



# 1420-7001

By

Dr. Abdul Majeed (조교수)

1<sup>st</sup> Semester, 2025

웹 프로그래밍

Web Programming



IT융합대학 컴퓨터공학부(컴퓨터공학전공)

# Summary of the Today's Lesson

- JS Functions
  - ▣ Without parameter
  - ▣ With parameter
  - ▣ Function Calls
- JS Objects
  - ▣ Creation of objects
  - ▣ Accessing object properties
- JS Events
  - ▣ onClick()
  - ▣ onMouseOver()
  - ▣ onMouseout()

```
> response
< ▼ Object 1
  ▶ config: Object
  ▼ data: Object
    ▼ assignToMap: Object
      ▼ 123: Array[4]
        0: 1
        1: 2
        2: 3
        3: 4
        length: 4
        ▶ __proto__: Array[0]
      ▶ 345: Array[4]
      ▶ 678: Array[4]
      ▶ __proto__: Object
    ▶ __proto__: Object
  ▶ headers: (d)
    status: 200
    statusText: "OK"
  ▶ __proto__: Object
> |
```



# Part I

## Advanced JS Codes

## Web Programming



# Part I (a)

JS Functions

Web Programming

# JS Coding Concepts-JS Function

- ⌘ A JavaScript function is a block of code designed to perform a particular task.
- ⌘ A JavaScript function is executed when "something" invokes it (calls it).

```
<html>
<body>
<h1>JavaScript Functions</h1>

<p>Call a function which performs a calculation and returns the result:</p>

<p id="demo"></p>

<script>
function myFunction(p1, p2) {
    return p1 * p2;
}

let result = myFunction(4, 3);
document.getElementById("demo").innerHTML = result;
</script>

</body>
</html>
```

## JavaScript Functions

Call a function which performs a calculation and returns the result:

12



Part I (b)


JS Objects

Web Programming

# JS Coding Concepts-JS Objects- Objects in real life

⌘ In real life, a car is an **object**.

⌘ A car has **properties** like weight and color, and **methods** like start and stop

Object	Properties	Methods
	<code>car.name = Fiat</code> <code>car.model = 500</code> <code>car.weight = 850kg</code> <code>car.color = white</code>	<code>car.start()</code> <code>car.drive()</code> <code>car.brake()</code> <code>car.stop()</code>

- ✓ All cars have the same **properties**, but the property **values** differ from car to car.
- ✓ All cars have the same **methods**, but the methods are performed **at different times**.



Part I (c)

JS Events

Web Programming



# JS Coding Concepts-JS Events

⌘ Here is a list of some common HTML events.

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

# Activity #: 07

```
<!DOCTYPE html>
<html>
<body>
  <p id = "output"> </p>
  <script>
    const fruit = {
      name: "Banana",
      price: 20,
    }
    document.getElementById("output").innerHTML =
      "The price of the " + fruit.name + " is " + fruit.price;
  </script>
</body>
</html>
```

- Identify the name of the object
- How many properties of the objects are there?
- What is the output of this code?

# Activity #: 07

```
<!DOCTYPE html>
<html>
<body>
  <p id = "output"> </p>
  <script>
    const fruit = {
      name: "Banana",
      price: 20,
    }
    document.getElementById("output").innerHTML =
      "The price of the " + fruit.name + " is " + fruit.price;
  </script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>
  <p id = "output"> </p>
  <script>
    const fruit = {
      name: "Banana",
      price: 20,
    }
    document.getElementById("output").innerHTML =
      "The price of the " + fruit["name"] + " is " + fruit["price"];
  </script>
</body>
</html>
```

- What is the difference in the code?
- What is the output of these codes?

# Activity #: 07

```
<!DOCTYPE html>
<html>
<body>
  <p id = "output"> </p>
  <script>
    const num = {
      10: "ten",
      20: "Twenty",
    }
    const x = 10;
    document.getElementById("output").innerHTML = num[x + 10];
  </script>
</body>
</html>
```

- What is the output of this code?

# A more detailed description of the object

```
<!DOCTYPE html>
<html>
<body>
  <p id = "output1"> </p>
  <p id = "output2"> </p>
  <script>
    const cars = {
      totalBrands: 50,
      audi: {
        model: "Q7",
        price: 10000000,
      },
      bmw: {
        model: "S20D",
        price: 8000000,
      }
    }
  </script>
</body>
</html>
```

```
document.getElementById("output1").innerHTML =
  "The model of Audi is " + cars.audi.model +
  " and its price is " + cars.audi.price;

document.getElementById("output2").innerHTML =
  "The model of BMW is " + cars["bmw"]["model"] +
  " and its price is " + cars["bmw"]["price"];
</script>
</body>
</html>
```

## Output

The model of Audi is Q7 and its price is 10000000

The model of BMW is S20D and its price is 8000000

# A more detailed description of the object

```
<html>
<body>
  <p id = "output"> </p>
  <script>
    const fruit = {
      name: "Watermelon",
      price: 150,
    }
    fruit.name = "Apple"; // Updating using the dot notation
    fruit["price"] = 200; // Updating using the bracket notation
    fruit.expiry = "5 days"; // Adding new property to the object.
    document.getElementById("output").innerHTML +=
      "The price of " + fruit.name +
      " is " + fruit.price +
      " and it expires in " + fruit.expiry;

  </script>
</body>
</html>
```

## Output

The price of Apple is 200 and it expires in 5 days

---

# Part II

## Displaying Custom Data via Charts

### Data Visualizations

# Displaying Data in Charts using ChartJs

---

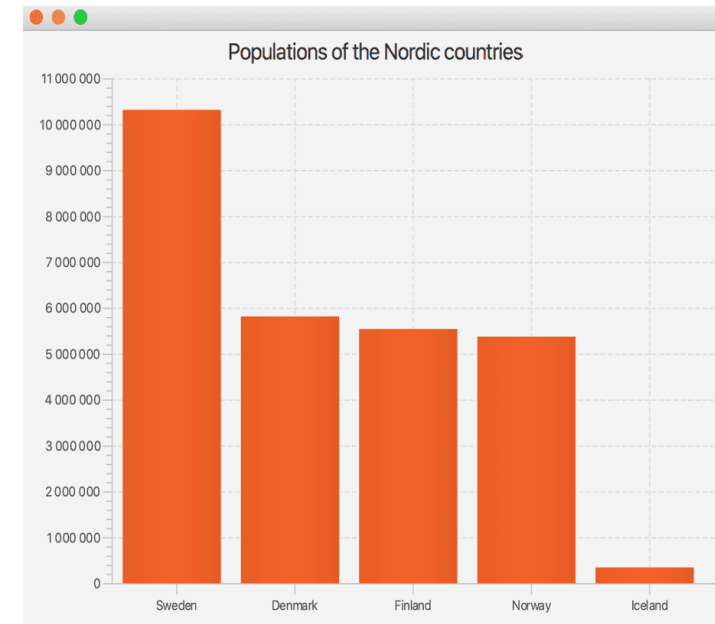
- In this discussion, we will learn to draw line graph using ChartJS and some static data.

<https://www.geeksforgeeks.org/what-is-data-visualization-and-why-is-it-important/>



# Data Visualizations-Basic Concepts

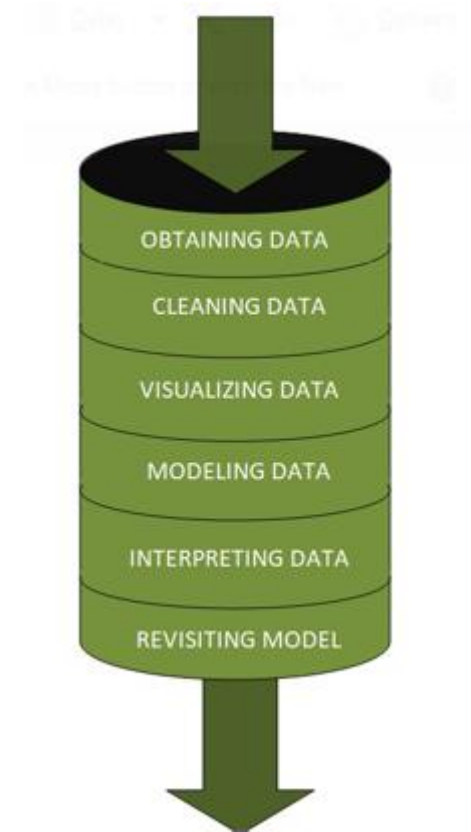
- The adage "a picture is worth a thousand words" describes the goal of data visualization appropriately.
- Data visualization seeks to present information in a concise, yet comprehensible form.
- Visualizations can emphasize important points and provide the user with useful things, such as summaries of data.



# Significance of the Data Visualizations

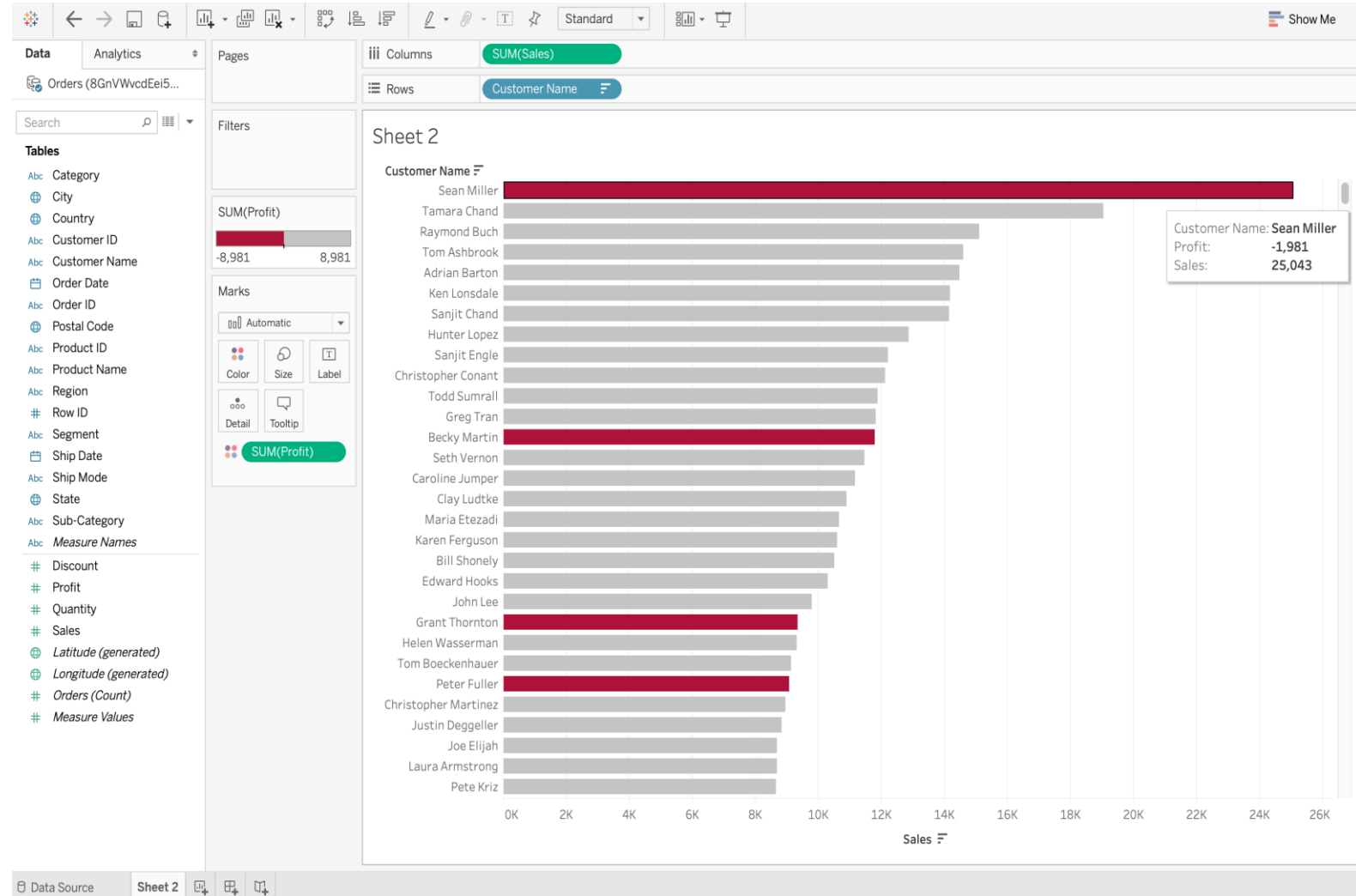
- 1.Fetching/Obtaining the Data
- 2.Scrubbing/Cleaning the Data
- 3.Data Visualization**
- 4.Modelling the Data
- 5.Interpreting the Data
- 6.Revision

- Data visualization is the graphical representation of information and data in a pictorial or graphical format(Example: charts, graphs, and maps).
- Data visualization tools provide an accessible way to see and understand trends, patterns in data, and outliers.
- Data visualization tools and technologies are essential to analysing massive amounts of information and making data-driven decisions. The concept of using pictures is to understand data that has been used for centuries. General types of data visualization are Charts, Tables, Graphs, Maps, Dashboards.



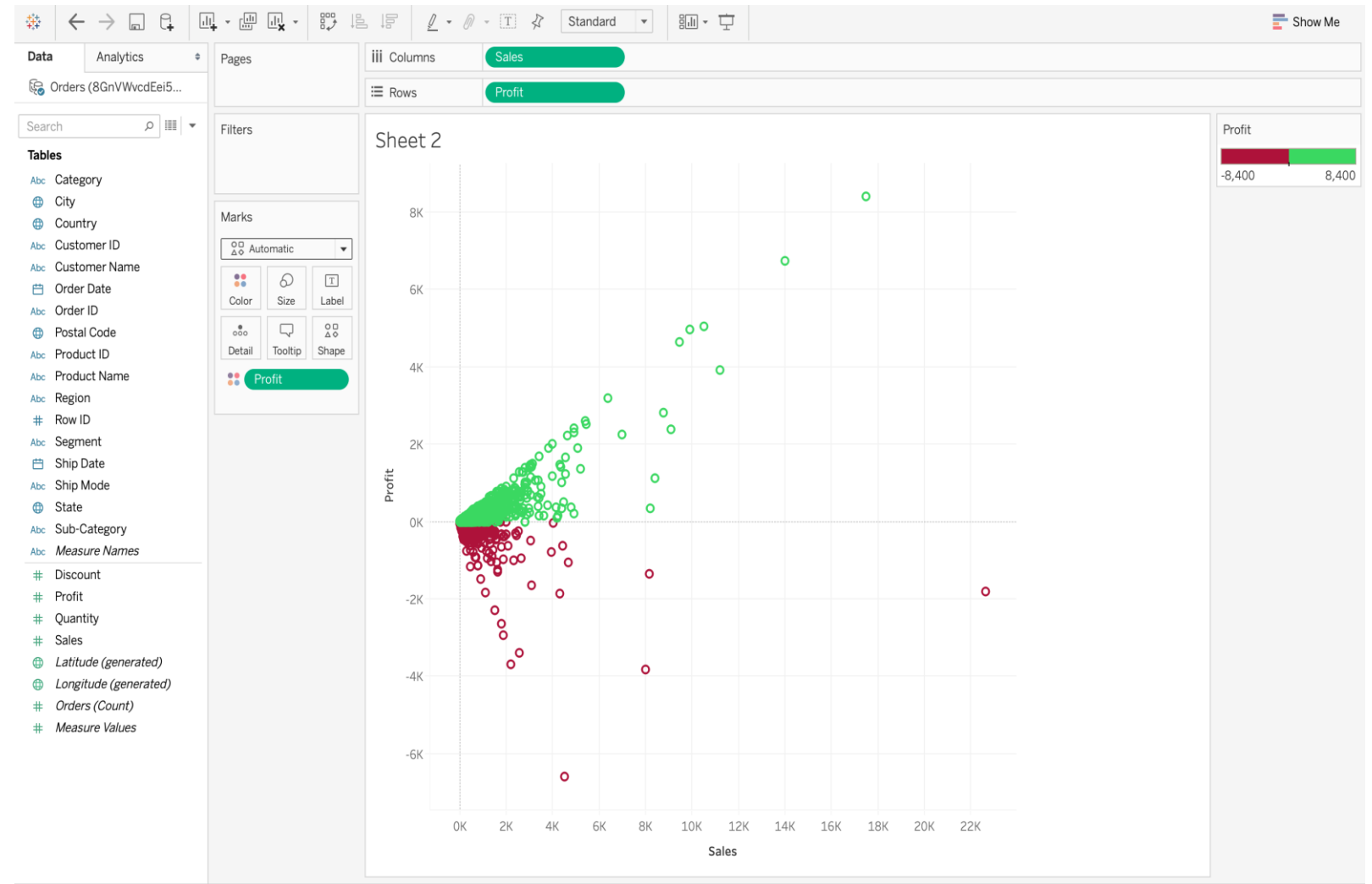
# Why Data Visualizations?

- ❑ Data Visualization Discovers the Trends in Data.



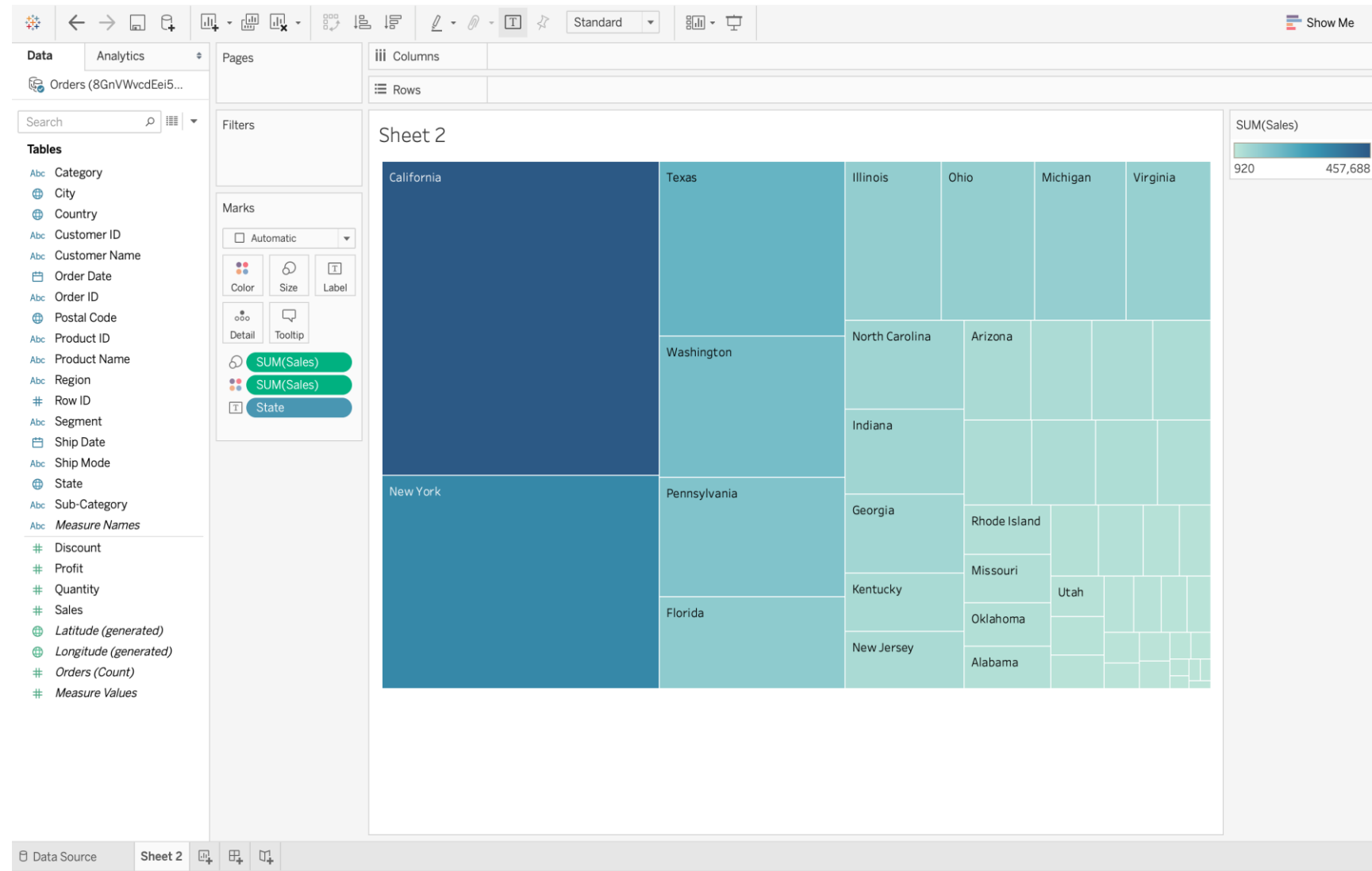
# Why Data Visualizations?

- ❑ Data Visualization Provides a Perspective on the Data.



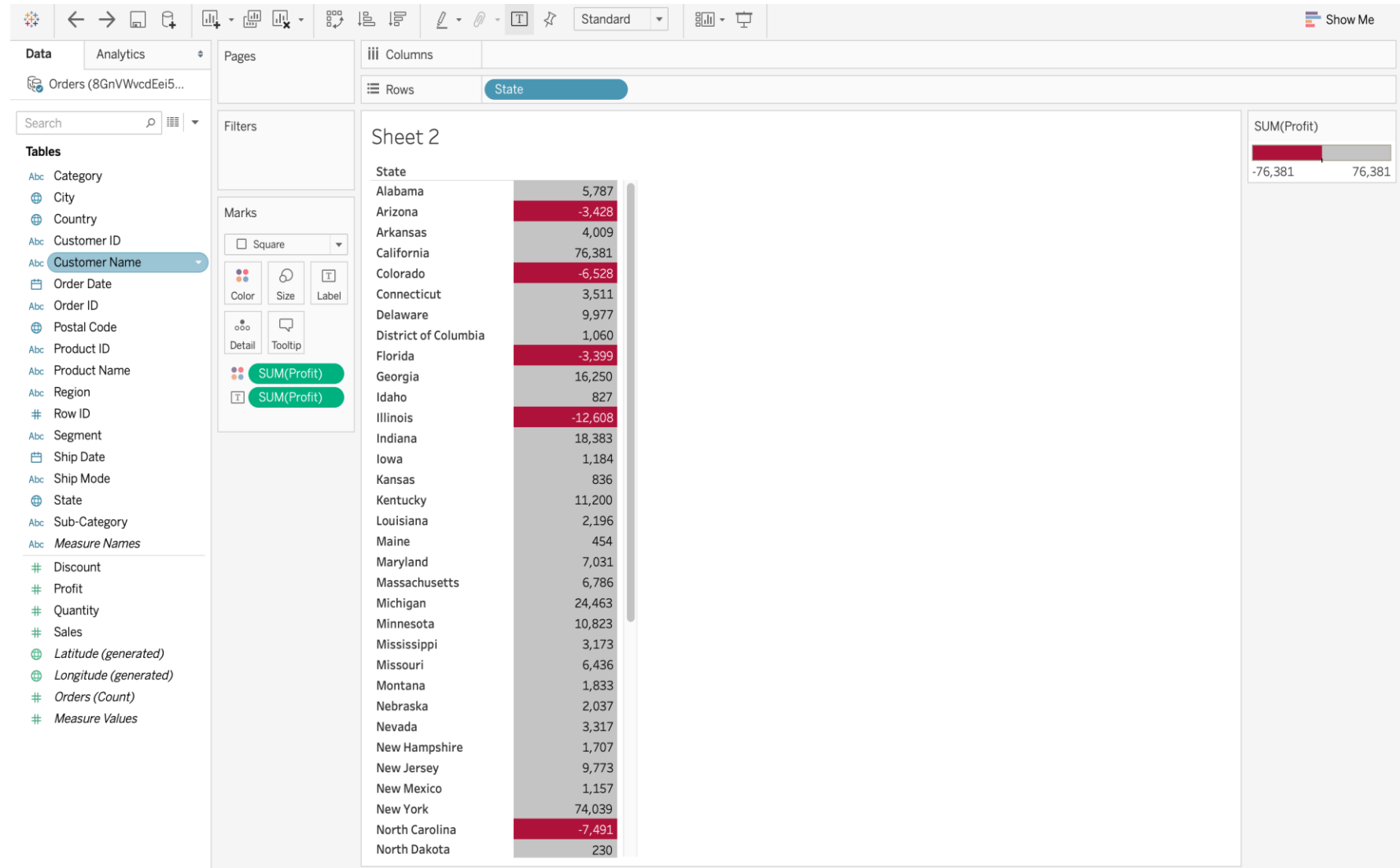
# Why Data Visualizations?

- ❑ Data Visualization Puts the Data into the Correct Context.



# Why Data Visualizations?

- ❑ Data Visualization Saves Time.



# Why Data Visualizations?

❑ Data Visualization Tells a Data Story.



# Why visualize Data?

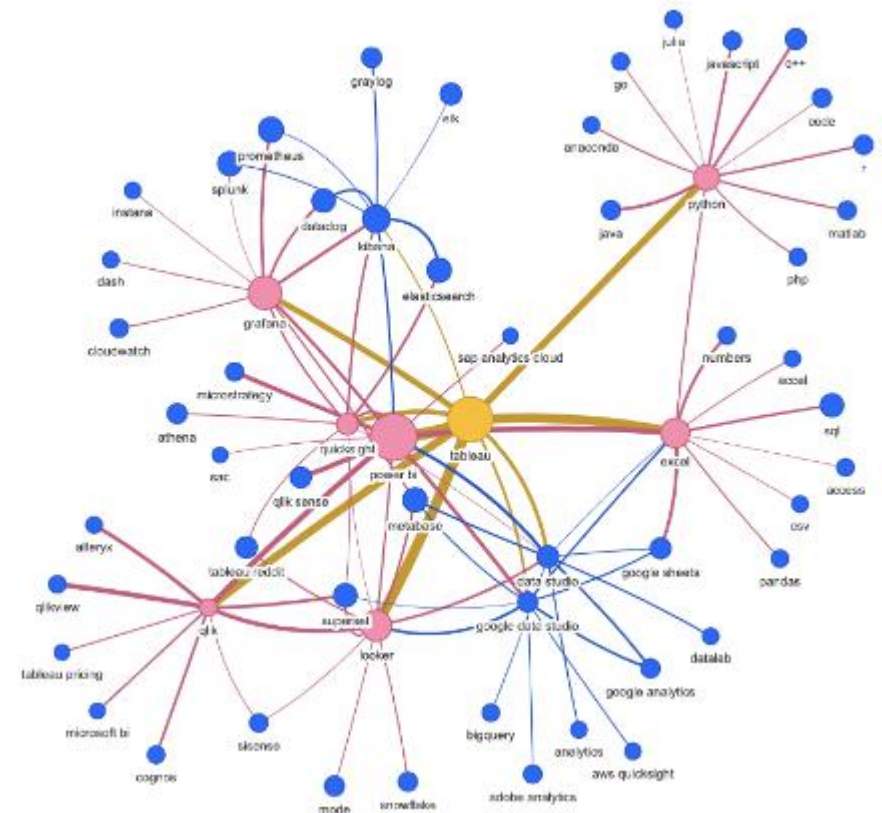
- 1) **Data Visualization** Discovers the Trends in Data.
- 2) **Data Visualization** Provides a Perspective on the Data.
- 3) **Data Visualization** Puts the Data into the Correct Context.
- 4) **Data Visualization** Saves Time.
- 5) **Data Visualization** Tells a Data Story.



# Top Data Visualization Tools

The following are the 10 best Data Visualization Tools.

- 1) Tableau
- 2) Looker
- 3) Zoho Analytics
- 4) Sisense
- 5) IBM Cognos Analytics
- 6) Qlik Sense
- 7) Domo
- 8) Microsoft Power BI
- 9) Klipfolio
- 10) SAP Analytics Cloud



# Top Data Visualization Libraries Available in Python, R, and JavaScript

## ✓ Python:

- Matplotlib
- Plotly
- ggplot
- Seaborn
- Altair
- Geoplotlib
- Bokeh

## ✓ R:

- ggplot2
- Plotly
- Leaflet
- Esquisse
- Lattice

## ✓ JavaScript:


- D3.js
- Chart.js
- Plotly



# **Data visualizations in JavaScript**

## **Coding Examples**

**Via third party libraries**



# **Data visualizations in JavaScript**

## **Coding Examples**

# Displaying Data in Charts-Line Chart

```
$(document).ready(function() {  
    //get canvas  
    var ctx = $("#line-chartcanvas");  
    var data = {  
        labels : ["match1", "match2", "match3", "match4", "match5"],  
        datasets : [  
            {  
                label : "TeamA score",  
                data : [10, 50, 25, 70, 40],  
                backgroundColor : "blue",  
                borderColor : "lightblue",  
                fill : false,  
                lineTension : 0,  
                pointRadius : 5  
            },  
            {  
                label : "TeamB score",  
                data : [20, 35, 40, 60, 50],  
                backgroundColor : "green",  
                borderColor : "lightgreen",  
                fill : false,  
                lineTension : 0,  
                pointRadius : 5  
            }  
        ]  
    };  
    var options = {  
        title : {  
            display : true,  
            position : "top",  
            text : "Line Graph",  
            fontSize : 18,  
            fontColor : "#111"  
        },  
        legend : {  
            display : true,  
            position : "bottom"  
        }  
    };  
    var chart = new Chart( ctx, {  
        type : "line",  
        data : data,  
        options : options  
    } );  
});
```

# Displaying Data in Charts-Line Chart

```
<!DOCTYPE html>
<html>
<head>
  <title>ChartJS - Line</title>

  <link href="css/default.css" rel="stylesheet">
</head>
<body>

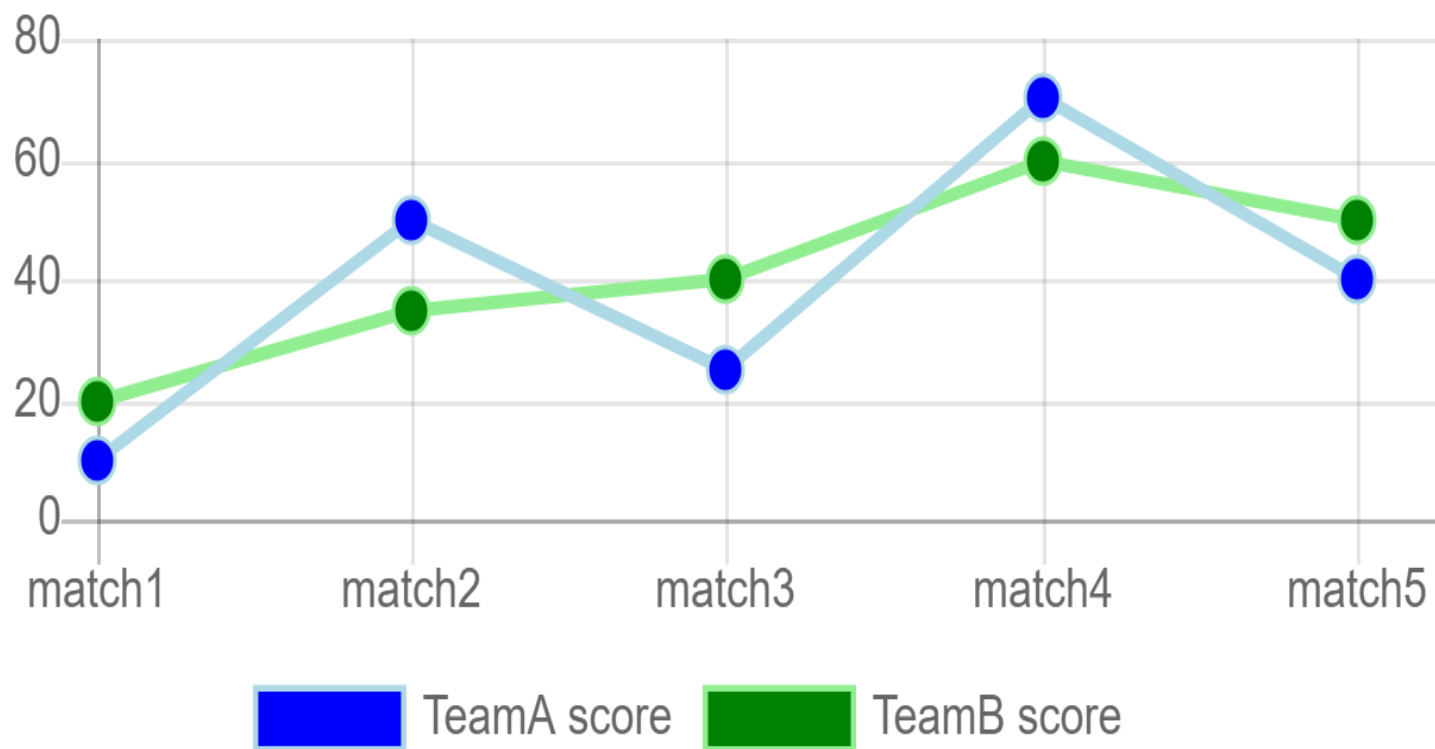
  <div class="chart-container">
    <canvas id="line-chartcanvas"></canvas>
  </div>

  <!-- javascript -->
  <script src="js/jquery.min.js"></script>
  <script src="js/Chart.min.js"></script>

  <script src="js/line.js"></script>

</body>
</html>
```

Line Graph



# Displaying Data in Charts-

## Line Chart [Min-Max]

```
var chart = new Chart( ctx, {  
  type : "line",  
  data : data,  
  options : options  
} );  
  
});
```

```
$(document).ready(function() {  
  //get canvas  
  var ctx = $("#line-chartcanvas");  
  var data = {  
    labels : ["match1", "match2", "match3", "match4", "match5"],  
    datasets : [  
      {  
        label : "TeamA score",  
        data : [10, 50, 25, 70, 40],  
        backgroundColor : "blue",  
        borderColor : "lightblue",  
        fill : false,  
        lineTension : 0,  
        pointRadius : 5  
      },  
      {  
        label : "TeamB score",  
        data : [20, 35, 40, 60, 50],  
        backgroundColor : "green",  
        borderColor : "lightgreen",  
        fill : false,  
        lineTension : 0,  
        pointRadius : 5  
      }  
    ]  
  };  
  
  var options = {  
    title : {  
      display : true,  
      position : "top",  
      text : "Line Graph",  
      fontSize : 18,  
      fontColor : "#111"  
    },  
    legend : {  
      display : true,  
      position : "bottom"  
    },  
    scales : {  
      yAxes : [{  
        ticks : {  
          max : 80,  
          min : -10,  
          stepSize : 10  
        }  
      }]  
    }  
  }  
});
```

# Displaying Data in Charts-Line Chart

```
<!DOCTYPE html>
<html>
<head>
  <title>ChartJS - Line - Min Max</title>

  <link href="css/default.css" rel="stylesheet">

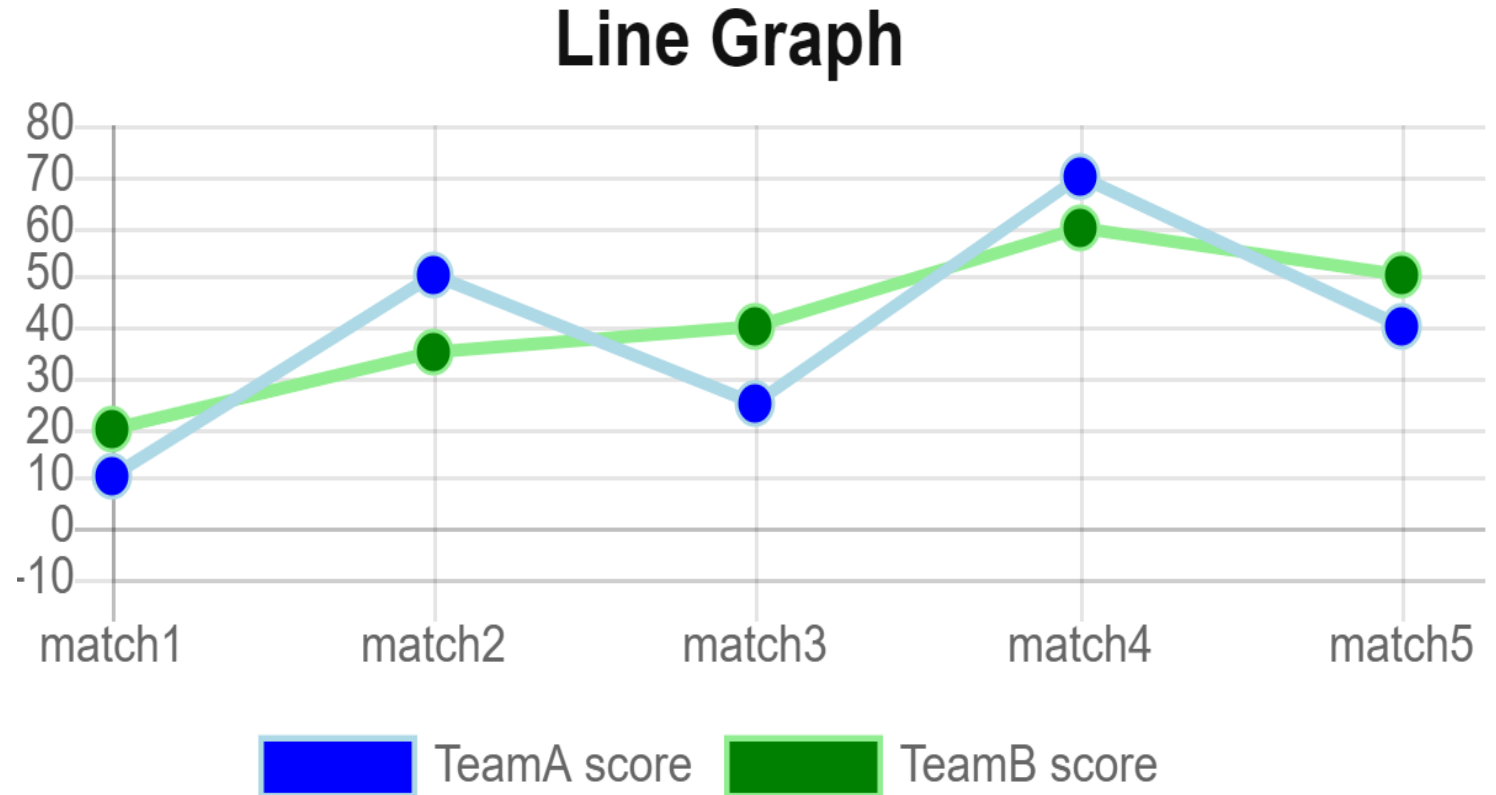
</head>
<body>

  <div class="chart-container">
    <canvas id="line-chartcanvas"></canvas>
  </div>

  <!-- javascript -->
  <script src="js/jquery.min.js"></script>
  <script src="js/Chart.min.js"></script>

  <script src="js/line-min-max.js"></script>

</body>
</html>
```





# Displaying Data in Charts-

## Bar Chart

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

var chart = new Chart( ctx, {
  type : "bar",
  data : data,
  options : options
});
```

```
var ctx = $("#bar-chartcanvas");

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)"
      ],
      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, 250, 0.3)",
        "rgba(50, 150, 250, 0.3)",
        "rgba(50, 150, 250, 0.3)",
        "rgba(50, 150, 250, 0.3)",
        "rgba(50, 150, 250, 0.3)"
      ],
      borderColor : [
        "rgba(50, 150, 250, 1)",
        "rgba(50, 150, 250, 1)",
        "rgba(50, 150, 250, 1)",
        "rgba(50, 150, 250, 1)",
        "rgba(50, 150, 250, 1)"
      ],
      borderWidth : 1
    }
  ]
};
```

# Displaying Data in Charts-**Bar Chart**

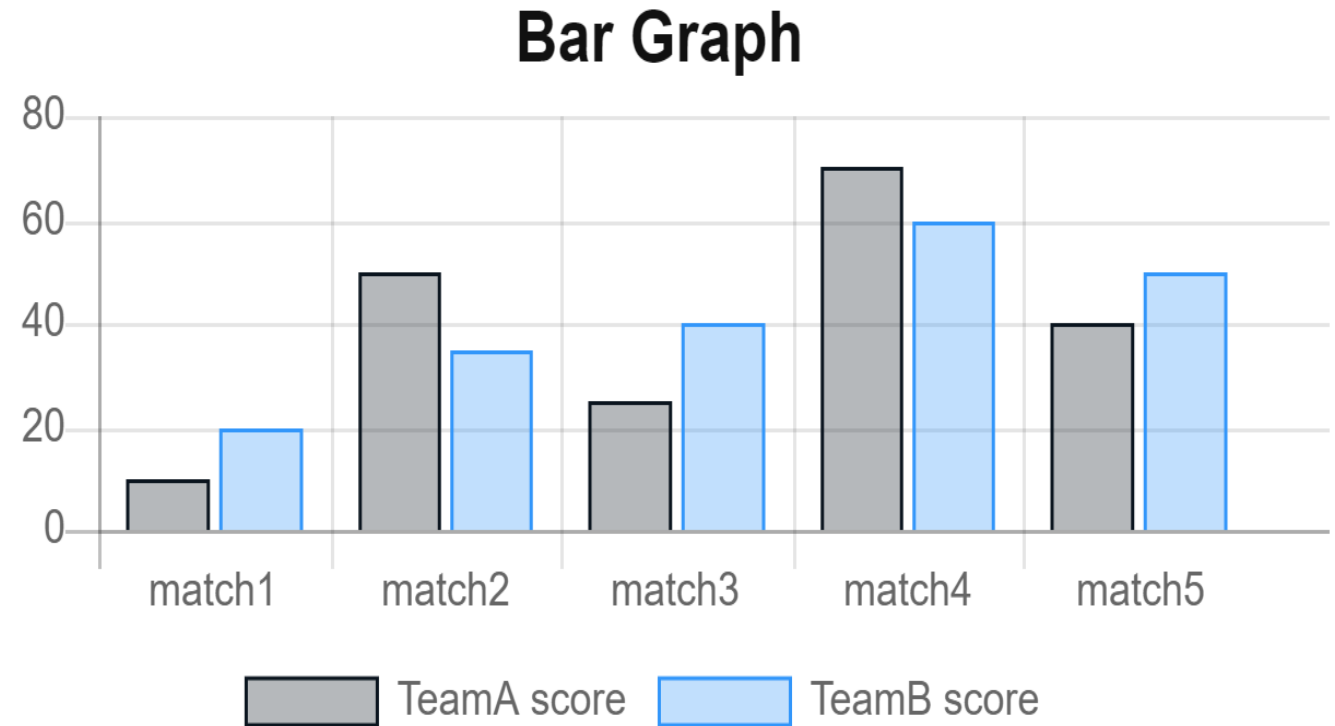
```
<!DOCTYPE html>
<html>
<head>
  <title>ChartJS - Bar</title>

  <link href="css/default.css" rel="stylesheet">
</head>
<body>

  <div class="chart-container">
    <canvas id="bar-chartcanvas"></canvas>
  </div>

  <!-- javascript -->
  <script src="js/jquery.min.js"></script>
  <script src="js/Chart.min.js"></script>

  <script src="js/bar.js"></script>
</body>
</html>
```



# Displaying Data in Charts-

## Multicolor Bar Chart

```
var options = {
  title : {
    display : true,
    position : "top",
    text : "Multicolor Bar Graph",
    fontSize : 18,
    fontColor : "#111"
  },
  legend : {
    display : false
  },
  scales : {
    yAxes : [{
      ticks : {
        min : 0
      }
    }]
  }
};

var chart = new Chart( ctx, {
  type : "bar",
  data : data,
  options : options
});
```

```
$(document).ready(function () {
  var ctx = $("#bar-chartcanvas");
  var data = {
    labels : ["match1", "match2", "match3", "match4", "match5"],
    datasets : [
      {
        label : "TeamA score",
        data : [10, 50, 25, 70, 40],
        backgroundColor : [
          "red",
          "green",
          "blue",
          "purple",
          "magenta"
        ],
        borderColor : [
          "#111",
          "#111",
          "#111",
          "#111",
          "#111"
        ],
        borderWidth : 1
      },
      {
        label : "TeamB score",
        data : [20, 35, 40, 60, 50],
        backgroundColor : [
          "aqua",
          "salmon",
          "darkgray",
          "pink",
          "coral"
        ],
        borderColor : [
          "#111",
          "#111",
          "#111",
          "#111",
          "#111"
        ],
        borderWidth : 1
      }
    ]
  };
});
```

# Displaying Data in Charts-Multicolor Bar Chart

```
<!DOCTYPE html>
<html>
<head>
  <title>ChartJS - Multicolor Bar graph</title>

  <link href="css/default.css" rel="stylesheet">
</head>
<body>

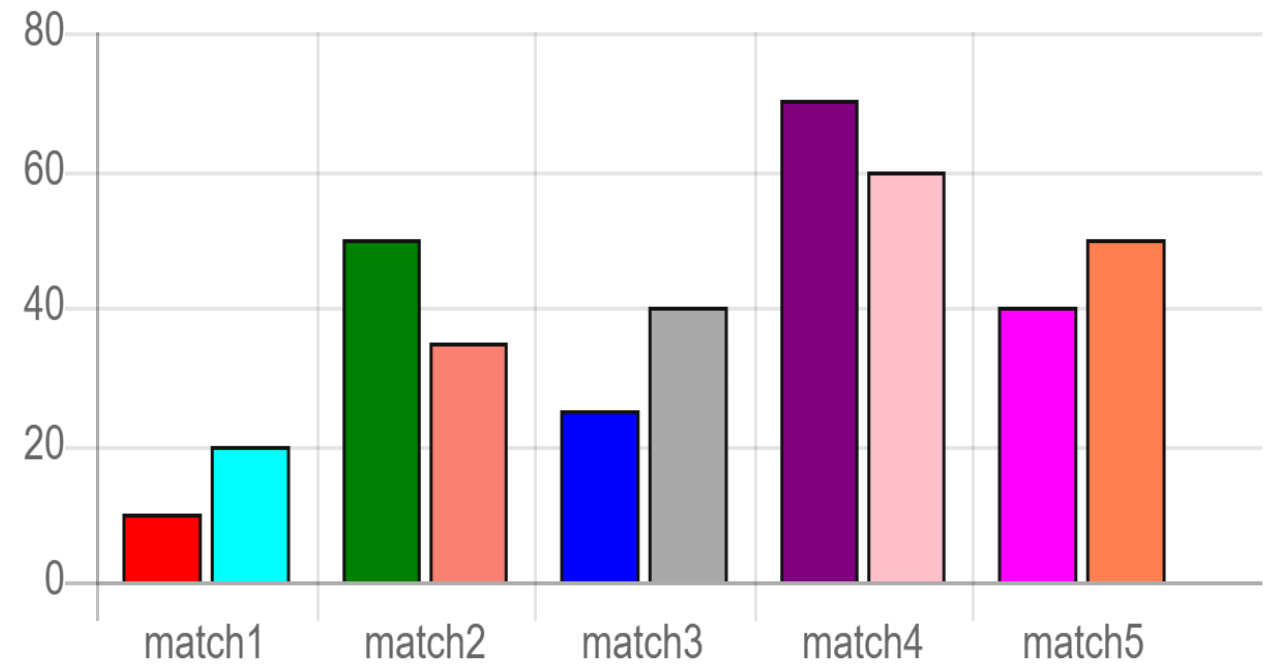
  <div class="chart-container">
    <canvas id="bar-chartcanvas"></canvas>
  </div>

  <!-- javascript -->
  <script src="js/jquery.min.js"></script>
  <script src="js/Chart.min.js"></script>

  <script src="js/bar-multicolor.js"></script>

</body>
</html>
```

Multicolor Bar Graph



# Displaying Data in Charts-

## Multicolor Bar Chart Random

```
var options = {
  title : {
    display : true,
    position : "top",
    text : "Random Multicolor Bar Graph",
    fontSize : 18,
    fontColor : "#111"
  },
  legend : {
    display : false
  },
  scales : {
    yAxes : [{
      ticks : {
        min : 0
      }
    }]
  }
};

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

/**
 * function to generate random color in hex form
 */
function getRandomColorHex() {
  var hex = "0123456789ABCDEF",
      color = "#";
  for (var i = 1; i <= 6; i++) {
    color += hex[Math.floor(Math.random() * 16)];
  }
  return color;
}

});
```

```
$(document).ready(function () {

  var ctx = $("#bar-chartcanvas");

  var data = {
    labels : ["match1", "match2", "match3", "match4", "match5"],
    datasets : [
      {
        label : "TeamA score",
        data : [10, 50, 25, 70, 40],
        backgroundColor : [
          getRandomColorHex(),
          getRandomColorHex(),
          getRandomColorHex(),
          getRandomColorHex(),
          getRandomColorHex()
        ],
        borderColor : [
          "#111",
          "#111",
          "#111",
          "#111",
          "#111"
        ],
        borderWidth : 1
      },
      {
        label : "TeamB score",
        data : [20, 35, 40, 60, 50],
        backgroundColor : [
          getRandomColorHex(),
          getRandomColorHex(),
          getRandomColorHex(),
          getRandomColorHex(),
          getRandomColorHex()
        ],
        borderColor : [
          "#111",
          "#111",
          "#111",
          "#111",
          "#111"
        ],
        borderWidth : 1
      }
    ]
  }
});
```

# Displaying Data in Charts-Multicolor Bar Chart Random

```
<!DOCTYPE html>
<html>
<head>
  <title>ChartJS - Random Multicolor Bar graph</title>

  <link href="css/default.css" rel="stylesheet">

</head>
<body>

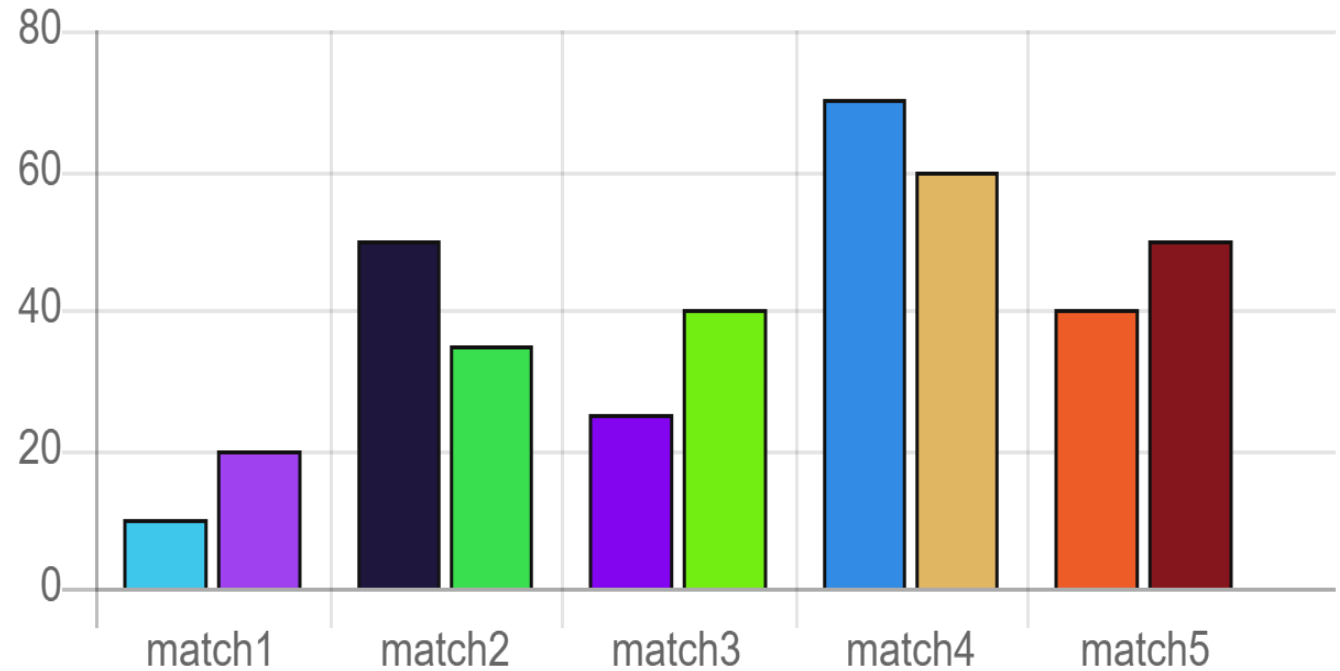
  <div class="chart-container">
    <canvas id="bar-chartcanvas"></canvas>
  </div>

  <!-- javascript -->
  <script src="js/jquery.min.js"></script>
  <script src="js/Chart.min.js"></script>

  <script src="js/bar-multicolor-random.js"></script>

</body>
</html>
```

Random Multicolor Bar Graph



# Displaying Data in Charts-

## Multicolor Pie Chart

```
        borderWidth : [1, 1, 1, 1, 1]
    }
}
};

var options = {
    title : {
        display : true,
        position : "top",
        text : "Pie Chart",
        fontSize : 18,
        fontColor : "#111"
    },
    legend : {
        display : true,
        position : "bottom"
    }
};
```

```
var chart1 = new Chart( ctx1, {
    type : "pie",
    data : data1,
    options : options
});
```

```
var chart2 = new Chart( ctx2, {
    type : "pie",
    data : data2,
    options : options
});
```

```
});
```

```
$(document).ready(function () {
    var ctx1 = $("#pie-chartcanvas-1");
    var ctx2 = $("#pie-chartcanvas-2");

    var data1 = {
        labels : ["match1", "match2", "match3", "match4", "match5"],
        datasets : [
            {
                label : "TeamA score",
                data : [10, 50, 25, 70, 40],
                backgroundColor : [
                    "#DEB887",
                    "#A9A9A9",
                    "#DC143C",
                    "#F4A460",
                    "#2E8B57"
                ],
                borderColor : [
                    "#CDA776",
                    "#989898",
                    "#CB252B",
                    "#E39371",
                    "#1D7A46"
                ],
                borderWidth : [1, 1, 1, 1, 1]
            }
        ]
    };

    var data2 = {
        labels : ["match1", "match2", "match3", "match4", "match5"],
        datasets : [
            {
                label : "TeamB score",
                data : [20, 35, 40, 60, 50],
                backgroundColor : [
                    "#FAEBD7",
                    "#DCDCDC",
                    "#E9967A",
                    "#F5DEB3",
                    "#9ACD32"
                ],
                borderColor : [
                    "#E9DAC6",
                    "#CBCBCB",
                    "#D88569",
                    "#E4CDA2",
                    "#4682B4"
                ],
                borderWidth : [1, 1, 1, 1, 1]
            }
        ]
    };
});
```

# Displaying Data in Charts-Multicolor Pie Chart

```
<!DOCTYPE html>
<html>
<head>
  <title>ChartJS - Pie</title>

  <link href="css/default.css" rel="stylesheet">
</head>
<body>

  <div class="chart-container">
    <div class="pie-chart-container">
      <canvas id="pie-chartcanvas-1"></canvas>
    </div>

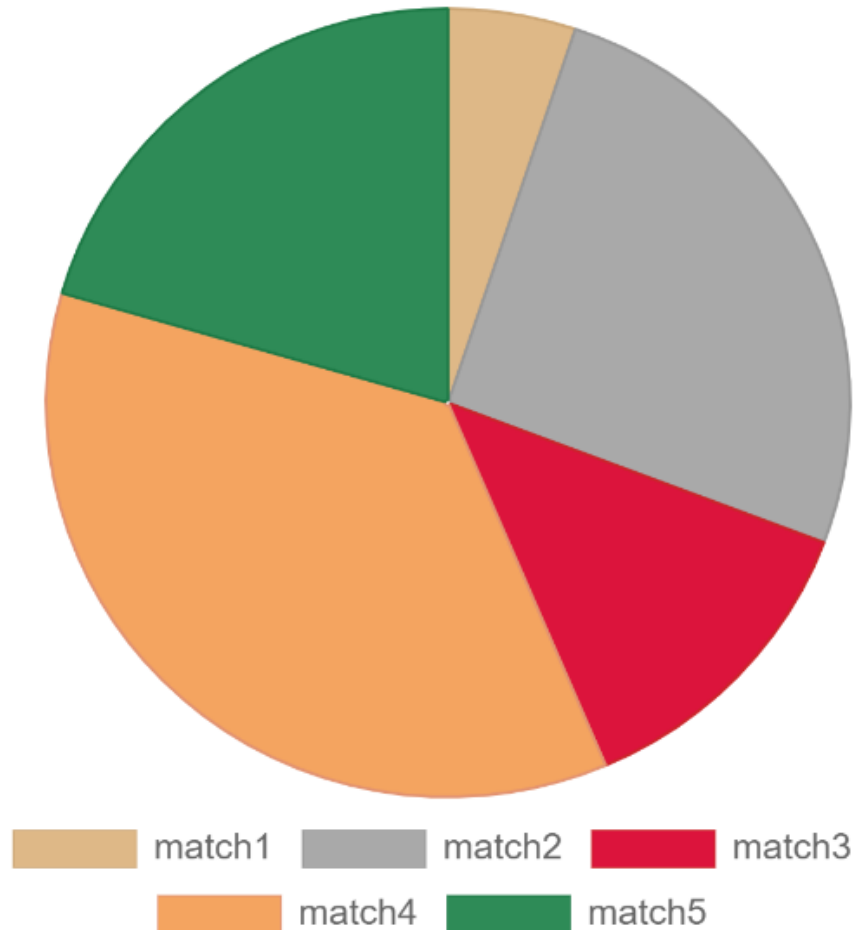
    <div class="pie-chart-container">
      <canvas id="pie-chartcanvas-2"></canvas>
    </div>
  </div>

  <!-- javascript -->
  <script src="js/jquery.min.js"></script>
  <script src="js/Chart.min.js"></script>

  <script src="js/pie.js"></script>

</body>
</html>
```

Pie Chart





# Displaying Data in Charts-Multicolor Pie Chart

```
<!DOCTYPE html>
<html>
<head>
  <title>ChartJS - Pie</title>

  <link href="css/default.css" rel="stylesheet">
</head>
<body>

  <div class="chart-container">
    <div class="pie-chart-container">
      <canvas id="pie-chartcanvas-1"></canvas>
    </div>

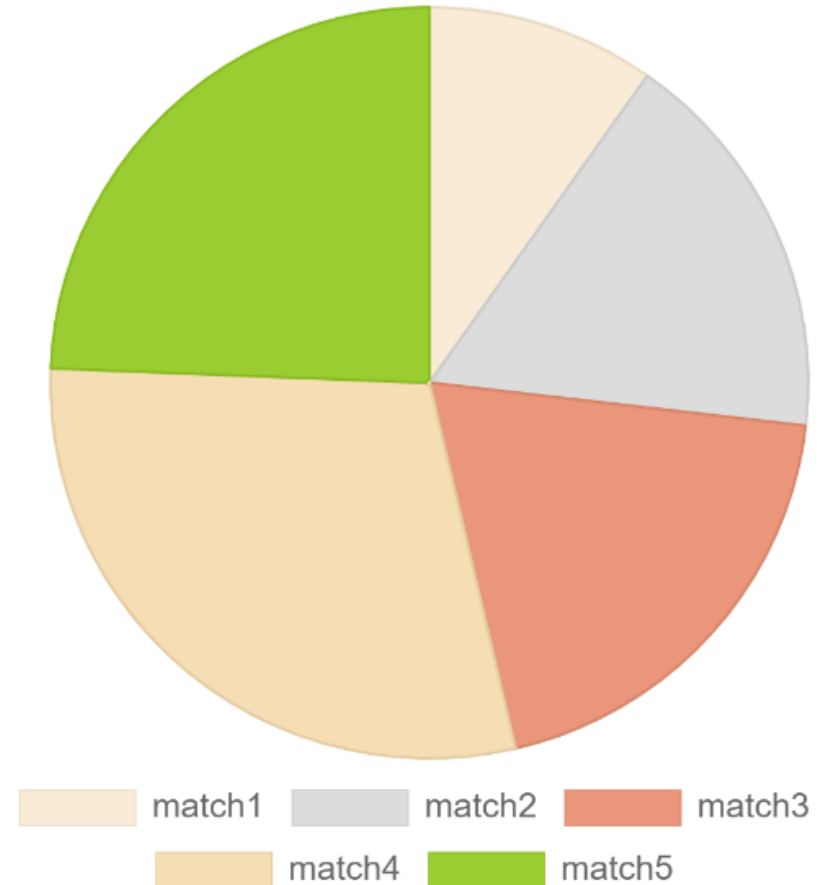
    <div class="pie-chart-container">
      <canvas id="pie-chartcanvas-2"></canvas>
    </div>
  </div>

  <!-- javascript -->
  <script src="js/jquery.min.js"></script>
  <script src="js/Chart.min.js"></script>

  <script src="js/pie.js"></script>

</body>
</html>
```

Pie Chart



# Displaying Data in Charts-

## Multicolor Doughnut Chart

```
//options
var options = {
  responsive: true,
  title: {
    display: true,
    position: "top",
    text: "Doughnut Chart",
    fontSize: 18,
    fontColor: "#111"
  },
  legend: {
    display: true,
    position: "bottom",
    labels: {
      fontColor: "#333",
      fontSize: 16
    }
  }
};

//create Chart class object
var chart1 = new Chart(ctx1, {
  type: "doughnut",
  data: data1,
  options: options
});

//create Chart class object
var chart2 = new Chart(ctx2, {
  type: "doughnut",
  data: data2,
  options: options
});
});
```

```
$(function() {
  //get the doughnut chart canvas
  var ctx1 = $("#doughnut-chartcanvas-1");
  var ctx2 = $("#doughnut-chartcanvas-2");
  //doughnut chart data
  var data1 = {
    labels: ["match1", "match2", "match3", "match4", "match5"],
    datasets: [
      {
        label: "TeamA Score",
        data: [10, 50, 25, 70, 40],
        backgroundColor: [
          "#DEB887",
          "#A9A9A9",
          "#DC143C",
          "#F4A460",
          "#2E8B57"
        ],
        borderColor: [
          "#CDA776",
          "#989898",
          "#CB252B",
          "#E39371",
          "#1D7A46"
        ],
        borderWidth: [1, 1, 1, 1, 1]
      }
    ]
  };

  //doughnut chart data
  var data2 = {
    labels: ["match1", "match2", "match3", "match4", "match5"],
    datasets: [
      {
        label: "TeamB Score",
        data: [20, 35, 40, 60, 50],
        backgroundColor: [
          "#FAEBD7",
          "#DCDCDC",
          "#E9967A",
          "#F5DEB3",
          "#9ACD32"
        ],
        borderColor: [
          "#E9DAC6",
          "#CBCBCB",
          "#D88569",
          "#E4CDA2",
          "#89BC21"
        ],
        borderWidth: [1, 1, 1, 1, 1]
      }
    ]
  };
});
```

# Displaying Data in Charts-Doughnut Chart

## Doughnut Chart

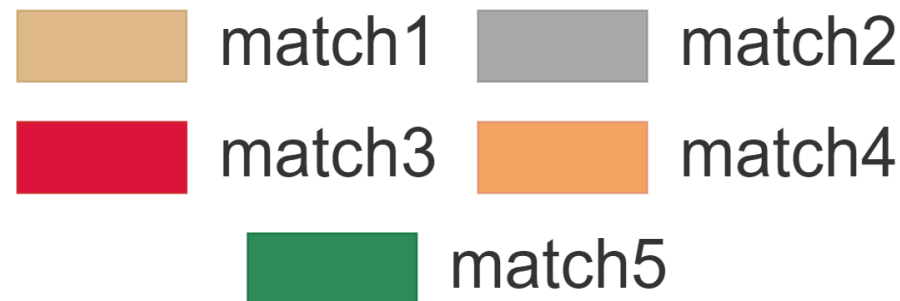
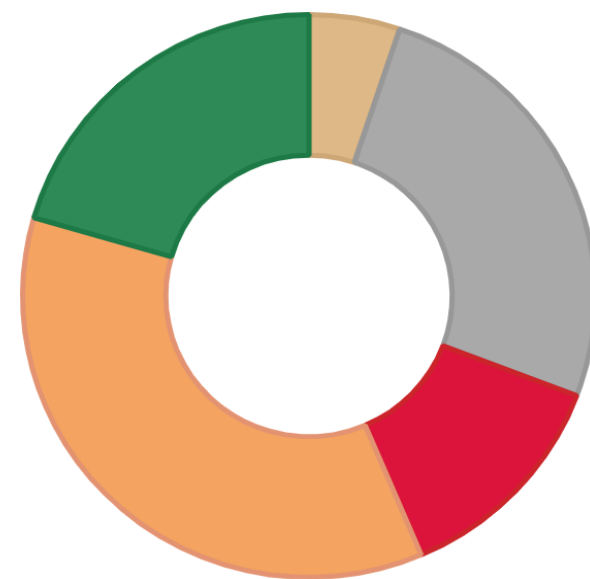
```
<!DOCTYPE html>
<html>
<head>
  <title>ChartJS - Doughnut</title>

  <link type="text/css" rel="stylesheet" href="css/default.css" />
</head>
<body>

  <div class="chart-container">
    <div class="doughnut-chart-container">
      <canvas id="doughnut-chartcanvas-1"></canvas>
    </div>

    <div class="doughnut-chart-container">
      <canvas id="doughnut-chartcanvas-2"></canvas>
    </div>
  </div>

  <!-- javascript -->
  <script src="js/jquery.min.js"></script>
  <script src="js/Chart.min.js"></script>
  <script src="js/doughnut.js"></script>
</body>
</html>
```



# Displaying Data in Charts-Doughnut Chart

## Doughnut Chart

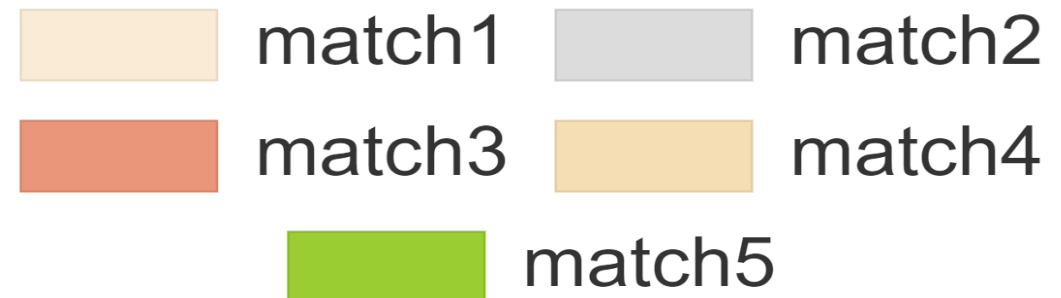
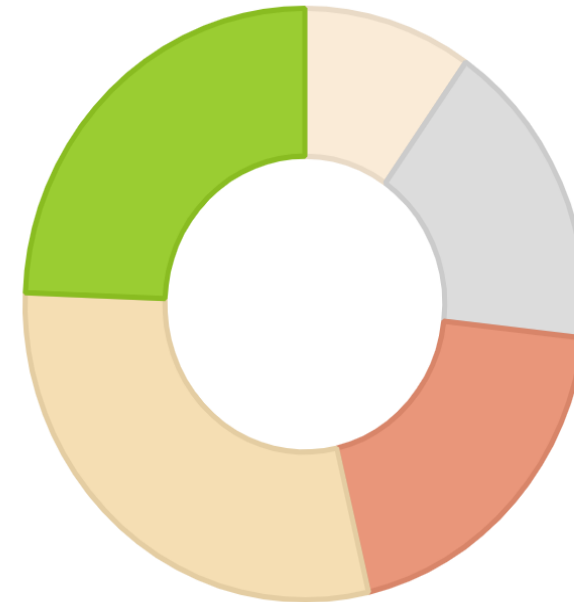
```
<!DOCTYPE html>
<html>
<head>
  <title>ChartJS - Doughnut</title>

  <link type="text/css" rel="stylesheet" href="css/default.css" />
</head>
<body>

  <div class="chart-container">
    <div class="doughnut-chart-container">
      <canvas id="doughnut-chartcanvas-1"></canvas>
    </div>

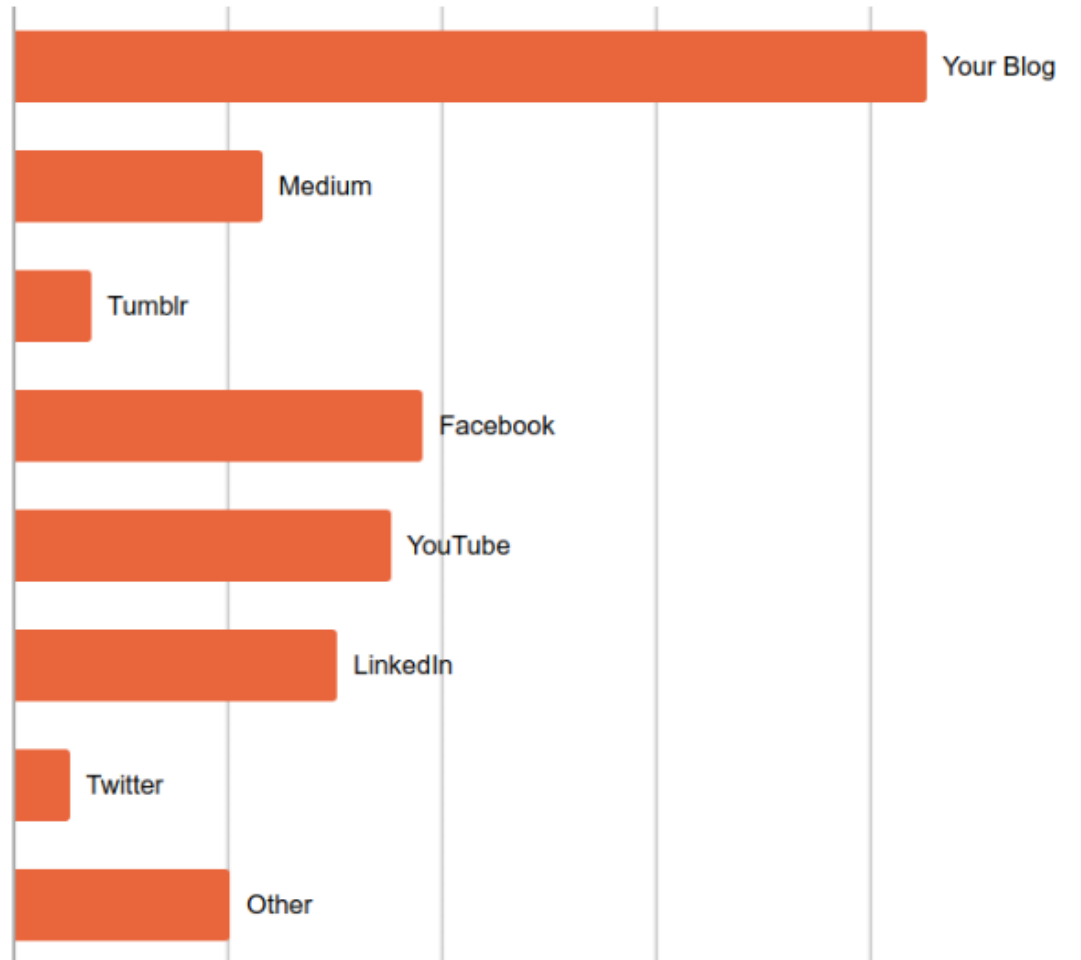
    <div class="doughnut-chart-container">
      <canvas id="doughnut-chartcanvas-2"></canvas>
    </div>
  </div>

  <!-- javascript -->
  <script src="js/jquery.min.js"></script>
  <script src="js/Chart.min.js"></script>
  <script src="js/doughnut.js"></script>
</body>
</html>
```



# Other Types of Visualizations

HTML Bar Graph Example Using Flexbox



<https://www.chartjs.org/docs/latest/getting-started/usage.html>

# Summary of the Today's Lesson

- Data visualizations
  - ▣ Introduction
  - ▣ Significance of data visualizations
  - ▣ Types of visualizations
  - ▣ Tools & libraries
- Creating charts in JS
  - ▣ Discussion of simple charts
  - ▣ Introduction of the visualization library
    - D3
    - chartJS

