltip.html)
  + [Elements](http://docs.google.com/elements.html)
* [Charts](http://docs.google.com/charts/)
  + [Line](http://docs.google.com/charts/line.html)
  + [Bar](http://docs.google.com/charts/bar.html)
  + [Radar](http://docs.google.com/charts/radar.html)
  + [Doughnut & Pie](http://docs.google.com/charts/doughnut.html)
  + [Polar Area](http://docs.google.com/charts/polar.html)
  + [Bubble](http://docs.google.com/charts/bubble.html)
  + [Scatter](http://docs.google.com/charts/scatter.html)
  + [Area](http://docs.google.com/charts/area.html)
  + [Mixed](http://docs.google.com/charts/mixed.html)
* [Axes](http://docs.google.com/axes/)
  + [Cartesian](http://docs.google.com/axes/cartesian/)
    - [Category](http://docs.google.com/axes/cartesian/category.html)
    - [Linear](http://docs.google.com/axes/cartesian/linear.html)
    - [Logarithmic](http://docs.google.com/axes/cartesian/logarithmic.html)
    - [Time](http://docs.google.com/axes/cartesian/time.html)
  + [Radial](http://docs.google.com/axes/radial/)
    - [Linear](http://docs.google.com/axes/radial/linear.html)
  + [Labelling](http://docs.google.com/axes/labelling.html)
  + [Styling](http://docs.google.com/axes/styling.html)
* [Developers](http://docs.google.com/developers/)
  + [Chart.js API](http://docs.google.com/developers/api.html)
  + [Updating Charts](http://docs.google.com/developers/updates.html)
  + [Plugins](http://docs.google.com/developers/plugins.html)
  + [New Charts](http://docs.google.com/developers/charts.html)
  + [New Axes](http://docs.google.com/developers/axes.html)
  + [Contributing](http://docs.google.com/developers/contributing.html)
* [Additional Notes](http://docs.google.com/notes/)
  + [Comparison Table](http://docs.google.com/notes/comparison.html)
  + [Popular Extensions](http://docs.google.com/notes/extensions.html)
  + [License](http://docs.google.com/notes/license.html)
* [Published with GitBook](https://www.gitbook.com)

[**Animations**](http://docs.google.com/)

Animations

Chart.js animates charts out of the box. A number of options are provided to configure how the animation looks and how long it takes

## Animation Configuration

The following animation options are available. The global options for are defined in Chart.defaults.global.animation.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Default | Description |
| duration | Number | 1000 | The number of milliseconds an animation takes. |
| easing | String | 'easeOutQuart' | Easing function to use. [more...](#gjdgxs) |
| onProgress | Function | null | Callback called on each step of an animation. [more...](#30j0zll) |
| onComplete | Function | null | Callback called at the end of an animation. [more...](#30j0zll) |

## Easing

Available options are:

* 'linear'
* 'easeInQuad'
* 'easeOutQuad'
* 'easeInOutQuad'
* 'easeInCubic'
* 'easeOutCubic'
* 'easeInOutCubic'
* 'easeInQuart'
* 'easeOutQuart'
* 'easeInOutQuart'
* 'easeInQuint'
* 'easeOutQuint'
* 'easeInOutQuint'
* 'easeInSine'
* 'easeOutSine'
* 'easeInOutSine'
* 'easeInExpo'
* 'easeOutExpo'
* 'easeInOutExpo'
* 'easeInCirc'
* 'easeOutCirc'
* 'easeInOutCirc'
* 'easeInElastic'
* 'easeOutElastic'
* 'easeInOutElastic'
* 'easeInBack'
* 'easeOutBack'
* 'easeInOutBack'
* 'easeInBounce'
* 'easeOutBounce'
* 'easeInOutBounce'

See [Robert Penner's easing equations](http://robertpenner.com/easing/).

## Animation Callbacks

The onProgress and onComplete callbacks are useful for synchronizing an external draw to the chart animation. The callback is passed a Chart.Animation instance:

{  
 // Chart object  
 chart: Chart,  
  
 // Current Animation frame number  
 currentStep: Number,  
  
 // Number of animation frames  
 numSteps: Number,  
  
 // Animation easing to use  
 easing: String,  
  
 // Function that renders the chart  
 render: Function,  
  
 // User callback  
 onAnimationProgress: Function,  
  
 // User callback  
 onAnimationComplete: Function  
}

The following example fills a progress bar during the chart animation.

var chart = new Chart(ctx, {  
 type: 'line',  
 data: data,  
 options: {  
 animation: {  
 onProgress: function(animation) {  
 progress.value = animation.animationObject.currentStep / animation.animationObject.numSteps;  
 }  
 }  
 }  
});

Another example usage of these callbacks can be found on [Github](https://github.com/chartjs/Chart.js/blob/master/samples/advanced/progress-bar.html): this sample displays a progress bar showing how far along the animation is.

results matching ""

No results matching ""