

## D3.js Lab Assignment

### Q1: Selecting & Modifying Elements

-----

Task: Use D3.js to select a paragraph (<p>) and change its text content.

Instructions:

1. Select a <p> element using d3.select().
2. Change its text content using .text("New Text").

### Q2: Appending Elements

-----

Task: Append an <h1> element dynamically to a <div> using D3.js.

Instructions:

1. Select the <div> where the new <h1> should be added.
2. Use .append("h1") to create an <h1> element inside the <div>.
3. Set the text of the <h1> element using .text("Hello, D3.js!").

### Q3: Styling Elements

-----

Task: Select multiple elements and change their styles dynamically.

Instructions:

1. Select all <p> elements using d3.selectAll().
2. Use .style() to change their color and font size.
3. Apply "blue" color and font size "20px".

### Q4: Data Binding

-----

Task: Bind an array of data (["Apple", "Banana", "Cherry", "Pineapple", "Mango"]) to an unordered list (<ul>) and create <li> items for each value.

Instructions:

1. Use .selectAll("li") to bind data to list items.
2. Use .data(array) to bind the dataset.
3. Use .enter().append("li") to create <li> elements.
4. Use .text(d => d) to set text content.

#### Q5: Creating an SVG Circle

---

Task: Create an SVG element and add a red circle inside it.

Instructions:

1. Create an `<svg>` element with width & height 200px.
2. Append a `<circle>` inside the `<svg>`.
3. Set attributes: `cx=100, cy=100, r=50, fill="red"`.

#### Q6: Using Scales

---

Task: Use a linear scale to map values from [0,100] to [0,500] and display the scaled value of 50.

Instructions:

1. Create a `d3.scaleLinear()` scale with domain [0,100] and range [0,500].
2. Compute the scaled value for 50 using `.scale(50)`.
3. Display the output inside a `<p>` element.

#### Q7: Adding an Axis

---

Task: Create an X-axis using D3.js and display it inside an SVG.

Instructions:

1. Define an `xScale` with a domain [0,100] mapped to [0,400].
2. Create an axis using `d3.axisBottom(xScale)`.
3. Append the axis to an `<svg>` and position it using `.attr("transform")`.

#### Q8: Creating a Bar Chart

---

Task: Create a simple bar chart using D3.js with values [10, 20, 30, 40, 50, 60, 70, 80].

Instructions:

1. Select an `<svg>` element.
2. Bind data to `rect` elements.
3. Set x position based on the index.
4. Set y position, width=40, and height proportional to data.
5. Use `.attr("fill", "steelblue")` to color the bars.

### Q9: Implementing Transitions

---

Task: Create a green circle and animate its radius from 20 to 50.

Instructions:

1. Create an `<svg>` with a `<circle>`.
2. Select the `<circle>` using D3.js.
3. Use `.transition().duration(2000).attr("r", 50)` to animate the radius.

### Q10: Creating a Line Chart

---

Task: Draw a simple line chart using D3.js with points {x: 10, y: 30}, {x: 20, y: 80}, {x: 30, y: 50}, {x: 40, y: 100}.

Instructions:

1. Define an `xScale` and `yScale` to map values to pixel coordinates.
2. Use `d3.line()` to create a line generator.
3. Use `.datum(data).attr("d", line)` to append the line to an `<svg>`.
4. Set `stroke="red"` and `fill="none"`.

- ① Write code in an HTML file and include the D3.js library (`<script src="https://d3js.org/d3.v7.min.js"></script>`).
- ② Complete each question in a separate `<script>` block or separate files.
- ③ Try modifying attributes and experimenting with different data.
- ④ For questions requiring SVG, ensure you create an SVG container.

**Note - Don't submit a zip file. Follow proper naming conventions.**