By OnlineInterviewQuestions.com

D3 JS interview questions

D3.js is defined as a JavaScript-based library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

Data visualization Interview Questions and answers.

Q1. Define D3.js?

D3.js is defined as a JavaScript-based library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

Q2. What does D3 stand for?

D3 stands for **Data-Driven Documents**

Q3. Who developed D3.js?

Mike Bostock wrote D3.js based on his work during his Ph.D. studies at the Stanford Visualization Group. Mike worked at the The New York Times for a while and is now independently working on D3.js.

Q4. Why use to D3.js?

You can use D3 js because

- D3.js lets you to build the data visualization framework
- D3.js focuses on binding data to DOM elements.
- D3.js is written in JavaScript and uses a functional style which means you can reuse code and add specific functions to your heart's content.

Q5. How D3.js identify on which elements to operate?

D3.js uses CSS-style selectors to identify elements on which to operate.
Example:

```
d3.selectAll("p").style("color", "white");
```

Q6. Explain selections in D3.js ?

D3 implements a declarative approach, operating on arbitrary sets of nodes called selections.
Also, Read **Neo4j interview questions**

Q7. Explain, what is the use of “Enter” and “Exit” selection in D3.js ?

In D3.js “Enter” selection is used to create new nodes for incoming data and “Exit” selection is used eliminate outgoing nodes that are no longer required.

Q8. List Type of sliders are available in D3.js?

There are 7 types of slider are available in D3.js, they are

- Default slider
- Slider with start value
- Slider with slide event: 0
- Slider with default axis
- Slider with custom axis
- Slider with min, max and step values
- Vertical slider

Q9. Explain transition in D3.js?

A transition is a selection-like interface for animating changes to the DOM. Instead of applying changes instantaneously, transitions smoothly interpolate the DOM from its current state to the desired target state over a given duration.

To apply a transition, select elements, call selection.transition and then make the desired changes.

For example:

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

```
    .style("background-color", "red");
```

Q10. List the command to interpolate two objects in D3.js?

d3.interpolateObject(a,b) command is used to interpolate two objects in d3.js

Q11. which is the correct way to use XML file for d3?

The correct way to use XML file for d3 is d3. xml(url[mimeType][,callback]) . This command is used to create a request for the XML file at the specified url.

Q12. What is the best way to create the stacked chart in d3 js?

The best **way to create the stacked chart in d3 js** with the help of the following steps:

- Enter the data in a worksheet and highlight the data.
- Click the Insert tab and click Chart. Click Area and click Stacked Area.

Q13. How to import XML data using d3.js?

You can import XML data using d3.js with parsed XML data objects. d3.?csv().

Q14. Which is not a valid scale in d3 js?

If input is greater than 10 or less than 0 in domain or the range is greater than 600 then it is not a valid scale in d3 js.

Q15. What is different between d3.scale.linear() and d3.scaleLinear().

To create scales showing the linear relationship between the output and input we use **d3.scale.linear()** and **d3.scaleLinear()**. The domain and range of the scale are set by default at the interval (0,1) which expresses the function y = x. It can be used for flipped range, nice bounds, round ranges, non-numerical range, camping, etc.

There is a minor **difference between d3.scale.linear() and d3.scaleLinear()** which is on the basis of its use.

- If you want to create d3.js linear scale on version 3 then use API d3.scale.linear()

- If you want to create d3.js linear scale on version 4 and 5 then use API d3.scaleLinear()

Q16. How to alter zoom modes in d3 js?

With the help of the d3.zoom function you can alter zoom modes in d3 js.

Q17. How to set initial zoom level in D3.js?

You can set initial zoom level in D3.js by setting d3.behavior.zoom().translate([100,50]).scale(.5);

Q18. How to resize an SVG when the window is resized in d3.js?

You can resize an SVG when the window is resized in d3.js by resize function, for example Svg = d3.select("#div_basicResize") .append("?svg").

Q19. How to get Mouse position in D3.js?

You can use **d3.mouse()** function in order to get the Mouse position in D3.js. This function will return the x-coordinate and y-?coordinate of the current event.

Q20. What is the difference between canvas and SVG in d3.js?

In **SVG** we can re-select a circle and modify it or access its properties. Whereas in **Canvas** you can't bind data to shapes within the canvas as the canvas only comprises pixels.

Q21. How to format the date in d3.js?

You can format the date in d3.js with the help of d3 time formatting, for example d3.time.format("%Y-%m-%d").

Q22. List types of loops available in D3.js with syntax?

The types of loops available in D3.js with syntax can be demonstrated by:

- Creating an Empty SVG - const svg = d3.create("svg");

- Creating a Circle Element - var circle = svg
- Function to Contract the Circle - function contractCircle()

Q23. Explain axes in D3.js?How to create d3.js axes without numbering?

In d3.js axes renders human-readable reference marks for scales. D3 provides the following functions to draw axes:

- d3.axisTop() - Creates top horizontal axis.
- d3.axisRight() - Creates vertical right-oriented axis.
- d3.axisBottom() - Creates bottom horizontal axis.
- d3.axisLeft() - Creates left vertical axis.

Q24. Explain Transition is D3.js?

In d3.js a **transition** is a selection-like interface for animating changes to the DOM therefore instead of applying changes instantaneously, transitions smoothly interpolate the DOM from its current state to the desired target state over a given duration.

Q25. How to calculate the area of the polygon in d3.js?

To calculate the area of the polygon in d3.js you can use **d3. polygonArea**. It returns the area of the specified polygon. If the vertices of the polygon are in counterclockwise order the returned area is positive; otherwise, it is negative, or zero.