* [Chart.js](http://docs.google.com/)
* [Getting Started](http://docs.google.com/getting-started/)
  + [Installation](http://docs.google.com/getting-started/installation.html)
  + [Integration](http://docs.google.com/getting-started/integration.html)
  + [Usage](http://docs.google.com/getting-started/usage.html)
* [General](http://docs.google.com/general/)
  + [Responsive](http://docs.google.com/general/responsive.html)
  + [Pixel Ratio](http://docs.google.com/general/device-pixel-ratio.html)
  + [Interactions](http://docs.google.com/general/interactions/)
    - [Events](http://docs.google.com/general/interactions/events.html)
    - [Modes](http://docs.google.com/general/interactions/modes.html)
  + [Options](http://docs.google.com/general/options.html)
  + [Colors](http://docs.google.com/general/colors.html)
  + [Fonts](http://docs.google.com/general/fonts.html)
* [Configuration](http://docs.google.com/)
  + [Animations](http://docs.google.com/animations.html)
  + [Layout](http://docs.google.com/layout.html)
  + [Legend](http://docs.google.com/legend.html)
  + [Title](http://docs.google.com/title.html)
  + [Tooltip](http://docs.google.com/tooltip.html)
  + [Elements](http://docs.google.com/elements.html)
* [Charts](http://docs.google.com/charts/)
  + [Line](http://docs.google.com/charts/line.html)
  + [Bar](http://docs.google.com/charts/bar.html)
  + [Radar](http://docs.google.com/charts/radar.html)
  + [Doughnut & Pie](http://docs.google.com/charts/doughnut.html)
  + [Polar Area](http://docs.google.com/charts/polar.html)
  + [Bubble](http://docs.google.com/charts/bubble.html)
  + [Scatter](http://docs.google.com/charts/scatter.html)
  + [Area](http://docs.google.com/charts/area.html)
  + [Mixed](http://docs.google.com/charts/mixed.html)
* [Axes](http://docs.google.com/axes/)
  + [Cartesian](http://docs.google.com/axes/cartesian/)
    - [Category](http://docs.google.com/axes/cartesian/category.html)
    - [Linear](http://docs.google.com/axes/cartesian/linear.html)
    - [Logarithmic](http://docs.google.com/axes/cartesian/logarithmic.html)
    - [Time](http://docs.google.com/axes/cartesian/time.html)
  + [Radial](http://docs.google.com/axes/radial/)
    - [Linear](http://docs.google.com/axes/radial/linear.html)
  + [Labelling](http://docs.google.com/axes/labelling.html)
  + [Styling](http://docs.google.com/axes/styling.html)
* [Developers](http://docs.google.com/developers/)
  + [Chart.js API](http://docs.google.com/developers/api.html)
  + [Updating Charts](http://docs.google.com/developers/updates.html)
  + [Plugins](http://docs.google.com/developers/plugins.html)
  + [New Charts](http://docs.google.com/developers/charts.html)
  + [New Axes](http://docs.google.com/developers/axes.html)
  + [Contributing](http://docs.google.com/developers/contributing.html)
* [Additional Notes](http://docs.google.com/notes/)
  + [Comparison Table](http://docs.google.com/notes/comparison.html)
  + [Popular Extensions](http://docs.google.com/notes/extensions.html)
  + [License](http://docs.google.com/notes/license.html)
* [Published with GitBook](https://www.gitbook.com)

[**Configuration**](http://docs.google.com/)

Configuration

The configuration is used to change how the chart behaves. There are properties to control styling, fonts, the legend, etc.

## Global Configuration

This concept was introduced in Chart.js 1.0 to keep configuration [DRY](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself), and allow for changing options globally across chart types, avoiding the need to specify options for each instance, or the default for a particular chart type.

Chart.js merges the options object passed to the chart with the global configuration using chart type defaults and scales defaults appropriately. This way you can be as specific as you would like in your individual chart configuration, while still changing the defaults for all chart types where applicable. The global general options are defined in Chart.defaults.global. The defaults for each chart type are discussed in the documentation for that chart type.

The following example would set the hover mode to 'nearest' for all charts where this was not overridden by the chart type defaults or the options passed to the constructor on creation.

Chart.defaults.global.hover.mode = 'nearest';  
  
// Hover mode is set to nearest because it was not overridden here  
var chartHoverModeNearest = new Chart(ctx, {  
 type: 'line',  
 data: data,  
});  
  
// This chart would have the hover mode that was passed in  
var chartDifferentHoverMode = new Chart(ctx, {  
 type: 'line',  
 data: data,  
 options: {  
 hover: {  
 // Overrides the global setting  
 mode: 'index'  
 }  
 }  
})

results matching ""

No results matching ""