15. D3JS Axis API.

D3 provides functions to draw axes. An axis is made of Lines, Ticks and Labels. An axis uses a Scale, so each axis will need to be given a scale to work with.

15.1. Configuring API

You can configure the API directly using the script below.

Example:

```
<script src = "https://d3js.org/d3-axis.v1.min.js"></script>
<body>
   <script>
   </script>
</body>
```

15.2. Axis API Methods

D3 provides the following significant functions to draw axes. They are described in brief as follows.

- ***d3.axisTop()*** − This method is used to create a top horizontal axis.
- ***d3.axisRight()*** − This method is used to create a vertical right-oriented axis.
- ***d3.axisBottom()*** − This method is used to create a bottom horizontal axis.
- ***d3.axisLeft()*** − It creates left vertical axis.

15.3. Working Example

Let us learn how to add the x and y-axis to a graph. To do this, we need to adhere to the steps given below.

Step 1: Define variables − Define SVG variables and data using the coding below.

```
var width = 400, height = 400;
var data = [100, 150, 200, 250, 280, 300];
var svg = d3.select("body")
   .append("svg")
   .attr("width", width)
   .attr("height", height);
```

Step 2: Create a scale linear function − Create a scale linear function for both x and y axis as defined below.

```
var xscale = d3.scaleLinear()
   .domain([0, d3.max(data)])
   .range([0, width - 100]);

var yscale = d3.scaleLinear()
   .domain([0, d3.max(data)])
   .range([height/2, 0]);
```

Here, we have created a linear scale and specified the domain and the range

Step 3: Add scales to x-axis − Now, we can add scales to the x-axis using the following code.

```
var x_axis = d3.axisBottom()
   .scale(xscale);
```

Here, we use d3.axisBottom to create our x-axis and provide it with the scale, which is defined earlier.

Step 4: Add scales to the y-axis − Use the following code to add scales to the y-axis.

```
var y_axis = d3.axisLeft()
   .scale(yscale);
```

Here, we use the d3.axisLeft to create our y-axis and provide it with the scale we defined above.

Step 5: Apply transformation − You can append a group element and insert the x, y axis, which is defined below.

```
svg.append("g")
   .attr("transform", "translate(50, 10)")
   .call(y_axis);
```

Step 6: Append group elements − Apply transition and group elements using the following code.

```
var xAxisTranslate = height/2 + 10;
svg.append("g")
   .attr("transform", "translate(50, " + xAxisTranslate  +")")
   .call(x_axis)
```

Step 7: Working Example − The complete code listing is given in the following code block. Create a webpage axes.html and add the following changes to it.

```
<html>
   <head>
```

```
        <script src = "https://d3js.org/d3.v4.min.js"></script>
        <style>
            svg text {
                fill: purple;
                font: 12px sans-serif;
                text-anchor: end;
            }
        </style>
    </head>

    <body>
        <script>
            var width = 400, height = 400;
            var data = [100, 120, 140, 160, 180, 200];
            var svg = d3.select("body")
                .append("svg")
                .attr("width", width)
                .attr("height", height);

            var xscale = d3.scaleLinear()
                .domain([0, d3.max(data)])
                .range([0, width - 100]);

            var yscale = d3.scaleLinear()
                .domain([0, d3.max(data)])
                .range([height/2, 0]);

            var x_axis = d3.axisBottom().scale(xscale);

            var y_axis = d3.axisLeft().scale(yscale);

            svg.append("g")
                .attr("transform", "translate(50, 10)")
                .call(y_axis);

            var xAxisTranslate = height/2 + 10;
```

```
            svg.append("g")
                .attr("transform", "translate(50, " + xAxisTranslate  +")")
                .call(x_axis)
        </script>
    </body>
</html>
```

Output: Now, request the browser and we will see the following changes.