FREE eBook

# LEARNING
# chart.js

Free unaffiliated eBook created from
**Stack Overflow contributors.**

#chart.js

# Table of Contents

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: chart-js

It is an unofficial and free chart.js ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official chart.js.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

# Chapter 1: Getting started with chart.js

## Remarks

Chart.js is a simple yet flexible open source JavaScript charting library for designers & developers.

For version information check out their GitHub

## Examples

### Installation or Setup

Chart.js can be included in several different ways:

**NPM**

Run the following command on your NPM project directory

```
npm install chart.js --save
```

**CDN**

Include a script tag in your HTML linking to the chart.js CDN

```
<html>
   <body>
      <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.1.6/Chart.bundle.min.js"></script>
   </body>
</html>
```

Latest version can be found at cdnjs.com/libraries/Chart.js.

**Local Copy**

A local copy can also be hosted on your server. You can get the lasted version from their GitHub.

```
<html>
   <body>
      <script type="text/javascript" src="/Path/To/Chart.bundle.min.js"></script>
   </body>
</html>
```

For more information about getting chart.js installed see www.chartjs.org/docs/.

### Minimal Chart Example

Depending on the version of Chart.JS you are using (the current one being 2.X), the syntax is different to create a minimal example of a bar chart (JSFiddle Demo for 2.X).

## Chart.js 2.X

```
<html>
    <body>
        <canvas id="myChart" width="400" height="400"></canvas>
        <script>
              var ctx = document.getElementById("myChart");
              var myChart = new Chart(ctx, {
                  type: 'bar',
                  data: {
                      labels: ["Group 1", "Group 2", "Group 3"],
                      datasets: [{
                          label: 'Groups',
                          data: [12, 19, 3]
                      }]
                  }
              });
        </script>
    </body>
</html>
```

A slightly more advanced version of this can be found in the chart.js documentation (JSFiddle Demo).

## Chart.js 1.X

However, if you need to use the legacy version, first take a look at the documentation on Github.

Now here is a minimal example of a bar chart (JSFiddle Demo) :

```
<html>
    <body>
        <canvas id="myChart" width="400" height="400"></canvas>
        <script>
            var ctx = document.getElementById("myChart");
            var myChart= new Chart(ctx).Bar({
                labels: ["Group 1", "Group 2", "Group 3"],
                datasets: [
                {
                    label: "Group",
                    data: [12, 19, 3]
                }]
            });
        </script>
    </body>
</html>
```

A slightly more advanced version of this can be found in the Github documentation (JSFiddle Demo).

Read Getting started with chart.js online: https://riptutorial.com/chart-js/topic/4274/getting-started-with-chart-js

# Chapter 2: Plugins

## Remarks

Starting with v2.1.0, you can create plugins for chart.js! Plugin official documentation

## Examples

### Plugins Introduction

Plugins are a way for a developer to modify a chart as it is being created. Chart.js calls all plugins at the following chart states:

- Start of initialization
- End of initialization
- Start of update
- After the chart scales have calculated
- Start of datasets update
- End of datasets update
- End of update (before render occurs)
- Start of draw
- End of draw
- Before datasets draw
- After datasets draw
- Resize
- Before an animation is started

**Creating a plugin**

To create a plugin, create a JavaScript object with appropriate named functions for any chart state you wish to modify (listed above). After you have your plugin object, pass it to `Chart.pluginService.register(PLUGIN_OBJECT_NAME);` to let Chart.js know to register the plugin.

**Minimal Plugin Example**

```
// Create the plugin object with functions for all the chart states
var simplePlugin = {
  beforeInit: function(chartInstance) {},
  afterInit: function(chartInstance) {},

  resize: function(chartInstance, newChartSize) {},

  beforeUpdate: function(chartInstance) {},
  afterScaleUpdate: function(chartInstance) {},
  beforeDatasetsUpdate: function(chartInstance) {},
  afterDatasetsUpdate: function(chartInstance) {},
  afterUpdate: function(chartInstance) {},
  // This is called at the start of a render. It is only called once, even if the animation will run for a number of frames. Use beforeDraw or afterDraw
  // to do something on each animation frame
  beforeRender: function(chartInstance) {},

  // Easing is for animation
  beforeDraw: function(chartInstance, easing) {},
  afterDraw: function(chartInstance, easing) {},

  // Before the datasets are drawn but after scales are drawn
  beforeDatasetsDraw: function(chartInstance, easing) {},
  afterDatasetsDraw: function(chartInstance, easing) {},

  destroy: function(chartInstance) {}
};

// Let Chart.js know about the new plugin
Chart.pluginService.register(simplePlugin);
```

Currently this minimal plugin does not do anything. To make this plugin useful one would need to add code to the functions that modifies the chart.

## Draw Horizonal Lines

Create horizontal lines with a label. This could be used to show notable values in the chart data (e.g. min, max, average). JSFiddle Demo

```
var horizonalLinePlugin = {
  afterDraw: function(chartInstance) {
    var yScale = chartInstance.scales["y-axis-0"];
    var canvas = chartInstance.chart;
    var ctx = canvas.ctx;
    var index;
    var line;
    var style;

    if (chartInstance.options.horizontalLine) {
      for (index = 0; index < chartInstance.options.horizontalLine.length; index++) {
        line = chartInstance.options.horizontalLine[index];

        if (!line.style) {
          style = "rgba(169,169,169, .6)";
        } else {
          style = line.style;
        }

        if (line.y) {
          yValue = yScale.getPixelForValue(line.y);
        } else {
          yValue = 0;
        }

        ctx.lineWidth = 3;

        if (yValue) {
          ctx.beginPath();
          ctx.moveTo(0, yValue);
          ctx.lineTo(canvas.width, yValue);
          ctx.strokeStyle = style;
          ctx.stroke();
        }

        if (line.text) {
          ctx.fillStyle = style;
          ctx.fillText(line.text, 0, yValue + ctx.lineWidth);
        }
      }
      return;
    };
  }
};
Chart.pluginService.register(horizonalLinePlugin);
```

Credit to L Bahr for the example

Read Plugins online: https://riptutorial.com/chart-js/topic/6510/plugins

# Credits

| S. No | Chapters | Contributors |
|---|---|---|
| 1 | Getting started with chart.js | Community, L Bahr, tektiv |
| 2 | Plugins | L Bahr, Lucas |