

18. D3JS Transitions API.

D3 Transitions take a selection of elements and for each element; it applies a transition to a part of the current definition of the element.

18.1. Configuring API

You can configure the API directly using the script below.

Example:

```
<script src = "https://d3js.org/d3-color.v1.min.js"></script>
<script src = "https://d3js.org/d3-dispatch.v1.min.js"></script>
<script src = "https://d3js.org/d3-ease.v1.min.js"></script>
<script src = "https://d3js.org/d3-interpolate.v1.min.js"></script>
<script src = "https://d3js.org/d3-selection.v1.min.js"></script>
<script src = "https://d3js.org/d3-timer.v1.min.js"></script>
<script src = "https://d3js.org/d3-transition.v1.min.js"></script>
<body>
  <script>
  </script>
</body>
```

18.2. Transition API Methods

Let us go through the Transition API methods in detail.

Selecting Elements

Let us discuss the various selecting elements in detail.

- **selection.transition([name])** – This method is used to return a new selection transition with the name. If a name is not specified, it returns null.

- **selection.interrupt([name])** – This method is used to interrupt the selected elements of the transition with the name and is defined below.

Example:

```
selection.interrupt().selectAll("*").interrupt();
```

- **d3.interrupt(node[, name])** – This method is used to interrupt the transition of the specified name on the specified node.
- **d3.transition([name])** – This method is used to return a new transition with the specified name.
- **transition.select(selector)** – This method is used to select the first element that matches the specified selector and returns a transition on the resulting selection, which is defined below.

Example:

```
transition  
  .selection()  
  .select(selector)  
  .transition(transition)
```

- **transition.selectAll(selector)** – This method is used to select all the elements that matches the specified selector and returns a transition on the resulting selection. It is defined below –

Example:

```
transition
  .selection()
  .selectAll(selector)
  .transition(transition)
```

- **transition.filter(filter)** – This method is used to select the elements matching the specified filter, they are defined below.

Example:

```
transition
  .selection()
  .filter(filter)
  .transition(transition)
```

- **transition.merge(other)** – This method is used to merge the transition with other transition. It is defined below.

Example:

```
transition
  .selection()
  .merge(other.selection())
  .transition(transition)
```

- **transition.transition()** – This method is used to return a new transition on the selected elements. It is scheduled to start when the transition stops. The new transition inherits this transition's name, duration and easing.

Example: Let us consider the following example.

```
d3.selectAll(".body")
    .transition()

    // fade to yellow.
    .style("fill", "yellow")
    .transition()

    // Wait for five second. Then change blue, and remove.
    .delay(5000)
    .style("fill", "blue")
    .remove();
```

Here, the body fades to yellow and starts just five seconds before the last transition.

- **d3.active(node[, name])** – This method is used to return the transition on the specified node with the name.

Timing Methods

Let us go through the transition timing API methods in detail.

- **transition.delay([value])** – This method is used to set the transition delay to the specified value. If a function is evaluated for each selected element, it is passed to the current datum 'd' and index 'i', with the context as the current DOM element. If a value is not specified, returns the current value of the delay for the first (non-null) element in the transition. It is defined below,

Example:

```
transition.delay(function(d, i) { return i * 10; });
```

- **transition.duration([value])** – This method is used to set the transition duration to the specified value. If a value is not specified, returns the current value of the duration for the first (non-null) element in the transition.
- **transition.ease([value])** – This method is used to ease the transition value for selected elements. The easing function is invoked for each frame of the animation and passed the normalized time 't' in the range [0, 1]. If a value is not specified, it returns the current easing function for the first (non-null) element in the transition.

In the next chapter, we will discuss the drag and drop concept in d3.js.