 [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)

* [Information](http://docs.google.com/docs/3.9.1/samples/information.html)
* Bar Charts
* Line Charts
* Other charts
* Area charts
* Scales
* Scale Options
* Legend
* Title
* Subtitle
* Tooltip
* Scriptable Options
* Animations
* Advanced
* Plugins
* [Utils](http://docs.google.com/docs/3.9.1/samples/utils.html)

[**#**](#gjdgxs) Utils

## [**#**](#30j0zll) Disclaimer

The Utils file contains multiple helper functions that the chart.js sample pages use to generate charts. These functions are subject to change, including but not limited to breaking changes without prior notice.

Because of this please don't rely on this file in production environments.

## [**#**](#1fob9te) Functions

import colorLib from '@kurkle/color';  
import {DateTime} from 'luxon';  
import 'chartjs-adapter-luxon';  
import {valueOrDefault} from '../../dist/helpers.mjs';  
// Adapted from http://indiegamr.com/generate-repeatable-random-numbers-in-js/  
var \_seed = Date.now();  
export function srand(seed) {  
 \_seed = seed;  
}  
export function rand(min, max) {  
 min = valueOrDefault(min, 0);  
 max = valueOrDefault(max, 0);  
 \_seed = (\_seed \* 9301 + 49297) % 233280;  
 return min + (\_seed / 233280) \* (max - min);  
}  
export function numbers(config) {  
 var cfg = config || {};  
 var min = valueOrDefault(cfg.min, 0);  
 var max = valueOrDefault(cfg.max, 100);  
 var from = valueOrDefault(cfg.from, []);  
 var count = valueOrDefault(cfg.count, 8);  
 var decimals = valueOrDefault(cfg.decimals, 8);  
 var continuity = valueOrDefault(cfg.continuity, 1);  
 var dfactor = Math.pow(10, decimals) || 0;  
 var data = [];  
 var i, value;  
 for (i = 0; i < count; ++i) {  
 value = (from[i] || 0) + this.rand(min, max);  
 if (this.rand() <= continuity) {  
 data.push(Math.round(dfactor \* value) / dfactor);  
 } else {  
 data.push(null);  
 }  
 }  
 return data;  
}  
export function points(config) {  
 const xs = this.numbers(config);  
 const ys = this.numbers(config);  
 return xs.map((x, i) => ({x, y: ys[i]}));  
}  
export function bubbles(config) {  
 return this.points(config).map(pt => {  
 pt.r = this.rand(config.rmin, config.rmax);  
 return pt;  
 });  
}  
export function labels(config) {  
 var cfg = config || {};  
 var min = cfg.min || 0;  
 var max = cfg.max || 100;  
 var count = cfg.count || 8;  
 var step = (max - min) / count;  
 var decimals = cfg.decimals || 8;  
 var dfactor = Math.pow(10, decimals) || 0;  
 var prefix = cfg.prefix || '';  
 var values = [];  
 var i;  
 for (i = min; i < max; i += step) {  
 values.push(prefix + Math.round(dfactor \* i) / dfactor);  
 }  
 return values;  
}  
const MONTHS = [  
 'January',  
 'February',  
 'March',  
 'April',  
 'May',  
 'June',  
 'July',  
 'August',  
 'September',  
 'October',  
 'November',  
 'December'  
];  
export function months(config) {  
 var cfg = config || {};  
 var count = cfg.count || 12;  
 var section = cfg.section;  
 var values = [];  
 var i, value;  
 for (i = 0; i < count; ++i) {  
 value = MONTHS[Math.ceil(i) % 12];  
 values.push(value.substring(0, section));  
 }  
 return values;  
}  
const COLORS = [  
 '#4dc9f6',  
 '#f67019',  
 '#f53794',  
 '#537bc4',  
 '#acc236',  
 '#166a8f',  
 '#00a950',  
 '#58595b',  
 '#8549ba'  
];  
export function color(index) {  
 return COLORS[index % COLORS.length];  
}  
export function transparentize(value, opacity) {  
 var alpha = opacity === undefined ? 0.5 : 1 - opacity;  
 return colorLib(value).alpha(alpha).rgbString();  
}  
export const CHART\_COLORS = {  
 red: 'rgb(255, 99, 132)',  
 orange: 'rgb(255, 159, 64)',  
 yellow: 'rgb(255, 205, 86)',  
 green: 'rgb(75, 192, 192)',  
 blue: 'rgb(54, 162, 235)',  
 purple: 'rgb(153, 102, 255)',  
 grey: 'rgb(201, 203, 207)'  
};  
const NAMED\_COLORS = [  
 CHART\_COLORS.red,  
 CHART\_COLORS.orange,  
 CHART\_COLORS.yellow,  
 CHART\_COLORS.green,  
 CHART\_COLORS.blue,  
 CHART\_COLORS.purple,  
 CHART\_COLORS.grey,  
];  
export function namedColor(index) {  
 return NAMED\_COLORS[index % NAMED\_COLORS.length];  
}  
export function newDate(days) {  
 return DateTime.now().plus({days}).toJSDate();  
}  
export function newDateString(days) {  
 return DateTime.now().plus({days}).toISO();  
}  
export function parseISODate(str) {  
 return DateTime.fromISO(str);  
}

[File on github (opens new window)](https://github.com/chartjs/Chart.js/blob/master/docs/scripts/utils.js)

## [**#**](#3znysh7) Components

Some of the samples make reference to a components object. This is an artifact of using a module bundler to build the samples. The creation of that components object is shown below. If chart.js is included as a browser script, these items are accessible via the Chart object, i.e Chart.Tooltip.

// Add Chart components needed in samples here.  
// Usable through `components[name]`.  
export {Tooltip} from '../../dist/chart.mjs';

[File on github (opens new window)](https://github.com/chartjs/Chart.js/blob/master/docs/scripts/components.js)

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

←  [Quadrants](http://docs.google.com/docs/3.9.1/samples/plugins/quadrants.html)