|  |  |
| --- | --- |
| **Ex. No. 05** | **DATA DRIVEN DOCUMENTS** |
| **09.08.2024** |

**AIM:**

To demonstrate data using D3.js.

**ALGORITHM:**

1. Start

2. Create a new html document.

3. Include the d3.js script in the head section.

4. Define a svg container and add its properties in a js script.

5. Create an object with numeric values to plot in bar chart.

6. Create an appropriate viewbox and add x-axis and y-axis.

7. Open the page and show the plot.

8. Stop

**PROGRAM:**

1. **Dynamic circle:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Static Circle with D3.js</title>

<script src="https://d3js.org/d3.v7.min.js"></script>

<style>

body {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background-color: #f0f0f0;

}

svg {

border: 1px solid #ccc; /\* Adding a border around the graph \*/

}

.circle {

transition: fill 0.3s, stroke 0.3s; /\* Smooth transition for hover effect \*/

}

</style>

</head>

<body>

<svg width="600" height="400"></svg>

<script>

const svg = d3.select("svg");

// Create a static circle with a dark outline and lighter fill

const circle = svg.append("circle")

.attr("class", "circle")

.attr("cx", 300) // Center X position

.attr("cy", 200) // Center Y position

.attr("r", 100) // Radius

.attr("fill", "#add8e6") // Lighter fill color (light blue)

.attr("stroke", "darkblue") // Dark outline color

.attr("stroke-width", 3); // Width of the outline

// Add hover effects

circle.on("mouseover", function() {

d3.select(this)

.attr("fill", "lightcoral") // Change fill color on hover

.attr("stroke", "red"); // Change stroke color on hover

})

.on("mouseout", function() {

d3.select(this)

.attr("fill", "#add8e6") // Reset fill color

.attr("stroke", "darkblue"); // Reset stroke color

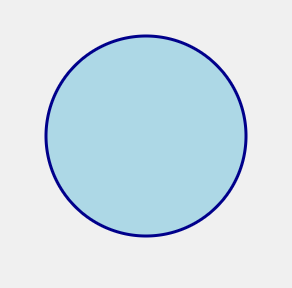
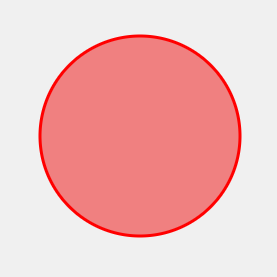
});

</script>

</body>

</html>

**Output:**

****

1. **Bar graph:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>D3.js Bar Chart</title>

<script src="https://d3js.org/d3.v7.min.js"></script>

<style>

.bar { fill: steelblue; transition: fill 0.3s; }

.bar:hover { fill: orange; }

.axis path, .axis line { fill: none; shape-rendering: crispEdges; }

.axis-label { font-size: 14px; font-weight: bold; }

svg { border: 1px solid #ccc; }

</style>

</head>

<body>

<svg width="600" height="400"></svg>

<script>

const data = Array.from({length: 10}, () => Math.floor(Math.random() \* 101));

const names = ["Annette", "Bob","Christy" ,"Deborah", "David", "Grace", "Hannah", "Ian","Jorryn","Levin"];

const svg = d3.select("svg"), margin = { top: 20, right: 30, bottom: 60, left: 60 };

const width = +svg.attr("width") - margin.left - margin.right;

const height = +svg.attr("height") - margin.top - margin.bottom;

const x = d3.scaleBand().domain(names).range([0, width]).padding(0.1);

const y = d3.scaleLinear().domain([0, d3.max(data)]).nice().range([height, 0]);

const g = svg.append("g").attr("transform", `translate(${margin.left},${margin.top})`);

g.append("g").attr("class", "axis axis--x").attr("transform", `translate(0,${height})`)

.call(d3.axisBottom(x));

g.append("g").attr("class", "axis axis--y")

.call(d3.axisLeft(y));

g.selectAll(".bar").data(data).enter().append("rect")

.attr("class", "bar").attr("x", (d, i) => x(names[i])).attr("y", d => y(d))

.attr("width", x.bandwidth()).attr("height", d => height - y(d));

svg.append("text").attr("class", "axis-label").attr("x", width / 2 + margin.left)

.attr("y", height + margin.top + 40).attr("text-anchor", "middle").text("Names");

svg.append("text").attr("class", "axis-label").attr("transform", "rotate(-90)")

.attr("x", -height / 2 - margin.top).attr("y", margin.left - 40)

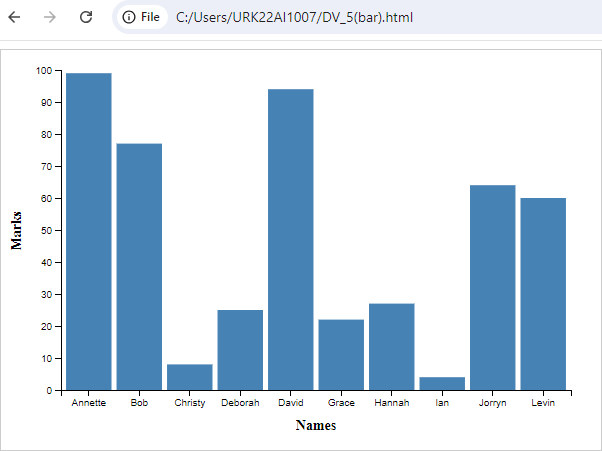
.attr("text-anchor", "middle").text("Marks");

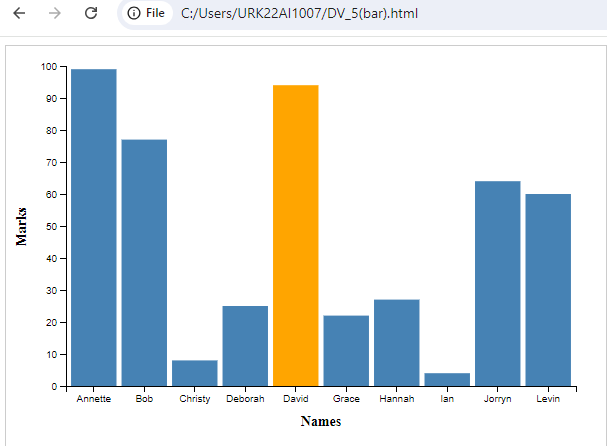
</script>

</body>

</html>

**Output:**

****

****

**RESULT:**

The above D3.js visualization of a dynamic circle and a bar graph is successfully plotted.