 [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
  + [Configuration](http://docs.google.com/docs/3.9.1/configuration/)
  + [Animations](http://docs.google.com/docs/3.9.1/configuration/animations.html)
  + [Canvas background](http://docs.google.com/docs/3.9.1/configuration/canvas-background.html)
  + [Data Decimation](http://docs.google.com/docs/3.9.1/configuration/decimation.html)
  + [Device Pixel Ratio](http://docs.google.com/docs/3.9.1/configuration/device-pixel-ratio.html)
  + [Elements](http://docs.google.com/docs/3.9.1/configuration/elements.html)
  + [Interactions](http://docs.google.com/docs/3.9.1/configuration/interactions.html)
  + [Layout](http://docs.google.com/docs/3.9.1/configuration/layout.html)
  + [Legend](http://docs.google.com/docs/3.9.1/configuration/legend.html)
  + [Locale](http://docs.google.com/docs/3.9.1/configuration/locale.html)
  + [Responsive Charts](http://docs.google.com/docs/3.9.1/configuration/responsive.html)
  + [Subtitle](http://docs.google.com/docs/3.9.1/configuration/subtitle.html)
  + [Title](http://docs.google.com/docs/3.9.1/configuration/title.html)
  + [Tooltip](http://docs.google.com/docs/3.9.1/configuration/tooltip.html)
* Chart Types
* Axes
* Developers

[**#**](#gjdgxs) Tooltip

## [**#**](#30j0zll) Tooltip Configuration

Namespace: options.plugins.tooltip, the global options for the chart tooltips is defined in Chart.defaults.plugins.tooltip.

WARNING

The bubble, doughnut, pie, polar area, and scatter charts override the tooltip defaults. To change the overrides for those chart types, the options are defined in Chart.overrides[type].plugins.tooltip.

| Name | Type | Default | Description |
| --- | --- | --- | --- |
| enabled | boolean | true | Are on-canvas tooltips enabled? |
| external | function | null | See [external tooltip](#1fob9te) section. |
| mode | string | interaction.mode | Sets which elements appear in the tooltip. [more...](http://docs.google.com/docs/3.9.1/configuration/interactions.html#modes). |
| intersect | boolean | interaction.intersect | If true, the tooltip mode applies only when the mouse position intersects with an element. If false, the mode will be applied at all times. |
| position | string | 'average' | The mode for positioning the tooltip. [more...](#3znysh7) |
| callbacks | object |  | See the [callbacks section](#2et92p0). |
| itemSort | function |  | Sort tooltip items. [more...](#tyjcwt) |
| filter | function |  | Filter tooltip items. [more...](#3dy6vkm) |
| backgroundColor | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | 'rgba(0, 0, 0, 0.8)' | Background color of the tooltip. |
| titleColor | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | '#fff' | Color of title text. |
| titleFont | Font | {weight: 'bold'} | See [Fonts](http://docs.google.com/docs/3.9.1/general/fonts.html). |
| titleAlign | string | 'left' | Horizontal alignment of the title text lines. [more...](#1t3h5sf) |
| titleSpacing | number | 2 | Spacing to add to top and bottom of each title line. |
| titleMarginBottom | number | 6 | Margin to add on bottom of title section. |
| bodyColor | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | '#fff' | Color of body text. |
| bodyFont | Font | {} | See [Fonts](http://docs.google.com/docs/3.9.1/general/fonts.html). |
| bodyAlign | string | 'left' | Horizontal alignment of the body text lines. [more...](#1t3h5sf) |
| bodySpacing | number | 2 | Spacing to add to top and bottom of each tooltip item. |
| footerColor | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | '#fff' | Color of footer text. |
| footerFont | Font | {weight: 'bold'} | See [Fonts](http://docs.google.com/docs/3.9.1/general/fonts.html). |
| footerAlign | string | 'left' | Horizontal alignment of the footer text lines. [more...](#1t3h5sf) |
| footerSpacing | number | 2 | Spacing to add to top and bottom of each footer line. |
| footerMarginTop | number | 6 | Margin to add before drawing the footer. |
| padding | [Padding](http://docs.google.com/docs/3.9.1/general/padding.html) | 6 | Padding inside the tooltip. |
| caretPadding | number | 2 | Extra distance to move the end of the tooltip arrow away from the tooltip point. |
| caretSize | number | 5 | Size, in px, of the tooltip arrow. |
| cornerRadius | number|object | 6 | Radius of tooltip corner curves. |
| multiKeyBackground | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | '#fff' | Color to draw behind the colored boxes when multiple items are in the tooltip. |
| displayColors | boolean | true | If true, color boxes are shown in the tooltip. |
| boxWidth | number | bodyFont.size | Width of the color box if displayColors is true. |
| boxHeight | number | bodyFont.size | Height of the color box if displayColors is true. |
| boxPadding | number | 1 | Padding between the color box and the text. |
| usePointStyle | boolean | false | Use the corresponding point style (from dataset options) instead of color boxes, ex: star, triangle etc. (size is based on the minimum value between boxWidth and boxHeight). |
| borderColor | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | 'rgba(0, 0, 0, 0)' | Color of the border. |
| borderWidth | number | 0 | Size of the border. |
| rtl | boolean |  | true for rendering the tooltip from right to left. |
| textDirection | string | canvas' default | This will force the text direction 'rtl' or 'ltr on the canvas for rendering the tooltips, regardless of the css specified on the canvas |
| xAlign | string | undefined | Position of the tooltip caret in the X direction. [more](#4d34og8) |
| yAlign | string | undefined | Position of the tooltip caret in the Y direction. [more](#4d34og8) |

### [**#**](#3znysh7) Position Modes

Possible modes are:

* 'average'
* 'nearest'

'average' mode will place the tooltip at the average position of the items displayed in the tooltip. 'nearest' will place the tooltip at the position of the element closest to the event position.

You can also define [custom position modes](#2s8eyo1).

### [**#**](#4d34og8) Tooltip Alignment

The xAlign and yAlign options define the position of the tooltip caret. If these parameters are unset, the optimal caret position is determined.

The following values for the xAlign setting are supported.

* 'left'
* 'center'
* 'right'

The following values for the yAlign setting are supported.

* 'top'
* 'center'
* 'bottom'

### [**#**](#1t3h5sf) Text Alignment

The titleAlign, bodyAlign and footerAlign options define the horizontal position of the text lines with respect to the tooltip box. The following values are supported.

* 'left' (default)
* 'right'
* 'center'

These options are only applied to text lines. Color boxes are always aligned to the left edge.

### [**#**](#tyjcwt) Sort Callback

Allows sorting of [tooltip items](#17dp8vu). Must implement at minimum a function that can be passed to [Array.prototype.sort (opens new window)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort). This function can also accept a third parameter that is the data object passed to the chart.

### [**#**](#3dy6vkm) Filter Callback

Allows filtering of [tooltip items](#17dp8vu). Must implement at minimum a function that can be passed to [Array.prototype.filter (opens new window)](https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Array/filter). This function can also accept a fourth parameter that is the data object passed to the chart.

## [**#**](#2et92p0) Tooltip Callbacks

Namespace: options.plugins.tooltip.callbacks, the tooltip has the following callbacks for providing text. For all functions, this will be the tooltip object created from the Tooltip constructor.

Namespace: data.datasets[].tooltip.callbacks, items marked with Yes in the column Dataset override can be overridden per dataset.

A [tooltip item context](#17dp8vu) is generated for each item that appears in the tooltip. This is the primary model that the callback methods interact with. For functions that return text, arrays of strings are treated as multiple lines of text.

| Name | Arguments | Return Type | Dataset override | Description |
| --- | --- | --- | --- | --- |
| beforeTitle | TooltipItem[] | string | string[] |  | Returns the text to render before the title. |
| title | TooltipItem[] | string | string[] |  | Returns text to render as the title of the tooltip. |
| afterTitle | TooltipItem[] | string | string[] |  | Returns text to render after the title. |
| beforeBody | TooltipItem[] | string | string[] |  | Returns text to render before the body section. |
| beforeLabel | TooltipItem | string | string[] | Yes | Returns text to render before an individual label. This will be called for each item in the tooltip. |
| label | TooltipItem | string | string[] | Yes | Returns text to render for an individual item in the tooltip. [more...](#3rdcrjn) |
| labelColor | TooltipItem | object | Yes | Returns the colors to render for the tooltip item. [more...](#26in1rg) |
| labelTextColor | TooltipItem | Color | Yes | Returns the colors for the text of the label for the tooltip item. |
| labelPointStyle | TooltipItem | object | Yes | Returns the point style to use instead of color boxes if usePointStyle is true (object with values pointStyle and rotation). Default implementation uses the point style from the dataset points. [more...](#lnxbz9) |
| afterLabel | TooltipItem | string | string[] | Yes | Returns text to render after an individual label. |
| afterBody | TooltipItem[] | string | string[] |  | Returns text to render after the body section. |
| beforeFooter | TooltipItem[] | string | string[] |  | Returns text to render before the footer section. |
| footer | TooltipItem[] | string | string[] |  | Returns text to render as the footer of the tooltip. |
| afterFooter | TooltipItem[] | string | string[] |  | Text to render after the footer section. |

### [**#**](#3rdcrjn) Label Callback

The label callback can change the text that displays for a given data point. A common example to show a unit. The example below puts a '$' before every row.

const chart = new Chart(ctx, {  
 type: 'line',  
 data: data,  
 options: {  
 plugins: {  
 tooltip: {  
 callbacks: {  
 label: function(context) {  
 let label = context.dataset.label || '';  
 if (label) {  
 label += ': ';  
 }  
 if (context.parsed.y !== null) {  
 label += new Intl.NumberFormat('en-US', { style: 'currency', currency: 'USD' }).format(context.parsed.y);  
 }  
 return label;  
 }  
 }  
 }  
 }  
 }  
});

### [**#**](#26in1rg) Label Color Callback

For example, to return a red box with a blue dashed border that has a border radius for each item in the tooltip you could do:

const chart = new Chart(ctx, {  
 type: 'line',  
 data: data,  
 options: {  
 plugins: {  
 tooltip: {  
 callbacks: {  
 labelColor: function(context) {  
 return {  
 borderColor: 'rgb(0, 0, 255)',  
 backgroundColor: 'rgb(255, 0, 0)',  
 borderWidth: 2,  
 borderDash: [2, 2],  
 borderRadius: 2,  
 };  
 },  
 labelTextColor: function(context) {  
 return '#543453';  
 }  
 }  
 }  
 }  
 }  
});

### [**#**](#lnxbz9) Label Point Style Callback

For example, to draw triangles instead of the regular color box for each item in the tooltip you could do:

const chart = new Chart(ctx, {  
 type: 'line',  
 data: data,  
 options: {  
 plugins: {  
 tooltip: {  
 usePointStyle: true,  
 callbacks: {  
 labelPointStyle: function(context) {  
 return {  
 pointStyle: 'triangle',  
 rotation: 0  
 };  
 }  
 }  
 }  
 }  
 }  
});

### [**#**](#17dp8vu) Tooltip Item Context

The tooltip items passed to the tooltip callbacks implement the following interface.

{  
 // The chart the tooltip is being shown on  
 chart: Chart  
 // Label for the tooltip  
 label: string,  
 // Parsed data values for the given `dataIndex` and `datasetIndex`  
 parsed: object,  
 // Raw data values for the given `dataIndex` and `datasetIndex`  
 raw: object,  
 // Formatted value for the tooltip  
 formattedValue: string,  
 // The dataset the item comes from  
 dataset: object  
 // Index of the dataset the item comes from  
 datasetIndex: number,  
 // Index of this data item in the dataset  
 dataIndex: number,  
 // The chart element (point, arc, bar, etc.) for this tooltip item  
 element: Element,  
}

## [**#**](#1fob9te) External (Custom) Tooltips

External tooltips allow you to hook into the tooltip rendering process so that you can render the tooltip in your own custom way. Generally this is used to create an HTML tooltip instead of an on-canvas tooltip. The external option takes a function which is passed a context parameter containing the chart and tooltip. You can enable external tooltips in the global or chart configuration like so:

const myPieChart = new Chart(ctx, {  
 type: 'pie',  
 data: data,  
 options: {  
 plugins: {  
 tooltip: {  
 // Disable the on-canvas tooltip  
 enabled: false,  
 external: function(context) {  
 // Tooltip Element  
 let tooltipEl = document.getElementById('chartjs-tooltip');  
 // Create element on first render  
 if (!tooltipEl) {  
 tooltipEl = document.createElement('div');  
 tooltipEl.id = 'chartjs-tooltip';  
 tooltipEl.innerHTML = '<table></table>';  
 document.body.appendChild(tooltipEl);  
 }  
 // Hide if no tooltip  
 const tooltipModel = context.tooltip;  
 if (tooltipModel.opacity === 0) {  
 tooltipEl.style.opacity = 0;  
 return;  
 }  
 // Set caret Position  
 tooltipEl.classList.remove('above', 'below', 'no-transform');  
 if (tooltipModel.yAlign) {  
 tooltipEl.classList.add(tooltipModel.yAlign);  
 } else {  
 tooltipEl.classList.add('no-transform');  
 }  
 function getBody(bodyItem) {  
 return bodyItem.lines;  
 }  
 // Set Text  
 if (tooltipModel.body) {  
 const titleLines = tooltipModel.title || [];  
 const bodyLines = tooltipModel.body.map(getBody);  
 let innerHtml = '<thead>';  
 titleLines.forEach(function(title) {  
 innerHtml += '<tr><th>' + title + '</th></tr>';  
 });  
 innerHtml += '</thead><tbody>';  
 bodyLines.forEach(function(body, i) {  
 const colors = tooltipModel.labelColors[i];  
 let style = 'background:' + colors.backgroundColor;  
 style += '; border-color:' + colors.borderColor;  
 style += '; border-width: 2px';  
 const span = '<span style="' + style + '"></span>';  
 innerHtml += '<tr><td>' + span + body + '</td></tr>';  
 });  
 innerHtml += '</tbody>';  
 let tableRoot = tooltipEl.querySelector('table');  
 tableRoot.innerHTML = innerHtml;  
 }  
 const position = context.chart.canvas.getBoundingClientRect();  
 const bodyFont = Chart.helpers.toFont(tooltipModel.options.bodyFont);  
 // Display, position, and set styles for font  
 tooltipEl.style.opacity = 1;  
 tooltipEl.style.position = 'absolute';  
 tooltipEl.style.left = position.left + window.pageXOffset + tooltipModel.caretX + 'px';  
 tooltipEl.style.top = position.top + window.pageYOffset + tooltipModel.caretY + 'px';  
 tooltipEl.style.font = bodyFont.string;  
 tooltipEl.style.padding = tooltipModel.padding + 'px ' + tooltipModel.padding + 'px';  
 tooltipEl.style.pointerEvents = 'none';  
 }  
 }  
 }  
 }  
});

See [samples](http://docs.google.com/docs/3.9.1/samples/tooltip/html.html) for examples on how to get started with external tooltips.

## [**#**](#35nkun2) Tooltip Model

The tooltip model contains parameters that can be used to render the tooltip.

{  
 chart: Chart,  
 // The items that we are rendering in the tooltip. See Tooltip Item Interface section  
 dataPoints: TooltipItem[],  
 // Positioning  
 xAlign: string,  
 yAlign: string,  
 // X and Y properties are the top left of the tooltip  
 x: number,  
 y: number,  
 width: number,  
 height: number,  
 // Where the tooltip points to  
 caretX: number,  
 caretY: number,  
 // Body  
 // The body lines that need to be rendered  
 // Each object contains 3 parameters  
 // before: string[] // lines of text before the line with the color square  
 // lines: string[], // lines of text to render as the main item with color square  
 // after: string[], // lines of text to render after the main lines  
 body: object[],  
 // lines of text that appear after the title but before the body  
 beforeBody: string[],  
 // line of text that appear after the body and before the footer  
 afterBody: string[],  
 // Title  
 // lines of text that form the title  
 title: string[],  
 // Footer  
 // lines of text that form the footer  
 footer: string[],  
 // colors to render for each item in body[]. This is the color of the squares in the tooltip  
 labelColors: Color[],  
 labelTextColors: Color[],  
 // 0 opacity is a hidden tooltip  
 opacity: number,  
 // tooltip options  
 options: Object  
}

## [**#**](#2s8eyo1) Custom Position Modes

New modes can be defined by adding functions to the Chart.Tooltip.positioners map.

Example:

import { Tooltip } from 'chart.js';  
/\*\*  
 \* Custom positioner  
 \* @function Tooltip.positioners.myCustomPositioner  
 \* @param elements {Chart.Element[]} the tooltip elements  
 \* @param eventPosition {Point} the position of the event in canvas coordinates  
 \* @returns {TooltipPosition} the tooltip position  
 \*/  
Tooltip.positioners.myCustomPositioner = function(elements, eventPosition) {  
 // A reference to the tooltip model  
 const tooltip = this;  
 /\* ... \*/  
 return {  
 x: 0,  
 y: 0  
 // You may also include xAlign and yAlign to override those tooltip options.  
 };  
};  
// Then, to use it...  
new Chart.js(ctx, {  
 data,  
 options: {  
 plugins: {  
 tooltip: {  
 position: 'myCustomPositioner'  
 }  
 }  
 }  
})

See [samples](http://docs.google.com/docs/3.9.1/samples/tooltip/position.html) for a more detailed example.

If you're using TypeScript, you'll also need to register the new mode:

declare module 'chart.js' {  
 interface TooltipPositionerMap {  
 myCustomPositioner: TooltipPositionerFunction<ChartType>;  
 }  
}

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

←  [Title](http://docs.google.com/docs/3.9.1/configuration/title.html)   [Area Chart](http://docs.google.com/docs/3.9.1/charts/area.html)  →