 [Chart.js](http://docs.google.com/docs/3.9.1/)

[Home](http://docs.google.com/docs/3.9.1/)

[API](http://docs.google.com/docs/3.9.1/api/)

[Samples](http://docs.google.com/docs/3.9.1/samples/)

Ecosystem Ecosystem

* [Awesome (opens new window)](https://github.com/chartjs/awesome)
* [Slack (opens new window)](https://chartjs-slack.herokuapp.com/)
* [Stack Overflow (opens new window)](https://stackoverflow.com/questions/tagged/chart.js)

[GitHub (opens new window)](https://github.com/chartjs/Chart.js)

[Home](http://docs.google.com/docs/3.9.1/)

[API](http://docs.google.com/docs/3.9.1/api/)

[Samples](http://docs.google.com/docs/3.9.1/samples/)

Ecosystem Ecosystem

* [Awesome (opens new window)](https://github.com/chartjs/awesome)
* [Slack (opens new window)](https://chartjs-slack.herokuapp.com/)
* [Stack Overflow (opens new window)](https://stackoverflow.com/questions/tagged/chart.js)

[GitHub (opens new window)](https://github.com/chartjs/Chart.js)

* [Information](http://docs.google.com/docs/3.9.1/samples/information.html)
* Bar Charts
* Line Charts
* Other charts
* Area charts
* Scales
* Scale Options
* Legend
* Title
* Subtitle
* Tooltip
* Scriptable Options
* Animations
  + [Delay](http://docs.google.com/docs/3.9.1/samples/animations/delay.html)
  + [Drop](http://docs.google.com/docs/3.9.1/samples/animations/drop.html)
  + [Loop](http://docs.google.com/docs/3.9.1/samples/animations/loop.html)
  + [Progressive Line](http://docs.google.com/docs/3.9.1/samples/animations/progressive-line.html)
  + [Progressive Line With Easing](http://docs.google.com/docs/3.9.1/samples/animations/progressive-line-easing.html)
* Advanced
* Plugins
* [Utils](http://docs.google.com/docs/3.9.1/samples/utils.html)

[**#**](#gjdgxs) Progressive Line With Easing

config animation data actions

const config = { type: 'line', data: { datasets: [{ borderColor: Utils.CHART\_COLORS.red, borderWidth: 1, radius: 0, data: data, }, { borderColor: Utils.CHART\_COLORS.blue, borderWidth: 1, radius: 0, data: data2, }] }, options: { animation, interaction: { intersect: false }, plugins: { legend: false, title: { display: true, text: () => easing.name } }, scales: { x: { type: 'linear' } } } };

const config = {  
 type: 'line',  
 data: {  
 datasets: [{  
 borderColor: Utils.CHART\_COLORS.red,  
 borderWidth: 1,  
 radius: 0,  
 data: data,  
 },  
 {  
 borderColor: Utils.CHART\_COLORS.blue,  
 borderWidth: 1,  
 radius: 0,  
 data: data2,  
 }]  
 },  
 options: {  
 animation,  
 interaction: {  
 intersect: false  
 },  
 plugins: {  
 legend: false,  
 title: {  
 display: true,  
 text: () => easing.name  
 }  
 },  
 scales: {  
 x: {  
 type: 'linear'  
 }  
 }  
 }  
};

let easing = helpers.easingEffects.easeOutQuad; let restart = false; const totalDuration = 5000; const duration = (ctx) => easing(ctx.index / data.length) \* totalDuration / data.length; const delay = (ctx) => easing(ctx.index / data.length) \* totalDuration; const previousY = (ctx) => ctx.index === 0 ? ctx.chart.scales.y.getPixelForValue(100) : ctx.chart.getDatasetMeta(ctx.datasetIndex).data[ctx.index - 1].getProps(['y'], true).y; const animation = { x: { type: 'number', easing: 'linear', duration: duration, from: NaN, // the point is initially skipped delay(ctx) { if (ctx.type !== 'data' || ctx.xStarted) { return 0; } ctx.xStarted = true; return delay(ctx); } }, y: { type: 'number', easing: 'linear', duration: duration, from: previousY, delay(ctx) { if (ctx.type !== 'data' || ctx.yStarted) { return 0; } ctx.yStarted = true; return delay(ctx); } } };

let easing = helpers.easingEffects.easeOutQuad;  
let restart = false;  
const totalDuration = 5000;  
const duration = (ctx) => easing(ctx.index / data.length) \* totalDuration / data.length;  
const delay = (ctx) => easing(ctx.index / data.length) \* totalDuration;  
const previousY = (ctx) => ctx.index === 0 ? ctx.chart.scales.y.getPixelForValue(100) : ctx.chart.getDatasetMeta(ctx.datasetIndex).data[ctx.index - 1].getProps(['y'], true).y;  
const animation = {  
 x: {  
 type: 'number',  
 easing: 'linear',  
 duration: duration,  
 from: NaN, // the point is initially skipped  
 delay(ctx) {  
 if (ctx.type !== 'data' || ctx.xStarted) {  
 return 0;  
 }  
 ctx.xStarted = true;  
 return delay(ctx);  
 }  
 },  
 y: {  
 type: 'number',  
 easing: 'linear',  
 duration: duration,  
 from: previousY,  
 delay(ctx) {  
 if (ctx.type !== 'data' || ctx.yStarted) {  
 return 0;  
 }  
 ctx.yStarted = true;  
 return delay(ctx);  
 }  
 }  
};

const data = []; const data2 = []; let prev = 100; let prev2 = 80; for (let i = 0; i < 1000; i++) { prev += 5 - Math.random() \* 10; data.push({x: i, y: prev}); prev2 += 5 - Math.random() \* 10; data2.push({x: i, y: prev2}); }

const data = [];  
const data2 = [];  
let prev = 100;  
let prev2 = 80;  
for (let i = 0; i < 1000; i++) {  
 prev += 5 - Math.random() \* 10;  
 data.push({x: i, y: prev});  
 prev2 += 5 - Math.random() \* 10;  
 data2.push({x: i, y: prev2});  
}

function restartAnims(chart) { chart.stop(); const meta0 = chart.getDatasetMeta(0); const meta1 = chart.getDatasetMeta(1); for (let i = 0; i < data.length; i++) { const ctx0 = meta0.controller.getContext(i); const ctx1 = meta1.controller.getContext(i); ctx0.xStarted = ctx0.yStarted = false; ctx1.xStarted = ctx1.yStarted = false; } chart.update(); } const actions = [ { name: 'easeOutQuad', handler(chart) { easing = helpers.easingEffects.easeOutQuad; restartAnims(chart); } }, { name: 'easeOutCubic', handler(chart) { easing = helpers.easingEffects.easeOutCubic; restartAnims(chart); } }, { name: 'easeOutQuart', handler(chart) { easing = helpers.easingEffects.easeOutQuart; restartAnims(chart); } }, { name: 'easeOutQuint', handler(chart) { easing = helpers.easingEffects.easeOutQuint; restartAnims(chart); } }, { name: 'easeInQuad', handler(chart) { easing = helpers.easingEffects.easeInQuad; restartAnims(chart); } }, { name: 'easeInCubic', handler(chart) { easing = helpers.easingEffects.easeInCubic; restartAnims(chart); } }, { name: 'easeInQuart', handler(chart) { easing = helpers.easingEffects.easeInQuart; restartAnims(chart); } }, { name: 'easeInQuint', handler(chart) { easing = helpers.easingEffects.easeInQuint; restartAnims(chart); } }, ];

function restartAnims(chart) {  
 chart.stop();  
 const meta0 = chart.getDatasetMeta(0);  
 const meta1 = chart.getDatasetMeta(1);  
 for (let i = 0; i < data.length; i++) {  
 const ctx0 = meta0.controller.getContext(i);  
 const ctx1 = meta1.controller.getContext(i);  
 ctx0.xStarted = ctx0.yStarted = false;  
 ctx1.xStarted = ctx1.yStarted = false;  
 }  
 chart.update();  
}  
const actions = [  
 {  
 name: 'easeOutQuad',  
 handler(chart) {  
 easing = helpers.easingEffects.easeOutQuad;  
 restartAnims(chart);  
 }  
 },  
 {  
 name: 'easeOutCubic',  
 handler(chart) {  
 easing = helpers.easingEffects.easeOutCubic;  
 restartAnims(chart);  
 }  
 },  
 {  
 name: 'easeOutQuart',  
 handler(chart) {  
 easing = helpers.easingEffects.easeOutQuart;  
 restartAnims(chart);  
 }  
 },  
 {  
 name: 'easeOutQuint',  
 handler(chart) {  
 easing = helpers.easingEffects.easeOutQuint;  
 restartAnims(chart);  
 }  
 },  
 {  
 name: 'easeInQuad',  
 handler(chart) {  
 easing = helpers.easingEffects.easeInQuad;  
 restartAnims(chart);  
 }  
 },  
 {  
 name: 'easeInCubic',  
 handler(chart) {  
 easing = helpers.easingEffects.easeInCubic;  
 restartAnims(chart);  
 }  
 },  
 {  
 name: 'easeInQuart',  
 handler(chart) {  
 easing = helpers.easingEffects.easeInQuart;  
 restartAnims(chart);  
 }  
 },  
 {  
 name: 'easeInQuint',  
 handler(chart) {  
 easing = helpers.easingEffects.easeInQuint;  
 restartAnims(chart);  
 }  
 },  
];

## [**#**](#30j0zll) Api

* [Chart](http://docs.google.com/docs/3.9.1/api/classes/Chart.html)
  + [getDatasetMeta](http://docs.google.com/docs/3.9.1/api/classes/Chart.html#getdatasetmeta)
* [Scale](http://docs.google.com/docs/3.9.1/api/classes/Scale.html)
  + [getPixelForValue](http://docs.google.com/docs/3.9.1/api/classes/Scale.html#getpixelforvalue)

## [**#**](#1fob9te) Docs

* [Animations](http://docs.google.com/docs/3.9.1/configuration/animations.html)
  + [animation](http://docs.google.com/docs/3.9.1/configuration/animations.html#animation)
    - delay
    - duration
    - easing
    - loop
  + [Easing](http://docs.google.com/docs/3.9.1/configuration/animations.html#easing)
* [Line](http://docs.google.com/docs/3.9.1/charts/line.html)
* [Options](http://docs.google.com/docs/3.9.1/general/options.html)
  + [Scriptable Options](http://docs.google.com/docs/3.9.1/general/options.html#scriptable-options)
    - [Data Context](http://docs.google.com/docs/3.9.1/general/options.html#data)

Last Updated: 8/3/2022, 12:46:38 PM

←  [Progressive Line](http://docs.google.com/docs/3.9.1/samples/animations/progressive-line.html)   [Data Decimation](http://docs.google.com/docs/3.9.1/samples/advanced/data-decimation.html)  →