* [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/)
  + [Cartesian](http://docs.google.com/cartesian/)
    - [Category](http://docs.google.com/cartesian/category.html)
    - [Linear](http://docs.google.com/cartesian/linear.html)
    - [Logarithmic](http://docs.google.com/cartesian/logarithmic.html)
    - [Time](http://docs.google.com/cartesian/time.html)
  + [Radial](http://docs.google.com/radial/)
    - [Linear](http://docs.google.com/radial/linear.html)
  + [Labelling](http://docs.google.com/labelling.html)
  + [Styling](http://docs.google.com/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)

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

Axes

Axes are an integral part of a chart. They are used to determine how data maps to a pixel value on the chart. In a cartesian chart, there is 1 or more X axis and 1 or more Y axis to map points onto the 2 dimensional canvas. These axes are know as ['cartesian axes'](http://docs.google.com/cartesian/#cartesian-axes).

In a radial chart, such as a radar chart or a polar area chart, there is a single axis that maps points in the angular and radial directions. These are known as ['radial axes'](http://docs.google.com/radial/#radial-axes).

Scales in Chart.js >V2.0 are significantly more powerful, but also different than those of v1.0.

* Multiple X & Y axes are supported.
* A built-in label auto-skip feature detects would-be overlapping ticks and labels and removes every nth label to keep things displaying normally.
* Scale titles are supported
* New scale types can be extended without writing an entirely new chart type

# Common Configuration

The following properties are common to all axes provided by Chart.js

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Default | Description |
| display | Boolean | true | If set to false the axis is hidden from view. Overrides *gridLines.display*, *scaleLabel.display*, and *ticks.display*. |
| callbacks | Object |  | Callback functions to hook into the axis lifecycle. [more...](#gjdgxs) |
| weight | Number | 0 | The weight used to sort the axis. Higher weights are further away from the chart area. |

## Callbacks

There are a number of config callbacks that can be used to change parameters in the scale at different points in the update process.

|  |  |  |
| --- | --- | --- |
| Name | Arguments | Description |
| beforeUpdate | axis | Callback called before the update process starts. |
| beforeSetDimensions | axis | Callback that runs before dimensions are set. |
| afterSetDimensions | axis | Callback that runs after dimensions are set. |
| beforeDataLimits | axis | Callback that runs before data limits are determined. |
| afterDataLimits | axis | Callback that runs after data limits are determined. |
| beforeBuildTicks | axis | Callback that runs before ticks are created. |
| afterBuildTicks | axis | Callback that runs after ticks are created. Useful for filtering ticks. |
| beforeTickToLabelConversion | axis | Callback that runs before ticks are converted into strings. |
| afterTickToLabelConversion | axis | Callback that runs after ticks are converted into strings. |
| beforeCalculateTickRotation | axis | Callback that runs before tick rotation is determined. |
| afterCalculateTickRotation | axis | Callback that runs after tick rotation is determined. |
| beforeFit | axis | Callback that runs before the scale fits to the canvas. |
| afterFit | axis | Callback that runs after the scale fits to the canvas. |
| afterUpdate | axis | Callback that runs at the end of the update process. |

## Updating Axis Defaults

The default configuration for a scale can be easily changed using the scale service. All you need to do is to pass in a partial configuration that will be merged with the current scale default configuration to form the new default.

For example, to set the minimum value of 0 for all linear scales, you would do the following. Any linear scales created after this time would now have a minimum of 0.

Chart.scaleService.updateScaleDefaults('linear', {  
 ticks: {  
 min: 0  
 }  
});

## Creating New Axes

To create a new axis, see the [developer docs](http://docs.google.com/developers/axes.html).

results matching ""

No results matching ""