Getting started with D3.js

* D3 stands for Data-Driven Documents.
* It is open-source JavaScript library developed by Mike Bostock to create custom interactive data visualizations in the web browser using SVG, HTML and CSS
* D3.js launched on 18th Feb 2011
* Latest version is 7.3, which is launched on 7th Jan 2022

D3 Features

* **Uses Web standards** :- D3 is an extremely powerful visualization tool to create interactive data visualization. It exploits the modern web standards: SVG, HTML and CSS to create data visualizat
* **Data driven**: D3 is data driven. It can use static data or fetch it from the remote server in different formats such as Arrays, objects, CSV, JSON, XML, to create different types of charts.
* **DOM Manipulation**: D3 allows you to manipulate DOM based on your data
* **Data driven element**: It empowers your data to dynamically generate elements and apply styles to the elements be it table, a graph or any other HTML elements and/or group of elements.
* **Dynamic Properties**:- DB gives the flexibility to provide dynamic Properties to most of its functions Properties can be specified as functions of data. That means your Data can drive your style and attribute

Types of visualization :- With DB, there are no standard visualization formats. But it enables you to create anything from a HTML table to a pie chart, from graphs and bar charts to geospatial ma

* Custom Visualization: Since D3 works with we standards, it gives you complete control over visualization features
* Transitions: D3 provides the transition() function. Internally, D3 works out the logic to interpolate btw your values and find the intermittent state
* Interaction and animation: D3 provides greate support for animation with function like duration(), delay and ease(). Animation from one state to another are fast and responsive to user interaction

Advantages

* D3.js is a JS library. So it can be used with JS framework of your choice. ReactJS
* D3 focuses on data. So it is the most appropriate and specialized data visualization
* D3 is open-source.
* It works with wof standards so no need to implement ~~tech~~ any other technology or plugin other than a browser to make use of D3.
* D3 works with web standards like HTML, CSS and SVG, there no new learning or debugging tool required to work on D3.
* D3 does not provide any specific feature, so it gives you complete control over your visualization to customize it the way you want. This gives it an edge over other popular tools like Tableau & QlikView
* Since D3 is lightweight, and works directly with wof standards it is extremely fast and works well with large datasets

Including D3.js into HTML file

1) Include D3 library from your project's folder
   - Download zip file from official website
   - Extract that to desired place
   - Mention src as the file address in script.

```
<script src="../d3.min.js"></script>
```

2) Include D3 library from CDN
   - The link for CDN is provided in official ~~bote~~ site.
   - Copy that and place CDN link to head section

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

* SVG:

What is SVG?

- SVG stands for scalable vector graphics
- SVG is used to define vector-based graphics for the web.
- SVG defines the graphics in XML format.
- Every element and every attribute in SVG files can be animated.
- SVG is a World Wide Web recommendation
- SVG integrates well with other W3C standards such as the DOM and XSL.

Example for SVG:

```
<html>
  <body>
<h1> This is a svg example <h1>
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green"
      stroke-width="4" fill="yellow" />
</svg> </body> </html>
```

—— X —— X ——

- An SVG file, short for ~~scalable~~ scalable vector graphic file, is a standard graphics file type used for rendering two-dimensional images on the internet.

- Unlike other popular image file formats, the SVG format stores images as vector, which is a type of graphic made up of points, lines, curves and shapes based on mathematical formulas.

⇒ Raster (JPG/PNG) vs Vector (SVG):

- We all are familiar with file formats like JPEG & PNG, these are raster-graphics formats, which means that they store image information in grid of coloured squares, also called as bitmap.
- The squares in this bitmap combine to form a coherent image, much like pixels on a computer screen.
- Raster graphics work well for highly detailed images like photographs in which the exact color of each pixel needs to be specified.
- Raster images have a fixed resolution, so increasing their size lowers the quality of the image.
- Vector graphics formats - SVG and PDF work differently. These formats store images as a set of points and lines between points.
- Mathematical formulas dictate the placement and shape of these points and lines, and maintain their spatial relationships when the image is scaled up or down. Vector graphic file also store

color information and can even display text.

=> FEATURES OF SVG:

(1) Infinite scalability:

- It is right that SVGs can be expanded or shrunk down to any size without a loss of quality. Image size and display type don't matter with SVGs - they always look the same.
- This is important because the size of web images differ by viewer, based on browser window dimensions, device, zoom, site layout, and responsive design. Your images must appear fully-rendered to every viewer, and SVGs make this a lot easier.
- Raster images, in contrast, appear pixelated when blown-up on our screens. While there are workarounds for this problem to keep the raster formula - like using different files of increasing size for the same image - they take over and more prone to errors. Raster images were ultimately are not designed for scaling.

(2) Customization:

SVGs give designers and developers a lot of control over their appearance. Rather than modifying the files directly in text editor, you can employ one to many SVG-compatible editing

programs to change your vector shapes, colors, text and even other visual effects like color gradients and shadows.

③ Scripting Compatibility:

- The SVG file was developed by the world wide web consortium as a standardized format for web graphics, designed to work with other web conventions like HTML, CSS, Javascript and the document object model.
- SVG images can be controlled with scripts. This opens the door for a huge range of dynamic display possibilities, from animations to dynamic charts to mobile-responsive images. This is not possible with PNG & JPEG.

④ Accessibility and search engine optimization:

- SVG files are text files, and this itself offers some advantages over raster formats. First, programmers can look at the XML code and quickly understand it.
- Also, if an SVG graphic contains text, the text information is stored in the file as literal text (not as shapes). This allows SVG to be interpreted by screen readers,
- SVG files can be indexed by search engines like Google, Bing, Yahoo. If you want to place a text-heavy infographic or

other SVG display on your page, including keywords text in the image can help your page rank and improve your SEO. PNGs and JPEGs are limited to metadata and alt text in this regard.

(5) Smaller file sizes:

- SVG files tend to store images more efficiently that common raster formats as long as the image is not too detailed. SVG files contain enough information to display vectors at any scale, whereas bitmaps require larger files for scaled-up versions of images - more pixels use up more file space.
- This is good for websites because smaller files load faster on browsers, so SVGs can increase overall page performance.

* SVG DOM:

- All of the SVG DOM interfaces that correspond directly to elements in the SVG language derive from the SVGElement interface.

→ PROPERTIES:

- Also inherits properties from:
  Document And Element Event Handlers, Element, Global Event Handlers, SVG Element Instance.

① DOMStringMap: object which provides a list of key/value pairs of named data attributes which correspond to custom data attributes attached to element.

② SVGAnimatedString: that reflects the value of the class attribute given to element, or the empty string if class is not present

③ SVGElement.ownerSVGElement: An SVGElement referring to the nearest ancestor <svg> element. null if the given element is the outermost <svg> element

④ SVGElement.style: representing the declarations of element's style attribute

⑤ SVGElement.tabIndex: The position of element in the tabbing order.

② METHODS:

- This interface has no methods, but inherits methods from: DocumentAndElementEventHandlers, Element, GlobalEventHandlers, SVGElementInstance.

→ EVENTS:

- Listen to these events using addEventListener() or by assigning an event listener to the equivalent on handler property defined.

(1) Abort:

- Fired when page loading is stopped before an SVG Element has been allowed to load completely

(2) Error:

- Fired when a SVG element does not load properly or when an error occurs during script execution.

(3) Resize:

- Fired when an SVG document is being resized. Also available via the onresize property.

(4) Scroll:

- Fired when a SVG document view is being shifted along X or/and Y axis.

(5) Unload:

- Fired when the DOM implementation removes an SVG document from a window or frame

SVG Tag Attributes

- **color:**

The color attribute is used to provide a potential indirect value for fill, stroke, stop-color, flood-color and lighting-color attributes.

```
<g color = "green">
    <rect width = "50" height = "50" fill =
     "currentcolor" />
    <circle r = "25" cx = "70" cy = "70" stroke =
     "currentcolor" fill = "none" stroke-width = "5"
     />
</g>
```

- **cx:**

The cx attribute defines the x-axis coordinate of a center point.

You can use this attribute with the following SVG elements:

→ <circle>
→ <ellipse>
→ <radialGradient>

SVG Tag Attributes

- **color:**

The color attribute is used to provide a potential indirect value for fill, stroke, stop-color, flood-color and lighting-color attributes.

```
<g color = "green">
    <rect width = "50" height = "50" fill =
     "currentcolor" />
    <circle r = "25" cx = "70" cy = "70" stroke =
     "currentcolor" fill = "none" stroke-width = "5"
     />
</g>
```

- **cx:**

The cx attribute defines the x-axis coordinate of a center point.

You can use this attribute with the following SVG elements:

→ <circle>
→ <ellipse>
→ <radialGradient>

- dx:

The dx attribute indicates a shift along the x-axis on the position of an element or its content.

You can use this attribute with the following SVG elements:

→ <feDropShadow>
→ <feOffset>
→ <text>
→ <tref>
→ <tspan>

Similarly dy.

- fill:

The fill attribute has two different meanings. For shapes and text it's a presentation attribute that defines the color used to paint the element; for animation it defines the final state of the animation.

You can use this attribute with the following SVG elements:

→ <circle>
→ <ellipse>

→ <path>
→ <polygon>
→ <polyline>
→ <rect>
→ <text>
→ <textPath>
→ <tref>
→ <tspan>

- x:
  The x attribute defines an x-axis coordinate in the user coordinate system.
- y:
  The y attribute defines a y-axis coordinate in the user coordinate system.
- style:
  The style attribute allows to style an element using CSS declarations. It functions identically to the style attribute in HTML.

```
<rect width="80" height="40" x="10" y="10"
  style="fill: skyblue; stroke: cadetblue;
  stroke-width: 2;" />
```

- transform:
The transform attribute defines a list of transform definitions that are applied to an element and the element's children.

You can use this attribute with any SVG element.

- font-family:
The font-family attribute indicates which font family will be used to render the text, specified as a prioritized list of font family names and/or generic font family names.

You can use this attribute with the following SVG elements:

→ <text>
→ <textPath>
→ <tref>
→ <tspan>

Basic Elements:-

* Selection:- → It is one of the core concept in D3.js
  → It allows us to select 1 or more elements in webpage
→ It also allows us to modify, append or remove elements in a relation to the pre-defined dataset.

Select Method:-

* select():- This method selects the HTML element based on the selectors
  → Tag selector:- Tag name is used to select the specific element
    Ex:- select('p') → selects the first p in the web
  → Class Selector:- Class name is used to select the specific Element.
    Ex:- select('.mydiv')
  → ID selector:- ID name is used to select the specific element
    Ex:- select('#mydiv')

SelectAll Method:- By default select() selects only the first occurance of the kind that has to be selected, to select all the occurance we use SelectAll method:-
  Syntax:- selectAll("selector")

Ex:- selectAll('div') -> Selects all the on the webpage.

DOM Manipulation using Select:

* **Styling:** We can add styles to the element by chaining style() method to the selected element, it takes 2 parameters i.e the property & the value.

Syntax:- d3.select("selector").style("property", "value")

Ex:- d3.select("#input").style("width", "30px")

* **Attributes:** We can specify attributes to the HTML elements using attr() method,

Syntax:- d3.select("selector").attr("attribute", "value")

Ex:- d3.select("#img").attr("~~src~~ alt", "Simple Img")

* **Inner Text:** We can insert texts into HTML elements

Syntax:- .text("paragraph")

Ex:- d3.select("p").text("Hello there...!")

* **Appending:** We can append different HTML elements into other elements

Syntax:- .append("tag")

Ex:- d3.select("div").append("p")