 [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
  + [Getting Started](http://docs.google.com/docs/3.9.1/getting-started/)
  + [Installation](http://docs.google.com/docs/3.9.1/getting-started/installation.html)
  + [Integration](http://docs.google.com/docs/3.9.1/getting-started/integration.html)
  + [Usage](http://docs.google.com/docs/3.9.1/getting-started/usage.html)
  + [3.x Migration Guide](http://docs.google.com/docs/3.9.1/getting-started/v3-migration.html)
* General
* Configuration
* Chart Types
* Axes
* Developers

[**#**](#gjdgxs) Integration

Chart.js can be integrated with plain JavaScript or with different module loaders. The examples below show how to load Chart.js in different systems.

## [**#**](#30j0zll) Script Tag

<script src="path/to/chartjs/dist/chart.js"></script>  
<script>  
 const myChart = new Chart(ctx, {...});  
</script>

## [**#**](#1fob9te) Common JS

const Chart = require('chart.js');  
const myChart = new Chart(ctx, {...});

## [**#**](#3znysh7) Bundlers (Webpack, Rollup, etc.)

Chart.js 3 is tree-shakeable, so it is necessary to import and register the controllers, elements, scales and plugins you are going to use.

For all available imports see the example below.

import {  
 Chart,  
 ArcElement,  
 LineElement,  
 BarElement,  
 PointElement,  
 BarController,  
 BubbleController,  
 DoughnutController,  
 LineController,  
 PieController,  
 PolarAreaController,  
 RadarController,  
 ScatterController,  
 CategoryScale,  
 LinearScale,  
 LogarithmicScale,  
 RadialLinearScale,  
 TimeScale,  
 TimeSeriesScale,  
 Decimation,  
 Filler,  
 Legend,  
 Title,  
 Tooltip,  
 SubTitle  
} from 'chart.js';  
Chart.register(  
 ArcElement,  
 LineElement,  
 BarElement,  
 PointElement,  
 BarController,  
 BubbleController,  
 DoughnutController,  
 LineController,  
 PieController,  
 PolarAreaController,  
 RadarController,  
 ScatterController,  
 CategoryScale,  
 LinearScale,  
 LogarithmicScale,  
 RadialLinearScale,  
 TimeScale,  
 TimeSeriesScale,  
 Decimation,  
 Filler,  
 Legend,  
 Title,  
 Tooltip,  
 SubTitle  
);  
const myChart = new Chart(ctx, {...});

A short registration format is also available to quickly register everything.

import { Chart, registerables } from 'chart.js';  
Chart.register(...registerables);

And finally there is a separate path to do just the above for you, in one line:

import Chart from 'chart.js/auto';

### [**#**](#2et92p0) Helper functions

If you want to use the helper functions, you will need to import these separately from the helpers package and use them as stand-alone functions.

Example of [Converting Events to Data Values](http://docs.google.com/docs/3.9.1/configuration/interactions.html#converting-events-to-data-values) using bundlers.

import Chart from 'chart.js/auto';  
import { getRelativePosition } from 'chart.js/helpers';  
const chart = new Chart(ctx, {  
 type: 'line',  
 data: data,  
 options: {  
 onClick: (e) => {  
 const canvasPosition = getRelativePosition(e, chart);  
 // Substitute the appropriate scale IDs  
 const dataX = chart.scales.x.getValueForPixel(canvasPosition.x);  
 const dataY = chart.scales.y.getValueForPixel(canvasPosition.y);  
 }  
 }  
});

## [**#**](#tyjcwt) Require JS

**Important:** RequireJS [can **not** load CommonJS module as is (opens new window)](https://requirejs.org/docs/commonjs.html#intro), so be sure to require one of the UMD builds instead (i.e. dist/chart.js, dist/chart.min.js, etc.).

require(['path/to/chartjs/dist/chart.min.js'], function(Chart){  
 const myChart = new Chart(ctx, {...});  
});

**Note:** in order to use the time scale, you need to make sure [one of the available date adapters (opens new window)](https://github.com/chartjs/awesome#adapters) and corresponding date library are fully loaded **after** requiring Chart.js. For this you can use nested requires:

require(['chartjs'], function(Chart) {  
 require(['moment'], function() {  
 require(['chartjs-adapter-moment'], function() {  
 new Chart(ctx, {...});  
 });  
 });  
});

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

←  [Installation](http://docs.google.com/docs/3.9.1/getting-started/installation.html)   [Usage](http://docs.google.com/docs/3.9.1/getting-started/usage.html)  →