ECA
CHECK-IN

Welcome to Hack Club!

Workshop: Charts with Chart.js

10/25/2021

BENEFITS

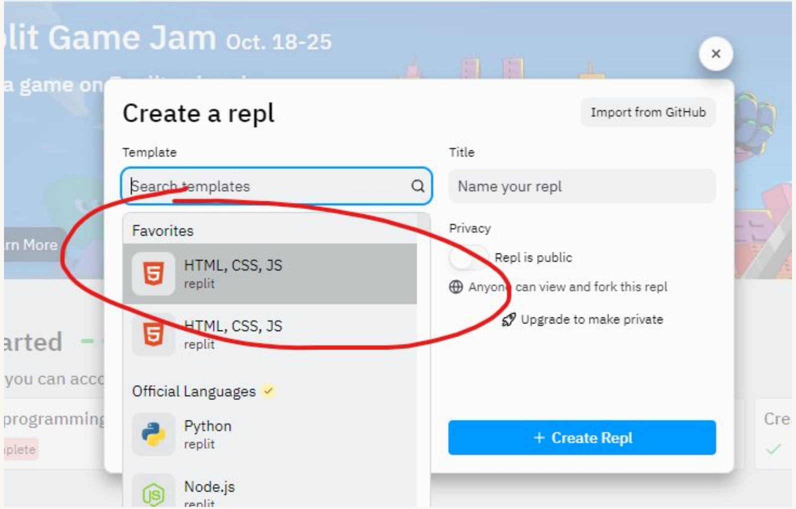
Chart.js

Objective

- Make a chart generator with chart.js
- Use our knowledge of HTML & CSS

Setting Up

Create a new repl (we're switching back)



Create a repl

Template

Search templates

Title

Name your repl

Privacy

Repl is public

Anyone can view and fork this repl

Upgrade to make private

Adding Libraries

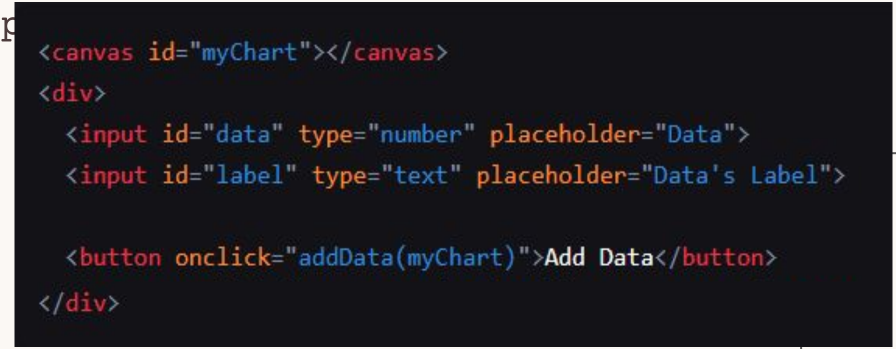
- Add the following line in the <head> to import chart.js

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

Setup Basic HTML

We will illustrate our charts on the <canvas> element.

1. Create a **canvas** element in the <body></body> tags and give it an id
2. Then, create a **div** container with 2 input fields and one button
   - Data value and label (inp
   - Submit (button)



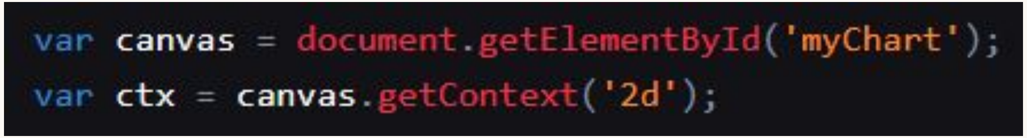
```html
<canvas id="myChart"></canvas>
<div>
  <input id="data" type="number" placeholder="Data">
  <input id="label" type="text" placeholder="Data's Label">

  <button onclick="addData(myChart)">Add Data</button>
</div>
```

Moving onto JavaScript

To be able to draw on the canvas (that we added in HTML), we have to “grab” it in JavaScript using the “id” that we assigned.

1. Grab the canvas element using **getElementById**
2. Setup the 2D context of the canvas using **.getContext(“2d”)**



```
var canvas = document.getElementById('myChart');
var ctx = canvas.getContext('2d');
```

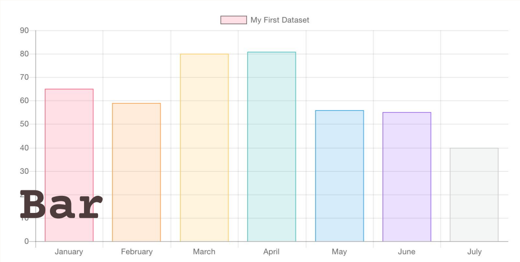
Starting with chart.js

- Creating the Chart in chart.js
- On the right is schema for defining chart
- We pass in the chart canvas(ctx)

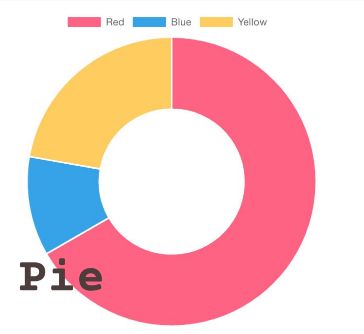
```
var myChart = new Chart(ctx, {
  type: 'bar',
  data: {
    datasets: [{
      label: '',
      backgroundColor: [
        'rgba(255, 99, 132, 0.2)',
        'rgba(54, 162, 235, 0.2)',
        'rgba(255, 206, 86, 0.2)',
        'rgba(75, 192, 192, 0.2)',
        'rgba(153, 102, 255, 0.2)',
        'rgba(255, 159, 64, 0.2)'
      ],
      borderColor: [
        'rgba(255, 99, 132, 1)',
        'rgba(54, 162, 235, 1)',
        'rgba(255, 206, 86, 1)',
        'rgba(75, 192, 192, 1)',
        'rgba(153, 102, 255, 1)',
        'rgba(255, 159, 64, 1)'
      ],
      borderWidth: 1
    }]
  }
});
```

Types of charts

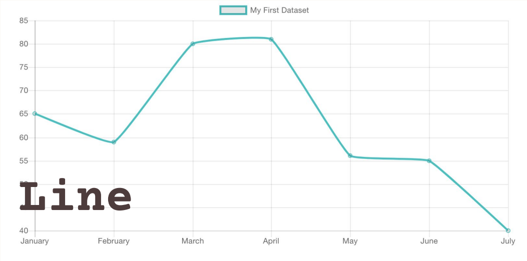
Chart.js offers 7 built-in chart types (you can make your own custom chart type too!)



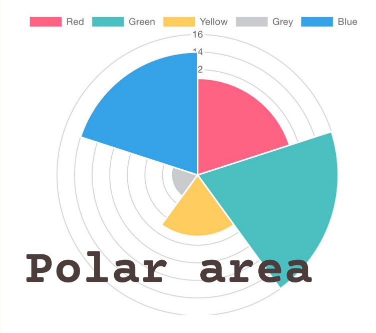
Bar



Pie



Line

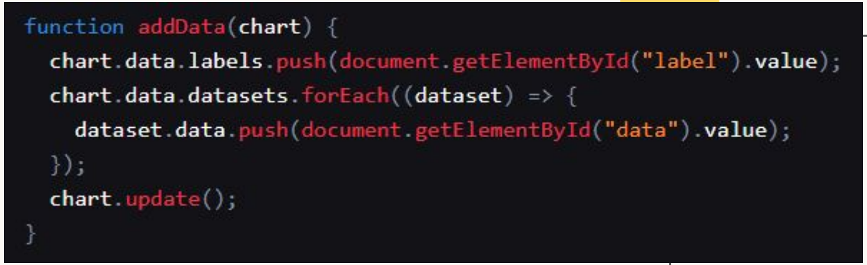


Polar area

Adding data to the chart

Now, we have our chart, but we need to have a way to add data.

1. Create an “addData” function that retrieves the input field data and pushes them to the chart
    - chart.data.labels.push()
        - Adds a new label to the chart
    - chart.data.datasets.forEach()
        - Looping through all the datasets and pushes the new data value into each dataset
    - chart.update()
        - Updates the chart data to the canvas



```
function addData(chart) {
  chart.data.labels.push(document.getElementById("label").value);
  chart.data.datasets.forEach((dataset) => {
    dataset.data.push(document.getElementById("data").value);
  });
  chart.update();
}
```

Hacking time

Possible Improvements

Delete data

Add a button that deletes a data value from the chart.

Save as image

Add a button that saves and downloads the chart as an image.

Fetch data from an external API

Collect data from an external API and visualize it with a chart in Chart.js! This is a little advanced, but here is an article to help.