 [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) Responsive Charts

When it comes to changing the chart size based on the window size, a major limitation is that the canvas *render* size (canvas.width and .height) can **not** be expressed with relative values, contrary to the *display* size (canvas.style.width and .height). Furthermore, these sizes are independent from each other and thus the canvas *render* size does not adjust automatically based on the *display* size, making the rendering inaccurate.

The following examples **do not work**:

* <canvas height="40vh" width="80vw">: **invalid** values, the canvas doesn't resize ([example (opens new window)](https://codepen.io/chartjs/pen/oWLZaR))
* <canvas style="height:40vh; width:80vw">: **invalid** behavior, the canvas is resized but becomes blurry ([example (opens new window)](https://codepen.io/chartjs/pen/WjxpmO))
* <canvas style="margin: 0 auto;">: **invalid** behavior, the canvas continually shrinks. Chart.js needs a dedicated container for each canvas and this styling should be applied there.

Chart.js provides a [few options](#30j0zll) to enable responsiveness and control the resize behavior of charts by detecting when the canvas *display* size changes and update the *render* size accordingly.

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

Namespace: options

| Name | Type | Default | Description |
| --- | --- | --- | --- |
| responsive | boolean | true | Resizes the chart canvas when its container does ([important note...](#1fob9te)). |
| maintainAspectRatio | boolean | true | Maintain the original canvas aspect ratio (width / height) when resizing. |
| aspectRatio | number | 1|2 | Canvas aspect ratio (i.e. width / height, a value of 1 representing a square canvas). Note that this option is ignored if the height is explicitly defined either as attribute or via the style. The default value varies by chart type; Radial charts (doughnut, pie, polarArea, radar) default to 1 and others default to 2. |
| onResize | function | null | Called when a resize occurs. Gets passed two arguments: the chart instance and the new size. |
| resizeDelay | number | 0 | Delay the resize update by the given amount of milliseconds. This can ease the resize process by debouncing the update of the elements. |

## [**#**](#1fob9te) Important Note

Detecting when the canvas size changes can not be done directly from the canvas element. Chart.js uses its parent container to update the canvas *render* and *display* sizes. However, this method requires the container to be **relatively positioned** and **dedicated to the chart canvas only**. Responsiveness can then be achieved by setting relative values for the container size ([example (opens new window)](https://codepen.io/chartjs/pen/YVWZbz)):

<div class="chart-container" style="position: relative; height:40vh; width:80vw">  
 <canvas id="chart"></canvas>  
</div>

The chart can also be programmatically resized by modifying the container size:

chart.canvas.parentNode.style.height = '128px';  
chart.canvas.parentNode.style.width = '128px';

Note that in order for the above code to correctly resize the chart height, the [maintainAspectRatio](#30j0zll) option must also be set to false.

## [**#**](#3znysh7) Printing Resizable Charts

CSS media queries allow changing styles when printing a page. The CSS applied from these media queries may cause charts to need to resize. However, the resize won't happen automatically. To support resizing charts when printing, you need to hook the [onbeforeprint (opens new window)](https://developer.mozilla.org/en-US/docs/Web/API/WindowEventHandlers/onbeforeprint) event and manually trigger resizing of each chart.

function beforePrintHandler () {  
 for (let id in Chart.instances) {  
 Chart.instances[id].resize();  
 }  
}

You may also find that, due to complexities in when the browser lays out the document for printing and when resize events are fired, Chart.js is unable to properly resize for the print layout. To work around this, you can pass an explicit size to .resize() then use an [onafterprint (opens new window)](https://developer.mozilla.org/en-US/docs/Web/API/WindowEventHandlers/onafterprint) event to restore the automatic size when done.

window.addEventListener('beforeprint', () => {  
 myChart.resize(600, 600);  
});  
window.addEventListener('afterprint', () => {  
 myChart.resize();  
});

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

←  [Locale](http://docs.google.com/docs/3.9.1/configuration/locale.html)   [Subtitle](http://docs.google.com/docs/3.9.1/configuration/subtitle.html)  →