

Chart.js

数行のコードで折れ線グラフや棒グラフ、円グラフやレーダーチャートなどを動的に表示することができる JavaScript ライブラリ。

グラフを作成できる JavaScript ライブラリとしては D3.js が有名。D3.js は複雑なグラフを描画できるが、グラフの描画処理を自分でコーディングしなければならないので、JavaScript 初級者には使うのが難しい。

Chart.js は D3.js のような複雑な表示をすることはできないが、グラフやチャートの作成に特化しているので、初心者も簡単に扱える。

Chart.js は様々な種類のチャート描画に対応している。

- ・ 折れ線グラフ
- ・ 棒グラフ
- ・ レーダーチャート
- ・ 極性面グラフ (鶏頭図)
- ・ 円グラフ・ドーナツ型チャート
- ・ バブルチャート

グラフを作るだけなら、Microsoft Excel でもできるが、Chart.js を使えば、データを自動更新して、ウェブ公開できるというメリットがある。

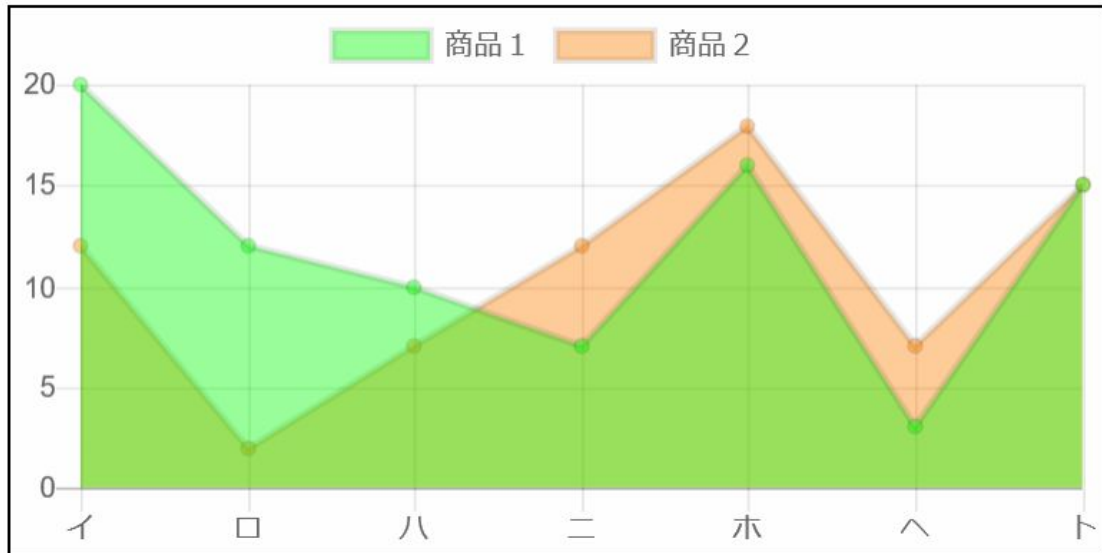
棒グラフ

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script src="Chart.min.js"></script>
  </head>
  <body>
    <canvas id="canvas" width=450 style="border:solid 1px"></canvas>
    <script>
var canvas = document.getElementById('canvas');
var ctx = canvas.getContext('2d');
var myChart = new Chart(ctx, {
  options: { responsive: false },
  type: 'bar',
  data: {
    labels: ['イ', 'ロ', 'ハ', 'ニ', 'ホ', 'ヘ', 'ト'],
    datasets: [{
      label: '商品1',
      data: [20, 12, 10, 7, 16, 3, 15],
      backgroundColor: "rgba(0,255,0,0.4)"
    }, {
      label: '商品2',
      data: [12, 2, 7, 12, 18, 7, 15],
      backgroundColor: "rgba(255,127,0,0.4)"
    }
  ]
});
    </script>
  </body>
</html>
```



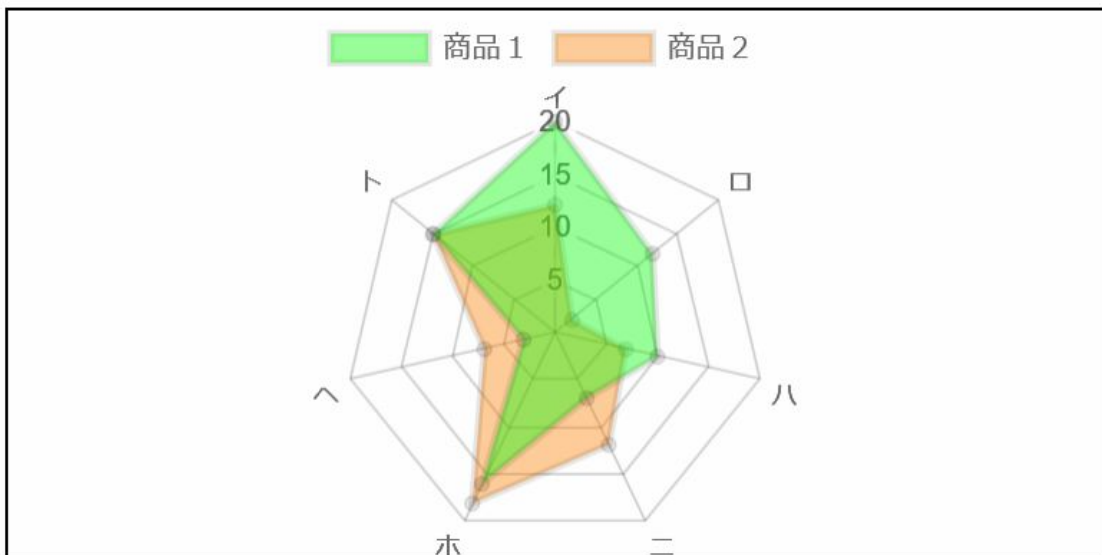
折れ線グラフ

type: 'bar' を type: 'line' に変えるだけ。label: '商品1/2', の後に lineTension: 0, を入れる。入れないと滑らかな曲線でつながることになる。



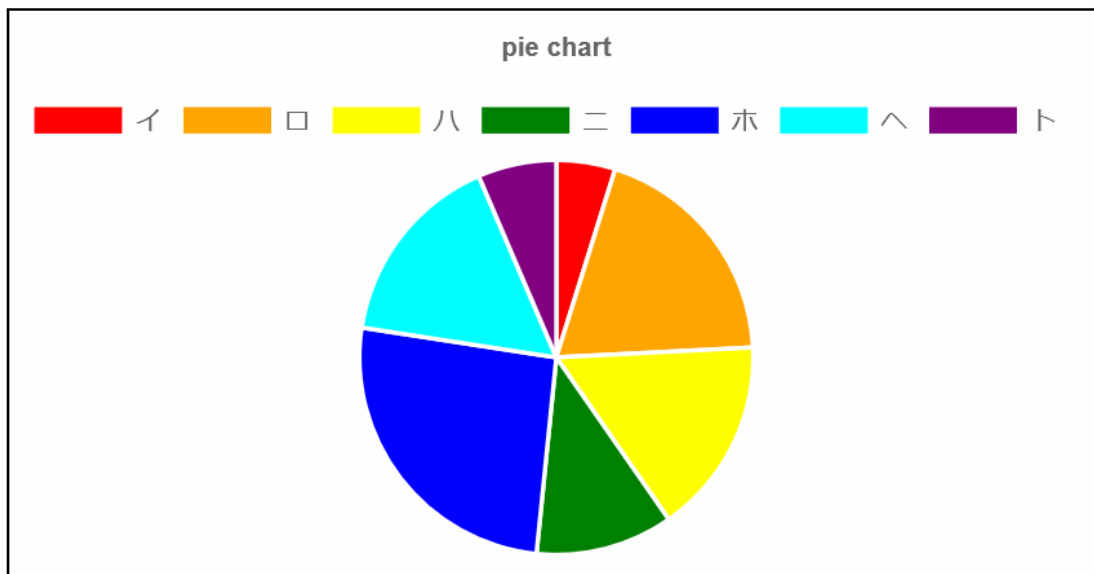
レーダーチャート

type: 'radar' に変えるだけ。



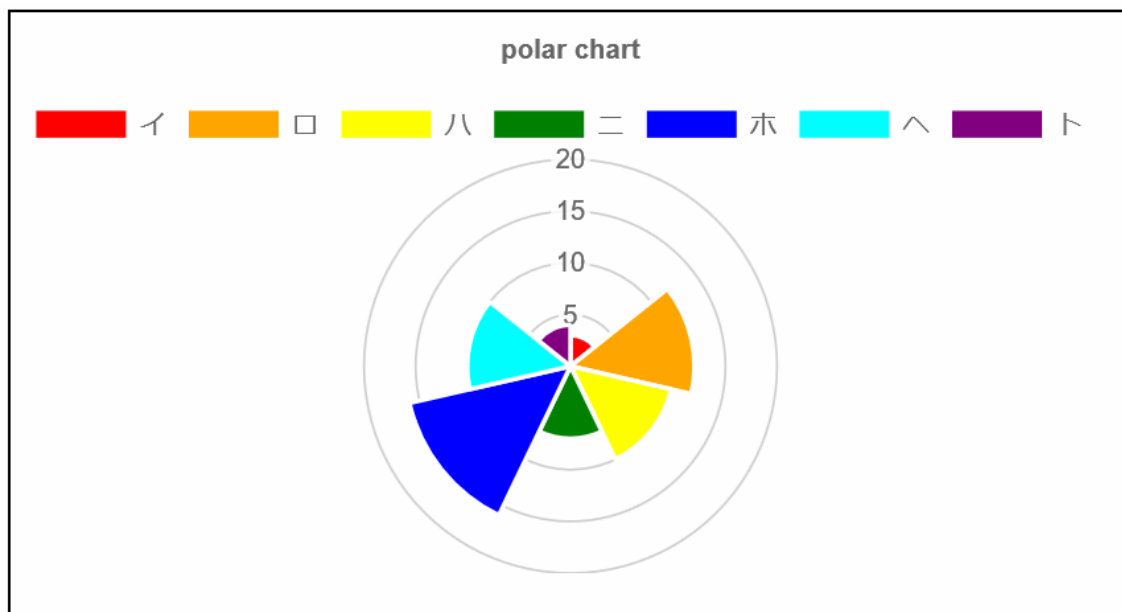
円グラフ

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script src="Chart.min.js"></script>
  </head>
  <body>
    <div style="width:500px; height:260px; border:solid 1px">
      <canvas id="canvas" width=500 height=250></canvas>
    </div>
    <script>
var canvas = document.getElementById('canvas');
var ctx = canvas.getContext('2d');
var myChart = new Chart(ctx, {
  options: { responsive: false,
            title: { display: true, text: 'pie chart' }
  },
  type: 'pie',
  data: {
    labels: ['イ', 'ロ', 'ハ', 'ニ', 'ホ', 'ヘ', 'ト'],
    datasets: [{
      data: [3, 12, 10, 7, 16, 10, 4],
      backgroundColor: [
        "red", "orange", "yellow", "green", "blue", "cyan", "purple" ]
    }]
  }
});
    </script>
  </body>
</html>
```



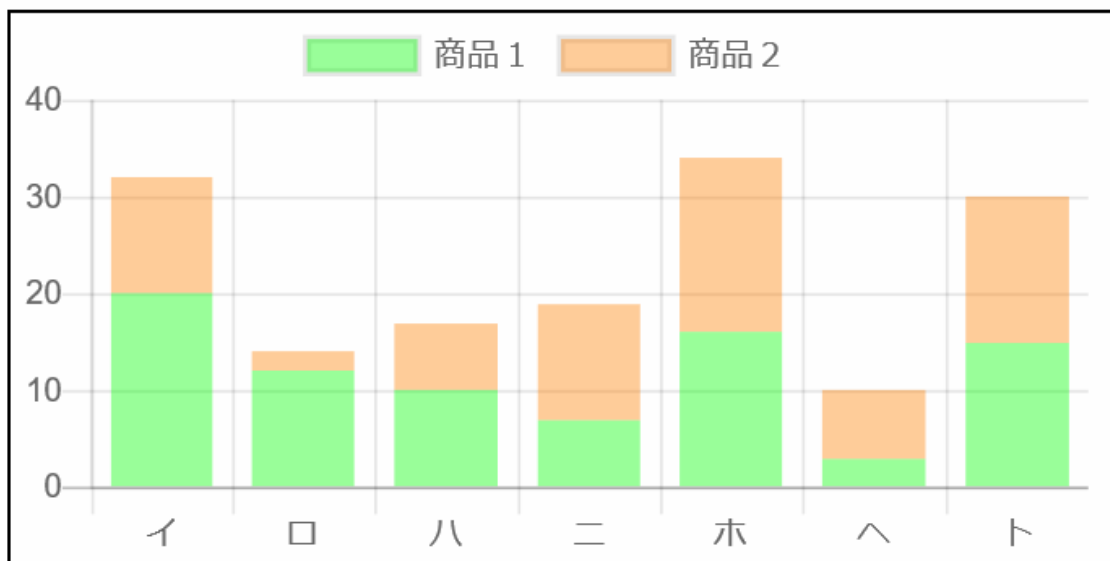
ポーラーチャート

前記のプログラムで type: 'polarArea' に変えるだけ。



積み上げグラフ

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script src="Chart.min.js"></script>
  </head>
  <body>
    <div style="width:400px; height:200px; border:solid 1px">
      <canvas id="canvas" width=400 height=200></canvas>
    </div>
    <script>
var canvas = document.getElementById('canvas');
var ctx = canvas.getContext('2d');
var myChart = new Chart(ctx, {
  type: 'bar',
  options: { responsive: false,
            scales: { xAxes:[{ stacked: true }],
                    yAxes:[{ stacked: true }]} },
  data: {
    labels: ['イ', 'ロ', 'ハ', 'ニ', 'ホ', 'ヘ', 'ト'],
    datasets: [{
      label: '商品1',
      data: [20, 12, 10, 7, 16, 3, 15],
      backgroundColor: "rgba(0,255,0,0.4)"
    }, {
      label: '商品2',
      data: [12, 2, 7, 12, 18, 7, 15],
      backgroundColor: "rgba(255,127,0,0.4)"
    }
  ]
}
});
    </script>
  </body>
</html>
```



グラフの各部の色や動作は、よりきめ細かく指定することもできる。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <script src="Chart.min.js"></script>
  </head>
  <body>
    <div style="width:400px; height:230px; border:solid 1px">
      <canvas id="canvas" width="400" height="220"></canvas>
    </div>
    <script>
var ctx = document.getElementById("canvas");
var myChart = new Chart(ctx, {
  type: 'line',
  data: {
    labels: ["20011 年", "2012 年", "2013 年", "2014 年", "2015 年", "2016 年"],
    datasets: [
      {
        label: "商品 1",
        fill: false,
        lineTension: 0,
        backgroundColor: "rgba(255,0,0,0.2)",
        borderColor: "rgba(255,0,0,1)",
        pointBorderColor: "rgba(255,0,0,1)",
        pointBackgroundColor: "#fff",
        pointRadius: 5,
        pointHoverRadius: 8,
        pointHoverBackgroundColor: "rgba(255,0,0,1)",
        pointHoverBorderColor: "rgba(255,0,0,1)",
        pointHitRadius: 15,
        data: [12, 19, 3, 5, 2, 3]
      },
      {
        label: "商品 2",
        fill: false,
        lineTension: 0,
        backgroundColor: "rgba(0,0,255,0.4)",
        borderColor: "rgba(0,0,255,1)",
        pointBorderColor: "rgba(0,0,255,1)",
        pointBackgroundColor: "#fff",
        pointRadius: 5,
        pointHoverRadius: 8,
        pointHoverBackgroundColor: "rgba(0,0,255,1)",
        pointHoverBorderColor: "rgba(0,0,255,1)",
        pointHitRadius: 10,
        data: [15, 15, 6, 8, 5, 6]
      }
    ]
  },
  options: {
    responsive: false,
    scales: {
      yAxes: [{
        ticks: {
          beginAtZero: true
        }
      }]
    },
    hover: {
      mode: 'single' //ホバ一時の動作 (single, label, dataset)
    }
  }
});
    </script>
  </body>
</html>
```

