 [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) Options

## [**#**](#30j0zll) Option resolution

Options are resolved from top to bottom, using a context dependent route.

### [**#**](#1fob9te) Chart level options

* options
* overrides[config.type]
* defaults

### [**#**](#3znysh7) Dataset level options

dataset.type defaults to config.type, if not specified.

* dataset
* options.datasets[dataset.type]
* options
* overrides[config.type].datasets[dataset.type]
* defaults.datasets[dataset.type]
* defaults

### [**#**](#2et92p0) Dataset animation options

* dataset.animation
* options.datasets[dataset.type].animation
* options.animation
* overrides[config.type].datasets[dataset.type].animation
* defaults.datasets[dataset.type].animation
* defaults.animation

### [**#**](#tyjcwt) Dataset element level options

Each scope is looked up with elementType prefix in the option name first, then without the prefix. For example, radius for point element is looked up using pointRadius and if that does not hit, then radius.

* dataset
* options.datasets[dataset.type]
* options.datasets[dataset.type].elements[elementType]
* options.elements[elementType]
* options
* overrides[config.type].datasets[dataset.type]
* overrides[config.type].datasets[dataset.type].elements[elementType]
* defaults.datasets[dataset.type]
* defaults.datasets[dataset.type].elements[elementType]
* defaults.elements[elementType]
* defaults

### [**#**](#3dy6vkm) Scale options

* options.scales
* overrides[config.type].scales
* defaults.scales
* defaults.scale

### [**#**](#1t3h5sf) Plugin options

A plugin can provide additionalOptionScopes array of paths to additionally look for its options in. For root scope, use empty string: ''. Most core plugins also take options from root scope.

* options.plugins[plugin.id]
* (options.[...plugin.additionalOptionScopes])
* overrides[config.type].plugins[plugin.id]
* defaults.plugins[plugin.id]
* (defaults.[...plugin.additionalOptionScopes])

## [**#**](#4d34og8) Scriptable Options

Scriptable options also accept a function which is called for each of the underlying data values and that takes the unique argument context representing contextual information (see [option context](http://docs.google.com/docs/3.9.1/general/options.html#option-context)). A resolver is passed as second parameter, that can be used to access other options in the same context.

Note

The context argument should be validated in the scriptable function, because the function can be invoked in different contexts. The type field is a good candidate for this validation.

Example:

color: function(context) {  
 const index = context.dataIndex;  
 const value = context.dataset.data[index];  
 return value < 0 ? 'red' : // draw negative values in red  
 index % 2 ? 'blue' : // else, alternate values in blue and green  
 'green';  
},  
borderColor: function(context, options) {  
 const color = options.color; // resolve the value of another scriptable option: 'red', 'blue' or 'green'  
 return Chart.helpers.color(color).lighten(0.2);  
}

## [**#**](#2s8eyo1) Indexable Options

Indexable options also accept an array in which each item corresponds to the element at the same index. Note that if there are less items than data, the items are looped over. In many cases, using a [function](#4d34og8) is more appropriate if supported.

Example:

color: [  
 'red', // color for data at index 0  
 'blue', // color for data at index 1  
 'green', // color for data at index 2  
 'black', // color for data at index 3  
 //...  
]

## [**#**](#17dp8vu) Option Context

The option context is used to give contextual information when resolving options and currently only applies to [scriptable options](#4d34og8). The object is preserved, so it can be used to store and pass information between calls.

There are multiple levels of context objects:

* chart
  + dataset
    - data
  + scale
    - tick
    - pointLabel (only used in the radial linear scale)
  + tooltip

Each level inherits its parent(s) and any contextual information stored in the parent is available through the child.

The context object contains the following properties:

### [**#**](#3rdcrjn) chart

* chart: the associated chart
* type: 'chart'

### [**#**](#26in1rg) dataset

In addition to [chart](#3rdcrjn)

* active: true if element is active (hovered)
* dataset: dataset at index datasetIndex
* datasetIndex: index of the current dataset
* index: same as datasetIndex
* mode: the update mode
* type: 'dataset'

### [**#**](#lnxbz9) data

In addition to [dataset](#26in1rg)

* active: true if element is active (hovered)
* dataIndex: index of the current data
* parsed: the parsed data values for the given dataIndex and datasetIndex
* raw: the raw data values for the given dataIndex and datasetIndex
* element: the element (point, arc, bar, etc.) for this data
* index: same as dataIndex
* type: 'data'

### [**#**](#35nkun2) scale

In addition to [chart](#3rdcrjn)

* scale: the associated scale
* type: 'scale'

### [**#**](#1ksv4uv) tick

In addition to [scale](#35nkun2)

* tick: the associated tick object
* index: tick index
* type: 'tick'

### [**#**](#44sinio) tooltip

In addition to [chart](#3rdcrjn)

* tooltip: the tooltip object
* tooltipItems: the items the tooltip is displaying

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

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