Visualisation de données avec Chart.js

Chart.js est une bibliothèque open-source permettant visualiser facilement les données en utilisant JavaScript dans des pages web HTML5 . Il prend en charge 8 types de graphiques réactifs différents (graphique à barres,lignes, camembert, graphique à courbes, etc.) avec des couleurs et des données configurées pour créer des graphiques.


Diagramme à bandes


Graphique en ligne


Camembert


Diagramme radar


Zone polaire


Tableau de beignets


Barres horizontales


Barres groupées


Tableaux mixtes


Graphique à bulles

Pour dessiner un graphique avec Chart.js:

1. Définissez *où* sur votre page pour dessiner le graphique.
2. Définissez le *type de graphique que* vous voulez dessiner.
3. *Fournissez la bibilothèque Chart.js avec des données* , des étiquettes et d'autres options.

Étape 1: Ajouter Chart.js dans la page html avec le style css

```
<head>

 <script src= "Chart.min.js" ></script>

<link rel= "stylesheet" type= "text/css" href= "style.css" >

</head>
```

Étape 2: Préparez une place dans votre code HTML pour afficher le graphique

Pour Chart.js , on ajoute un élément <canvas> (canevas HTML5) en définissant les attributs width et la height pour définir les proportions de votre graphique.

<canvas id= "myChart" width= "1600" height= "900" ></canvas>

Étape 3: Préparer les données

Nous allons créer un graphique linéaire réactif simple visualisant la population mondiale au cours des 500 dernières années, et une prédiction pour 2050:

Populations historiques et prédites dans le monde (en millions)

Pays	**1500**	**1600**	**1700**	**1750**	**1800**	**1850**	**1900**	**1950**	**1999**	**2050**
Afrique	86	114	106	106	107	111	133	221	783	2478
Asie	282	350	411	502	635	809	947	1402	3700	5267
·L'Europe	168	170	178	190	203	276	408	547	675	734
Amérique latine	40	20	dix	16	24	38	74	167	508	784
Amérique du Nord	6	3	2	2	7	26	82	172	312	433

Nous allons utiliser 6 tableaux au total: un pour toutes les étiquettes de l'année à afficher le long de l'axe X (1500-2050), et un tableau pour chaque pays, contenant les données sur la population.

script.js .

// les labels le long des axes X
var years = [1500 , 1600 , 1700 , 1750 , 1800 , 1850 , 1900 , 1950 , 1999 , 2050];
// For drawing the lines
var africa = [86 , 114 , 106 , 106 , 107 , 111 , 133 , 221 , 783 , 2478]; var asia = [282 , 350 , 411 , 502 , 635 , 809 , 947 , 1402 , 3700 , 5267];
var europe = [168 , 170 , 178 , 190 , 203 , 276 , 408 , 547 , 675 , 734]; var latinAmerica = [40 , 20 , 10 , 16 , 24 , 38 , 74 , 167 , 508 , 784]; var northAmerica = [6 , 3 , 2 , 2 , 7 , 26 , 82 , 172 , 312 , 433];

Étape 4: Tracez une ligne!

Dans le fichier script.js , ajoutez ces lignes de JavaScript:

var ctx = document.getElementById ("myChart") *.getContext("2d");*
ou bien
var ctx = document.getElementById ("myChart") ;

Syntaxe générale pour dessiner un graphique

```
var myChart = new Chart(ctx, {
  type: //chart type,
  data: // chart data,
  options: // chart options  facultatif
});
```

Cas d'une ligne :

var myChart = new Chart (ctx , { type : 'line' , data : { labels : years , datasets : [{ data : africa }] } });
// uniquement données africa pour tracer la ligne.

Étape 5: Style de la ligne

On peut donner un nom (label) à la première ligne en ajoutant après les data: africa , label: "Africa" :

{ data: africa, label: "Africa" }

Pour définir la couleur de la bordure et supprimer la grande zone grise sous le graphique, ajoutez une autre virgule après le label: "Africa" et ajoutez ces deux lignes:

borderColor: "#3e95cd", fill: false

Étape 6: Ajouter le reste des données

Pour visualise la population de différentes régions, on peut utiliser *toutes* nos variables de région (asia , europe , etc.), et nommez les lignes en conséquence.

{ data : africa , label : "Africa" , borderColor : "#3e95cd" , fill : false },

{ data : asia , label : "Asia" , borderColor : "#3e95cd" , fill : false },

{ data : europe , label : "Europe" , borderColor : "#3e95cd" , fill : false },

{ data : latinAmerica , label : "Latin America" , borderColor : "#3e95cd" , fill : false },

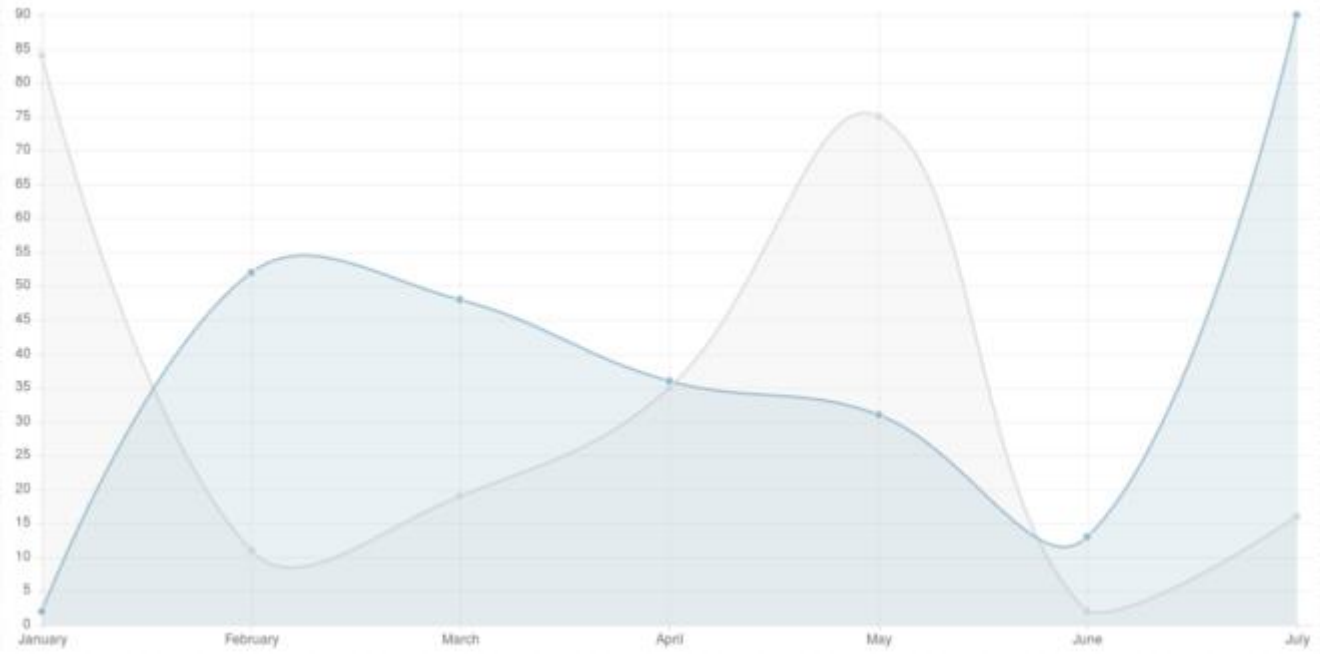
{ data : northAmerica , label : "North America" , borderColor : "#3e95cd" , fill : false }

Changez la valeur borderColor pour chacune des nouvelles régions: #8e5ea2 , #3cba9f , #e8c3b9 , #c45850 .

1. Graphique linéaire (Linear Chart)

Les graphiques line sont créés en définissant le type sur la line . Par défaut, les lignes sont pourvues d'un remplissage transparent foncé couvrant la zone située entre la ligne et l'axe des x. (fill: false).

Si vous voulez supprimer des remplissages pour tous vos graphiques linéaires, il suffit de changer la valeur par défaut globale pour les remplissages: Chart.defaults.global.elements.line.fill = false; .



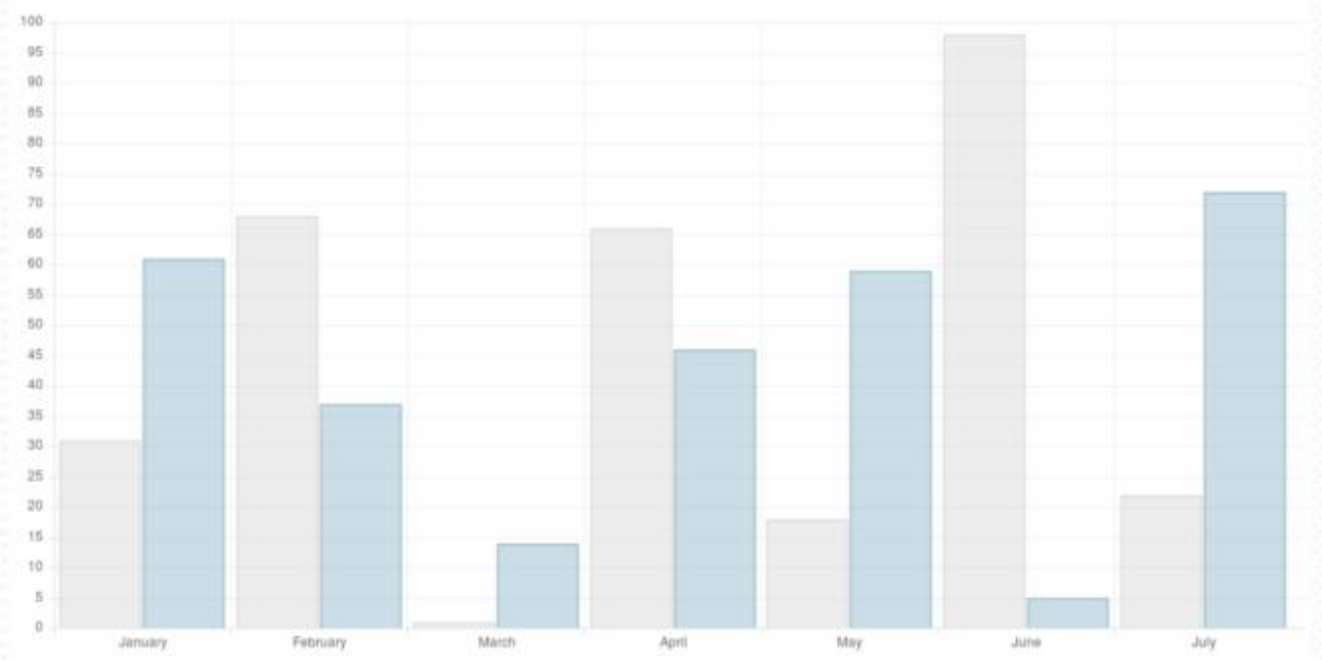
Exemple de code pour créer le graphique :

```
<canvas id="line-chart" width="800" height="450"></canvas>
new Chart(document.getElementById("line-chart"), {
  type: 'line',
  data: {
    labels: [1500,1600,1700,1750,1800,1850,1900,1950,1999,2050],
    datasets: [{
        data: [86,114,106,106,107,111,133,221,783,2478],
        label: "Africa",
        borderColor: "#3e95cd",
        fill: false
      }, {
        data: [282,350,411,502,635,809,947,1402,3700,5267],
        label: "Asia",
        borderColor: "#8e5ea2",
        fill: false
      }, {
        data: [168,170,178,190,203,276,408,547,675,734],
        label: "Europe",
        borderColor: "#3cba9f",
        fill: false
      }, {
        data: [40,20,10,16,24,38,74,167,508,784],
        label: "Latin America",
        borderColor: "#e8c3b9",
        fill: false
      }, {
        data: [6,3,2,2,7,26,82,172,312,433],
        label: "North America",
        borderColor: "#c45850",
        fill: false
      }
    ]
  },
```

```
  options: {
    title: {
      display: true,
      text: 'World population per region (in millions)'
    }
  }
});
```

2. Diagramme à barres (BarChart)

Les graphiques à barres sont créés en définissant le type bar. Les couleurs des barres sont définies en passant une couleur à backgroundColor (toutes les barres auront la même couleur), ou un tableau de couleurs.



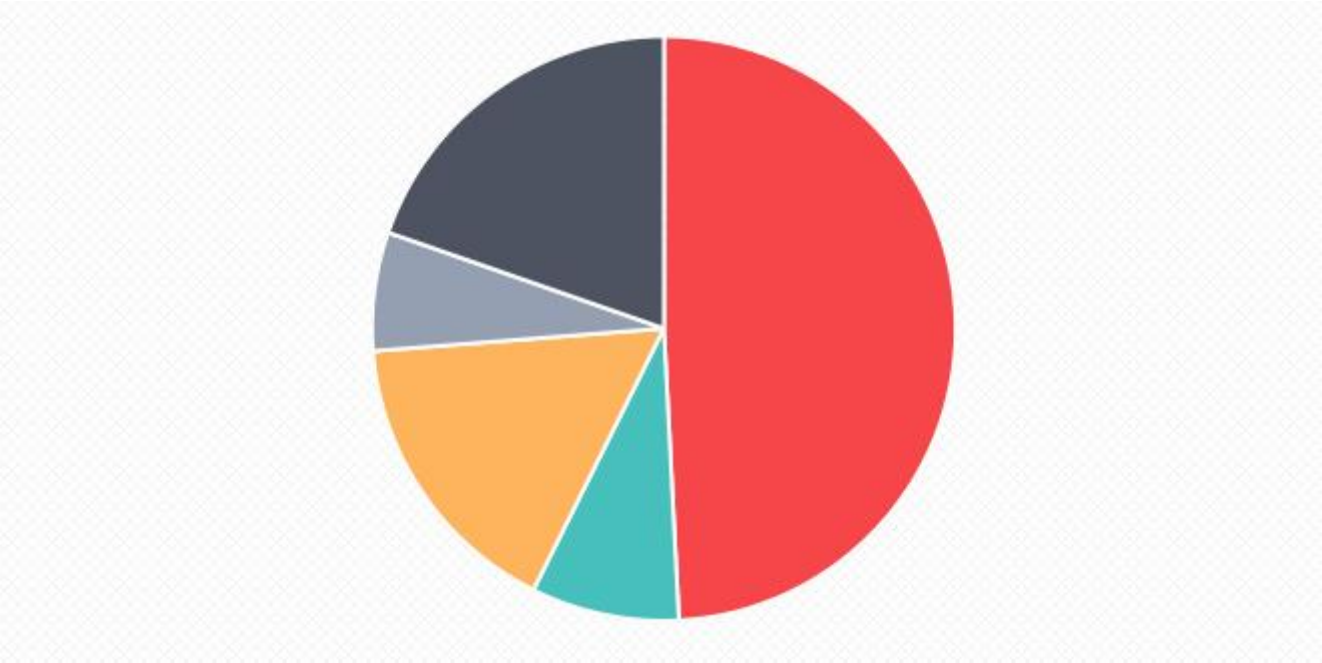
Exemple de code pour créer le graphique :

```
<canvas id="bar-chart" width="800" height="450"></canvas>
// Bar chart
new Chart(document.getElementById("bar-chart"), {
    type: 'bar',
    data: {
      labels: ["Africa", "Asia", "Europe", "Latin America", "North America"],
      datasets: [
        {
          label: "Population (millions)",
          backgroundColor: ["#3e95cd", "#8e5ea2","#3cba9f","#e8c3b9","#c45850"],
          data: [2478,5267,734,784,433]
        }
      ]
    },
    options: {
```

```
    legend: { display: false },
    title: {
      display: true,
      text: 'Predicted world population (millions) in 2050'
    }
  }
});
```

3. Diagramme à secteurs (PieChart)

Les pie sont créés en définissant le type sur pie .

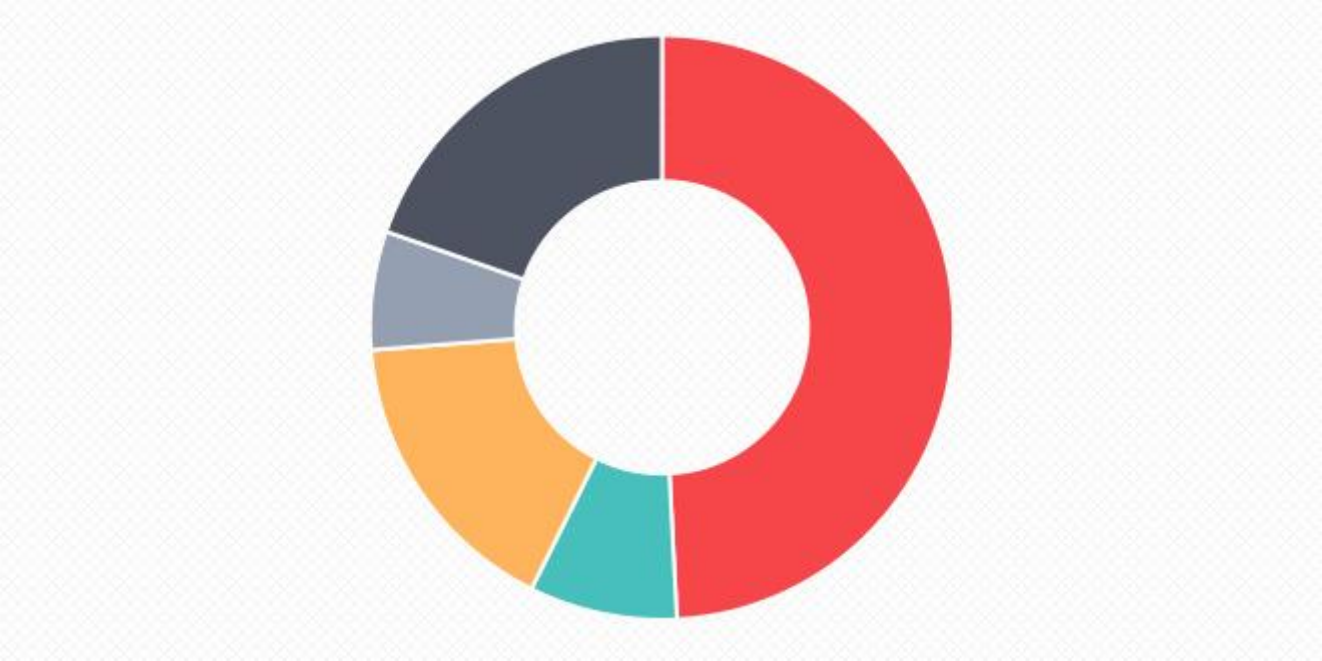


Exemple de code pour créer le graphique :

```
<canvas id="pie-chart" width="800" height="450"></canvas>
new Chart(document.getElementById("pie-chart"), {
  type: 'pie',
  data: {
    labels: ["Africa", "Asia", "Europe", "Latin America", "North America"],
    datasets: [{
      label: "Population (millions)",
      backgroundColor: ["#3e95cd", "#8e5ea2","#3cba9f","#e8c3b9","#c45850"],
      data: [2478,5267,734,784,433]
    }]
  },
  options: {
    title: {
      display: true,
      text: 'Predicted world population (millions) in 2050'
    }
  }
});
```

6. Diagramme de beignets (Doughnut chart)

Les diagrammes de donut sont créés en définissant le type sur doughnut , identiques aux camemberts



Exemple de code pour créer le graphique :

```
<canvas id="doughnut-chart" width="800" height="450"></canvas>
new Chart(document.getElementById("doughnut-chart"), {
    type: 'doughnut',
    data: {
      labels: ["Africa", "Asia", "Europe", "Latin America", "North America"],
      datasets: [
        {
          label: "Population (millions)",
          backgroundColor: ["#3e95cd", "#8e5ea2","#3cba9f","#e8c3b9","#c45850"],
          data: [2478,5267,734,784,433]
        }
      ]
    },
    options: {
      title: {
        display: true,
        text: 'Predicted world population (millions) in 2050'
      }
    }
});
```

7. Graphique à barres horizontal

Les graphiques à barres horizontales sont créés en définissant le type sur horizontalBar .

```
<canvas id="bar-chart-horizontal" width="800" height="450"></canvas>
new Chart(document.getElementById("bar-chart-horizontal"), {
    type: 'horizontalBar',
    data: {
      labels: ["Africa", "Asia", "Europe", "Latin America", "North America"],
      datasets: [
        {
          label: "Population (millions)",
          backgroundColor: ["#3e95cd", "#8e5ea2","#3cba9f","#e8c3b9","#c45850"],
          data: [2478,5267,734,784,433]
        }
      ]
    },
    options: {
      legend: { display: false },
      title: {
        display: true,
        text: 'Predicted world population (millions) in 2050'
      }
    }
});
```

8. Graphique à barres groupées (Brouped Bar Chart)

Un diagramme à barres groupées n'est pas un type de graphique unique, mais vous devez configurer vos données différemment par rapport aux graphiques à barres de type bar .

Lorqu'on transmet plus d'un objet aux datasets de datasets , Chart.js crée un nouveau groupe de barres pour chaque objet. Définir la couleur pour ce groupe de barres est ensuite fait en passant une couleur à backgroundColor .

Exemple de code pour créer le graphique :

```
<canvas id="bar-chart-grouped" width="800" height="450"></canvas>
new Chart(document.getElementById("bar-chart-grouped"), {
    type: 'bar',
    data: {
      labels: ["1900", "1950", "1999", "2050"],
      datasets: [
        {
          label: "Africa",
          backgroundColor: "#3e95cd",
          data: [133,221,783,2478]
        }, {
          label: "Europe",
          backgroundColor: "#8e5ea2",
          data: [408,547,675,734]
        }
      ]
    },
    options: {
      title: {
        display: true,
        text: 'Population growth (millions)'
```

```
    }
  }
});
```

9. Tableau mixte (Mixed chart)

Vous pouvez mélanger plusieurs graphiques et les superposer les uns sur les autres.

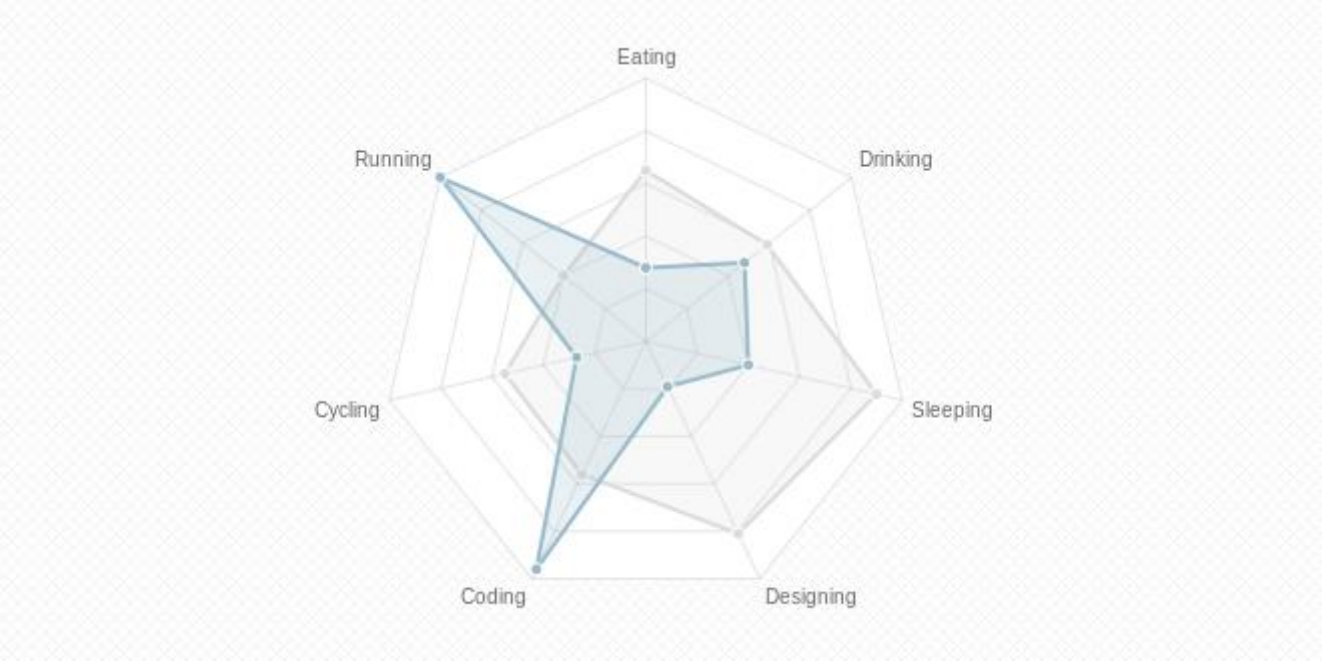
Pour produire le graphique ci-dessus, par exemple, nous avons quatre objets de données: deux mis en bar , et deux mis en line , tandis que le type de l'objet Graphique est mis à la bar .

Exemple de code pour créer le graphique :

```
<canvas id="mixed-chart" width="800" height="450"></canvas>
new Chart(document.getElementById("mixed-chart"), {
  type: 'bar',
  data: {
    labels: ["1900", "1950", "1999", "2050"],
    datasets: [{
        label: "Europe",
        type: "line",
        borderColor: "#8e5ea2",
        data: [408,547,675,734],
        fill: false
      }, {
        label: "Africa",
        type: "line",
        borderColor: "#3e95cd",
        data: [133,221,783,2478],
        fill: false
      }, {
        label: "Europe",
        type: "bar",
        backgroundColor: "rgba(0,0,0,0.2)",
        data: [408,547,675,734],
      }, {
        label: "Africa",
        type: "bar",
        backgroundColor: "rgba(0,0,0,0.2)",
        backgroundColorHover: "#3e95cd",
        data: [133,221,783,2478]
      }
    ]
  },
  options: {
    title: {
      display: true,
      text: 'Population growth (millions): Europe & Africa'
    },
    legend: { display: false }
  }
});
```

10. Diagramme radar (Radar Chart)

Radar affiche les données multivariées sous la forme d' un tableau à deux dimensions de plus de trois variables représentée sur des axes à partir du même point .Les graphiques radar, également connus sous le nom de graphiques en ligne, de graphiques en étoile, de graphiques en étoile, sont créés en définissant le type sur radar . Les graphiques radar nécessitent généralement plus d'espace vertical que les autres graphiques pour être lisibles, vous devrez donc modifier les proportions du graphique.



Exemple de code pour créer le graphique :

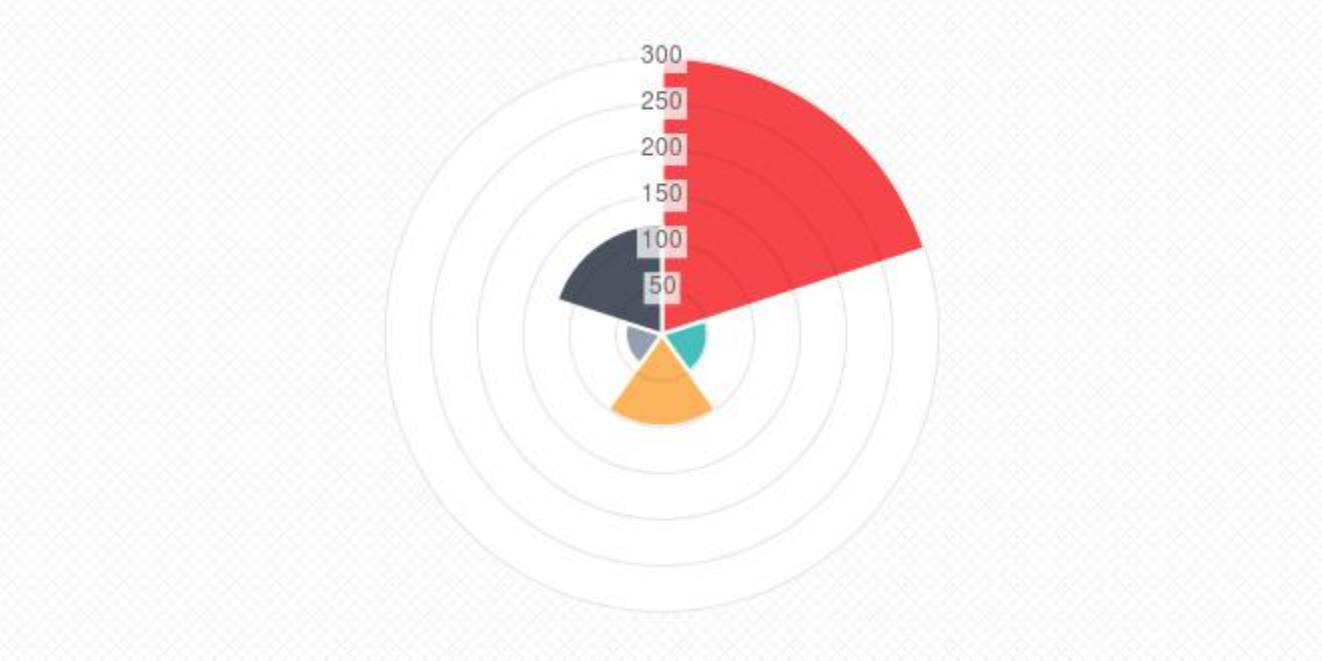
```
<canvas id="radar-chart" width="800" height="600"></canvas>
new Chart(document.getElementById("radar-chart"), {
   type: 'radar',
   data: {
     labels: ["Africa", "Asia", "Europe", "Latin America", "North America"],
     datasets: [
      {
        label: "1950",
        fill: true,
        backgroundColor: "rgba(179,181,198,0.2)",
        borderColor: "rgba(179,181,198,1)",
        pointBorderColor: "#fff",
        pointBackgroundColor: "rgba(179,181,198,1)",
        data: [8.77,55.61,21.69,6.62,6.82]
      }, {
        label: "2050",
        fill: true,
        backgroundColor: "rgba(255,99,132,0.2)",
        borderColor: "rgba(255,99,132,1)",
        pointBorderColor: "#fff",
        pointBackgroundColor: "rgba(255,99,132,1)",
        pointBorderColor: "#fff",
        data: [25.48,54.16,7.61,8.06,4.45]
      }
    ]
```

```
  },
  options: {
    title: {
      display: true,
      text: 'Distribution in % of world population'
    }
  }
});
```

11. Carte de zone polaire (Polar Area Chart)

Polar Area affiche les données multivariées sous la forme d' un tableau à deux dimensions de plus de trois variables représentée sur des axes à partir du même point .

Un diagramme de zone polaire est créé en définissant le type sur polarArea . Les diagrammes de secteurs polaires sont étroitement liés aux diagrammes à secteurs, à la différence qu'en plus des angles représentant la taille relative des points de données, le rayon de chaque élément est défini par rapport à sa valeur



Exemple de code pour créer le graphique :

```
<canvas id="polar-chart" width="800" height="450"></canvas>
new Chart(document.getElementById("polar-chart"), {
  type: 'polarArea',
  data: {
    labels: ["Africa", "Asia", "Europe", "Latin America", "North America"],
    datasets: [
      {
        label: "Population (millions)",
        backgroundColor: ["#3e95cd", "#8e5ea2","#3cba9f","#e8c3b9","#c45850"],
        data: [2478,5267,734,784,433]
      }
    ]
```

```
    },
    options: {
      title: {
        display: true,
        text: 'Predicted world population (millions) in 2050'
      }
    }
});
```

Exemple de construction d'un diagramme à barres (graphique à barres) avec ChartJS en jQuery

Les options à définir pour le graphique

Nous allons créer une variable d'objet options et définir ses propriétés responsive, title, legend et scale.

Nous allons définir responsive à true pour rendre le graphique réactif.

Pour créer un **titre** pour le graphique à barres, nous allons définir ce qui suit pour l'objet de données de titre.

• display: true afin de faire apparaître le titre.

• position: top (haut) pour que le titre apparaisse en haut du graphique à barres.

• text: pour le graphique à barres

• fontSize: la taille de la police du titre.

• fontColor: la couleur de police du titre.

Pour créer une **légende** pour le graphique à barres, définissez la propriété l**egend**.

• display: true pour afficher la légende.

• position: bottom (en bas) bas, ce qui définit la position de la légende.

• label: concerne la couleur et la taille de la police.

Enfin, pour que l'axe des ordonnées commence à 0, nous définissons la propriété **scale.**

```
//options
var options = {
    responsive: true,
    title: {
        display: true,
        position: "top",
        text: "Bar Graph",
        fontSize: 18,
        fontColor: "#111"
    },
    legend: {
        display: true,
```

```
        position: "bottom",
        labels: {
            fontColor: "#333",
            fontSize: 16
        }
    },
    scales: {
        yAxes: [{
            ticks: {
                min: 0
            }
        }]
    }
};
```

Les données

On crée une variable de données qui tiendra le score des deux équipes - TeamA et TeamB pour les 5 matchs.

• **labels**: il s'agit d'un tableau contenant le nom de la correspondance.

• **datasets**: il s'agit d'un tableau de deux objets, un pour chaque score d'équipe.

Chaque élément d'objet des jeux de données contient les propriétés suivantes.

• l**abel**: l'étiquette à afficher lorsque la souris survole la barre.

• **data:** un tableau contenant les scores de chaque match.

• **backgroundColor:** un tableau de valeur hexadécimale ou de nom de couleur pour les barres.

• **borderColor:** un tableau de valeur hexadécimale ou de nom de couleur pour la barre.

• **borderWidth:** il s'agit de la largeur de bordure des barres.

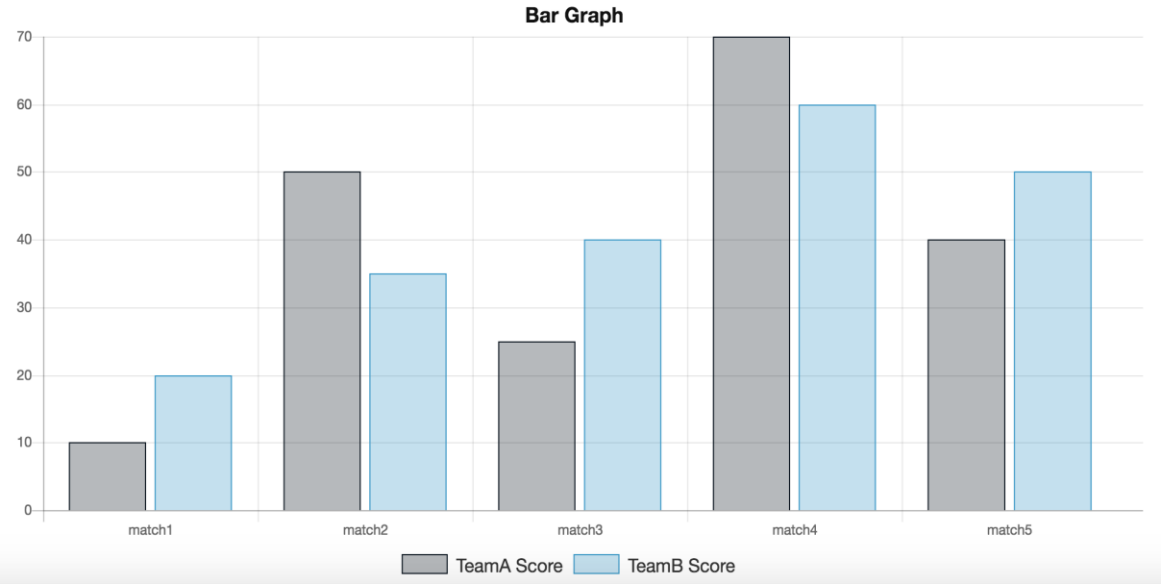
```
<!DOCTYPE html>

<html>

        <head>

                <title>ChartJS - BarGraph</title>

                <style type="text/css">

                        #chart-container {

                                width: 640px;

                                height: auto;
```

```
                }
            </style>

            <script type="text/javascript" src="js/Chart.min.js"></script>
            <script type="text/javascript" src="js/jquery-1.7.2.min.js"></script>
            <script type="text/javascript"  language="javascript" >
        $(function(){

//get the bar chart canvas
var ctx = $("#bar-chartcanvas");
//bar chart data
var data = {
    labels: ["match1", "match2", "match3", "match4", "match5"],
    datasets: [
        {
            label: "TeamA Score",
            data: [10, 50, 25, 70, 40],
            backgroundColor: [
                "rgba(10,20,30,0.3)",
                "rgba(10,20,30,0.3)",
                "rgba(10,20,30,0.3)",
                "rgba(10,20,30,0.3)",
                "rgba(10,20,30,0.3)"
                //backgroundColor: [   "red",   "red",   "red",  "red",   "red"   ]
            ],
            borderColor: [
                "rgba(10,20,30,1)",
```

```
            "rgba(10,20,30,1)",
            "rgba(10,20,30,1)",
            "rgba(10,20,30,1)",
            "rgba(10,20,30,1)"
        ],
        borderWidth: 1
    },
    {
        label: "TeamB Score",
        data: [20, 35, 40, 60, 50],
        backgroundColor: [
            "rgba(50,150,200,0.3)",
            "rgba(50,150,200,0.3)",
            "rgba(50,150,200,0.3)",
            "rgba(50,150,200,0.3)",
            "rgba(50,150,200,0.3)"
        ],
        borderColor: [
            "rgba(50,150,200,1)",
            "rgba(50,150,200,1)",
            "rgba(50,150,200,1)",
            "rgba(50,150,200,1)",
            "rgba(50,150,200,1)"
        ],
        borderWidth: 1
    }
]
```

```
};


//options
var options = {
    responsive: true,
    title: {
        display: true,
        position: "top",
        text: "Bar Graph",
        fontSize: 18,
        fontColor: "#111"
    },
    legend: {
        display: true,
        position: "bottom",
        labels: {
            fontColor: "#333",
            fontSize: 16
        }
    },
    scales: {
        yAxes: [{
            ticks: {
                min: 0
            }
        }]
    }
```

```
    };

    //create Chart class object
    var chart = new Chart(ctx, {
        type: "bar",
        data: data,
        options: options
    });
});

                </script>

        </head>
        <body>
                <div id="chart-container">
                        <canvas id= "bar-chartcanvas" width= "1600" height= "900" ></canvas>
                </div>




        </body>
</html>
```



Exemple de construction d'un histogramme à partir d'une base de données via PHP

Le code PHP récupère les données d'une table Client et les convertit en format json , celles-ci sont retournées au client pour visualiser le graphe

Data.php

```
<?php

$mysqli = new mysqli("localhost","root","", "base");

if(!$mysqli){

die("Connection failed: " . $mysqli->error);}

//query to get data from the table

$query = "SELECT nom, solde FROM client order by nom";

//execute query

$result = $mysqli->query($query);

$data = array();

    //check if there is any data returned by the SQL Query

    if ($result->num_rows > 0) {

      //Converting the results into an associative array

      while($row = $result->fetch_assoc()) {

        $jsonArrayItem = array();
```

```
        $jsonArrayItem['nom'] = $row['nom'];
        $jsonArrayItem['solde'] = $row['solde'];
        //append the above created object into the main array.
        array_push($data, $jsonArrayItem);
        //  ou bien  $data[] = $jsonArrayItem;
      }
    }
$result->close(); $mysqli->close();
//now print the data
print json_encode($data); // print_r (json_encode($data));
/*returns a string like so:
[{"col1":"col1_value","col2":"col2_value"},{"col1":"col1_value","col2":"col2_value"}]
*/
?>
```

JQUERY AFFICHANT LE GRAPHE

```
<head>
<title>ChartJS - BarGraph</title>
<script type="text/javascript" src="js/jquery-1.3.1.min.js"></script>
<script type="text/javascript" src="js/Chart.min.js"></script>
</head>
<body>
                <div id="chart-container">
                  <canvas id="mycanvas"></canvas>
                </div>
<script type="text/javascript"  language="javascript" >
$(document).ready(function(){
$.ajax({                url: "data.php",
```

```
            method: "GET",
            success: function(data) {
                var nom = [];
                var solde = [];
              var json_obj = JSON.parse(data); //parse JSON
             for(var i in json_obj )  {
            nom.push("Nom " + json_obj [i].nom);
            solde.push(json_obj[i].solde);          }


var chartdata = {
        labels: nom,
        datasets : [
            {
            label: 'Player Score',
            backgroundColor: 'rgba(200, 200, 200, 0.75)',
            borderColor: 'rgba(200, 200, 200, 0.75)',
            hoverBackgroundColor: 'rgba(200, 200, 200, 1)',
            hoverBorderColor: 'rgba(200, 200, 200, 1)',
            data: solde
            }
            ]
            };
        var ctx = $("#mycanvas");
         var barGraph = new Chart(ctx, {type: 'bar',data:chartdata });
            },
            error: function(data) { console.log(data);}
        });
```

```
});
        </script>
    </body>
```