 [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
  + [Events](http://docs.google.com/docs/3.9.1/samples/legend/events.html)
  + [HTML Legend](http://docs.google.com/docs/3.9.1/samples/legend/html.html)
  + [Point Style](http://docs.google.com/docs/3.9.1/samples/legend/point-style.html)
  + [Position](http://docs.google.com/docs/3.9.1/samples/legend/position.html)
  + [Alignment and Title Position](http://docs.google.com/docs/3.9.1/samples/legend/title.html)
* Title
* Subtitle
* Tooltip
* Scriptable Options
* Animations
* Advanced
* Plugins
* [Utils](http://docs.google.com/docs/3.9.1/samples/utils.html)

[**#**](#gjdgxs) HTML Legend

This example shows how to create a custom HTML legend using a plugin and connect it to the chart in lieu of the default on-canvas legend.

plugin data setup

const getOrCreateLegendList = (chart, id) => { const legendContainer = document.getElementById(id); let listContainer = legendContainer.querySelector('ul'); if (!listContainer) { listContainer = document.createElement('ul'); listContainer.style.display = 'flex'; listContainer.style.flexDirection = 'row'; listContainer.style.margin = 0; listContainer.style.padding = 0; legendContainer.appendChild(listContainer); } return listContainer; }; const htmlLegendPlugin = { id: 'htmlLegend', afterUpdate(chart, args, options) { const ul = getOrCreateLegendList(chart, options.containerID); // Remove old legend items while (ul.firstChild) { ul.firstChild.remove(); } // Reuse the built-in legendItems generator const items = chart.options.plugins.legend.labels.generateLabels(chart); items.forEach(item => { const li = document.createElement('li'); li.style.alignItems = 'center'; li.style.cursor = 'pointer'; li.style.display = 'flex'; li.style.flexDirection = 'row'; li.style.marginLeft = '10px'; li.onclick = () => { const {type} = chart.config; if (type === 'pie' || type === 'doughnut') { // Pie and doughnut charts only have a single dataset and visibility is per item chart.toggleDataVisibility(item.index); } else { chart.setDatasetVisibility(item.datasetIndex, !chart.isDatasetVisible(item.datasetIndex)); } chart.update(); }; // Color box const boxSpan = document.createElement('span'); boxSpan.style.background = item.fillStyle; boxSpan.style.borderColor = item.strokeStyle; boxSpan.style.borderWidth = item.lineWidth + 'px'; boxSpan.style.display = 'inline-block'; boxSpan.style.height = '20px'; boxSpan.style.marginRight = '10px'; boxSpan.style.width = '20px'; // Text const textContainer = document.createElement('p'); textContainer.style.color = item.fontColor; textContainer.style.margin = 0; textContainer.style.padding = 0; textContainer.style.textDecoration = item.hidden ? 'line-through' : ''; const text = document.createTextNode(item.text); textContainer.appendChild(text); li.appendChild(boxSpan); li.appendChild(textContainer); ul.appendChild(li); }); } };

const getOrCreateLegendList = (chart, id) => {  
 const legendContainer = document.getElementById(id);  
 let listContainer = legendContainer.querySelector('ul');  
 if (!listContainer) {  
 listContainer = document.createElement('ul');  
 listContainer.style.display = 'flex';  
 listContainer.style.flexDirection = 'row';  
 listContainer.style.margin = 0;  
 listContainer.style.padding = 0;  
 legendContainer.appendChild(listContainer);  
 }  
 return listContainer;  
};  
const htmlLegendPlugin = {  
 id: 'htmlLegend',  
 afterUpdate(chart, args, options) {  
 const ul = getOrCreateLegendList(chart, options.containerID);  
 // Remove old legend items  
 while (ul.firstChild) {  
 ul.firstChild.remove();  
 }  
 // Reuse the built-in legendItems generator  
 const items = chart.options.plugins.legend.labels.generateLabels(chart);  
 items.forEach(item => {  
 const li = document.createElement('li');  
 li.style.alignItems = 'center';  
 li.style.cursor = 'pointer';  
 li.style.display = 'flex';  
 li.style.flexDirection = 'row';  
 li.style.marginLeft = '10px';  
 li.onclick = () => {  
 const {type} = chart.config;  
 if (type === 'pie' || type === 'doughnut') {  
 // Pie and doughnut charts only have a single dataset and visibility is per item  
 chart.toggleDataVisibility(item.index);  
 } else {  
 chart.setDatasetVisibility(item.datasetIndex, !chart.isDatasetVisible(item.datasetIndex));  
 }  
 chart.update();  
 };  
 // Color box  
 const boxSpan = document.createElement('span');  
 boxSpan.style.background = item.fillStyle;  
 boxSpan.style.borderColor = item.strokeStyle;  
 boxSpan.style.borderWidth = item.lineWidth + 'px';  
 boxSpan.style.display = 'inline-block';  
 boxSpan.style.height = '20px';  
 boxSpan.style.marginRight = '10px';  
 boxSpan.style.width = '20px';  
 // Text  
 const textContainer = document.createElement('p');  
 textContainer.style.color = item.fontColor;  
 textContainer.style.margin = 0;  
 textContainer.style.padding = 0;  
 textContainer.style.textDecoration = item.hidden ? 'line-through' : '';  
 const text = document.createTextNode(item.text);  
 textContainer.appendChild(text);  
 li.appendChild(boxSpan);  
 li.appendChild(textContainer);  
 ul.appendChild(li);  
 });  
 }  
};

const NUM\_DATA = 7; const NUM\_CFG = {count: NUM\_DATA, min: 0, max: 100}; const data = { labels: Utils.months({count: NUM\_DATA}), datasets: [ { label: 'Dataset: 1', data: Utils.numbers(NUM\_CFG), borderColor: Utils.CHART\_COLORS.red, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.red, 0.5), fill: false, }, { label: 'Dataset: 1', data: Utils.numbers(NUM\_CFG), borderColor: Utils.CHART\_COLORS.blue, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.blue, 0.5), fill: false, }, ], };

const NUM\_DATA = 7;  
const NUM\_CFG = {count: NUM\_DATA, min: 0, max: 100};  
const data = {  
 labels: Utils.months({count: NUM\_DATA}),  
 datasets: [  
 {  
 label: 'Dataset: 1',  
 data: Utils.numbers(NUM\_CFG),  
 borderColor: Utils.CHART\_COLORS.red,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.red, 0.5),  
 fill: false,  
 },  
 {  
 label: 'Dataset: 1',  
 data: Utils.numbers(NUM\_CFG),  
 borderColor: Utils.CHART\_COLORS.blue,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.blue, 0.5),  
 fill: false,  
 },  
 ],  
};

const config = { type: 'line', data: data, options: { plugins: { htmlLegend: { // ID of the container to put the legend in containerID: 'legend-container', }, legend: { display: false, } } }, plugins: [htmlLegendPlugin], };

const config = {  
 type: 'line',  
 data: data,  
 options: {  
 plugins: {  
 htmlLegend: {  
 // ID of the container to put the legend in  
 containerID: 'legend-container',  
 },  
 legend: {  
 display: false,  
 }  
 }  
 },  
 plugins: [htmlLegendPlugin],  
};

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

* [Data structures (labels)](http://docs.google.com/docs/3.9.1/general/data-structures.html)
* [Line](http://docs.google.com/docs/3.9.1/charts/line.html)
* [Legend](http://docs.google.com/docs/3.9.1/configuration/legend.html)
  + display: false
* [Plugins](http://docs.google.com/docs/3.9.1/developers/plugins.html)

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

←  [Events](http://docs.google.com/docs/3.9.1/samples/legend/events.html)   [Point Style](http://docs.google.com/docs/3.9.1/samples/legend/point-style.html)  →