 [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
  + [Bar Chart](http://docs.google.com/docs/3.9.1/samples/scriptable/bar.html)
  + [Bubble Chart](http://docs.google.com/docs/3.9.1/samples/scriptable/bubble.html)
  + [Line Chart](http://docs.google.com/docs/3.9.1/samples/scriptable/line.html)
  + [Pie Chart](http://docs.google.com/docs/3.9.1/samples/scriptable/pie.html)
  + [Polar Area Chart](http://docs.google.com/docs/3.9.1/samples/scriptable/polar.html)
  + [Radar Chart](http://docs.google.com/docs/3.9.1/samples/scriptable/radar.html)
* Animations
* Advanced
* Plugins
* [Utils](http://docs.google.com/docs/3.9.1/samples/utils.html)

[**#**](#gjdgxs) Radar Chart

options data setup

function getLineColor(ctx) { return Utils.color(ctx.datasetIndex); } function alternatePointStyles(ctx) { const index = ctx.dataIndex; return index % 2 === 0 ? 'circle' : 'rect'; } function makeHalfAsOpaque(ctx) { return Utils.transparentize(getLineColor(ctx)); } function make20PercentOpaque(ctx) { return Utils.transparentize(getLineColor(ctx), 0.8); } function adjustRadiusBasedOnData(ctx) { const v = ctx.parsed.y; return v < 10 ? 5 : v < 25 ? 7 : v < 50 ? 9 : v < 75 ? 11 : 15; } const config = { type: 'radar', data: data, options: { plugins: { legend: false, tooltip: false, }, elements: { line: { backgroundColor: make20PercentOpaque, borderColor: getLineColor, }, point: { backgroundColor: getLineColor, hoverBackgroundColor: makeHalfAsOpaque, radius: adjustRadiusBasedOnData, pointStyle: alternatePointStyles, hoverRadius: 15, } } } };

function getLineColor(ctx) {  
 return Utils.color(ctx.datasetIndex);  
}  
function alternatePointStyles(ctx) {  
 const index = ctx.dataIndex;  
 return index % 2 === 0 ? 'circle' : 'rect';  
}  
function makeHalfAsOpaque(ctx) {  
 return Utils.transparentize(getLineColor(ctx));  
}  
function make20PercentOpaque(ctx) {  
 return Utils.transparentize(getLineColor(ctx), 0.8);  
}  
function adjustRadiusBasedOnData(ctx) {  
 const v = ctx.parsed.y;  
 return v < 10 ? 5  
 : v < 25 ? 7  
 : v < 50 ? 9  
 : v < 75 ? 11  
 : 15;  
}  
const config = {  
 type: 'radar',  
 data: data,  
 options: {  
 plugins: {  
 legend: false,  
 tooltip: false,  
 },  
 elements: {  
 line: {  
 backgroundColor: make20PercentOpaque,  
 borderColor: getLineColor,  
 },  
 point: {  
 backgroundColor: getLineColor,  
 hoverBackgroundColor: makeHalfAsOpaque,  
 radius: adjustRadiusBasedOnData,  
 pointStyle: alternatePointStyles,  
 hoverRadius: 15,  
 }  
 }  
 }  
};

function generateData() { return Utils.numbers({ count: DATA\_COUNT, min: 0, max: 100 }); } const data = { labels: [['Eating', 'Dinner'], ['Drinking', 'Water'], 'Sleeping', ['Designing', 'Graphics'], 'Coding', 'Cycling', 'Running'], datasets: [{ data: generateData() }] };

function generateData() {  
 return Utils.numbers({  
 count: DATA\_COUNT,  
 min: 0,  
 max: 100  
 });  
}  
const data = {  
 labels: [['Eating', 'Dinner'], ['Drinking', 'Water'], 'Sleeping', ['Designing', 'Graphics'], 'Coding', 'Cycling', 'Running'],  
 datasets: [{  
 data: generateData()  
 }]  
};

const DATA\_COUNT = 7; Utils.srand(110); const actions = [ { name: 'Randomize', handler(chart) { chart.data.datasets.forEach(dataset => { dataset.data = generateData(); }); chart.update(); } }, ];

const DATA\_COUNT = 7;  
Utils.srand(110);  
const actions = [  
 {  
 name: 'Randomize',  
 handler(chart) {  
 chart.data.datasets.forEach(dataset => {  
 dataset.data = generateData();  
 });  
 chart.update();  
 }  
 },  
];

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

* [Options](http://docs.google.com/docs/3.9.1/general/options.html)
  + [Scriptable Options](http://docs.google.com/docs/3.9.1/general/options.html#scriptable-options)
* [Radar](http://docs.google.com/docs/3.9.1/charts/radar.html)

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

←  [Polar Area Chart](http://docs.google.com/docs/3.9.1/samples/scriptable/polar.html)   [Delay](http://docs.google.com/docs/3.9.1/samples/animations/delay.html)  →