 [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
  + [Accessibility](http://docs.google.com/docs/3.9.1/general/accessibility.html)
  + [Colors](http://docs.google.com/docs/3.9.1/general/colors.html)
  + [Data structures](http://docs.google.com/docs/3.9.1/general/data-structures.html)
  + [Fonts](http://docs.google.com/docs/3.9.1/general/fonts.html)
  + [Options](http://docs.google.com/docs/3.9.1/general/options.html)
  + [Padding](http://docs.google.com/docs/3.9.1/general/padding.html)
  + [Performance](http://docs.google.com/docs/3.9.1/general/performance.html)
* Configuration
* Chart Types
* Axes
* Developers

[**#**](#gjdgxs) Data structures

The data property of a dataset can be passed in various formats. By default, that data is parsed using the associated chart type and scales.

If the labels property of the main data property is used, it has to contain the same amount of elements as the dataset with the most values. These labels are used to label the index axis (default x axes). The values for the labels have to be provided in an array. The provided labels can be of the type string or number to be rendered correctly. In case you want multiline labels you can provide an array with each line as one entry in the array.

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

type: 'bar',  
data: {  
 datasets: [{  
 data: [20, 10],  
 }],  
 labels: ['a', 'b']  
}

When the data is an array of numbers, values from labels array at the same index are used for the index axis (x for vertical, y for horizontal charts).

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

type: 'line',  
data: {  
 datasets: [{  
 data: [{x: 10, y: 20}, {x: 15, y: null}, {x: 20, y: 10}]  
 }]  
}

type: 'line',  
data: {  
 datasets: [{  
 data: [{x:'2016-12-25', y:20}, {x:'2016-12-26', y:10}]  
 }]  
}

type: 'bar',  
data: {  
 datasets: [{  
 data: [{x:'Sales', y:20}, {x:'Revenue', y:10}]  
 }]  
}

This is also the internal format used for parsed data. In this mode, parsing can be disabled by specifying parsing: false at chart options or dataset. If parsing is disabled, data must be sorted and in the formats the associated chart type and scales use internally.

The values provided must be parsable by the associated scales or in the internal format of the associated scales. A common mistake would be to provide integers for the category scale, which uses integers as an internal format, where each integer represents an index in the labels array. null can be used for skipped values.

## [**#**](#3znysh7) Object[] using custom properties

type: 'bar',  
data: {  
 datasets: [{  
 data: [{id: 'Sales', nested: {value: 1500}}, {id: 'Purchases', nested: {value: 500}}]  
 }]  
},  
options: {  
 parsing: {  
 xAxisKey: 'id',  
 yAxisKey: 'nested.value'  
 }  
}

When using the pie/doughnut, radar or polarArea chart type, the parsing object should have a key item that points to the value to look at. In this example, the doughnut chart will show two items with values 1500 and 500.

type: 'doughnut',  
data: {  
 datasets: [{  
 data: [{id: 'Sales', nested: {value: 1500}}, {id: 'Purchases', nested: {value: 500}}]  
 }]  
},  
options: {  
 parsing: {  
 key: 'nested.value'  
 }  
}

If the key contains a dot, it needs to be escaped with a double slash:

type: 'line',  
data: {  
 datasets: [{  
 data: [{ 'data.key': 'one', 'data.value': 20 }, { 'data.key': 'two', 'data.value': 30 }]  
 }]  
},  
options: {  
 parsing: {  
 xAxisKey: 'data\\.key',  
 yAxisKey: 'data\\.value'  
 }  
}

WARNING

When using object notation in a radar chart you still need a labels array with labels for the chart to show correctly.

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

type: 'pie',  
data: {  
 datasets: [{  
 data: {  
 January: 10,  
 February: 20  
 }  
 }]  
}

In this mode, property name is used for index scale and value for value scale. For vertical charts, index scale is x and value scale is y.

## [**#**](#tyjcwt) Dataset Configuration

| Name | Type | Description |
| --- | --- | --- |
| label | string | The label for the dataset which appears in the legend and tooltips. |
| clip | number|object | 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} |
| order | number | The drawing order of dataset. Also affects order for stacking, tooltip and legend. |
| stack | string | The ID of the group to which this dataset belongs to (when stacked, each group will be a separate stack). Defaults to dataset type. |
| parsing | boolean|object | How to parse the dataset. The parsing can be disabled by specifying parsing: false at chart options or dataset. If parsing is disabled, data must be sorted and in the formats the associated chart type and scales use internally. |
| hidden | boolean | Configure the visibility of the dataset. Using hidden: true will hide the dataset from being rendered in the Chart. |

### [**#**](#3dy6vkm) parsing

const data = [{x: 'Jan', net: 100, cogs: 50, gm: 50}, {x: 'Feb', net: 120, cogs: 55, gm: 75}];  
const cfg = {  
 type: 'bar',  
 data: {  
 labels: ['Jan', 'Feb'],  
 datasets: [{  
 label: 'Net sales',  
 data: data,  
 parsing: {  
 yAxisKey: 'net'  
 }  
 }, {  
 label: 'Cost of goods sold',  
 data: data,  
 parsing: {  
 yAxisKey: 'cogs'  
 }  
 }, {  
 label: 'Gross margin',  
 data: data,  
 parsing: {  
 yAxisKey: 'gm'  
 }  
 }]  
 },  
};

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

←  [Colors](http://docs.google.com/docs/3.9.1/general/colors.html)   [Fonts](http://docs.google.com/docs/3.9.1/general/fonts.html)  →