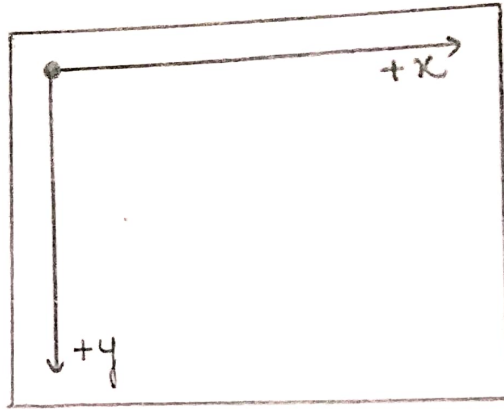


-: POSITIONING AND STYLING OF ELEMENTS :-

• Positioning Elements:

- SVG is positioned by a grid system, similar to the cartesian co-ordinate system.
- However, in SVG $(0,0)$ is the top-left corner.
- The x -axis proceeds horizontally from left to right starting at 0. The y -axis also starts from 0 and extends downwards.

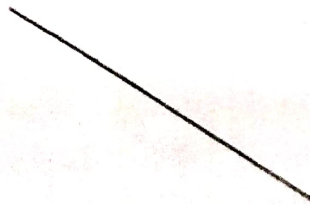


line:-

The SVG line is one of the simplest in the library. It draws a straight line from one point to another.

```
<line x1="10" y1="10" x2="100" y2="100" stroke-width="1" stroke="red"/>
```

Output:



Transform:-

SVG introduces a new attribute, transform to support transformation. The possible values are one or more of the following.

- Translate - It takes two options, tx refers translation along x-axis and ty refers translation along y-axis.
For ex: translate(30,30)

```
<rect x="20"
      y="20"
      width="60"
      height="60"
      fill="green"
      transform="translate(30 30)">
```

- Rotate - It takes 3 options, angle refers rotation angle, cx and cy refers to the center of the rotation in the x and y-axis.

It cx and cy are not specified, then it defaults to the current origin of the co-ordinate system. For ex - rotate(60).

```
<rect x="20"
      y="20"
      width="60"
      height="60"
      fill="blue"
      transform="translate(60 60) rotate(45)">
```

Note: Transformation can be applied on group as well.

```
<svg width="300" height="300">
  <g transform="translate(60,60) rotate(30)">
    <rect x="20"
          y="20"
          width="60"
          height="30">
  </rect>
```



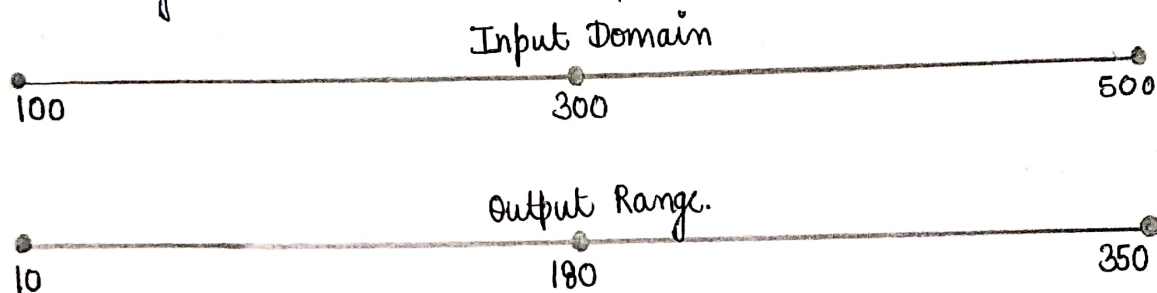
```

<circle cx = "0"
      cy = "0"
      r = "30"
      fill = "red" />
</g>

```

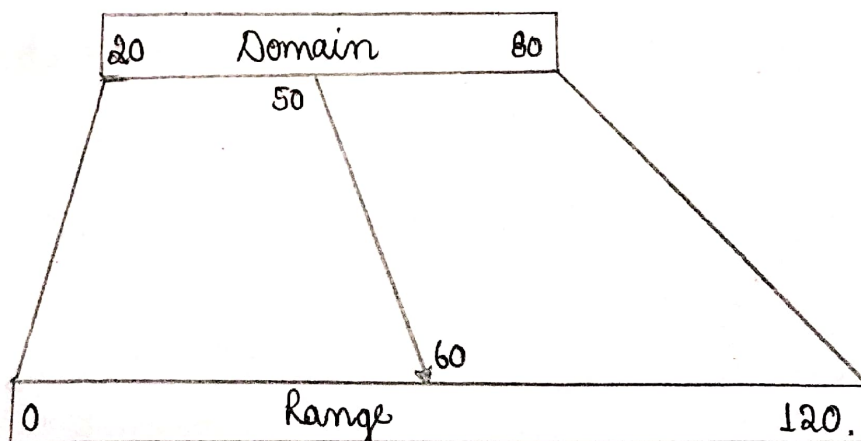
• Scale:

- Scales are functions that map from an input domain to an output range.
- Input is data-driven.
- Output range controls visual properties.



Scale types:

Ordinal scale
 Linear scale
 Log scale
 Power, scale
 Time range



Linear Scale :

```

var data = [31, 22, 50, 36, 80, 42];
var x = d3.scale.linear()
  .domain([20, d3.max(data)])
  .range([0, 120]);

```

• Transition of Elements :-

- Transition is the process of changing from one state to another of an item. D3.js provides a transition() method to perform transition in the HTML page.
- The transition() method is available for all selectors and it starts the transition process. This method supports most of the selection methods such as - attr(), style() etc. But it does not support the append() and data() methods.
- It provides methods specific to transition like duration(), delay() etc. A simple transition can be defined as follows:

Eg:

```
d3.select("body")  
  .transition()  
  .style("background-color", "lightblue");
```

• Styling the Elements :

Creating elements and Setting Attributes

- attr() - Sets an attribute on the selected element.

Two big things that attributes are for are:

- Position your element
- Set your element's size.

```
var svg = d3.create("svg");  
var circle = svg.append("circle")  
  .attr("cx", 85)  
  .attr("cy", 75)  
  .attr("r", 50);
```