 [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

config animation data

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 }, 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  
 },  
 scales: {  
 x: {  
 type: 'linear'  
 }  
 }  
 }  
};

const totalDuration = 10000; const delayBetweenPoints = totalDuration / data.length; 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: delayBetweenPoints, from: NaN, // the point is initially skipped delay(ctx) { if (ctx.type !== 'data' || ctx.xStarted) { return 0; } ctx.xStarted = true; return ctx.index \* delayBetweenPoints; } }, y: { type: 'number', easing: 'linear', duration: delayBetweenPoints, from: previousY, delay(ctx) { if (ctx.type !== 'data' || ctx.yStarted) { return 0; } ctx.yStarted = true; return ctx.index \* delayBetweenPoints; } } };

const totalDuration = 10000;  
const delayBetweenPoints = totalDuration / data.length;  
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: delayBetweenPoints,  
 from: NaN, // the point is initially skipped  
 delay(ctx) {  
 if (ctx.type !== 'data' || ctx.xStarted) {  
 return 0;  
 }  
 ctx.xStarted = true;  
 return ctx.index \* delayBetweenPoints;  
 }  
 },  
 y: {  
 type: 'number',  
 easing: 'linear',  
 duration: delayBetweenPoints,  
 from: previousY,  
 delay(ctx) {  
 if (ctx.type !== 'data' || ctx.yStarted) {  
 return 0;  
 }  
 ctx.yStarted = true;  
 return ctx.index \* delayBetweenPoints;  
 }  
 }  
};

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});  
}

## [**#**](#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
* [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

←  [Loop](http://docs.google.com/docs/3.9.1/samples/animations/loop.html)   [Progressive Line With Easing](http://docs.google.com/docs/3.9.1/samples/animations/progressive-line-easing.html)  →