 [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
* Axes
  + [Axes](http://docs.google.com/docs/3.9.1/axes/)
  + Cartesian
    - [Cartesian Axes](http://docs.google.com/docs/3.9.1/axes/cartesian/)
    - [Category Axis](http://docs.google.com/docs/3.9.1/axes/cartesian/category.html)
    - [Linear Axis](http://docs.google.com/docs/3.9.1/axes/cartesian/linear.html)
    - [Logarithmic Axis](http://docs.google.com/docs/3.9.1/axes/cartesian/logarithmic.html)
    - [Time Cartesian Axis](http://docs.google.com/docs/3.9.1/axes/cartesian/time.html)
    - [Time Series Axis](http://docs.google.com/docs/3.9.1/axes/cartesian/timeseries.html)
  + Radial
  + [Labeling Axes](http://docs.google.com/docs/3.9.1/axes/labelling.html)
  + [Styling](http://docs.google.com/docs/3.9.1/axes/styling.html)
* Developers

[**#**](#gjdgxs) Cartesian Axes

Axes that follow a cartesian grid are known as 'Cartesian Axes'. Cartesian axes are used for line, bar, and bubble charts. Four cartesian axes are included in Chart.js by default.

* [linear](http://docs.google.com/docs/3.9.1/axes/cartesian/linear.html)
* [logarithmic](http://docs.google.com/docs/3.9.1/axes/cartesian/logarithmic.html)
* [category](http://docs.google.com/docs/3.9.1/axes/cartesian/category.html)
* [time](http://docs.google.com/docs/3.9.1/axes/cartesian/time.html)
* [timeseries](http://docs.google.com/docs/3.9.1/axes/cartesian/timeseries.html)

## [**#**](#30j0zll) Visual Components

A cartesian axis is composed of visual components that can be individually configured. These components are:

* [border](#1fob9te)
* [grid lines](#3znysh7)
* [tick](#2et92p0)
* [tick mark](#2et92p0)
* [title](#tyjcwt)

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

The axis border is drawn at the edge of the axis, beside the chart area. In the image below, it is drawn in red.

config setup

const config = { type: 'line', data, options: { scales: { x: { grid: { borderColor: 'red' } } } } };

const config = {  
 type: 'line',  
 data,  
 options: {  
 scales: {  
 x: {  
 grid: {  
 borderColor: 'red'  
 }  
 }  
 }  
 }  
};

const labels = Utils.months({count: 7}); const data = { labels: labels, datasets: [{ label: 'My First dataset', backgroundColor: 'rgba(54, 162, 235, 0.5)', borderColor: 'rgb(54, 162, 235)', borderWidth: 1, data: [10, 20, 30, 40, 50, 0, 5], }] };

const labels = Utils.months({count: 7});  
const data = {  
 labels: labels,  
 datasets: [{  
 label: 'My First dataset',  
 backgroundColor: 'rgba(54, 162, 235, 0.5)',  
 borderColor: 'rgb(54, 162, 235)',  
 borderWidth: 1,  
 data: [10, 20, 30, 40, 50, 0, 5],  
 }]  
};

### [**#**](#3znysh7) Grid lines

The grid lines for an axis are drawn on the chart area. In the image below, they are red.

config setup

const config = { type: 'line', data, options: { scales: { x: { grid: { color: 'red', borderColor: 'grey', tickColor: 'grey' } } } } };

const config = {  
 type: 'line',  
 data,  
 options: {  
 scales: {  
 x: {  
 grid: {  
 color: 'red',  
 borderColor: 'grey',  
 tickColor: 'grey'  
 }  
 }  
 }  
 }  
};

const labels = Utils.months({count: 7}); const data = { labels: labels, datasets: [{ label: 'My First dataset', backgroundColor: 'rgba(54, 162, 235, 0.5)', borderColor: 'rgb(54, 162, 235)', borderWidth: 1, data: [10, 20, 30, 40, 50, 0, 5], }] };

const labels = Utils.months({count: 7});  
const data = {  
 labels: labels,  
 datasets: [{  
 label: 'My First dataset',  
 backgroundColor: 'rgba(54, 162, 235, 0.5)',  
 borderColor: 'rgb(54, 162, 235)',  
 borderWidth: 1,  
 data: [10, 20, 30, 40, 50, 0, 5],  
 }]  
};

### [**#**](#2et92p0) Ticks and Tick Marks

Ticks represent data values on the axis that appear as labels. The tick mark is the extension of the grid line from the axis border to the label. In this example, the tick mark is drawn in red while the tick label is drawn in blue.

config setup

const config = { type: 'line', data, options: { scales: { x: { grid: { tickColor: 'red' }, ticks: { color: 'blue', } } } } };

const config = {  
 type: 'line',  
 data,  
 options: {  
 scales: {  
 x: {  
 grid: {  
 tickColor: 'red'  
 },  
 ticks: {  
 color: 'blue',  
 }  
 }  
 }  
 }  
};

const labels = Utils.months({count: 7}); const data = { labels: labels, datasets: [{ label: 'My First dataset', backgroundColor: 'rgba(54, 162, 235, 0.5)', borderColor: 'rgb(54, 162, 235)', borderWidth: 1, data: [10, 20, 30, 40, 50, 0, 5], }] };

const labels = Utils.months({count: 7});  
const data = {  
 labels: labels,  
 datasets: [{  
 label: 'My First dataset',  
 backgroundColor: 'rgba(54, 162, 235, 0.5)',  
 borderColor: 'rgb(54, 162, 235)',  
 borderWidth: 1,  
 data: [10, 20, 30, 40, 50, 0, 5],  
 }]  
};

### [**#**](#tyjcwt) Title

The title component of the axis is used to label the data. In the example below, it is shown in red.

config setup

const config = { type: 'line', data, options: { scales: { x: { title: { color: 'red', display: true, text: 'Month' } } } } };

const config = {  
 type: 'line',  
 data,  
 options: {  
 scales: {  
 x: {  
 title: {  
 color: 'red',  
 display: true,  
 text: 'Month'  
 }  
 }  
 }  
 }  
};

const labels = Utils.months({count: 7}); const data = { labels: labels, datasets: [{ label: 'My First dataset', backgroundColor: 'rgba(54, 162, 235, 0.5)', borderColor: 'rgb(54, 162, 235)', borderWidth: 1, data: [10, 20, 30, 40, 50, 0, 5], }] };

const labels = Utils.months({count: 7});  
const data = {  
 labels: labels,  
 datasets: [{  
 label: 'My First dataset',  
 backgroundColor: 'rgba(54, 162, 235, 0.5)',  
 borderColor: 'rgb(54, 162, 235)',  
 borderWidth: 1,  
 data: [10, 20, 30, 40, 50, 0, 5],  
 }]  
};

## [**#**](#3dy6vkm) Common Configuration

Note

These are only the common options supported by all cartesian axes. Please see the specific axis documentation for all the available options for that axis.

### [**#**](#1t3h5sf) Common options to all cartesian axes

Namespace: options.scales[scaleId]

| Name | Type | Default | Description |
| --- | --- | --- | --- |
| bounds | string | 'ticks' | Determines the scale bounds. [more...](http://docs.google.com/docs/3.9.1/axes/cartesian/#scale-bounds) |
| position | string | object |  | Position of the axis. [more...](http://docs.google.com/docs/3.9.1/axes/cartesian/#axis-position) |
| stack | string |  | Stack group. Axes at the same position with same stack are stacked. |
| stackWeight | number | 1 | Weight of the scale in stack group. Used to determine the amount of allocated space for the scale within the group. |
| axis | string |  | Which type of axis this is. Possible values are: 'x', 'y'. If not set, this is inferred from the first character of the ID which should be 'x' or 'y'. |
| offset | boolean | false | If true, extra space is added to the both edges and the axis is scaled to fit into the chart area. This is set to true for a bar chart by default. |
| title | object |  | Scale title configuration. [more...](http://docs.google.com/docs/3.9.1/axes/labelling.html#scale-title-configuration) |

### [**#**](#4d34og8) Common options to all axes

Namespace: options.scales[scaleId]

| Name | Type | Default | Description |
| --- | --- | --- | --- |
| type | string |  | Type of scale being employed. Custom scales can be created and registered with a string key. This allows changing the type of an axis for a chart. |
| alignToPixels | boolean | false | Align pixel values to device pixels. |
| backgroundColor | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) |  | Background color of the scale area. |
| display | boolean|string | true | Controls the axis global visibility (visible when true, hidden when false). When display: 'auto', the axis is visible only if at least one associated dataset is visible. |
| grid | object |  | Grid line configuration. [more...](http://docs.google.com/docs/3.9.1/axes/styling.html#grid-line-configuration) |
| min | number |  | User defined minimum number for the scale, overrides minimum value from data. [more...](http://docs.google.com/docs/3.9.1/axes/#axis-range-settings) |
| max | number |  | User defined maximum number for the scale, overrides maximum value from data. [more...](http://docs.google.com/docs/3.9.1/axes/#axis-range-settings) |
| reverse | boolean | false | Reverse the scale. |
| stacked | boolean|string | false | Should the data be stacked. [more...](http://docs.google.com/docs/3.9.1/axes/#stacking) |
| suggestedMax | number |  | Adjustment used when calculating the maximum data value. [more...](http://docs.google.com/docs/3.9.1/axes/#axis-range-settings) |
| suggestedMin | number |  | Adjustment used when calculating the minimum data value. [more...](http://docs.google.com/docs/3.9.1/axes/#axis-range-settings) |
| ticks | object |  | Tick configuration. [more...](http://docs.google.com/docs/3.9.1/axes/#tick-configuration) |
| weight | number | 0 | The weight used to sort the axis. Higher weights are further away from the chart area. |

### [**#**](#2s8eyo1) Axis Position

An axis can either be positioned at the edge of the chart, at the center of the chart area, or dynamically with respect to a data value.

To position the axis at the edge of the chart, set the position option to one of: 'top', 'left', 'bottom', 'right'. To position the axis at the center of the chart area, set the position option to 'center'. In this mode, either the axis option must be specified or the axis ID has to start with the letter 'x' or 'y'. This is so chart.js knows what kind of axis (horizontal or vertical) it is. To position the axis with respect to a data value, set the position option to an object such as:

{  
 x: -20  
}

This will position the axis at a value of -20 on the axis with ID "x". For cartesian axes, only 1 axis may be specified.

### [**#**](#17dp8vu) Scale Bounds

The bounds property controls the scale boundary strategy (bypassed by min/max options).

* 'data': makes sure data are fully visible, labels outside are removed
* 'ticks': makes sure ticks are fully visible, data outside are truncated

### [**#**](#3rdcrjn) Tick Configuration

Note

These are only the common tick options supported by all cartesian axes. Please see specific axis documentation for all of the available options for that axis.

### [**#**](#26in1rg) Common tick options to all cartesian axes

Namespace: options.scales[scaleId].ticks

| Name | Type | Default | Description |
| --- | --- | --- | --- |
| align | string | 'center' | The tick alignment along the axis. Can be 'start', 'center', 'end', or 'inner'. inner alignment means align start for first tick and end for the last tick of horizontal axis |
| crossAlign | string | 'near' | The tick alignment perpendicular to the axis. Can be 'near', 'center', or 'far'. See [Tick Alignment](http://docs.google.com/docs/3.9.1/axes/cartesian/#tick-alignment) |
| sampleSize | number | ticks.length | The number of ticks to examine when deciding how many labels will fit. Setting a smaller value will be faster, but may be less accurate when there is large variability in label length. |
| autoSkip | boolean | true | If true, automatically calculates how many labels can be shown and hides labels accordingly. Labels will be rotated up to maxRotation before skipping any. Turn autoSkip off to show all labels no matter what. |
| autoSkipPadding | number | 3 | Padding between the ticks on the horizontal axis when autoSkip is enabled. |
| includeBounds | boolean | true | Should the defined min and max values be presented as ticks even if they are not "nice". |
| labelOffset | number | 0 | Distance in pixels to offset the label from the centre point of the tick (in the x-direction for the x-axis, and the y-direction for the y-axis). *Note: this can cause labels at the edges to be cropped by the edge of the canvas* |
| maxRotation | number | 50 | Maximum rotation for tick labels when rotating to condense labels. Note: Rotation doesn't occur until necessary. *Note: Only applicable to horizontal scales.* |
| minRotation | number | 0 | Minimum rotation for tick labels. *Note: Only applicable to horizontal scales.* |
| mirror | boolean | false | Flips tick labels around axis, displaying the labels inside the chart instead of outside. *Note: Only applicable to vertical scales.* |
| padding | number | 0 | Padding between the tick label and the axis. When set on a vertical axis, this applies in the horizontal (X) direction. When set on a horizontal axis, this applies in the vertical (Y) direction. |
| maxTicksLimit | number | 11 | Maximum number of ticks and gridlines to show. |

### [**#**](#lnxbz9) Common tick options to all axes

Namespace: options.scales[scaleId].ticks

| Name | Type | Scriptable | Default | Description |
| --- | --- | --- | --- | --- |
| backdropColor | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | Yes | 'rgba(255, 255, 255, 0.75)' | Color of label backdrops. |
| backdropPadding | [Padding](http://docs.google.com/docs/3.9.1/general/padding.html) |  | 2 | Padding of label backdrop. |
| callback | function |  |  | Returns the string representation of the tick value as it should be displayed on the chart. See [callback](http://docs.google.com/docs/3.9.1/axes/labelling.html#creating-custom-tick-formats). |
| display | boolean |  | true | If true, show tick labels. |
| color | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | Yes | Chart.defaults.color | Color of ticks. |
| font | Font | Yes | Chart.defaults.font | See [Fonts](http://docs.google.com/docs/3.9.1/general/fonts.html) |
| major | object |  | {} | [Major ticks configuration](http://docs.google.com/docs/3.9.1/axes/styling.html#major-tick-configuration). |
| padding | number |  | 3 | Sets the offset of the tick labels from the axis |
| showLabelBackdrop | boolean | Yes | true for radial scale, false otherwise | If true, draw a background behind the tick labels. |
| textStrokeColor | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | Yes | `` | The color of the stroke around the text. |
| textStrokeWidth | number | Yes | 0 | Stroke width around the text. |
| z | number |  | 0 | z-index of tick layer. Useful when ticks are drawn on chart area. Values <= 0 are drawn under datasets, > 0 on top. |

### [**#**](#35nkun2) Tick Alignment

The alignment of ticks is primarily controlled using two settings on the tick configuration object: align and crossAlign. The align setting configures how labels align with the tick mark along the axis direction (i.e. horizontal for a horizontal axis and vertical for a vertical axis). The crossAlign setting configures how labels align with the tick mark in the perpendicular direction (i.e. vertical for a horizontal axis and horizontal for a vertical axis). In the example below, the crossAlign setting is used to left align the labels on the Y axis.

config setup

const config = { type: 'bar', data, options: { indexAxis: 'y', scales: { y: { ticks: { crossAlign: 'far', } } } } };

const config = {  
 type: 'bar',  
 data,  
 options: {  
 indexAxis: 'y',  
 scales: {  
 y: {  
 ticks: {  
 crossAlign: 'far',  
 }  
 }  
 }  
 }  
};

const labels = Utils.months({count: 7}); const data = { labels: labels, datasets: [{ label: 'My First dataset', 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, data: [65, 59, 80, 81, 56, 55, 40], }] };

const labels = Utils.months({count: 7});  
const data = {  
 labels: labels,  
 datasets: [{  
 label: 'My First dataset',  
 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,  
 data: [65, 59, 80, 81, 56, 55, 40],  
 }]  
};

Note

The crossAlign setting is only effective when these preconditions are met:

* tick rotation is 0
* axis position is 'top', 'left', 'bottom' or 'right'

### [**#**](#1ksv4uv) Axis ID

The properties dataset.xAxisID or dataset.yAxisID have to match to scales property. This is especially needed if multi-axes charts are used.

const myChart = new Chart(ctx, {  
 type: 'line',  
 data: {  
 datasets: [{  
 // This dataset appears on the first axis  
 yAxisID: 'first-y-axis'  
 }, {  
 // This dataset appears on the second axis  
 yAxisID: 'second-y-axis'  
 }]  
 },  
 options: {  
 scales: {  
 'first-y-axis': {  
 type: 'linear'  
 },  
 'second-y-axis': {  
 type: 'linear'  
 }  
 }  
 }  
});

## [**#**](#44sinio) Creating Multiple Axes

With cartesian axes, it is possible to create multiple X and Y axes. To do so, you can add multiple configuration objects to the xAxes and yAxes properties. When adding new axes, it is important to ensure that you specify the type of the new axes as default types are **not** used in this case.

In the example below, we are creating two Y axes. We then use the yAxisID property to map the datasets to their correct axes.

const myChart = new Chart(ctx, {  
 type: 'line',  
 data: {  
 datasets: [{  
 data: [20, 50, 100, 75, 25, 0],  
 label: 'Left dataset',  
 // This binds the dataset to the left y axis  
 yAxisID: 'left-y-axis'  
 }, {  
 data: [0.1, 0.5, 1.0, 2.0, 1.5, 0],  
 label: 'Right dataset',  
 // This binds the dataset to the right y axis  
 yAxisID: 'right-y-axis'  
 }],  
 labels: ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun']  
 },  
 options: {  
 scales: {  
 'left-y-axis': {  
 type: 'linear',  
 position: 'left'  
 },  
 'right-y-axis': {  
 type: 'linear',  
 position: 'right'  
 }  
 }  
 }  
});

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

←  [Axes](http://docs.google.com/docs/3.9.1/axes/)   [Category Axis](http://docs.google.com/docs/3.9.1/axes/cartesian/category.html)  →