01. D3JS Introduction.

D3 stands for **Data-Driven Documents**. D3.js is a JavaScript library for manipulating documents based on data. *D3.js* is a dynamic, interactive, online data visualizations framework used in a large number of websites. *D3.js* is written by *Mike Bostock*, created as a successor to an earlier visualization toolkit called **Protovis**. This tutorial will give you a complete knowledge on *D3.js* framework. This is an introductory tutorial, which covers the basics of *Data-Driven Documents* and explains how to deal with its various components and sub-components.

1.1. Data Visualization

Data visualization is the presentation of data in a pictorial or graphical format. The primary goal of data visualization is to communicate information clearly and efficiently via statistical graphics, plots and information graphics.

Data visualization helps us to communicate our insights quickly and effectively. Any type of data, which is represented by a visualization allows users to compare the data, generate analytic reports, understand patterns and thus helps them to take the decision. **Data visualizations can be interactive**, so that users analyze specific data in the chart. Well, Data visualizations can be developed and integrated in regular websites and even mobile applications using different JavaScript frameworks.

1.2. What is D3.js?

D3.js is a JavaScript library used to create interactive visualizations in the browser. The D3.js library allows us to manipulate elements of a webpage in the context of a data set. These elements can be **HTML**, **SVG**, or **Canvas** elements and can be introduced, removed, or edited according to the contents of the data set. It is a library for manipulating the DOM objects. D3.js can be a valuable aid in data exploration, it gives you control over your data's representation and lets you add interactivity.

1.3. Why do we need D3.js?

D3.js is one of the premier framework when compare to other libraries. This is because it works on the web and its data visualizations are par excellence. Another reason it has worked so well is owing to its

flexibility. Since it works seamlessly with the existing web technologies and can manipulate any part of the document object model, it is as flexible as the **Client Side Web Technology Stack** (***HTML, CSS, and SVG***). It has a great community support and is easier to learn.

1.4. D3js Features

D3.js is one of the best data visualization framework and it can be used to generate simple as well as complex visualizations along with user interaction and transition effects. Some of its salient features are listed below:

- Extremely flexible.
- Easy to use and fast.
- Supports large datasets.
- Declarative programming.
- Code re-usability.
- Has wide variety of curve generating functions.
- Associates data to an element or group of elements in the HTML page.

1.5. D3js Benefits

D3.js is an open source project and works without any *plugin*. It requires very less code and comes up with the following benefits:

- Great data visualization.
- It is modular. You can download a small piece of *D3.js*, which you want to use. No need to load the whole library every time.
- Easy to build a charting component.
- **DOM** manipulation.

1.6. D3js Installation

We need to include the D3.js library into your HTML webpage in order to use D3.js to create data visualization. We can do it in the following two ways:

- Include the ***D3.js library*** from your project's folder.
- Reference ***D3.js library from CDN*** (Content Delivery Network).

Download D3.js Library

D3.js is an open-source library and the source code of the library is freely available on the web at https://d3js.org/ website. Visit the D3.js website and **download the latest version of D3.js** (d3.zip). As of now, the latest version is 5.0.0.

After the download is complete, unzip the file and look for ***d3.min.js***. This is the minified version of the D3.js source code. Copy the ***d3.min.js*** file and paste it into your project's root folder or any other folder, where you want to keep all the library files. Include the d3.min.js file in your HTML page as shown below.

Example: Let us consider the following example.

```
<!DOCTYPE html>
<html lang = "en">
   <head>
      <script src = "/path/to/d3.min.js"></script>
   </head>
   <body>
      <script>
         // write your d3 code here..
      </script>
   </body>
</html>
```

D3.js is a JavaScript code, so we should write all our D3 code within "script" tag. We may need to manipulate the existing DOM elements, so it is advisable to write the D3 code just before the end of the "body" tag.

Reference D3 Library from CDN

We can use the D3.js library by linking it directly into our HTML page from the **Content Delivery Network (CDN)**. CDN is a network of servers where files are hosted and are delivered to a user based on their geographic location. If we use the CDN, we do not need to download the source code.

Include the D3.js library using the CDN URL https://d3js.org/d3.v4.min.js into our page as shown below.

Example: Let us consider the following example.

```
<!DOCTYPE html>
<html lang = "en">
   <head>
      <script src = "https://d3js.org/d3.v5.min.js"></script>
   </head>
   <body>
      <script>
         // write your d3 code here..
      </script>
   </body>
</html>
```

1.7. D3js Concepts

D3.js is an open source JavaScript library for :

- Data-driven manipulation of the **Document Object Model** (DOM).
- Working with data and shapes.

- Laying out visual elements for linear, hierarchical, network and geographic data.
- Enabling smooth transitions between user interface (UI) states.
- Enabling effective user interaction.

1.8. Web Standards

Before we can start using *D3.js* to create visualizations, we need to get familiar with web standards. The following web standards are heavily used in *D3.js*.

- **HyperText Markup Language (HTML)**
- **Document Object Model (DOM)**
- **Cascading Style Sheets (CSS)**
- **Scalable Vector Graphics (SVG)**
- **JavaScript**

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

HyperText Markup Language (HTML)

As we know, HTML is used to structure the content of the webpage. It is stored in a text file with the extension ".html".

Example: A typical bare-bones HTML example looks like this

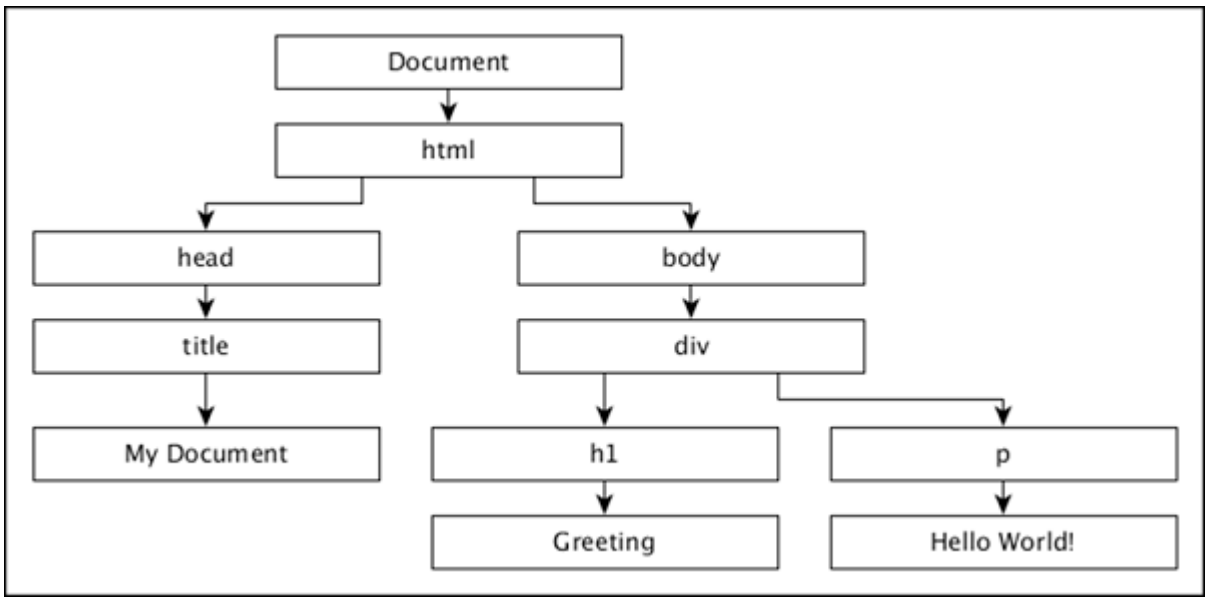
```
<!DOCTYPE html>
<html lang = "en">
   <head>
      <meta charset = "UTF-8">
      <title></title>
   </head>

   <body>
   </body>
</html>
```

Document Object Model (DOM)

When a HTML page is loaded by a browser, it is converted to a hierarchical structure. Every tag in HTML is converted to an element / object in the DOM with a parent-child hierarchy. It makes our HTML more logically structured. Once the DOM is formed, it becomes easier to manipulate (add/modify/remove) the elements on the page.

Example: Let us understand the DOM using the following HTML document −

```
<!DOCTYPE html>
<html lang = "en">
   <head>
      <title>My Document</title>
   </head>
   <body>
      <div>
         <h1>Greeting</h1>
         <p>Hello World!</p>
      </div>
   </body>
</html>
```



Cascading Style Sheets (CSS)

While HTML gives a structure to the webpage, CSS styles makes the webpage more pleasant to look at. CSS is a Style Sheet Language used to describe the presentation of a document written in HTML or XML (including XML dialects like SVG or XHTML). CSS describes how elements should be rendered on a webpage.

Scalable Vector Graphics (SVG)

SVG is a way to render images on the webpage. SVG is not a direct image, but is just a way to create images using text. As its name suggests, it is a Scalable Vector. It scales itself according to the size of the browser, so resizing your browser will not distort the image. All browsers support SVG except IE 8 and below. Data visualizations are visual representations and it is convenient to use SVG to render visualizations using the D3.js.

Think of SVG as a canvas on which we can paint different shapes. So to start with, let us create an SVG tag −

Example:

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

The default measurement for SVG is pixels, so we do not need to specify if our unit is pixel. Now, if we want to draw a rectangle, we can draw it using the code below −

Example:

```
<svg width = "500" height = "500">
   <rect x = "0" y = "0" width = "300" height = "200"></rect>
</svg>
```

We can draw other shapes in SVG such as − Line, Circle, Ellipse, Text and Path.

Example:

```
<svg width = "500" height = "500">
   <rect x = "0" y = "0" width = "300" height = "200" fill =
"yellow"></rect>
</svg>
```

Just like styling HTML elements, styling SVG elements is simple. Let us set the background color of the rectangle to yellow. For that, we need to add an attribute "fill" and specify the value as yellow as shown above.

JavaScript

JavaScript is a loosely typed client side scripting language that executes in the user's browser. JavaScript interacts with HTML elements (DOM elements) in order to make the web user interface interactive. JavaScript implements the **ECMAScript** Standards, which includes core features based on **ECMA-262** specifications as well as other features, which are not based on the ECMAScript standards. JavaScript knowledge is a prerequisite for D3.js.