* [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/configuration/)
  + [Animations](http://docs.google.com/configuration/animations.html)
  + [Layout](http://docs.google.com/configuration/layout.html)
  + [Legend](http://docs.google.com/configuration/legend.html)
  + [Title](http://docs.google.com/configuration/title.html)
  + [Tooltip](http://docs.google.com/configuration/tooltip.html)
  + [Elements](http://docs.google.com/configuration/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/)
  + [Chart.js API](http://docs.google.com/api.html)
  + [Updating Charts](http://docs.google.com/updates.html)
  + [Plugins](http://docs.google.com/plugins.html)
  + [New Charts](http://docs.google.com/charts.html)
  + [New Axes](http://docs.google.com/axes.html)
  + [Contributing](http://docs.google.com/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)

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

Plugins

Plugins are the most efficient way to customize or change the default behavior of a chart. They have been introduced at [version 2.1.0](https://github.com/chartjs/Chart.js/releases/tag/2.1.0) (global plugins only) and extended at [version 2.5.0](https://github.com/chartjs/Chart.js/releases/tag/v2.5.0) (per chart plugins and options).

## Using plugins

Plugins can be shared between chart instances:

var plugin = { /\* plugin implementation \*/ };  
  
// chart1 and chart2 use "plugin"  
var chart1 = new Chart(ctx, {  
 plugins: [plugin]  
});  
  
var chart2 = new Chart(ctx, {  
 plugins: [plugin]  
});  
  
// chart3 doesn't use "plugin"  
var chart3 = new Chart(ctx, {});

Plugins can also be defined directly in the chart plugins config (a.k.a. *inline plugins*):

var chart = new Chart(ctx, {  
 plugins: [{  
 beforeInit: function(chart, options) {  
 //..  
 }  
 }]  
});

However, this approach is not ideal when the customization needs to apply to many charts.

## Global plugins

Plugins can be registered globally to be applied on all charts (a.k.a. *global plugins*):

Chart.plugins.register({  
 // plugin implementation  
});

Note: *inline* plugins can't be registered globally.

## Configuration

### Plugin ID

Plugins must define a unique id in order to be configurable.

This id should follow the [npm package name convention](https://docs.npmjs.com/files/package.json#name):

* can't start with a dot or an underscore
* can't contain any non-URL-safe characters
* can't contain uppercase letters
* should be something short, but also reasonably descriptive

If a plugin is intended to be released publicly, you may want to check the [registry](https://www.npmjs.com/search?q=chartjs-plugin-) to see if there's something by that name already. Note that in this case, the package name should be prefixed by chartjs-plugin- to appear in Chart.js plugin registry.

### Plugin options

Plugin options are located under the options.plugins config and are scoped by the plugin ID: options.plugins.{plugin-id}.

var chart = new Chart(ctx, {  
 config: {  
 foo: { ... }, // chart 'foo' option  
 plugins: {  
 p1: {  
 foo: { ... }, // p1 plugin 'foo' option  
 bar: { ... }  
 },  
 p2: {  
 foo: { ... }, // p2 plugin 'foo' option  
 bla: { ... }  
 }  
 }  
 }  
});

#### Disable plugins

To disable a global plugin for a specific chart instance, the plugin options must be set to false:

Chart.plugins.register({  
 id: 'p1',  
 // ...  
});  
  
var chart = new Chart(ctx, {  
 config: {  
 plugins: {  
 p1: false // disable plugin 'p1' for this instance  
 }  
 }  
});

## Plugin Core API

Available hooks (as of version 2.6):

* beforeInit
* afterInit
* beforeUpdate *(cancellable)*
* afterUpdate
* beforeLayout *(cancellable)*
* afterLayout
* beforeDatasetsUpdate *(cancellable)*
* afterDatasetsUpdate
* beforeDatasetUpdate *(cancellable)*
* afterDatasetUpdate
* beforeRender *(cancellable)*
* afterRender
* beforeDraw *(cancellable)*
* afterDraw
* beforeDatasetsDraw *(cancellable)*
* afterDatasetsDraw
* beforeDatasetDraw *(cancellable)*
* afterDatasetDraw
* beforeEvent *(cancellable)*
* afterEvent
* resize
* destroy

results matching ""

No results matching ""