 [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
  + [Custom Tooltip Content](http://docs.google.com/docs/3.9.1/samples/tooltip/content.html)
  + [External HTML Tooltip](http://docs.google.com/docs/3.9.1/samples/tooltip/html.html)
  + [Interaction Modes](http://docs.google.com/docs/3.9.1/samples/tooltip/interactions.html)
  + [Point Style](http://docs.google.com/docs/3.9.1/samples/tooltip/point-style.html)
  + [Position](http://docs.google.com/docs/3.9.1/samples/tooltip/position.html)
* Scriptable Options
* Animations
* Advanced
* Plugins
* [Utils](http://docs.google.com/docs/3.9.1/samples/utils.html)

[**#**](#gjdgxs) External HTML Tooltip

This sample shows how to use the external tooltip functionality to generate an HTML tooltip.

config setup external

const config = { type: 'line', data: data, options: { interaction: { mode: 'index', intersect: false, }, plugins: { title: { display: true, text: 'Chart.js Line Chart - External Tooltips' }, tooltip: { enabled: false, position: 'nearest', external: externalTooltipHandler } } } };

const config = {  
 type: 'line',  
 data: data,  
 options: {  
 interaction: {  
 mode: 'index',  
 intersect: false,  
 },  
 plugins: {  
 title: {  
 display: true,  
 text: 'Chart.js Line Chart - External Tooltips'  
 },  
 tooltip: {  
 enabled: false,  
 position: 'nearest',  
 external: externalTooltipHandler  
 }  
 }  
 }  
};

const DATA\_COUNT = 7; const NUMBER\_CFG = {count: DATA\_COUNT, min: -100, max: 100, decimals: 0}; const data = { labels: Utils.months({count: DATA\_COUNT}), datasets: [ { label: 'Dataset 1', data: Utils.numbers(NUMBER\_CFG), fill: false, borderColor: Utils.CHART\_COLORS.red, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.red, 0.5), }, { label: 'Dataset 2', data: Utils.numbers(NUMBER\_CFG), fill: false, borderColor: Utils.CHART\_COLORS.blue, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.blue, 0.5), }, ] };

const DATA\_COUNT = 7;  
const NUMBER\_CFG = {count: DATA\_COUNT, min: -100, max: 100, decimals: 0};  
const data = {  
 labels: Utils.months({count: DATA\_COUNT}),  
 datasets: [  
 {  
 label: 'Dataset 1',  
 data: Utils.numbers(NUMBER\_CFG),  
 fill: false,  
 borderColor: Utils.CHART\_COLORS.red,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.red, 0.5),  
 },  
 {  
 label: 'Dataset 2',  
 data: Utils.numbers(NUMBER\_CFG),  
 fill: false,  
 borderColor: Utils.CHART\_COLORS.blue,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.blue, 0.5),  
 },  
 ]  
};

const getOrCreateTooltip = (chart) => { let tooltipEl = chart.canvas.parentNode.querySelector('div'); if (!tooltipEl) { tooltipEl = document.createElement('div'); tooltipEl.style.background = 'rgba(0, 0, 0, 0.7)'; tooltipEl.style.borderRadius = '3px'; tooltipEl.style.color = 'white'; tooltipEl.style.opacity = 1; tooltipEl.style.pointerEvents = 'none'; tooltipEl.style.position = 'absolute'; tooltipEl.style.transform = 'translate(-50%, 0)'; tooltipEl.style.transition = 'all .1s ease'; const table = document.createElement('table'); table.style.margin = '0px'; tooltipEl.appendChild(table); chart.canvas.parentNode.appendChild(tooltipEl); } return tooltipEl; }; const externalTooltipHandler = (context) => { // Tooltip Element const {chart, tooltip} = context; const tooltipEl = getOrCreateTooltip(chart); // Hide if no tooltip if (tooltip.opacity === 0) { tooltipEl.style.opacity = 0; return; } // Set Text if (tooltip.body) { const titleLines = tooltip.title || []; const bodyLines = tooltip.body.map(b => b.lines); const tableHead = document.createElement('thead'); titleLines.forEach(title => { const tr = document.createElement('tr'); tr.style.borderWidth = 0; const th = document.createElement('th'); th.style.borderWidth = 0; const text = document.createTextNode(title); th.appendChild(text); tr.appendChild(th); tableHead.appendChild(tr); }); const tableBody = document.createElement('tbody'); bodyLines.forEach((body, i) => { const colors = tooltip.labelColors[i]; const span = document.createElement('span'); span.style.background = colors.backgroundColor; span.style.borderColor = colors.borderColor; span.style.borderWidth = '2px'; span.style.marginRight = '10px'; span.style.height = '10px'; span.style.width = '10px'; span.style.display = 'inline-block'; const tr = document.createElement('tr'); tr.style.backgroundColor = 'inherit'; tr.style.borderWidth = 0; const td = document.createElement('td'); td.style.borderWidth = 0; const text = document.createTextNode(body); td.appendChild(span); td.appendChild(text); tr.appendChild(td); tableBody.appendChild(tr); }); const tableRoot = tooltipEl.querySelector('table'); // Remove old children while (tableRoot.firstChild) { tableRoot.firstChild.remove(); } // Add new children tableRoot.appendChild(tableHead); tableRoot.appendChild(tableBody); } const {offsetLeft: positionX, offsetTop: positionY} = chart.canvas; // Display, position, and set styles for font tooltipEl.style.opacity = 1; tooltipEl.style.left = positionX + tooltip.caretX + 'px'; tooltipEl.style.top = positionY + tooltip.caretY + 'px'; tooltipEl.style.font = tooltip.options.bodyFont.string; tooltipEl.style.padding = tooltip.options.padding + 'px ' + tooltip.options.padding + 'px'; };

const getOrCreateTooltip = (chart) => {  
 let tooltipEl = chart.canvas.parentNode.querySelector('div');  
 if (!tooltipEl) {  
 tooltipEl = document.createElement('div');  
 tooltipEl.style.background = 'rgba(0, 0, 0, 0.7)';  
 tooltipEl.style.borderRadius = '3px';  
 tooltipEl.style.color = 'white';  
 tooltipEl.style.opacity = 1;  
 tooltipEl.style.pointerEvents = 'none';  
 tooltipEl.style.position = 'absolute';  
 tooltipEl.style.transform = 'translate(-50%, 0)';  
 tooltipEl.style.transition = 'all .1s ease';  
 const table = document.createElement('table');  
 table.style.margin = '0px';  
 tooltipEl.appendChild(table);  
 chart.canvas.parentNode.appendChild(tooltipEl);  
 }  
 return tooltipEl;  
};  
const externalTooltipHandler = (context) => {  
 // Tooltip Element  
 const {chart, tooltip} = context;  
 const tooltipEl = getOrCreateTooltip(chart);  
 // Hide if no tooltip  
 if (tooltip.opacity === 0) {  
 tooltipEl.style.opacity = 0;  
 return;  
 }  
 // Set Text  
 if (tooltip.body) {  
 const titleLines = tooltip.title || [];  
 const bodyLines = tooltip.body.map(b => b.lines);  
 const tableHead = document.createElement('thead');  
 titleLines.forEach(title => {  
 const tr = document.createElement('tr');  
 tr.style.borderWidth = 0;  
 const th = document.createElement('th');  
 th.style.borderWidth = 0;  
 const text = document.createTextNode(title);  
 th.appendChild(text);  
 tr.appendChild(th);  
 tableHead.appendChild(tr);  
 });  
 const tableBody = document.createElement('tbody');  
 bodyLines.forEach((body, i) => {  
 const colors = tooltip.labelColors[i];  
 const span = document.createElement('span');  
 span.style.background = colors.backgroundColor;  
 span.style.borderColor = colors.borderColor;  
 span.style.borderWidth = '2px';  
 span.style.marginRight = '10px';  
 span.style.height = '10px';  
 span.style.width = '10px';  
 span.style.display = 'inline-block';  
 const tr = document.createElement('tr');  
 tr.style.backgroundColor = 'inherit';  
 tr.style.borderWidth = 0;  
 const td = document.createElement('td');  
 td.style.borderWidth = 0;  
 const text = document.createTextNode(body);  
 td.appendChild(span);  
 td.appendChild(text);  
 tr.appendChild(td);  
 tableBody.appendChild(tr);  
 });  
 const tableRoot = tooltipEl.querySelector('table');  
 // Remove old children  
 while (tableRoot.firstChild) {  
 tableRoot.firstChild.remove();  
 }  
 // Add new children  
 tableRoot.appendChild(tableHead);  
 tableRoot.appendChild(tableBody);  
 }  
 const {offsetLeft: positionX, offsetTop: positionY} = chart.canvas;  
 // Display, position, and set styles for font  
 tooltipEl.style.opacity = 1;  
 tooltipEl.style.left = positionX + tooltip.caretX + 'px';  
 tooltipEl.style.top = positionY + tooltip.caretY + 'px';  
 tooltipEl.style.font = tooltip.options.bodyFont.string;  
 tooltipEl.style.padding = tooltip.options.padding + 'px ' + tooltip.options.padding + 'px';  
};

## [**#**](#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)
* [Tooltip](http://docs.google.com/docs/3.9.1/configuration/tooltip.html)
  + [External (Custom) Tooltips](http://docs.google.com/docs/3.9.1/configuration/tooltip.html#external-custom-tooltips)

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

←  [Custom Tooltip Content](http://docs.google.com/docs/3.9.1/samples/tooltip/content.html)   [Interaction Modes](http://docs.google.com/docs/3.9.1/samples/tooltip/interactions.html)  →