Chart.js

❮ Previous Next ❯

Chart.js is an free JavaScript library for making HTML-based charts. It is one of the simplest visualization libraries for JavaScript, and comes with the many built-in chart types:

- Scatter Plot
- Line Chart
- Bar Chart
- Pie Chart
- Donut Chart
- Bubble Chart
- Area Chart
- Radar Chart
- Mixed Chart

How to Use Chart.js?

1. Add a link to the providing CDN (Content Delivery Network):

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.4/Chart.js">
</script>
```

2. Add a <canvas> to where in the HTML you want to draw the chart:

```
<canvas id="myChart" style="width:100%;max-width:700px"></canvas>
```

The canvas element must have a unique id.

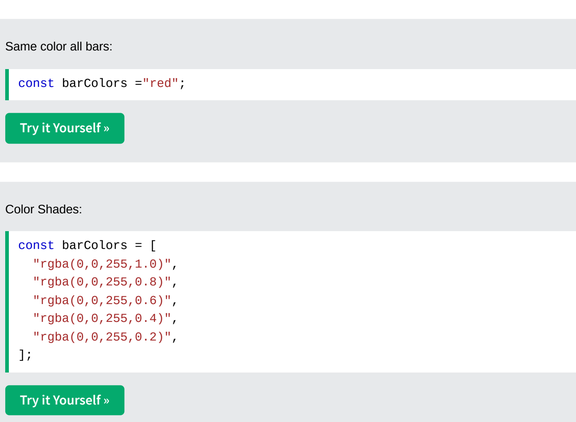
Typical Bar Chart Syntax:

```
const myChart = new Chart("myChart", {
  type: "bar",
  data: {},
  options: {}
});
```

Typical Line Chart Syntax:

```
const myChart = new Chart("myChart", {
  type: "line",
  data: {},
  options: {}
});
```

Bar Charts



Source Code

```
const xValues = ["Italy", "France", "Spain", "USA", "Argentina"];
const yValues = [55, 49, 44, 24, 15];
const barColors = ["red", "green","blue","orange","brown"];

new Chart("myChart", {
  type: "bar",
  data: {
    labels: xValues,
    datasets: [{
      backgroundColor: barColors,
      data: yValues
    }]
  },
  options: {...}
});
```

Try it Yourself »

Color only one bar:

```
const barColors = ["blue"];
```

Try it Yourself »

Same color all bars:

```
const barColors ="red";
```

Try it Yourself »

Color Shades:

```
const barColors = [
  "rgba(0,0,255,1.0)",
  "rgba(0,0,255,0.8)",
  "rgba(0,0,255,0.6)",
  "rgba(0,0,255,0.4)",
  "rgba(0,0,255,0.2)",
];
```

Try it Yourself »

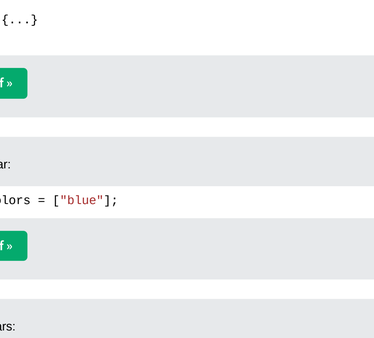
Horizontal Bars

Just change type from "bar" to "horizontalBar":

```
type: "horizontalBar",
```

Try it Yourself »

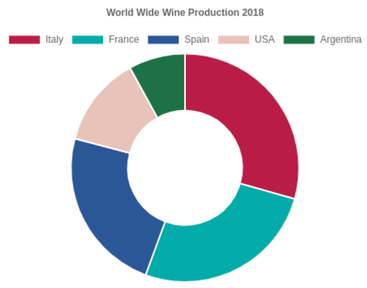
Pie Charts



Example

```
new Chart("myChart", {
  type: "pie",
  data: {
    labels: xValues,
    datasets: [{
      backgroundColor: barColors,
      data: yValues
    }]
  },
  options: {
    title: {
      display: true,
      text: "World Wide Wine Production"
    }
  }
});
```

Doughnut Charts



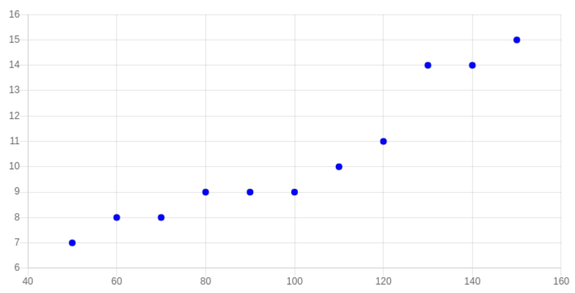
Just change type from "pie" to "doughnut":

```
type: "doughnut";
```

Try it Yourself »

Scatter Plots

House Prices vs. Size



Source Code

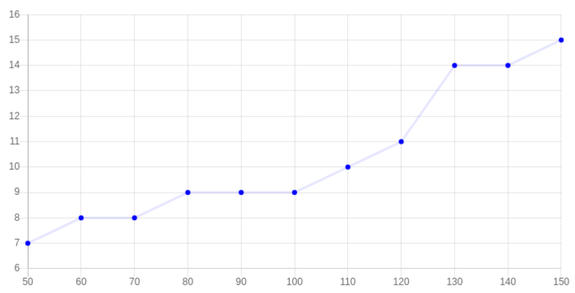
```
const xyValues = [
  {x:50, y:7},
  {x:60, y:8},
  {x:70, y:8},
  {x:80, y:9},
  {x:90, y:9},
  {x:100, y:9},
  {x:110, y:10},
  {x:120, y:11},
  {x:130, y:14},
  {x:140, y:14},
  {x:150, y:15}
];

new Chart("myChart", {
  type: "scatter",
  data: {
    datasets: [{
      pointRadius: 4,
      pointBackgroundColor: "rgba(0,0,255,1)",
      data: xyValues
    }]
  },
  options:{...}
});
```

Try it Yourself »

Line Graphs

House Prices vs. Size



Source Code

```
const xValues = [50,60,70,80,90,100,110,120,130,140,150];
const yValues = [7,8,8,9,9,9,10,11,14,14,15];

new Chart("myChart", {
  type: "line",
  data: {
    labels: xValues,
    datasets: [{
      backgroundColor:"rgba(0,0,255,1.0)",
      borderColor: "rgba(0,0,255,0.1)",
      data: yValues
    }]
  },
  options:{...}
});
```

Try it Yourself »

If you set the borderColor to zero, you can **scatter plot** the line graph:

```
borderColor: "rgba(0,0,0,0)",
```

Try it Yourself »

Multiple Lines

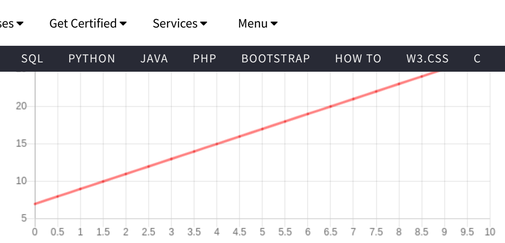
Source Code

```
const xValues = [100,200,300,400,500,600,700,800,900,1000];

new Chart("myChart", {
  type: "line",
  data: {
    labels: xValues,
    datasets: [{
      data: [860,1140,1060,1060,1070,1110,1330,2210,7830,2478],
      borderColor: "red",
      fill: false
    },{
      data: [1600,1700,1700,1900,2000,2700,4000,5000,6000,7000],
      borderColor: "green",
      fill: false
    },{
      data: [300,700,2000,5000,6000,4000,2000,1000,200,100],
      borderColor: "blue",
      fill: false
    }]
  },
  options: {
    legend: {display: false}
  }
});
```

Try it Yourself »

Linear Graphs



Source Code

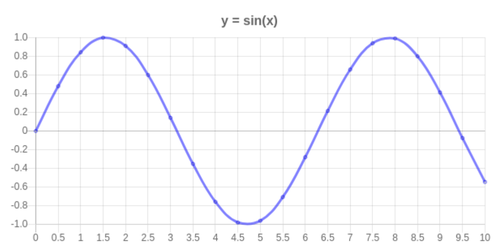
```
const xValues = [];
const yValues = [];
generateData("x * 2 + 7", 0, 10, 0.5);

new Chart("myChart", {
  type: "line",
  data: {
    labels: xValues,
    datasets: [{
      fill: false,
      pointRadius: 1,
      borderColor: "rgba(255,0,0,0.5)",
      data: yValues
    }]
  },
  options: {...}
});

function generateData(value, i1, i2, step = 1) {
  for (let x = i1; x <= i2; x += step) {
    yValues.push(eval(value));
    xValues.push(x);
  }
}
```

Try it Yourself »

Function Graphs



Same as Linear Graph. Just change the generateData parameter(s):

```
generateData("Math.sin(x)", 0, 10, 0.5);
```

Try it Yourself »

❮ Previous
Log in to track progress
Next ❯