EXPERIMENT 8 D11AD 46

AIM: Set Up a D3.js Environment, Select Elements in D3, Modify Elements in D3,Data Loading in D3

D3.js (Data-Driven Documents) is a JavaScript library for manipulating documents based on data. It enables developers to bind arbitrary data to a Document Object Model (DOM), and then apply data-driven transformations to the document. D3.js allows you to create interactive and dynamic data visualizations for the web.

Here are some key concepts and features of D3.js:

1. **Data Binding**: D3.js allows you to bind data to elements in the DOM, making it easy to create visual representations of the data. You can bind arrays, objects, or even functions to DOM elements.
2. **DOM Manipulation**: D3.js provides powerful methods for manipulating the DOM based on data. You can create, update, and remove DOM elements dynamically in response to changes in the data.
3. **Data Visualization**: D3.js is commonly used for creating data visualizations such as charts, graphs, maps, and interactive dashboards. It provides a wide range of built-in methods for generating common types of visualizations, as well as the flexibility to create custom visualizations.
4. **Scalability**: D3.js is designed to handle large datasets efficiently. It leverages the power of modern web browsers to efficiently render and manipulate large amounts of data.
5. **Transitions and Animations**: D3.js allows you to create smooth transitions and animations to enhance the interactivity and usability of your visualizations. You can animate changes in data, attributes, styles, and more.
6. **SVG and Canvas Support**: D3.js supports both Scalable Vector Graphics (SVG) and HTML5 Canvas for rendering visualizations. SVG is well-suited for creating interactive and responsive graphics, while Canvas is more suitable for rendering complex graphics with high performance.
7. **Modularity and Extensibility**: D3.js is modular and extensible, allowing you to use only the parts of the library that you need for your project. It also provides a rich ecosystem of plugins and extensions for extending its functionality.
8. **Community and Documentation**: D3.js has a large and active community of developers who contribute to its development and maintenance. The library is well-documented, with comprehensive API documentation, tutorials, and examples available to help you get started.

Overall, D3.js is a powerful tool for creating data-driven web applications and visualizations. Whether you're building simple charts or complex interactive dashboards, D3.js provides the flexibility and functionality you need to bring your data to life on the web.

```
%%html
<!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 Example</title>
    <script src="https://d3js.org/d3.v7.min.js"></script>
    <style>
        .bar {
            fill: steelblue;
        }
        .bar:hover {
            fill: orange;
        }
    </style>
</head>
<body>
    <svg id="chart"></svg>

    <script>
        var data = [10, 20, 30, 40, 50];

        var svgWidth = 500;
        var svgHeight = 300;

        var barWidth = svgWidth / data.length;
```

```
        var svg = d3.select("#chart")
            .attr("width", svgWidth)
            .attr("height", svgHeight);

        var bars = svg.selectAll("rect")
            .data(data)
            .enter()
            .append("rect")
            .attr("class", "bar")
            .attr("x", function(d, i) { return i * barWidth; })
            .attr("y", function(d) { return svgHeight - d * 5; })
            .attr("width", barWidth - 1)
            .attr("height", function(d) { return d * 5; });

        var texts = svg.selectAll("text")
            .data(data)
            .enter()
            .append("text")
            .text(function(d) { return d; })
            .attr("x", function(d, i) { return i * barWidth + barWidth / 2; })
            .attr("y", function(d) { return svgHeight - (d * 5) + 15; })
            .attr("text-anchor", "middle")
            .attr("font-family", "sans-serif")
            .attr("font-size", "12px")
            .attr("fill", "white");
    </script>
</body>
</html>
```