06. D3JS Animations.

D3.js supports animation through transition. We can do animation with proper use of transition. Transitions are a limited form of Key Frame Animation with only two key frames – start and end. The starting key frame is typically the current state of the DOM, and the ending key frame is a set of attributes, styles and other properties you specify. Transitions are well suited for transitioning to a new view without a complicated code that depends on the starting view.

Example 1: Let us consider the following code in "transition_color.html" page.

```
<!DOCTYPE html>
<html>
   <head>
      <script src = "https://d3js.org/d3.v4.min.js"></script>
   </head>

   <body>
      <h3>Simple transitions</h3>
      <script>
         d3.select("body").style("background-color", "lightblue")
         // make the background-color lightblue.transition()
         .style("background-color", "gray");
         // make the background-color gray
      </script>
   </body>
</html>
```

Here, the Background color of the document changed from white to light gray and then to gray.

6.1. The duration() Method

The duration() method allows property changes to occur smoothly over a specified duration rather than instantaneously. Let us make the transition which takes 5 seconds using the following code.

Example:

```
<!DOCTYPE html>
<html>
   <head>
      <script src = "https://d3js.org/d3.v4.min.js"></script>
   </head>

   <body>
      <h3>Simple transitions</h3>
      <script>
         d3.selectAll("h3").transition()
           .style("color","green").duration(5000);
      </script>
   </body>
</html>
```

Here, the transitions occurred smoothly and evenly. We can also assign RGB color code value directly using the following method.

Example:

```
d3.selectAll("h3").transition()
  .style("color","rgb(0,150,120)").duration(5000);
```

Now, each color number slowly, smoothly and evenly goes from 0 to 150. To get the accurate blending of in-between frames from the start frame value to the end frame value, D3.js uses an internal interpolate method. The syntax is given below:

Example:

```
d3.interpolate(a, b)
```

D3 also supports the following interpolation types :

- **interpolateNumber** − support numerical values.
- **interpolateRgb** − support colors.
- **interpolateString** − support string.

D3.js takes care of using the proper interpolate method and in advanced cases, we can use the interpolate methods directly to get our desired result. We can even create a new interpolate method, if needed.

6.2. The delay() Method

The delay() method allows a transition to take place after a certain period of time.

Example: Consider the following code in "transition_delay.html".

```
<!DOCTYPE html>
<html>
   <head>
      <script type = "text/javascript" src =
"https://d3js.org/d3.v4.min.js"></script>
   </head>

   <body>
      <h3> Simple transitions </h3>
      <script>
         d3.selectAll("h3").transition()
            .style("font-size","28px").delay(2000).duration(2000);
      </script>
   </body>
</html>
```

6.3. Lifecycle of Transition

Transition has a four-phased lifecycle:

1. **The transition is scheduled.**
2. **The transition starts.**
3. **The transition runs.**
4. **The transition ends.**

Let us go through each of these one by one in detail.

The Transition is scheduled

A transition is scheduled when it is created. When we call ***selection.transition***, we are scheduling a transition. This is also when we call attr(), style() and other transition methods to define the ending key frame.

The Transition Starts

A transition starts based on its delay, which was specified when the transition was scheduled. If no delay was specified, then the transition starts as soon as possible, which is typically after a few milliseconds.

If the transition has a delay, then the starting value should be set only when the transition starts. We can do this by listening to the start event:

Example:

```
<!DOCTYPE html>
d3.select("body")
   .transition()
   .delay(200)
   .each("start", function() {
           d3.select(this).style("color", "green");
         })
   .style("color", "red");
```

The Transition Runs

When the transition runs, it repeatedly invoked with values of transition ranging from 0 to 1. In addition to delay and duration, transitions have easing to control timing. Easing distorts time, such as for slow-in and slow-out. Some easing functions may temporarily give values of t greater than 1 or less than 0.

The Transition Ends

The transition ending time is always exactly 1, so that the ending value is set exactly when the transition ends. A transition ends based on the sum of its delay and duration. When a transition ends, the end event is dispatched.