 [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)

* [Chart.js](http://docs.google.com/docs/3.9.1/)
* Getting Started
* General
* Configuration
* Chart Types
  + [Area Chart](http://docs.google.com/docs/3.9.1/charts/area.html)
  + [Bar Chart](http://docs.google.com/docs/3.9.1/charts/bar.html)
  + [Bubble Chart](http://docs.google.com/docs/3.9.1/charts/bubble.html)
  + [Doughnut and Pie Charts](http://docs.google.com/docs/3.9.1/charts/doughnut.html)
  + [Line Chart](http://docs.google.com/docs/3.9.1/charts/line.html)
  + [Mixed Chart Types](http://docs.google.com/docs/3.9.1/charts/mixed.html)
  + [Polar Area Chart](http://docs.google.com/docs/3.9.1/charts/polar.html)
  + [Radar Chart](http://docs.google.com/docs/3.9.1/charts/radar.html)
  + [Scatter Chart](http://docs.google.com/docs/3.9.1/charts/scatter.html)
* Axes
* Developers

[**#**](#gjdgxs) Line Chart

A line chart is a way of plotting data points on a line. Often, it is used to show trend data, or the comparison of two data sets.

config setup

const config = { type: 'line', data: data, };

const config = {  
 type: 'line',  
 data: data,  
};

const labels = Utils.months({count: 7}); const data = { labels: labels, datasets: [{ label: 'My First Dataset', data: [65, 59, 80, 81, 56, 55, 40], fill: false, borderColor: 'rgb(75, 192, 192)', tension: 0.1 }] };

const labels = Utils.months({count: 7});  
const data = {  
 labels: labels,  
 datasets: [{  
 label: 'My First Dataset',  
 data: [65, 59, 80, 81, 56, 55, 40],  
 fill: false,  
 borderColor: 'rgb(75, 192, 192)',  
 tension: 0.1  
 }]  
};

## [**#**](#30j0zll) Dataset Properties

Namespaces:

* data.datasets[index] - options for this dataset only
* options.datasets.line - options for all line datasets
* options.elements.line - options for all [line elements](http://docs.google.com/docs/3.9.1/configuration/elements.html#line-configuration)
* options.elements.point - options for all [point elements](http://docs.google.com/docs/3.9.1/configuration/elements.html#point-configuration)
* options - options for the whole chart

The line chart allows a number of properties to be specified for each dataset. These are used to set display properties for a specific dataset. For example, the colour of a line is generally set this way.

| Name | Type | [Scriptable](http://docs.google.com/docs/3.9.1/general/options.html#scriptable-options) | [Indexable](http://docs.google.com/docs/3.9.1/general/options.html#indexable-options) | Default |
| --- | --- | --- | --- | --- |
| [backgroundColor](#1fob9te) | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | Yes | - | 'rgba(0, 0, 0, 0.1)' |
| [borderCapStyle](#1fob9te) | string | Yes | - | 'butt' |
| [borderColor](#1fob9te) | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | Yes | - | 'rgba(0, 0, 0, 0.1)' |
| [borderDash](#1fob9te) | number[] | Yes | - | [] |
| [borderDashOffset](#1fob9te) | number | Yes | - | 0.0 |
| [borderJoinStyle](#1fob9te) | 'round'|'bevel'|'miter' | Yes | - | 'miter' |
| [borderWidth](#1fob9te) | number | Yes | - | 3 |
| [clip](#3znysh7) | number|object|false | - | - | undefined |
| [cubicInterpolationMode](#2et92p0) | string | Yes | - | 'default' |
| [data](#tyjcwt) | object|object[]| number[]|string[] | - | - | **required** |
| [drawActiveElementsOnTop](#3znysh7) | boolean | Yes | Yes | true |
| [fill](#1fob9te) | boolean|string | Yes | - | false |
| [hoverBackgroundColor](#1fob9te) | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | Yes | - | undefined |
| [hoverBorderCapStyle](#1fob9te) | string | Yes | - | undefined |
| [hoverBorderColor](#1fob9te) | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | Yes | - | undefined |
| [hoverBorderDash](#1fob9te) | number[] | Yes | - | undefined |
| [hoverBorderDashOffset](#1fob9te) | number | Yes | - | undefined |
| [hoverBorderJoinStyle](#1fob9te) | 'round'|'bevel'|'miter' | Yes | - | undefined |
| [hoverBorderWidth](#1fob9te) | number | Yes | - | undefined |
| [indexAxis](#3znysh7) | string | - | - | 'x' |
| [label](#3znysh7) | string | - | - | '' |
| [order](#3znysh7) | number | - | - | 0 |
| [pointBackgroundColor](#3dy6vkm) | Color | Yes | Yes | 'rgba(0, 0, 0, 0.1)' |
| [pointBorderColor](#3dy6vkm) | Color | Yes | Yes | 'rgba(0, 0, 0, 0.1)' |
| [pointBorderWidth](#3dy6vkm) | number | Yes | Yes | 1 |
| [pointHitRadius](#3dy6vkm) | number | Yes | Yes | 1 |
| [pointHoverBackgroundColor](#1t3h5sf) | Color | Yes | Yes | undefined |
| [pointHoverBorderColor](#1t3h5sf) | Color | Yes | Yes | undefined |
| [pointHoverBorderWidth](#1t3h5sf) | number | Yes | Yes | 1 |
| [pointHoverRadius](#1t3h5sf) | number | Yes | Yes | 4 |
| [pointRadius](#3dy6vkm) | number | Yes | Yes | 3 |
| [pointRotation](#3dy6vkm) | number | Yes | Yes | 0 |
| [pointStyle](#3dy6vkm) | [pointStyle](http://docs.google.com/docs/3.9.1/configuration/elements.html#types) | Yes | Yes | 'circle' |
| [segment](#4d34og8) | object | - | - | undefined |
| [showLine](#1fob9te) | boolean | - | - | true |
| [spanGaps](#1fob9te) | boolean|number | - | - | undefined |
| [stack](#3znysh7) | string | - | - | 'line' |
| [stepped](#2s8eyo1) | boolean|string | - | - | false |
| [tension](#1fob9te) | number | - | - | 0 |
| [xAxisID](#3znysh7) | string | - | - | first x axis |
| [yAxisID](#3znysh7) | string | - | - | first y axis |

All these values, if undefined, fallback to the scopes described in [option resolution](http://docs.google.com/general/options)

### [**#**](#3znysh7) General

| Name | Description |
| --- | --- |
| clip | How to clip relative to chartArea. Positive value allows overflow, negative value clips that many pixels inside chartArea. 0 = clip at chartArea. Clipping can also be configured per side: clip: {left: 5, top: false, right: -2, bottom: 0} |
| drawActiveElementsOnTop | Draw the active points of a dataset over the other points of the dataset |
| indexAxis | The base axis of the dataset. 'x' for horizontal lines and 'y' for vertical lines. |
| label | The label for the dataset which appears in the legend and tooltips. |
| order | The drawing order of dataset. Also affects order for stacking, tooltip and legend. [more](http://docs.google.com/docs/3.9.1/charts/mixed.html#drawing-order) |
| stack | The ID of the group to which this dataset belongs to (when stacked, each group will be a separate stack). [more](#17dp8vu) |
| xAxisID | The ID of the x-axis to plot this dataset on. |
| yAxisID | The ID of the y-axis to plot this dataset on. |

### [**#**](#3dy6vkm) Point Styling

The style of each point can be controlled with the following properties:

| Name | Description |
| --- | --- |
| pointBackgroundColor | The fill color for points. |
| pointBorderColor | The border color for points. |
| pointBorderWidth | The width of the point border in pixels. |
| pointHitRadius | The pixel size of the non-displayed point that reacts to mouse events. |
| pointRadius | The radius of the point shape. If set to 0, the point is not rendered. |
| pointRotation | The rotation of the point in degrees. |
| pointStyle | Style of the point. [more...](http://docs.google.com/docs/3.9.1/configuration/elements.html#point-styles) |

All these values, if undefined, fallback first to the dataset options then to the associated [elements.point.\*](http://docs.google.com/docs/3.9.1/configuration/elements.html#point-configuration) options.

### [**#**](#1fob9te) Line Styling

The style of the line can be controlled with the following properties:

| Name | Description |
| --- | --- |
| backgroundColor | The line fill color. |
| borderCapStyle | Cap style of the line. See [MDN (opens new window)](https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D/lineCap). |
| borderColor | The line color. |
| borderDash | Length and spacing of dashes. See [MDN (opens new window)](https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D/setLineDash). |
| borderDashOffset | Offset for line dashes. See [MDN (opens new window)](https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D/lineDashOffset). |
| borderJoinStyle | Line joint style. See [MDN (opens new window)](https://developer.mozilla.org/en-US/docs/Web/API/CanvasRenderingContext2D/lineJoin). |
| borderWidth | The line width (in pixels). |
| fill | How to fill the area under the line. See [area charts](http://docs.google.com/docs/3.9.1/charts/area.html). |
| tension | Bezier curve tension of the line. Set to 0 to draw straightlines. This option is ignored if monotone cubic interpolation is used. |
| showLine | If false, the line is not drawn for this dataset. |
| spanGaps | If true, lines will be drawn between points with no or null data. If false, points with null data will create a break in the line. Can also be a number specifying the maximum gap length to span. The unit of the value depends on the scale used. |

If the value is undefined, the values fallback to the associated [elements.line.\*](http://docs.google.com/docs/3.9.1/configuration/elements.html#line-configuration) options.

### [**#**](#1t3h5sf) Interactions

The interaction with each point can be controlled with the following properties:

| Name | Description |
| --- | --- |
| pointHoverBackgroundColor | Point background color when hovered. |
| pointHoverBorderColor | Point border color when hovered. |
| pointHoverBorderWidth | Border width of point when hovered. |
| pointHoverRadius | The radius of the point when hovered. |

### [**#**](#2et92p0) cubicInterpolationMode

The following interpolation modes are supported.

* 'default'
* 'monotone'

The 'default' algorithm uses a custom weighted cubic interpolation, which produces pleasant curves for all types of datasets.

The 'monotone' algorithm is more suited to y = f(x) datasets: it preserves monotonicity (or piecewise monotonicity) of the dataset being interpolated, and ensures local extremums (if any) stay at input data points.

If left untouched (undefined), the global options.elements.line.cubicInterpolationMode property is used.

### [**#**](#4d34og8) Segment

Line segment styles can be overridden by scriptable options in the segment object. Currently all of the border\* and backgroundColor options are supported. The segment styles are resolved for each section of the line between each point. undefined fallbacks to main line styles.

TIP

To be able to style gaps, you need the [spanGaps](#1fob9te) option enabled.

Context for the scriptable segment contains the following properties:

* type: 'segment'
* p0: first point element
* p1: second point element
* p0DataIndex: index of first point in the data array
* p1DataIndex: index of second point in the data array
* datasetIndex: dataset index

[Example usage](http://docs.google.com/docs/3.9.1/samples/line/segments.html)

### [**#**](#2s8eyo1) Stepped

The following values are supported for stepped.

* false: No Step Interpolation (default)
* true: Step-before Interpolation (eq. 'before')
* 'before': Step-before Interpolation
* 'after': Step-after Interpolation
* 'middle': Step-middle Interpolation

If the stepped value is set to anything other than false, tension will be ignored.

## [**#**](#3rdcrjn) Default Options

It is common to want to apply a configuration setting to all created line charts. The global line chart settings are stored in Chart.overrides.line. Changing the global options only affects charts created after the change. Existing charts are not changed.

For example, to configure all line charts with spanGaps = true you would do:

Chart.overrides.line.spanGaps = true;

## [**#**](#tyjcwt) Data Structure

All of the supported [data structures](http://docs.google.com/docs/3.9.1/general/data-structures.html) can be used with line charts.

## [**#**](#17dp8vu) Stacked Area Chart

Line charts can be configured into stacked area charts by changing the settings on the y-axis to enable stacking. Stacked area charts can be used to show how one data trend is made up of a number of smaller pieces.

const stackedLine = new Chart(ctx, {  
 type: 'line',  
 data: data,  
 options: {  
 scales: {  
 y: {  
 stacked: true  
 }  
 }  
 }  
});

## [**#**](#26in1rg) Vertical Line Chart

A vertical line chart is a variation on the horizontal line chart. To achieve this you will have to set the indexAxis property in the options object to 'y'. The default for this property is 'x' and thus will show horizontal lines.

config setup

const config = { type: 'line', data: data, options: { indexAxis: 'y', scales: { x: { beginAtZero: true } } } };

const config = {  
 type: 'line',  
 data: data,  
 options: {  
 indexAxis: 'y',  
 scales: {  
 x: {  
 beginAtZero: true  
 }  
 }  
 }  
};

const labels = Utils.months({count: 7}); const data = { labels: labels, datasets: [{ axis: 'y', label: 'My First Dataset', data: [65, 59, 80, 81, 56, 55, 40], fill: false, backgroundColor: [ 'rgba(255, 99, 132, 0.2)', 'rgba(255, 159, 64, 0.2)', 'rgba(255, 205, 86, 0.2)', 'rgba(75, 192, 192, 0.2)', 'rgba(54, 162, 235, 0.2)', 'rgba(153, 102, 255, 0.2)', 'rgba(201, 203, 207, 0.2)' ], borderColor: [ 'rgb(255, 99, 132)', 'rgb(255, 159, 64)', 'rgb(255, 205, 86)', 'rgb(75, 192, 192)', 'rgb(54, 162, 235)', 'rgb(153, 102, 255)', 'rgb(201, 203, 207)' ], borderWidth: 1 }] };

const labels = Utils.months({count: 7});  
const data = {  
 labels: labels,  
 datasets: [{  
 axis: 'y',  
 label: 'My First Dataset',  
 data: [65, 59, 80, 81, 56, 55, 40],  
 fill: false,  
 backgroundColor: [  
 'rgba(255, 99, 132, 0.2)',  
 'rgba(255, 159, 64, 0.2)',  
 'rgba(255, 205, 86, 0.2)',  
 'rgba(75, 192, 192, 0.2)',  
 'rgba(54, 162, 235, 0.2)',  
 'rgba(153, 102, 255, 0.2)',  
 'rgba(201, 203, 207, 0.2)'  
 ],  
 borderColor: [  
 'rgb(255, 99, 132)',  
 'rgb(255, 159, 64)',  
 'rgb(255, 205, 86)',  
 'rgb(75, 192, 192)',  
 'rgb(54, 162, 235)',  
 'rgb(153, 102, 255)',  
 'rgb(201, 203, 207)'  
 ],  
 borderWidth: 1  
 }]  
};

### [**#**](#lnxbz9) Config Options

The configuration options for the vertical line chart are the same as for the [line chart](#35nkun2). However, any options specified on the x-axis in a line chart, are applied to the y-axis in a vertical line chart.

## [**#**](#1ksv4uv) Internal data format

{x, y}

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

←  [Doughnut and Pie Charts](http://docs.google.com/docs/3.9.1/charts/doughnut.html)   [Mixed Chart Types](http://docs.google.com/docs/3.9.1/charts/mixed.html)  →