

## Utilisation de la librairie ChartJS

### Présentation :

Chart.js est une bibliothèque JavaScript populaire et gratuite pour la visualisation de données. Elle permet de créer des graphiques interactifs et personnalisables pour les applications web. Voici une présentation détaillée de cette bibliothèque :

### **Caractéristiques**

- Types de graphiques : Chart.js offre une variété de types de graphiques, tels que des graphiques en ligne, des graphiques en barres, des graphiques en secteurs, des graphiques en radar, des graphiques en bulles, des graphiques en surface, des graphiques en étoile, des graphiques mixtes, etc.
- Intégration avec HTML : Chart.js utilise l'élément `<canvas>` pour afficher les graphiques, ce qui signifie qu'il est facile à intégrer dans des pages web.
- Personnalisation : La bibliothèque offre de nombreuses options de personnalisation, telles que la modification des couleurs, des polices, des bordures, des animations, etc.
- Responsivité : Les graphiques créés avec Chart.js sont entièrement responsives, ce qui signifie qu'ils s'adaptent automatiquement à la taille de l'écran.
- Support des navigateurs modernes : Chart.js est compatible avec tous les navigateurs modernes, y compris Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, etc.
- Animation : Les graphiques peuvent être animés pour offrir une expérience utilisateur plus interactive.
- Interactions : Les utilisateurs peuvent interagir avec les graphiques en passant la souris sur les éléments, en cliquant sur les éléments, etc.
- Légende et tooltips : Les graphiques peuvent inclure des légendes et des tooltips pour fournir des informations supplémentaires sur les données.
- Mise à jour dynamique : Les données peuvent être mises à jour dynamiquement pour afficher des informations en temps réel.

### Installation :

1. Ajoutez le lien suivant dans la section `<head>` de votre fichier HTML :

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

2. Création du Canvas

Créez un élément `<canvas>` dans votre fichier HTML où vous souhaitez afficher le graphique :

```
<div>  
    <canvas id="myChart"></canvas>  
</div>
```

### 3. Configuration du Graphique

Créez un script pour configurer et afficher le graphique :

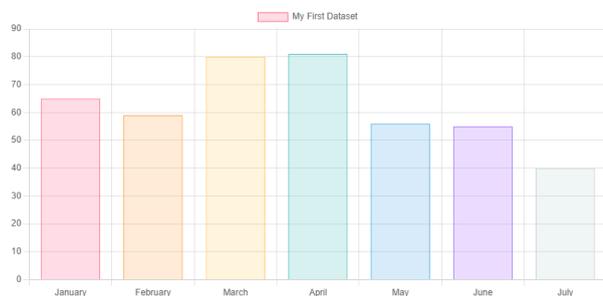
```
<script>
  const ctx = document.getElementById('myChart').getContext('2d');
  new Chart(ctx, {
    type: 'bar',
    data: {
      labels: ['Red', 'Blue', 'Yellow', 'Green', 'Purple',
'Orange'],
      datasets: [{
        label: '# of Votes',
        data: [12, 19, 3, 5, 2, 3],
        borderWidth: 1
      }]
    },
    options: {
      scales: {
        y: {
          beginAtZero: true
        }
      }
    }
  });
</script>
```

### 4. Affichage du Graphique

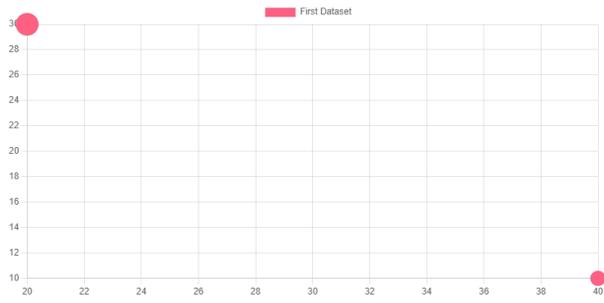
Exécutez le code et vous devriez voir un graphique en barres avec les données spécifiées. Vous pouvez personnaliser les options et les données pour adapter le graphique à vos besoins.

Voici les différents types de graphiques possibles à afficher :

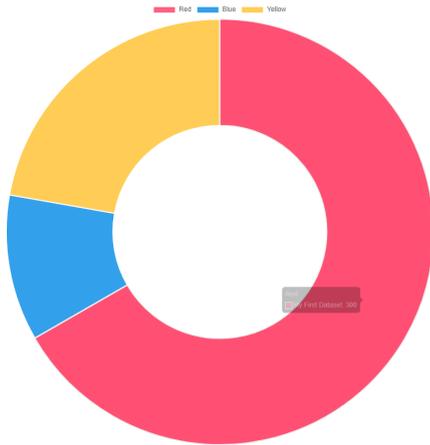
#### Bar Chart :



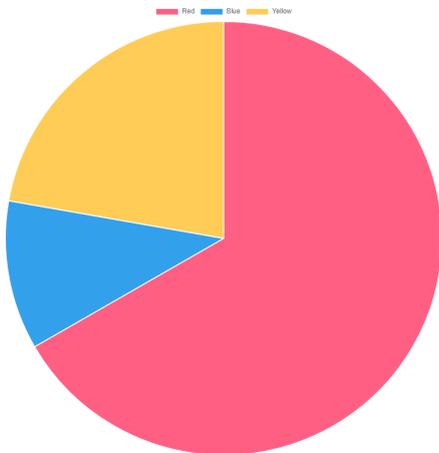
**Bubble Chart :**



**Doughnut :**



**Pie :**



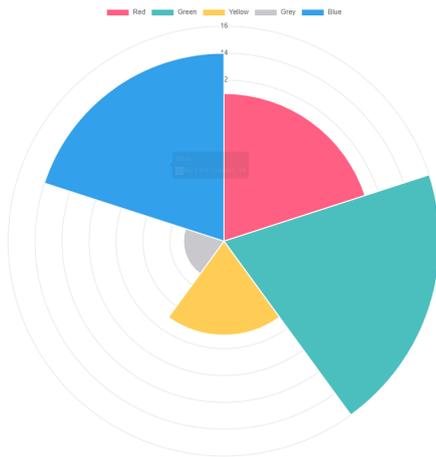
**Line Chart :**



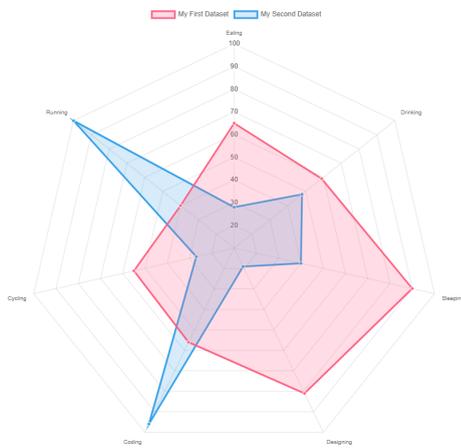
**Mixed Chart Types :**



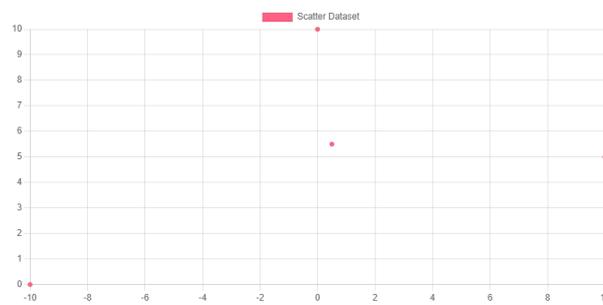
**Polar Area Chart :**



**Radar Chart :**



**Scatter Chart :**



## Résumé :

Chart.js est une bibliothèque JavaScript puissante et flexible pour la visualisation de données. Elle offre une variété de types de graphiques, de nombreuses options de personnalisation, et une intégration facile avec les applications web. Avec sa communauté active et ses ressources abondantes, Chart.js est un excellent choix pour les développeurs qui cherchent à créer des graphiques interactifs et attractifs.