Husain Mehdi
F20DV
Lab 2
Demonstrated 25 Feb 2022 to Amit Parekh

EX1

```html
<!DOCTYPE html>
<html lang="en">

<body>

    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
    <style>
        div {
            display: inline-block;
            width: 10px;
            height: 10px;
            margin: 1px;
            padding: 10px;
        }
    </style>
    <script>

        var body = d3.select('body');
        var myColor1 = d3.scaleSequential().domain([1, 10]).interpolator(d3.interpolateCool);

        // Set Dimensions
        const xSize = 600; const ySize = 600;
        const margin = 40;
        const xMax = xSize - margin * 2;
        const yMax = ySize - margin * 2;

        // Create Random Points
        const numPoints = 100;
        const data = [];
        for (let i = 0; i < numPoints; i++) { data.push({ x: i / 100, y: Math.sin(6.2 * i / 100) }); }




        // Get the 'limits' of the data - the full extent (mins and max)
        // so the plotted data fits perfectly
        const xExtent = d3.extent(data, d => { return d.x });
        const yExtent = d3.extent(data, d => { return d.y });

        // Append SVG Object to the Page
        const svg = d3.select("body")
            .append("svg")
            .attr('width', xSize)
            .attr('height', ySize)
            .append("g")
            .attr("transform", "translate(" + margin + "," + margin + ")");

        // X Axis
        const x = d3.scaleLinear()
            .domain([xExtent[0], xExtent[1]])
            .range([0, xMax]);

        // bottom
        svg.append("g")
            .attr("transform", "translate(0," + yMax + ")")
            .call(d3.axisBottom(x))
```

```
56          // top
57          svg.append("g")
58              .call(d3.axisTop(x));
59
60          // Y Axis
61          const y = d3.scaleLinear()
62              .domain([yExtent[0], yExtent[1]])
63              .range([yMax, 0]);
64
65          // left y axis
66          svg.append("g")
67              .call(d3.axisLeft(y));
68
69          // right y axis
70          svg.append("g")
71              .attr("transform", `translate(${yMax},0)`)
72              .call(d3.axisRight(y));
73
74          // Add the line
75          svg.append("path")
76              .datum(data)
77              .attr("fill", "none")
78              .attr("stroke", "steelblue")
79              .attr("stroke-width", 1.5)
80              .attr("d", d3.line()
81                  .x(function (d) { return x(d.x) })
82                  .y(function (d) { return y(d.y) })
83              );
84
85          var triangle = d3.symbol()
86              .type(d3.symbolTriangle)
87              .size(50)
88
89          svg.selectAll("dot")
90              .data(data)
91              .enter()
92              .append("circle")
93              .attr("cx", function (d) { return x(d.x) })
94              .attr("cy", function (d) { return y(d.y) })
```

```
110             transform-origin: 0% 0%;
111             position:relative;
112            }
113            .pulse:hover
114            {
115             -webkit-animation-name: pulsar;
116             -webkit-animation-duration: 1s;
117             -webkit-animation-iteration-count: infinite;
118             -webkit-animation-direction: alternate;
119             animation-name: pulsar;
120             animation-duration: 1s;
121             animation-iteration-count: infinite;
122             animation-direction: alternate;
123             -webkit-transform-origin: 0% 0%;
124             transform-origin: 0% 0%;
125            }
126            @keyframes pulsar {
127             from {
128             }
129             to {
130
131             r: 5%;
132             transform-origin: 0% 0%;
133             z-index:-100;
134             }
135            }
136            </style>
137            <svg class='pulse' width='30' height='30'></svg>
138
139    </body>
140
```

EX2

```html
<!DOCTYPE html>
<html lang="en">

<body>
    <style>
        .b {
            padding: 20px;
        }

        .c {
            background-color: yellow;
            padding: 20px;
            display: none;
        }


        .b:hover .c {
            position: absolute;
            display: inline-block;
        }
    </style>
    <script src='https://d3js.org/d3.v7.min.js'></script>


    <script>
        var myData = ['a', 'b', 'c', 'd'];

        d3.select("body")
            .selectAll("div")
            .data(myData)
            .enter()
            .append('div')
            .attr("class", "b")
            .text(function (d, i) {
                return d;
            })
            .append('div')
            .attr("class", "c")
            .text("hovertext")
```

EX3

```html
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>

    <script>
        d3.select('body')
            .append('div')
            .text("lorum ipsum")
            .style('width', '100px')
            .style('height', '20px')
            .style('background-color', 'green')
            .style("color", "black");

        d3.selectAll("div")
            .on("mouseover", function (event) {
                d3.select(this)
                    .style("background-color", "orange")
                    .style("height", "100px")
                    .style("color", "red")
                    .style("border-style", "double")
                    .style("border-width", "5px");

                // Get current event info
                // Note: d3.event (event) passed as the first argument to all listeners
                console.log(event);

                // Get x & y co-ordinates
                // Note: d3.mouse was removed in d3v6, you should use d3.pointer(event)
                console.log(d3.pointer(event));
            })
            .on("mouseout", function () {
                d3.select(this)
                    .style("background-color", "steelblue")
                    .style("height", "20px")
                    .style("color", "blue")
                    .style("border-style", "none");
            });
    </script>
</body>
```

EX4

```html
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>

    <script>
        var svg = d3.select("body")
            .append("svg")
            .attr("width", 400)
            .attr("height", 400)
            .style("border", '1px solid green');

        svg.append("circle")
            .attr("cx", 200)
            .attr("cy", 200)
            .attr("r", 10)
            .attr("fill", "green");

        d3.selectAll("circle")
            .on("mouseover", function (event) {
                d3.select(this)
                    .attr("r", 100)
                    .attr("fill", "red");

                // Get current event info
                // Note: d3.event (event) passed as the first argument to all listeners
                console.log(event);

                // Get x & y co-ordinates
                // Note: d3.mouse was removed in d3v6, you should use d3.pointer(event)
                console.log(d3.pointer(event));
            })
            .on("mouseout", function () {
                d3.select(this)
                    .attr("r", 10)
                    .attr("fill", "green")
            });
    </script>
</body>
```

EX5

```html
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>

    <script>

        var div = d3.select("body").append("div")
            .attr("class", "tooltip")
            .style("opacity", 0);

        var svg = d3.select("body")
            .append("svg")
            .attr("width", 400)
            .attr("height", 400)
            .style("border", '1px solid green')
            .on('mousemove', (event) => {
                var coords = d3.pointer(event);
                console.log(coords[0], coords[1]) // log the mouse x,y position
                div.transition()
                    .duration(200)
                    .style("opacity", .9);
                div.html(coords[0] + ", " + coords[1])
                    .style("left", (coords[0]) + "px")
                    .style("top", (coords[1]) + "px");
            });

        svg.append("circle")
            .attr("cx", 200)
            .attr("cy", 200)
            .attr("r", 10)
            .attr("fill", "green")
```

```
34
35
36          d3.selectAll("circle")
37              .on("mouseover", function (event) {
38                  d3.select(this)
39                      .attr("r", 100)
40                      .attr("fill", "red")
41                  e => console.log(d3.pointer(e));
42
43                  // Get current event info
44                  // Note: d3.event (event) passed as the first argument to all listeners
45                  console.log(event);
46
47                  // Get x & y co-ordinates
48                  // Note: d3.mouse was removed in d3v6, you should use d3.pointer(event)
49                  console.log(d3.pointer(event));
50              })
51              .on("mouseout", function () {
52                  d3.select(this)
53                      .attr("r", 10)
54                      .attr("fill", "green");
55              });
56      </script>
57      <style>
58          div.tooltip {
59              position: absolute;
60              text-align: center;
61              width: 60px;
62              height: 28px;
63              padding: 2px;
64              font: 12px sans-serif;
65              background: lightsteelblue;
66              border: 0px;
67              border-radius: 8px;
68              pointer-events: none;
69          }
70      </style>
71  </body>
```

EX6

```
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
    <script>
        d3.select('body')
            .append("div")
            .style('width', '100px')
            .style('height', '100px')
            .style('background-color', 'blue')
            .transition()
            .duration(1000)
            .style("background-color", "red")
            .transition()
            .duration(2000)
            .style("background-color", "green");
    </script>
</body>
```

EX7

```
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
    <script>
        d3.select('body')
            .append("div")
            .style('width', '100px')
            .style('height', '100px')
            .style('background-color', 'blue')
            .transition()
            .duration(1000)
            .style("background-color", "red")
            .style('width', '50px')
            .style('height', '50px')
            .transition()
            .duration(2000)
            .style("background-color", "green")
            .style('width', '200px')
            .style('height', '200px');
    </script>
</body>
```

EX8

```
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
    <script>
        d3.select('body')
            .append("div")
            .style('width', '100px')
            .style('height', '100px')
            .style('background-color', 'blue');



        d3.selectAll("div")
            .on("mouseover", function (event) {
            d3.select(this).transition()
            .duration(1000)
            .style("background-color", "red")
            .style('width', '50px')
            .style('height', '50px');
            })
            .on("mouseout", function () {
            d3.select(this).transition()
            .duration(2000)
            .style("background-color", "green")
            .style('width', '200px')
            .style('height', '200px')})

    </script>
</body>
```

EX9

```html
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
    <script>
        d3.select('body')
            .append("div")
            .style('width', '100px')
            .style('height', '100px')
            .style('background-color', 'blue')
            .style('transform', 'scale(1.0)')
            .transition()
            .ease(d3.easeBounce)
            .duration(1000)
            .style("background-color", "red")
            .style('transform', 'scale(0.5)')

        d3.select('body')
            .append("div")
            .style('width', '100px')
            .style('height', '100px')
            .style('background-color', 'blue')
            .style('transform', 'scale(1.0)')
            .transition()
            .ease(d3.easeQuadOut)
            .duration(1000)
            .style("background-color", "red")
            .style('transform', 'scale(0.5)')

        d3.select('body')
            .append("div")
            .style('width', '100px')
            .style('height', '100px')
            .style('background-color', 'blue')
            .style('transform', 'scale(1.0)')
            .transition()
            .ease(d3.easeElasticIn.amplitude(1).period(0.3))
            .duration(1000)
            .style("background-color", "red")
            .style('transform', 'scale(0.5)')
    </script>
</body>
```

EX10

```html
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
    <script>

        var svg = d3.select("body")
            .append("svg")
            .attr("width", 400)
            .attr("height", 400)
            .style("border", '1px solid green');

        svg.append("circle")
            .attr("cx", 200)
            .attr("cy", 200)
            .attr("r", 10)
            .attr("fill", "green");

        d3.selectAll("circle")
            .on("mouseover", function (event) {
                d3.select(this)
                    .transition()
                    .ease(d3.easeBounce)
                    .duration(1000)
                    .attr("r", 100)
                    .attr("fill", "red");
            })
            .on("mouseout", function () {
                d3.select(this)
                    .transition()
                    .ease(d3.easeBounce)
                    .duration(1000)
                    .attr("r", 10)
                    .attr("fill", "green")
            });

    </script>
</body>
```

EX11

```
<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
    <script>
        var svg = d3.select("body")
            .append("svg")
            .attr("width", 400)
            .attr("height", 400)
            .style("border", '1px solid green');

        svg.append("circle")
            .attr("cx", 200)
            .attr("cy", 200)
            .attr("r", 10)
            .attr("fill", "green");

        svg.append("text")
            .attr("x", 200)
            .attr("y", 350)
            .text("hi");

        d3.selectAll("circle")
            .on("mouseover", function (event) {
                d3.select(this)
                    .transition()
                    .ease(d3.easeBounce)
                    .duration(1000)
                    .attr("r", 100)
                    .attr("fill", "red");
            })
            .on("mouseout", function () {
                d3.select(this)
                    .transition()
                    .ease(d3.easeBounce)
                    .duration(1000)
                    .attr("r", 10)
                    .attr("fill", "green")
            });

            d3.selectAll("text")
            .on("mouseover", function (event) {
                d3.select(this)
                    .transition()
                    .ease(d3.easeBounce)
                    .duration(1000)
                    .attr("font-size", 100)
                    .attr("fill", "red");
            })
            .on("mouseout", function () {
                d3.select(this)
                    .transition()
                    .ease(d3.easeBounce)
                    .duration(1000)
                    .attr("font-size", 10)
                    .attr("fill", "green")
            });
```

EX12

```
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
    <script>

        var svg = d3.select("body")
            .append("svg")
            .attr("width", 500)
            .attr("height", 500);

        var bar1 = svg.append("rect")
            .attr("fill", "blue")
            .attr("x", 100)
            .attr("y", 20)
            .attr("height", 20)
            .attr("width", 10)

        var bar2 = svg.append("rect")
            .attr("fill", "blue")
            .attr("x", 120)
            .attr("y", 20)
            .attr("height", 20)
            .attr("width", 10)

        var bar3 = svg.append("rect")
            .attr("fill", "blue")
            .attr("x", 140)
            .attr("y", 20)
            .attr("height", 20)
            .attr("width", 10)


        update();

        function update() {
            bar1.transition()
                .ease(d3.easeLinear)
                .duration(2000)
                .attr("height", 100)

            bar2.transition()
                .ease(d3.easeLinear)
                .duration(2000)
                .delay(2000)
                .attr("height", 100)


            bar3.transition()
                .ease(d3.easeLinear)
                .duration(2000)
                .delay(4000)
                .attr("height", 100)
        }
    </script>
</body>
```

Ex13

```
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
    <script>

        var svg = d3.select("body")
            .append("svg")
            .attr("width", 500)
            .attr("height", 500);

        var bar1 = svg.append("rect")
            .attr("fill", "blue")
            .attr("x", 100)
            .attr("y", 20)
            .attr("height", 20)
            .attr("width", 10)

        var bar2 = svg.append("rect")
            .attr("fill", "blue")
            .attr("x", 120)
            .attr("y", 20)
            .attr("height", 20)
            .attr("width", 10)

        var bar3 = svg.append("rect")
            .attr("fill", "blue")
            .attr("x", 140)
            .attr("y", 20)
            .attr("height", 20)
            .attr("width", 10)


        update();

        function update() {
            bar1.transition()
                .ease(d3.easeLinear)
                .duration(2000)
                .attr("height", 100)
                .end()
                .then(function() {
                    bar1.transition()
                        .ease(d3.easeLinear)
                        .duration(2000)
                        .attr("height", 20);})

            bar2.transition()
                .ease(d3.easeLinear)
                .duration(2000)
                .delay(2000)
                .attr("height", 100)
                .end()
                .then(function() {
```

```
57                          .ease(d3.easeLinear)
58                          .duration(2000)
59                          .attr("height", 20);})
60
61              bar3.transition()
62                  .ease(d3.easeLinear)
63                  .duration(2000)
64                  .delay(4000)
65                  .attr("height", 100)
66                  .end()
67                  .then(function() {
68                      bar3.transition()
69                          .ease(d3.easeLinear)
70                          .duration(2000)
71                          .attr("height", 20);})
72          }
73
74      </script>
75  </body>
```

EX14

```html
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
    <script>

        var svg = d3.select("body")
            .append("svg")
            .attr("width", 500)
            .attr("height", 500);

        var bar1 = svg.append("rect")
            .attr("fill", "blue")
            .attr("x", 100)
            .attr("y", 20)
            .attr("height", 20)
            .attr("width", 10)

        var bar2 = svg.append("rect")
            .attr("fill", "blue")
            .attr("x", 120)
            .attr("y", 20)
            .attr("height", 20)
            .attr("width", 10)

        var bar3 = svg.append("rect")
            .attr("fill", "blue")
            .attr("x", 140)
            .attr("y", 20)
            .attr("height", 20)
            .attr("width", 10)


        update();

        function update() {
            bar1.transition()
                .ease(d3.easeLinear)
                .duration(2000)
                .attr("height", 100)
                .attr("fill", "red")
                .end()
                .then(function () {
                    bar1.transition()
                        .ease(d3.easeLinear)
                        .duration(2000)
                        .attr("height", 20)
                        .attr("fill", "blue")
                        ;
                })

            bar2.transition()
                .ease(d3.easeLinear)
```

```
55                  .duration(2000)
56                  .delay(2000)
57                  .attr("height", 100)
58                  .attr("fill", "red")
59                  .end()
60                  .then(function () {
61                      bar2.transition()
62                          .ease(d3.easeLinear)
63                          .duration(2000)
64                          .attr("height", 20)
65                          .attr("fill", "blue")
66                          ;
67                  })
68
69              bar3.transition()
70                  .ease(d3.easeLinear)
71                  .duration(2000)
72                  .delay(4000)
73                  .attr("height", 100)
74                  .attr("fill", "red")
75                  .end()
76                  .then(function () {
77                      bar3.transition()
78                          .ease(d3.easeLinear)
79                          .duration(2000)
80                          .attr("height", 20)
81                          .attr("fill", "blue")
82                          ;
83                  })
84          }
85
86      </script>
87  </body>
```

EX15

```
<body>

    <style>
        .bar {
            fill: steelblue;
        }

        .highlight {
            fill: orange;
        }
    </style>

    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>

    <svg width="600" height="500"></svg>

    <script>
        var svg = d3.select("svg");
        var margin = 200;
        var width = svg.attr("width") - margin;
        var height = svg.attr("height") - margin;

        svg.append("text")
            .attr("transform", "translate(100,0)")
            .attr("x", 50)
            .attr("y", 50)
            .attr("font-size", "24px")
            .text("Stock Price")

        var x = d3.scaleBand().range([0, width]).padding(0.4);
        var y = d3.scaleLinear().range([height, 0]);
        var g = svg.append("g")
            .attr("transform", "translate(" + 100 + "," + 100 + ")");

        d3.csv("https://raw.githubusercontent.com/HusainMehdi/DataVis/main/csvfile.csv").then(function (data) {
            x.domain(data.map(function (d) { return d.year; }));
            y.domain([0, d3.max(data, function (d) { return d.value; })]);
            g.append("g")
                .attr("transform", "translate(0," + height + ")")
                .call(d3.axisBottom(x))
                .append("text")
                .attr("y", height - 250)
                .attr("x", width - 100)
                .attr("text-anchor", "end")
                .attr("stroke", "black")
                .text("Year");
            g.append("g")
                .call(d3.axisLeft(y).tickFormat(function (d) {
                    return "$" + d;
                }).ticks(10))
                .append("text")
                .attr("transform", "rotate(-90)")
                .attr("y", 6)
                .attr("dy", "-5.1em")
                .attr("text-anchor", "end")
```

```
            .attr("stroke", "black")
            .text("Stock Price");
        g.selectAll(".bar")
            .data(data)
            .enter().append("rect")
            .attr("class", "bar")
            // .on(….. ) - call mouse events here…

            .on("mouseover", function onMouseOver(d, i) {
                d3.select(this).attr('class', 'highlight');
                d3.select(this)
                    .transition() // adds animation
                    .duration(400)
                    .attr('width', x.bandwidth() + 5)
                    .attr("y", function (d) { return y(d.value) - 10; })
                    .attr("height", function (d) { return height - y(d.value) + 10; });
                g.append("text")
                    .attr('class', 'val')
                    .attr('x', function () {
                        return x(d.year);
                    })
                    .attr('y', function () {
                        return y(d.value) - 15;
                    })
                    .text(function (d) { return '$' + i.value; }); // Value of the text
            })

            .on("mouseout", function onMouseOut(d, i) {
                // use the text label class to remove label on mouseout
                d3.select(this).attr('class', 'bar');
                d3.select(this)
                    .transition() // adds animation
                    .duration(400)
                    .attr('width', x.bandwidth())
                    .attr("y", function (d) { return y(i.value); })
                    .attr("height", function (d) { return height - y(i.value); });
                d3.selectAll('.val')
                    .remove()
            })


            .attr("x", function (d) { return x(d.year); })
            .attr("y", function (d) { return y(d.value); })
            .attr("width", x.bandwidth())
            .transition()
            .ease(d3.easeLinear)
            .duration(400)
            .delay(function (d, i) {
                return i * 50;
            })
            .attr("height", function (d) { return height - y(d.value); });
    });


</script>
</body>
```

EX16

```html
<!DOCTYPE html>
<html lang="en">

<body>

    <style>
        .bar {
            fill: steelblue;
        }

        .highlight {
            fill: orange;
        }
    </style>

    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>

    <svg width="600" height="500"></svg>

    <script>
        var svg = d3.select("svg");
        var margin = 200;
        var width = svg.attr("width") - margin;
        var height = svg.attr("height") - margin;

        svg.append("text")
            .attr("transform", "translate(100,0)")
            .attr("x", 50)
            .attr("y", 50)
            .attr("font-size", "24px")
            .text("Stock Price")

        var x = d3.scaleBand().range([0, width]).padding(0.4);
        var y = d3.scaleLinear().range([height, 0]);
        var g = svg.append("g")
            .attr("transform", "translate(" + 100 + "," + 100 + ")");

        d3.csv("https://raw.githubusercontent.com/HusainMehdi/DataVis/main/csvfile.csv").then(function (data) {
            x.domain(data.map(function (d) { return d.year; }));
            y.domain([0, d3.max(data, function (d) { return d.value; })]);
            g.append("g")
                .attr("transform", "translate(0," + height + ")")
                .call(d3.axisBottom(x))
                .append("text")
                .attr("y", height - 250)
                .attr("x", width - 100)
                .attr("text-anchor", "end")
                .attr("stroke", "black")
                .text("Year");
            g.append("g")
                .call(d3.axisLeft(y).tickFormat(function (d) {
                    return "$" + d;
                }).ticks(10))
                .append("text")
                .attr("transform", "rotate(-90)")
                .attr("y", 6)
```

```
            .attr("dy", "-5.1em")
            .attr("text-anchor", "end")
            .attr("stroke", "black")
            .text("Stock Price");
        g.selectAll(".bar")
            .data(data)
            .enter().append("rect")
            .attr("class", "bar")
            // .on(….. ) - call mouse events here…

            .on("mouseover", function onMouseOver(d, i) {
                d3.select(this).attr('class', 'highlight');
                d3.select(this)
                    .transition() // adds animation
                    .duration(400)
                    .attr('width', x.bandwidth() + 5)
                    .attr("y", function (d) { return y(d.value) - 10; })
                    .attr("height", function (d) { return height - y(d.value) + 10; });

                g.append("text")
                    .attr('class', 'val')
                    .attr('x', function (d) {
                        return x(i.year);
                    })
                    .attr('y', function (d) {
                        return y(i.value) - 15;
                    })
                    .text(function (d) { return '$' + i.value; }); // Value of the text
            })

            .on("mouseout", function onMouseOut(d, i) {
                // use the text label class to remove label on mouseout
                d3.select(this).attr('class', 'bar');
                d3.select(this)
                    .transition() // adds animation
                    .duration(400)
                    .attr('width', x.bandwidth())
                    .attr("y", function (d) { return y(i.value); })
                    .attr("height", function (d) { return height - y(i.value); });
                d3.selectAll('.val')
                    .remove()
            })


            .attr("x", function (d) { return x(d.year); })
            .attr("y", function (d) { return y(d.value); })
            .attr("width", x.bandwidth())
            .transition()
            .ease(d3.easeLinear)
            .duration(400)
            .delay(function (d, i) {
                return i * 50;
            })
            .attr("height", function (d) { return height - y(d.value); });
    });
```

EX17

```html
<!DOCTYPE html>
<html lang="en">

<body>

    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>

    <svg width="600" height="500"></svg>

    <script>
        var svg = d3.select("svg");
        var margin = 200;
        var width = svg.attr("width") - margin;
        var height = svg.attr("height") - margin;

        var myColor1 = d3.scaleLinear().domain([1, 10])
            .range(["steelblue", "indianred"]);

        var myColor2 = d3.scaleLinear().domain([1, 10])
            .range(["darkblue", "darkred"]);

        var values = [];

        svg.append("text")
            .attr("transform", "translate(100,0)")
            .attr("x", 50)
            .attr("y", 50)
            .attr("font-size", "24px")
            .text("Stock Price")

        var x = d3.scaleBand().range([0, width]).padding(0.4);
        var y = d3.scaleLinear().range([height, 0]);
        var g = svg.append("g")
            .attr("transform", "translate(" + 100 + "," + 100 + ")");

        d3.csv("https://raw.githubusercontent.com/HusainMehdi/DataVis/main/csvfile.csv").then(function (data) {

            for (var i = 0; i < data.length; i++) {
                values.push(data[i].value);
            }

            var scale = d3.scaleLinear()
                .domain([d3.min(values), d3.max(values)])
                .range([0, 10]);

            x.domain(data.map(function (d) { return d.year; }));
            y.domain([0, d3.max(data, function (d) { return d.value; })]);
            g.append("g")
                .attr("transform", "translate(0," + height + ")")
                .call(d3.axisBottom(x))
                .append("text")
                .attr("y", height - 250)
                .attr("x", width - 100)
                .attr("text-anchor", "end")
                .attr("stroke", "black")
                .text("Year");
```

```
            .text("Year");
        g.append("g")
            .call(d3.axisLeft(y).tickFormat(function (d) {
                return "$" + d;
            }).ticks(10))
            .append("text")
            .attr("transform", "rotate(-90)")
            .attr("y", 6)
            .attr("dy", "-5.1em")
            .attr("text-anchor", "end")
            .attr("stroke", "black")
            .text("Stock Price");
        g.selectAll(".bar")
            .data(data)
            .enter().append("rect")
            .attr("class", "bar")
            .style("fill", function (d) {
                return myColor1(scale(d.value));
            })
            // .on(….. ) – call mouse events here…

            .on("mouseover", function onMouseOver(d, i) {
                d3.select(this).attr('class', 'highlight');
                d3.select(this)
                    .transition() // adds animation
                    .duration(400)
                    .attr('width', x.bandwidth() + 5)
                    .attr("y", function (d) { return y(d.value) - 10; })
                    .attr("height", function (d) { return height - y(d.value) + 10; })
                    .style("fill", function (d) {
                        return myColor2(scale(d.value))
                    });


                g.append("text")
                    .attr('class', 'val')
                    .attr('x', function (d) {
                        return x(i.year);
                    })
                    .attr('y', function (d) {
                        return y(i.value) - 15;
                    })
                    .text(function (d) { return '$' + i.value; }); // Value of the text
            })

            .on("mouseout", function onMouseOut(d, i) {
                // use the text label class to remove label on mouseout
                d3.select(this).attr('class', 'bar');
                d3.select(this)
                    .transition() // adds animation
                    .duration(400)
                    .attr('width', x.bandwidth())
                    .attr("y", function (d) { return y(i.value); })
                    .attr("height", function (d) { return height - y(i.value); })
                    .style("fill", function (d) {
                        return myColor1(scale(d.value));
```

```
                    });
                d3.selectAll('.val')
                    .remove()
            })


            .attr("x", function (d) { return x(d.year); })
            .attr("y", function (d) { return y(d.value); })
            .attr("width", x.bandwidth())
            .transition()
            .ease(d3.easeLinear)
            .duration(400)
            .delay(function (d, i) {
                return i * 50;
            })
            .attr("height", function (d) { return height - y(d.value); });
    });


</script>
</body>
```

EX18

```html
<!DOCTYPE html>
<html lang="en">

<body>

    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>

    <!-- Add buttons -->
    <button onclick="update(data1)">Variable 1</button>
    <button onclick="update(data2)">Variable 2</button>
    <button onclick="update(data3)">Variable 3</button>

    <script>
        // create 2 data_set
        const data1 = [
            { group: "A", value: 5 },
            { group: "B", value: 20 },
            { group: "C", value: 9 }
        ];
        const data2 = [
            { group: "A", value: 10 },
            { group: "B", value: 2 },
            { group: "C", value: 22 }
        ];
        const data3 = [
            { group: "A", value: 6 },
            { group: "B", value: 12 },
            { group: "C", value: 18 }
        ];

        // set the dimensions and margins of the graph
        const margin = { top: 30, right: 30, bottom: 70, left: 60 };
        const width = 460 - margin.left - margin.right;
        const height = 400 - margin.top - margin.bottom;

        // append the svg object to the body of the page
        var svg = d3.select('body')
            .append('div')
            .append("svg")
            .attr("width", width + margin.left + margin.right)
            .attr("height", height + margin.top + margin.bottom)
            .append("g")
            .attr("transform",
                "translate(" + margin.left + "," + margin.top + ")");

        // X axis
        var x = d3.scaleBand()
            .range([0, width])
            .domain(data1.map(function (d) { return d.group; }))
            .padding(0.2);
        svg.append("g")
            .attr("transform", "translate(0," + height + ")")
            .call(d3.axisBottom(x))

        // Add Y axis
        var y = d3.scaleLinear()
```

```
            .domain([0, 20])
            .range([height, 0]);
        svg.append("g")
            .attr("class", "myYaxis")
            .call(d3.axisLeft(y));

        // A function that create / update the plot for a given variable:
        function update(data) {
            var u = svg.selectAll("rect")
                .data(data)
            u.enter()
                .append("rect")
                .merge(u)
                .transition()
                .duration(1000)
                .attr("x", function (d) { return x(d.group); })
                .attr("y", function (d) { return y(d.value); })
                .attr("width", x.bandwidth())
                .attr("height", function (d) { return height - y(d.value); })
                .attr("fill", "#69b3a2")
        }
        // Initialize the plot with the first dataset
        update(data1)
    </script>
</body>
```

EX19

```html
<!DOCTYPE html>
<html lang="en">

<body>

    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>

    <!-- Add buttons -->
    <button onclick="update(data1, 'goldenrod' )">Variable 1</button>
    <button onclick="update(data2, 'lightgreen')">Variable 2</button>
    <button onclick="update(data3, 'lightsalmon')">Variable 3</button>

    <script>
        // create 2 data_set
        const data1 = [
            { group: "A", value: 5 },
            { group: "B", value: 20 },
            { group: "C", value: 9 }
        ];
        const data2 = [
            { group: "A", value: 10 },
            { group: "B", value: 2 },
            { group: "C", value: 22 }
        ];
        const data3 = [
            { group: "A", value: 6 },
            { group: "B", value: 12 },
            { group: "C", value: 18 }
        ];

        // set the dimensions and margins of the graph
        const margin = { top: 30, right: 30, bottom: 70, left: 60 };
        const width = 460 - margin.left - margin.right;
        const height = 400 - margin.top - margin.bottom;

        // append the svg object to the body of the page
        var svg = d3.select('body')
            .append('div')
            .append("svg")
            .attr("width", width + margin.left + margin.right)
            .attr("height", height + margin.top + margin.bottom)
            .append("g")
            .attr("transform",
                "translate(" + margin.left + "," + margin.top + ")");

        // X axis
        var x = d3.scaleBand()
            .range([0, width])
            .domain(data1.map(function (d) { return d.group; }))
            .padding(0.2);
        svg.append("g")
            .attr("transform", "translate(0," + height + ")")
            .call(d3.axisBottom(x))

```

```
        // Add Y axis
        var y = d3.scaleLinear()
            .domain([0, 20])
            .range([height, 0]);
        svg.append("g")
            .attr("class", "myYaxis")
            .call(d3.axisLeft(y));

        // A function that create / update the plot for a given variable:
        function update(data, colour) {
            var u = svg.selectAll("rect")
                .data(data)
            u.enter()
                .append("rect")
                .merge(u)
                .transition()
                .duration(1000)
                .attr("x", function (d) { return x(d.group); })
                .attr("y", function (d) { return y(d.value); })
                .attr("width", x.bandwidth())
                .attr("height", function (d) { return height - y(d.value); })
                .attr("fill", colour)
        }
        // Initialize the plot with the first dataset
        update(data1, 'goldenrod')
    </script>
</body>
```

EX20

```html
<!DOCTYPE html>
<html lang="en">

<body>

    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>

    <!-- Add buttons -->
    <button onclick="update(data1, 'goldenrod' )">Variable 1</button>
    <button onclick="update(data2, 'lightgreen')">Variable 2</button>
    <button onclick="update(data3, 'lightsalmon')">Variable 3</button>

    <script>
        // create 2 data_set
        const data1 = [
            { group: "A", value: 5 },
            { group: "B", value: 20 },
            { group: "C", value: 9 }
        ];
        const data2 = [
            { group: "A", value: 10 },
            { group: "B", value: 2 },
            { group: "C", value: 22 }
        ];
        const data3 = [
            { group: "A", value: 6 },
            { group: "B", value: 12 },
            { group: "C", value: 18 }
        ];

        // set the dimensions and margins of the graph
        const margin = { top: 30, right: 30, bottom: 70, left: 60 };
        const width = 460 - margin.left - margin.right;
        const height = 400 - margin.top - margin.bottom;

        // append the svg object to the body of the page
        var svg = d3.select('body')
            .append('div')
            .append("svg")
            .attr("width", width + margin.left + margin.right)
            .attr("height", height + margin.top + margin.bottom)
            .append("g")
            .attr("transform",
                "translate(" + margin.left + "," + margin.top + ")");

        // X axis
        var x = d3.scaleBand()
            .range([0, width])
            .domain(data1.map(function (d) { return d.group; }))
            .padding(0.2);
        svg.append("g")
            .attr("transform", "translate(0," + height + ")")
            .call(d3.axisBottom(x))

        // Add Y axis
        var y = d3.scaleLinear()
```

```
        var y = d3.scaleLinear()
            .domain([0, 20])
            .range([height, 0]);
        svg.append("g")
            .attr("class", "myYaxis")
            .call(d3.axisLeft(y));

        // A function that create / update the plot for a given variable:
        function update(data, colour) {
            var u = svg.selectAll("rect")
                .data(data)

            u.enter()
                .append("rect")
                .merge(u)
                .attr("x", function (d) {
                    console.log(d.value);
                    return 1;
                })
                .attr("y", "1")
                .on("mouseover", function onMouseOver(d, i) {
                    console.log("jhg");

                    svg.append("text")
                        .attr('class', 'val')
                        .attr('x', function (d) { return x(i.group); })
                        .attr('y', function (d) { console.log(y(i.value)); return y(i.value); })
                        .text(function (d) {
                            console.log(i.value);
                            return i.value;
                        });
                })
                .on("mouseout", function onMouseOut(d, i) {
                    console.log("hfgcvjbjbchjk");
                    svg.selectAll('.val')
                        .remove()
                })
                .transition()
                .duration(1000)
                .attr("x", function (d) { return x(d.group); })
                .attr("y", function (d) { return y(d.value); })
                .attr("width", x.bandwidth())
                .attr("height", function (d) { return height - y(d.value); })
                .attr("fill", colour)
                .attr("class", "bar")
        }

        // Initialize the plot with the first dataset
        update(data1, 'goldenrod')
    </script>
</body>
```

EX21

```html
<!DOCTYPE html>
<html lang="en">

<body>

    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>

    <!-- Add buttons -->
    <button onclick="update(data1, 'goldenrod' )">Variable 1</button>
    <button onclick="update(data2, 'lightgreen')">Variable 2</button>
    <button onclick="update(data3, 'lightsalmon')">Variable 3</button>

    <script>
        // create 2 data_set
        const data1 = [
            { group: "A", value: 5 },
            { group: "B", value: 20 },
            { group: "C", value: 9 }
        ];
        const data2 = [
            { group: "A", value: 10 },
            { group: "B", value: 2 },
            { group: "C", value: 22 }
        ];
        const data3 = [
            { group: "A", value: 6 },
            { group: "B", value: 12 },
            { group: "C", value: 18 }
        ];

        // set the dimensions and margins of the graph
        const margin = { top: 30, right: 30, bottom: 70, left: 60 };
        const width = 460 - margin.left - margin.right;
        const height = 400 - margin.top - margin.bottom;

        // append the svg object to the body of the page
        var svg = d3.select('body')
            .append('div')
            .append("svg")
            .attr("width", width + margin.left + margin.right)
            .attr("height", height + margin.top + margin.bottom)
            .append("g")
            .attr("transform",
                "translate(" + margin.left + "," + margin.top + ")");

        // X axis
        var x = d3.scaleBand()
            .range([0, width])
            .domain(data1.map(function (d) { return d.group; }))
            .padding(0.2);
        svg.append("g")
            .attr("transform", "translate(0," + height + ")")
            .call(d3.axisBottom(x))
        svg.append("g")
            .attr("transform", "translate(0," + 0 + ")")
            .call(d3.axisTop(x))
```

```
57
58          // Add Y axis
59          var y = d3.scaleLinear()
60              .domain([0, 20])
61              .range([height, 0]);
62          svg.append("g")
63              .attr("class", "myYaxis")
64              .call(d3.axisLeft(y));
65          svg.append("g")
66              .attr("class", "myYaxis")
67              .attr("transform", "translate(" + width + ", 0)")
68              .call(d3.axisRight(y));
69
70          // A function that create / update the plot for a given variable:
71          function update(data, colour) {
72              var u = svg.selectAll("rect")
73                  .data(data)
74              u.enter()
75                  .append("rect")
76                  .merge(u)
77                  .transition()
78                  .duration(1000)
79                  .attr("x", function (d) { return x(d.group); })
80                  .attr("y", function (d) { return y(d.value); })
81                  .attr("width", x.bandwidth())
82                  .attr("height", function (d) { return height - y(d.value); })
83                  .attr("fill", colour)
84          }
85          // Initialize the plot with the first dataset
86          update(data1, 'goldenrod')
87      </script>
88  </body>
```

EX22

```html
<!DOCTYPE html>
<html lang="en">

<body>

    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>

    <!-- Add buttons -->
    <button onclick="update(data1, 'goldenrod' )">Variable 1</button>
    <button onclick="update(data2, 'lightgreen')">Variable 2</button>
    <button onclick="update(data3, 'lightsalmon')">Variable 3</button>

    <script>
        // create 2 data_set
        const data1 = [
            { group: "A", value: 5 },
            { group: "B", value: 20 },
            { group: "C", value: 9 }
        ];
        const data2 = [
            { group: "A", value: 10 },
            { group: "B", value: 2 },
            { group: "C", value: 22 },
            { group: "D", value: 6 }
        ];
        const data3 = [
            { group: "A", value: 6 },
            { group: "B", value: 12 },
            { group: "C", value: 18 }
        ];

        // set the dimensions and margins of the graph
        const margin = { top: 30, right: 30, bottom: 70, left: 60 };
        const width = 460 - margin.left - margin.right;
        const height = 400 - margin.top - margin.bottom;

        // append the svg object to the body of the page
        var svg = d3.select('body')
            .append('div')
            .append("svg")
            .attr("width", width + margin.left + margin.right)
            .attr("height", height + margin.top + margin.bottom)
            .append("g")
            .attr("transform",
                "translate(" + margin.left + "," + margin.top + ")");



        // A function that create / update the plot for a given variable:
        function update(data, colour) {
            var u = svg.selectAll("rect")
                .data(data)

                svg.selectAll("g").remove();
```

```
        // X axis
        var x = d3.scaleBand()
            .range([0, width])
            .domain(data.map(function (d) { return d.group; }))
            .padding(0.2);
        svg.append("g")
            .attr("transform", "translate(0," + height + ")")
            .call(d3.axisBottom(x))
        svg.append("g")
            .attr("transform", "translate(0," + 0 + ")")
            .call(d3.axisTop(x))

        // Add Y axis
        var y = d3.scaleLinear()
            .domain([0, 20])
            .range([height, 0]);
        svg.append("g")
            .attr("class", "myYaxis")
            .call(d3.axisLeft(y));
        svg.append("g")
            .attr("class", "myYaxis")
            .attr("transform", "translate(" + width + ", 0)")
            .call(d3.axisRight(y));

            u.exit().remove();

            u.enter()
                .append("rect")
                .merge(u)
                .transition()
                .duration(1000)
                .attr("x", function (d) { return x(d.group); })
                .attr("y", function (d) { return y(d.value); })
                .attr("width", x.bandwidth())
                .attr("height", function (d) { return height - y(d.value); })
                .attr("fill", colour)


        }
        // Initialize the plot with the first dataset
        update(data1, 'goldenrod')
    </script>
</body>
```

EX23

```html
<!DOCTYPE html>
<html lang="en">

<body>

    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>

    <!-- Add buttons -->
    <button onclick="update(data1, 'goldenrod' )">Variable 1</button>
    <button onclick="update(data2, 'lightgreen')">Variable 2</button>


    <script>
        // create 2 data_set
  // Set Dimensions
  const xSize = 600; const ySize = 600;
        const margin = 40;
        const xMax = xSize - margin * 2;
        const yMax = ySize - margin * 2;

        // Create Random Points
        const numPoints = 100;
        const data1 = [];
        for (let i = 0; i < numPoints; i++) { data1.push({ x: i / 100, y: Math.sin(6.2 * i / 100) }); }

        const data2 = [];
        for (let i = 0; i < numPoints; i++) { data2.push({ x: i / 100, y: Math.cos(6.2 * i / 100) }); }


        // Get the 'limits' of the data - the full extent (mins and max)
        // so the plotted data fits perfectly
        const xExtent = d3.extent(data1, d => { return d.x });
        const yExtent = d3.extent(data1, d => { return d.y });

        // Append SVG Object to the Page
        const svg = d3.select("body")
            .append("svg")
            .attr('width', xSize)
            .attr('height', ySize)
            .append("g")
            .attr("transform", "translate(" + margin + "," + margin + ")");

        // X Axis
        const x = d3.scaleLinear()
            .domain([xExtent[0], xExtent[1]])
            .range([0, xMax]);

        // bottom
        svg.append("g")
            .attr("transform", "translate(0," + yMax + ")")
            .call(d3.axisBottom(x))
            .attr('color', 'green'); // make bottom axis green

        // top
        svg.append("g")
            .call(d3.axisTop(x));
```

```
56              .call(d3.axisTop(x));
57
58          // Y Axis
59          const y = d3.scaleLinear()
60              .domain([yExtent[0], yExtent[1]])
61              .range([yMax, 0]);
62
63          // left y axis
64          svg.append("g")
65              .call(d3.axisLeft(y));
66
67          // right y axis
68          svg.append("g")
69              .attr("transform", `translate(${yMax},0)`)
70              .call(d3.axisRight(y));
71
72
73
74          // A function that create / update the plot for a given variable:
75          function update(data, colour) {
76              var u = svg.selectAll("rect")
77                  .data(data)
78
79
80                  svg.selectAll(".line").remove();
81
82
83          // Add the line
84          svg.append("path")
85              .datum(data)
86              .attr("fill", "none")
87              .attr("class", "line")
88              .attr("stroke", "steelblue")
89              .attr("stroke-width", 1.5)
90              .attr("d", d3.line()
91                  .x(function (d) { return x(d.x) })
92                  .y(function (d) { return y(d.y) })
93              );
94
95
96
97          }
98          // Initialize the plot with the first dataset
99          update(data1, 'goldenrod')
100     </script>
101 </body>
```

EX24

```html
<!DOCTYPE html>
<html lang="en">

<body>
    <p>1) Script tag is missing "https:", resulting in error: "lab2-24.html:8 Uncaught ReferenceError: d3 is not defined"</p>
    <p>2) Afterwards the output is a list containing the values 20% between the values in the first and second lists of funtion "intr"</p>
    <script type='text/javascript' src="https://d3js.org/d3.v7.min.js"></script>
    <script>
    let intr = d3.interpolate( [20, 40, 4], [1, 12, 10])
    console.log("Type of returned function is: ", typeof (intr) );
    console.log( intr(0.2) )
    </script>
</body>
```

EX25

```html
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>


    <p>The returned value is 'rgb(128, 64, 0)', 0.5 the way between green [rgb(0, 128, 0)], and red [rgb(256, 0, 0)] </p>


    <script>
    let cc = d3.interpolate('red', 'green')
    console.log( cc(0.5) );

    d3.select("p").style("color",function(){return cc(0.5)})

    </script>
</body>
```

EX26

```html
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>


    <p></p>


    <script>
    let intrDate = d3.interpolateDate(new Date("01/01/2000"), new Date("02/22/2022"))
    console.log( intrDate(0.5) );

    d3.select("p").style("color",function(){return intrDate(0.5)})
    .text("The day between 01/01/2000 and 02/22/2022 is " + intrDate(0.5))

    </script>
</body>
```

EX27

```
<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>

    <button onclick="update(pie(dataset3.apples))">Variable 1</button>
    <button onclick="update(pie(dataset2.apples))">Variable 2</button>

    <script>
        var dataset = {
            apples: [5345, 2879, 1997, 2437, 4045],
        };
        var dataset2 = {
            apples: [23, 56, 2, 67, 30],
        };
        var dataset3 = {
            apples: [1, 2, 3, 4, 5],
        };

        var width = 460,
            height = 300,
            radius = Math.min(width, height) / 2;

        var color = d3.scaleOrdinal().range(d3.schemeSet3);

        var pie = d3.pie()
            .sort(null);

        var arc = d3.arc()
            .innerRadius(radius - 100)
            .outerRadius(radius - 50);

        var svg = d3.select("body").append("svg")
            .attr("width", width)
            .attr("height", height)
            .append("g")
            .attr("transform", "translate(" + width / 2 + "," + height / 2 + ")");

        function update(data) {
            svg.selectAll("path").remove();

            var path = svg.selectAll("path")
                .data(data)
                .enter()
                .append("path")
                .attr("fill", function (d, i) { return color(i); })
                .attr("d", arc)
                .transition()
                .duration(1000)
                .attrTween("d", function (d) {
                    var i = d3.interpolate(d.endAngle, d.startAngle);
                    return function (t) {
                        d.startAngle = i(t);
                        return arc(d);
                    }
                });
        }
        update(pie(dataset2.apples));
    </script>
</body>
```

EX28

```html
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
    <script>
        var width = 400, height = 400;

        var myColor = d3.scaleSequential().domain([0, 25]).interpolator(d3.interpolateViridis);

        // setup svg
        d3.select('body').append('svg').attr('width', width).attr('height', height);

        // generate some random data
        var numNodes = 100;
        var nodes = d3.range(numNodes).map(function (d) {
            return { radius: Math.random() * 25 }
        })

        console.log(nodes[0]);


        var scale = d3.scaleLinear()
                .domain([d3.min(nodes), d3.max(nodes)])
                .range([0, 10]);

        var simulation = d3.forceSimulation(nodes)
            .force('charge', d3.forceManyBody().strength(5))
            .force('center', d3.forceCenter(width / 2, height / 2))
            .force('collision', d3.forceCollide().radius(function (d) {
                return d.radius
            }))
            .on('tick', ticked);

        function ticked() {
            var u = d3.select('svg')
                .selectAll('circle')
                .data(nodes)
                .join('circle')
                .attr('fill', function (d) {
                    return myColor(d.radius)
                })
                .attr('r', function (d) {
                    return d.radius
                })
                .attr('cx', function (d) {
                    return d.x
                })
                .attr('cy', function (d) {
                    return d.y
                })
        }
        console.log('ready..');


    </script>
</body>
```

EX29

```html
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
    <script>
        var width = 400, height = 400;

        var myColor = d3.scaleSequential().domain([0, 25]).interpolator(d3.interpolateViridis);

        // setup svg
        d3.select('body').append('svg').attr('width', width).attr('height', height);

        // generate some random data
        var numNodes = 100;
        // var nodes = d3.range(numNodes).map(function (d) {
        //     return { radius: Math.random() * 25 }
        // })

        const nodes = [
            { radius: 5 },
            { radius: 25 },
            { radius: 9 },
            { radius: 1 },
            { radius: 2 },
            { radius: 3 },
            { radius: 4 },
            { radius: 5 },
            { radius: 6 },
            { radius: 7 },
            { radius: 8 },
            { radius: 9 },
            { radius: 10 },
            { radius: 11 },
            { radius: 12 },
            { radius: 13 },
            { radius: 14 },
            { radius: 15 },
            { radius: 16 },
            { radius: 17 },
            { radius: 18 },
            { radius: 19 },
            { radius: 20 },
            { radius: 21 },
            { radius: 22 },
            { radius: 23 },
            { radius: 24 },
            { radius: 25 }
        ];

        console.log(nodes)


        var scale = d3.scaleLinear()
                .domain([d3.min(nodes), d3.max(nodes)])
                .range([0, 10]);

        var simulation = d3.forceSimulation(nodes)
            .force('charge', d3.forceManyBody().strength(5))
```

```
            .force('center', d3.forceCenter(width / 2, height / 2))
            .force('collision', d3.forceCollide().radius(function (d) {
                return d.radius
            }))
            .on('tick', ticked);

        function ticked() {
            var u = d3.select('svg')
                .selectAll('circle')
                .data(nodes)
                .join('circle')
                .attr('fill', function (d) {
                    return myColor(d.radius)
                })
                .attr('r', function (d) {
                    return d.radius
                })
                .attr('cx', function (d) {
                    return d.x
                })
                .attr('cy', function (d) {
                    return d.y
                })
        }
        console.log('ready..');


    </script>
</body>
```

EX30

```html
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
    <script>

        var div = d3.select("body").append("div")
            .attr("class", "tooltip")
            .style("opacity", 0);

        var width = 400, height = 400;

        var myColor = d3.scaleSequential().domain([0, 25]).interpolator(d3.interpolateViridis);

        // setup svg
        d3.select('body').append('svg').attr('width', width).attr('height', height);


        // generate some random data
        var numNodes = 100;
        // var nodes = d3.range(numNodes).map(function (d) {
        //     return { radius: Math.random() * 25 }
        // })

        const nodes = [
            { radius: 5 },
            { radius: 25 },
            { radius: 9 },
            { radius: 1 },
            { radius: 2 },
            { radius: 3 },
            { radius: 4 },
            { radius: 5 },
            { radius: 6 },
            { radius: 7 },
            { radius: 8 },
            { radius: 9 },
            { radius: 10 },
            { radius: 11 },
            { radius: 12 },
            { radius: 13 },
            { radius: 14 },
            { radius: 15 },
            { radius: 16 },
            { radius: 17 },
            { radius: 18 },
            { radius: 19 },
            { radius: 20 },
            { radius: 21 },
            { radius: 22 },
            { radius: 23 },
            { radius: 24 },
            { radius: 25 }
        ];

        console.log(nodes)


```

```
        var scale = d3.scaleLinear()
            .domain([d3.min(nodes), d3.max(nodes)])
            .range([0, 10]);

        var simulation = d3.forceSimulation(nodes)
            .force('charge', d3.forceManyBody().strength(5))
            .force('center', d3.forceCenter(width / 2, height / 2))
            .force('collision', d3.forceCollide().radius(function (d) {
                return d.radius
            }))
            .on('tick', ticked);

        function ticked() {
            var u = d3.select('svg')
                .selectAll('circle')
                .data(nodes)
                .join('circle')
                .attr('fill', function (d) {
                    return myColor(d.radius)
                })
                .attr('r', function (d) {
                    return d.radius
                })
                .attr('cx', function (d) {
                    return d.x
                })
                .attr('cy', function (d) {
                    return d.y
                })
                .on('mousemove', (event, d) => {
                    console.log(d)
                    var coords = d3.pointer(event);
                    console.log(coords[0], coords[1]) // log the mouse x,y position
                    div.transition()
                        .duration(200)
                        .style("opacity", .9);
                    div.html("Radius: " + d.radius)
                        .style("left", (coords[0]) + "px")
                        .style("top", (coords[1]) + "px");
                });
        }
        console.log('ready..');


    </script>
    <style>
        div.tooltip {
            position: absolute;
            text-align: center;
            width: 60px;
            height: 28px;
            padding: 2px;
            font: 12px sans-serif;
            background: lightsteelblue;
            border: 0px;
            border-radius: 8px;
            pointer-events: none;
        }
    </style>
</body>
```

EX31

```html
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
    <script>

        var div = d3.select("body").append("div")
            .attr("class", "tooltip")
            .style("opacity", 0);

        var width = 400, height = 400;

        var myColor = d3.scaleSequential().domain([0, 25]).interpolator(d3.interpolateViridis);

        var myColor2 = d3.scaleLinear().domain([1, 25])
            .range(["pink", "goldenrod"]);

        // setup svg
        d3.select('body').append('svg').attr('width', width).attr('height', height);


        // generate some random data
        var numNodes = 100;
        // var nodes = d3.range(numNodes).map(function (d) {
        //     return { radius: Math.random() * 25 }
        // })

        const nodes = [
            { radius: 5 },
            { radius: 25 },
            { radius: 9 },
            { radius: 1 },
            { radius: 2 },
            { radius: 3 },
            { radius: 4 },
            { radius: 5 },
            { radius: 6 },
            { radius: 7 },
            { radius: 8 },
            { radius: 9 },
            { radius: 10 },
            { radius: 11 },
            { radius: 12 },
            { radius: 13 },
            { radius: 14 },
            { radius: 15 },
            { radius: 16 },
            { radius: 17 },
            { radius: 18 },
            { radius: 19 },
            { radius: 20 },
            { radius: 21 },
            { radius: 22 },
            { radius: 23 },
            { radius: 24 },
            { radius: 25 }
        ];

```

```
63          var scale = d3.scaleLinear()
64              .domain([d3.min(nodes), d3.max(nodes)])
65              .range([0, 10]);
66
67          var simulation = d3.forceSimulation(nodes)
68              .force('charge', d3.forceManyBody().strength(5))
69              .force('center', d3.forceCenter(width / 2, height / 2))
70              .force('collision', d3.forceCollide().radius(function (d) {
71                  return d.radius
72              }))
73              .on('tick', ticked);
74
75          function ticked() {
76              var u = d3.select('svg')
77                  .selectAll('circle')
78                  .data(nodes)
79                  .join('circle')
80                  .attr('fill', function (d) {
81                      return myColor(d.radius)
82                  })
83                  .attr('r', function (d) {
84                      return d.radius
85                  })
86                  .attr('cx', function (d) {
87                      return d.x
88                  })
89                  .attr('cy', function (d) {
90                      return d.y
91                  })
92                  .on('mousemove', (event, d) => {
93                      console.log(d)
94                      var coords = d3.pointer(event);
95                      console.log(coords[0], coords[1]) // log the mouse x,y position
96                      div.transition()
97                          .duration(200)
98                          .style("opacity", .9);
99                      div.html("Radius: " + d.radius)
100                         .style("left", (coords[0]) + "px")
101                         .style("top", (coords[1]) + "px");
102                 })
103                 .on("mouseover", function onMouseOver(d, i) {
104                     d3.select(this).attr('class', 'highlight');
105                     d3.select(this)
106                         .transition() // adds animation
107                         .duration(400)
108                         .style("fill", myColor2(i.radius));
109                 })
110                 .on("mouseout", function onMouseOut(d, i) {
111                     d3.select(this).attr('class', 'highlight');
112                     d3.select(this)
113                         .transition() // adds animation
114                         .duration(400)
115                         .style("fill", myColor(i.radius));
116                 });
117         }
118         console.log('ready..');
119
120
121     </script>
```

```
122     <style>
123         div.tooltip {
124             position: absolute;
125             text-align: center;
126             width: 60px;
127             height: 28px;
128             padding: 2px;
129             font: 12px sans-serif;
130             background: lightsteelblue;
131             border: 0px;
132             border-radius: 8px;
133             pointer-events: none;
134         }
135     </style>
136 </body>
```

EX32

```
<!DOCTYPE html>
<html lang="en">

<body>
    <script type='text/javascript' src='https://d3js.org/d3.v7.min.js'></script>
    <script>

        var div = d3.select("body").append("div")
            .attr("class", "tooltip")
            .style("opacity", 0);

        var width = 400, height = 400;

        var myColor = d3.scaleSequential().domain([0, 25]).interpolator(d3.interpolateViridis);

        var myColor2 = d3.scaleLinear().domain([1, 25])
            .range(["pink", "goldenrod"]);

        // setup svg
        d3.select('body').append('svg').attr('width', width).attr('height', height);


        // generate some random data
        var numNodes = 100;
        // var nodes = d3.range(numNodes).map(function (d) {
        //     return { radius: Math.random() * 25 }
        // })

        const nodes = [
            { radius: 5 },
            { radius: 25 },
            { radius: 9 },
            { radius: 1 },
            { radius: 2 },
            { radius: 3 },
            { radius: 4 },
            { radius: 5 },
            { radius: 6 },
            { radius: 7 },
            { radius: 8 },
            { radius: 9 },
            { radius: 10 },
            { radius: 11 },
            { radius: 12 },
            { radius: 13 },
            { radius: 14 },
            { radius: 15 },
            { radius: 16 },
            { radius: 17 },
            { radius: 18 },
            { radius: 19 },
            { radius: 20 },
            { radius: 21 },
            { radius: 22 },
            { radius: 23 },
            { radius: 24 },
            { radius: 25 }
        ];
```

```
        var links = [
            { source: nodes[1], target: nodes[26]},
            { source: nodes[27], target: nodes[26]},
            { source: nodes[1], target: nodes[27]},

            { source: nodes[0], target: nodes[2]},
            { source: nodes[2], target: nodes[3]},
            { source: nodes[3], target: nodes[4]},
            { source: nodes[4], target: nodes[5]},
            { source: nodes[5], target: nodes[6]},
            { source: nodes[6], target: nodes[7]},
            { source: nodes[7], target: nodes[0]},
            { source: nodes[0], target: nodes[8]},
            { source: nodes[8], target: nodes[9]},
            { source: nodes[10], target: nodes[11]},
            { source: nodes[11], target: nodes[12]},

            { source: nodes[2], target: nodes[6]},
            { source: nodes[4], target: nodes[0]}
        ]

        console.log(nodes)


        var scale = d3.scaleLinear()
            .domain([d3.min(nodes), d3.max(nodes)])
            .range([0, 10]);

        var simulation = d3.forceSimulation(nodes)
            .force('charge', d3.forceManyBody().strength(5))
            .force('center', d3.forceCenter(width / 2, height / 2))
            .force('collision', d3.forceCollide().radius(function (d) {
                return d.radius
            }))
            .force('link', d3.forceLink().links(links).distance(10))
            .on('tick', ticked);

        function ticked() {
            var u = d3.select('svg')
                .selectAll('circle')
                .data(nodes)
                .join('circle')
                .attr('fill', function (d) {
                    return myColor(d.radius)
                })
                .attr('r', function (d) {
                    return d.radius
                })
                .attr('cx', function (d) {
                    return d.x
                })
                .attr('cy', function (d) {
                    return d.y
                })
                .on('mousemove', (event, d) => {
                    console.log(d)
                    var coords = d3.pointer(event);
                    console.log(coords[0], coords[1]) // log the mouse x,y position
                    div.transition()
```

```
                    .duration(200)
                    .style("opacity", .9);
                div.html("Radius: " + d.radius)
                    .style("left", (coords[0]) + "px")
                    .style("top", (coords[1]) + "px");
            })
            .on("mouseover", function onMouseOver(d, i) {
                d3.select(this).attr('class', 'highlight');
                d3.select(this)
                    .transition() // adds animation
                    .duration(400)
                    .style("fill", myColor2(i.radius));
            })
            .on("mouseout", function onMouseOut(d, i) {
                d3.select(this).attr('class', 'highlight');
                d3.select(this)
                    .transition() // adds animation
                    .duration(400)
                    .style("fill", myColor(i.radius));
            });
        }
        console.log('ready..');


    </script>
    <style>
        div.tooltip {
            position: absolute;
            text-align: center;
            width: 60px;
            height: 28px;
            padding: 2px;
            font: 12px sans-serif;
            background: lightsteelblue;
            border: 0px;
            border-radius: 8px;
            pointer-events: none;
        }
    </style>
</body>
```