* [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/)
  + [Responsive](http://docs.google.com/responsive.html)
  + [Pixel Ratio](http://docs.google.com/device-pixel-ratio.html)
  + [Interactions](http://docs.google.com/interactions/)
    - [Events](http://docs.google.com/interactions/events.html)
    - [Modes](http://docs.google.com/interactions/modes.html)
  + [Options](http://docs.google.com/options.html)
  + [Colors](http://docs.google.com/colors.html)
  + [Fonts](http://docs.google.com/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/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)

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

Responsive Charts

When it comes to change 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](https://codepen.io/chartjs/pen/oWLZaR))
* <canvas style="height:40vh; width:80vw">: **invalid** behavior, the canvas is resized but becomes blurry ([example](https://codepen.io/chartjs/pen/WjxpmO))

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

## Configuration Options

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Default | Description |
| responsive | Boolean | true | Resizes the chart canvas when its container does ([important note...](#30j0zll)). |
| responsiveAnimationDuration | Number | 0 | Duration in milliseconds it takes to animate to new size after a resize event. |
| maintainAspectRatio | Boolean | true | Maintain the original canvas aspect ratio (width / height) when resizing. |
| onResize | Function | null | Called when a resize occurs. Gets passed two arguments: the chart instance and the new size. |

## 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](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';

## Printing Resizeable 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, one needs to hook the [onbeforeprint](https://developer.mozilla.org/en-US/docs/Web/API/WindowEventHandlers/onbeforeprint) event and manually trigger resizing of each chart.

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

results matching ""

No results matching ""