HTML Interview Questions

1. What is HTML?

HTML stands for HyperText Markup Language and is the language of the internet. It is the standard text formatting language used for creating and displaying pages on the Internet

HTML documents are made up of the elements and the tags that format it for proper display on pages.

2. What are HTML tags?

We use HTML tags for placing the elements in the proper and appropriate format. Tags use the symbols <, and > to set them apart from the HTML content.

The HTML tags need not be closed always. For example, in the case of images, the closing tags are not required as <img> tag.

3. What are HTML Attributes?

Attributes are the properties that can be added to an HTML tag. These attributes change the way the tag behaves or is displayed. For example, a <img> tag has an src attribute, which you use to add the source from which the image should be displayed.

We add attributes right after the name of the HTML tag, inside the brackets. We can only add the attributes to opening or self-closing tags, but never be in closing tags.

4. What is a marquee in HTML?

Marquee is used for scrolling text on a web page. It scrolls the image or text up, down, left, or right automatically. To apply for a marquee, you have to use </marquee> tags.

5. How do you separate a section of texts in HTML?

We separate a section of texts in HTML using the below tags:

-
 tag – It is used to separate the line of text. It breaks the current line and shifts the flow of the text to a new line.
- <p> tag–This tag is used to write a paragraph of text.
- <blockquote> tag–This tag is used to define large quoted sections.

6. Define the list types in HTML?

The list types in HTML are as below:

- Ordered list–The ordered list uses <ol> tag and displays elements in a numbered format.
- Unordered list–The unordered list uses <ul> tag and displays elements in a bulleted format.
- Definition list–The definition list uses <dl>, <dt>, <dd> tags and displays elements in definition form like in a dictionary.

7. How do you align list elements in an HTML file?

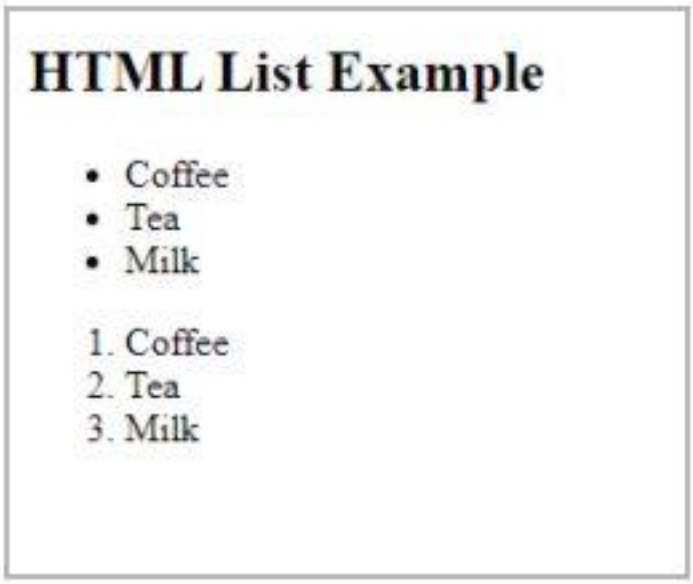
We can align the list elements in an HTML file by using indents. If you indent each nested list in further than the parent list, you can easily align and determine the various lists and the elements that it contains.

8. Differentiate between an Ordered list and an Unordered list?

An unordered list uses <ul> tags and each element of the list is written between <li> tags. The list items are displayed as bullets rather than numbers.

An ordered list uses <ol> tags and each element of the list is written between <li> tags. The list items are displayed as numbers rather than bullet points.

```
<!DOCTYPE html>

<html>

  <body>

    <h2>HTML List Example</h2>

    <ul>

      <li>Coffee</li>

      <li>Tea</li>

      <li>Milk</li>

    </ul>

    <ol>

      <li>Coffee</li>

      <li>Tea</li>

      <li>Milk</li>

    </ol>

  </body>

</html>
```



9. How do you display a table in an HTML webpage?

The HTML <table> tag is used to display data in a tabular format. It is also used to manage the layout of the page, for example, header section, navigation bar, body content, footer section. Given below are the list of HTML tags used for displaying a table in an HTML webpage:

Tag	Description
<table>	It defines a table.
<tr>	It defines a row in a table.
<th>	It defines a header cell in a table.
<td>	It defines a cell in a table.

<caption>	It defines the table caption.
<colgroup>	It specifies a group of one or more columns in a table for formatting.
<col>	It is used with <colgroup> element to specify column properties for each column.
<tbody>	It is used to group the body content in a table.
<thead>	It is used to group the header content in a table.
<tfooter>	It is used to group the footer content in a table.

10. How would you display the given table on an HTML webpage?

5 pcs	10	5
1 pcs	50	5

The HTML Code for the problem depicted above is:

```
<table>
  <tr>
    <td>50 pcs</td>
    <td>100</td>
    <td>500</td>
  </tr>
  <tr>
    <td>10 pcs</td>
    <td>5</td>
    <td>50</td>
  </tr>
</table>
```

11. How do we insert a comment in HTML?

We can insert a comment in HTML by beginning with a lesser than sign and ending with a greater than sign. For example, “<!-“ and “->.”

12. How do you insert a copyright symbol in HTML?

You can insert a copyright symbol by using © or © in an HTML file.

13. What is white space in HTML?

An empty sequence of space characters is called the white space in HTML. This white space is considered as a single space character in the HTML.

White space helps the browser to merge multiple spaces into one single space, and so taking care of indentation becomes easier. White space helps in better organizing the content and tags, making them readable and easy to understand.

14. How do you create links to different sections within the same HTML web page?

We use the <a> tag, along with referencing through the use of the # symbol, to create several links to different sections within the same web page.

15. How do you create a hyperlink in HTML?

We use the anchor tag <a> to create a hyperlink in HTML that links one page to another page. The hyperlink can be added to images too.

16. Define an image map?

An image map in HTML helps in linking with the different kinds of web pages using a single image. It can be used for defining shapes in the images that are made part of the image mapping process.

17. Why do we use a style sheet in HTML?

A style sheet helps in creating a well-defined template for an HTML webpage that is both consistent as well as portable. We can link a single style sheet template to various web pages, which makes it easier to maintain and change the look of the website.

18. What is semantic HTML?

Semantic HTML is a coding style. It is the use of HTML markup to reinforce the semantics or meaning of the content.

For example: In semantic HTML   tag is not used for bold statement as well as   tag is not used for italic. Instead of these we use and tags.

19. What is SVG in HTML?

HTML SVG is used to describe the vector or raster graphics. SVG images and their behaviors are defined in XML text files.

We mostly use it for vector type diagrams like pie charts, 2-Dimensional graphs in an X, Y coordinate system.

```
<svg width="100" height="100">

  <circle cx="50" cy="50" r="40" stroke="yellow" stroke-width="4" fill="red" />

</svg>
```

20. What would happen if there is no text between the HTML tags?

There would be nothing to format if there is no text present between the tags. Therefore, nothing will appear on the screen.

Some tags, such as the tags without a closing tag like the <img> tag, do not require any text between them.

21. How do you create nested web pages in HTML?

Nested web pages basically mean a webpage within a webpage. We can create nested web pages in HTML using the built-in iframe tag. The HTML <iframe> tag defines an inline frame. For example:

```
<!DOCTYPE html>
```

```
<html>
  <body>
    <h2>HTML Iframes example</h2>
    <p>
      specify the size of the iframe using the height and width attributes:
    </p>
    <iframe src="https://simplilearn.com/" height="600" width="800"></iframe>
  </body>
</html>
```

22. How do you add buttons in HTML?

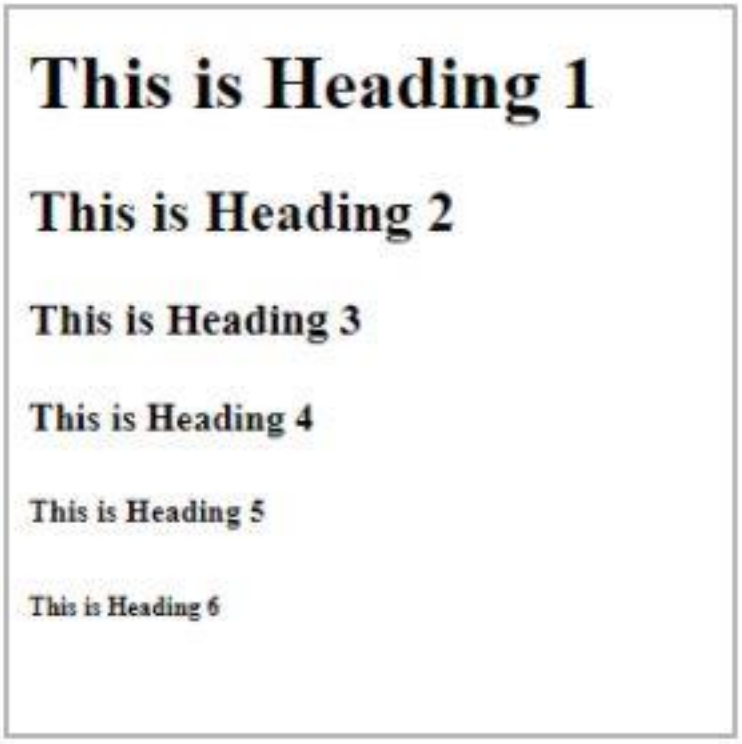
We can use the built-in Button tag in HTML to add buttons to an HTML web page.

```
<!DOCTYPE html>
<html>
  <body>
    <h2>HTML Button Tag Example</h2>
    <button name="button" type="button">CLICK ME</button>
  </body>
</html>
```

23. What are the different types of headings in HTML?

There are six types of heading tags in HTML which are defined with the <h1> to <h6> tags. Each type of heading tag displays a different text size from another. <h1> is the largest heading tag and <h6> is the smallest. For example:

```
<!DOCTYPE html>

<html>

  <body>

    <h1>This is Heading 1</h1>

    <h2>This is Heading 2</h2>

    <h3>This is Heading 3</h3>

    <h4>This is Heading 4</h4>

    <h5>This is Heading 5</h5>

    <h6>This is Heading 6</h6>

  </body>

</html>
```



24. How do you insert an image in the HTML webpage?

You can insert an image in the HTML webpage by using the following code:

```
<!DOCTYPE html>
<html>
  <body>
    <h2>HTML Image Example</h2>
    <img src="tulip.jpeg" />
  </body>
</html>
```

25. What is the alt attribute in HTML?

The alt attribute is used for displaying a text in place of an image whenever the image cannot be loaded due to any technical issue.

```
<!DOCTYPE html>

<html>

  <body>

    <h2>HTML Alt Example</h2>

    <img src="tulip.jpeg" alt="Tulip Garden" />

  </body>

</html>
```

26. How are hyperlinks inserted in the HTML webpage?

You can insert a hyperlink in the HTML webpage by using the following code:

```
<!DOCTYPE html>

<html>

  <body>

    <h2>HTML Hyperlink Example</h2>

    <a href="url">link text</a>

  </body>
```

```
</html>
```

27. How do you add colour to the text in HTML?

You can add colour to the text in HTML by using the following code:

```
<!DOCTYPE html>

<html>

  <body>

    <h2>HTML Color Text Example</h2>

    <h1 style="color: Red">Hello HTML</h1>

    <p style="color: Blue">Line 1</p>

    <p style="color: Green">Line 2</p>

  </body>

</html>
```

28. How do you add CSS styling in HTML?

There are three ways to include the CSS with HTML:

- Inline CSS: It is used when less amount of styling is needed or in cases where only a single element has to be styled. To use inline styles add the style attribute in the relevant tag.
- External Style Sheet: This is used when the style is applied to many elements or HTML pages. Each page must link to the style sheet using the <link> tag:

```
<head>
```

```
  <link rel="stylesheet" type="text/css" href="mystyle.css" />
</head>
```

- Internal Style Sheet: It is used when a single HTML document has a unique style and several elements need to be styled to follow the format. Internal styles sheet is added in the head section of an HTML page, by using the <style> tag:

```
<head>
  <style type="text/css">
    hr {
      color: sienna;
    }
    p {
      margin-left: 20px;
    }
    body {
      background-image: url("images/back40.gif");
    }
  </style>
</head>
```

29. What hierarchy do the style sheets follow?

If a single selector includes three different style definitions, the definition that is closest to the actual tag takes precedence. Inline style takes priority over embedded style sheets, which takes priority over external style sheets.

30. How do you add JavaScript to an HTML webpage?

JavaScript is used for making HTML web pages more interactive, and user-friendly. It is a scripting language that allows you to interact with certain elements on the page, based on user input. As with CSS, there are three major ways of including JavaScript:

- Inline:

You can add JavaScript to your HTML elements directly whenever a certain event occurs. We can add the JavaScript code using attributes of the HTML tags that support it. Here is an example that shows an alert with a message when the user clicks on it:

<button onclick="alert('Click the Button!');">

Click!

</button>

- Script block:

You can define a script block anywhere on the HTML code, which will get executed as soon as the browser reaches that part of the document. This is why script blocks are usually added at the bottom of HTML documents.

<html>

```
  <script>
    var x = 1;
    var y = 2;
    var result = x + y;
    alert("X + Y is equal to " + result);
  </script>
</html>
```

- External JavaScript file:

You can also import the JavaScript code from a separate file and keep your HTML code clutter-free. This is especially useful if there is a large amount of scripting added to an HTML webpage.

```
<html>
  <script src="my-script.js"></script>
</html>
```

As you get prepared for your job interview, we hope that these HTML Interview Questions have provided more insight into what types of questions you are likely to come across.

Basic CSS Interview Questions

Let us begin with the basic CSS interview questions!

1. Name some CSS frameworks.

CSS frameworks are libraries that make web page styling easier. Some of them are Foundation, Bootstrap, Gumby, Ukit, Semantic UI, etc.

2. What do you understand by the universal sector?

A universal selector is a selector that matches any element type's name instead of selecting elements of a particular type.

Example:

```
<style>

* {

  color: blue;

  font-size: 10px;

}

</style>
```

3. Tell us about the use of the ruleset.

The ruleset is used for the identification of selectors, which can be attached with other selectors. The two parts of a ruleset are:

- Declaration block: contains one or more semicolon-separated declarations
- Sector: indicates the HTML element needed to be styled

4. What are the elements of the CSS Box Model?

The CSS box model defines the layout and design of CSS elements. The elements are content (like text and images, padding (the area around content), border (the area around padding), and margin (the area around the border).

5. Differentiate between CSS3 and CSS2.

The main difference between CSS3 and CSS2 is that CSS divides different sections into modules and supports many browsers. It also contains new General Sibling Combinators responsible for matching similar elements.

6. How can CSS be integrated into an HTML page?

There are three ways of integrating CSS into HTML: using style tags in the head section, using inline-styling, writing CSS in a separate file, and linking into the HTML page by the link tag.

7. Explain a few advantages of CSS.

With CSS, different documents can be controlled using a single site, styles can be grouped in complex situations using selectors and grouping methods, and multiple HTML elements can have classes.

8. What is meant by RGB stream?

RGB represents colors in CSS. The three streams are namely Red, Green, and Blue. The intensity of colors is represented using numbers 0 to 256. This allows CSS to have a spectrum of visible colors.

9. What was the purpose of developing CSS?

CSS was developed to define the visual appearances of websites. It allows developers to separate the structure and content of a website that was not possible before.

10. What is the difference between a class and an ID?

Ans. Class is a way of using HTML elements for styling. They are not unique and have multiple elements. Whereas ID is unique and it can be assigned to a single element.

Intermediate CSS Interview Questions

In the next section, let us learn some intermediate level CSS interview questions!

1. Define z-index.

This is one of the most frequently asked CSS interview questions. Z-index is used to specify the stack order of elements that overlap each other. Its default value is zero and can take both negative and positive values. A higher z-index value is stacked above the lower index element. It takes the following values- auto, number, initial, and inherit.

2. What are the benefits of CSS Sprites?

With CSS sprites, loading multiple images is not an issue.

- Blinking is not seen.
- Advanced downloading of assets does not take place until needed.

3. How can you target h3 and h2 with the same styling?

Multiple elements can be targeted by separating with a comma:

h2, h3 {color: red;}

4. Name media types allowed by CSS.

The different media types allowed by CSS are:

- speech
- audio
- visual
- tactile media
- continuous or paged media
- grip media or bitmap
- interactive media

5. How can you use CSS to control image repetition?

The background-repeat property is used to control the image. Example:

```
h3 {

background-repeat: none;

 }
```

6. Tell us about the property used for image scroll controlling?

The background-attachment property is used to set whether the background image is fixed or it scrolls with the rest of the page. Example for a fixed background-image:

```
body {

  background-image: url('url_of_image');

  background-repeat: no-repeat;

  background-attachment: fixed;

}
```

7. Name some font-related CSS attributes.

The font-related attributes are Font- style, variant, weight, family, size, etc.

8. Define contextual selectors.

In CSS, contextual selectors allow developers to specify styles of different parts of the document. Styles can be assigned directly to specific HTML tags or create independent classes and assign tags to them.

9. Explain responsive web design.

Responsive Design is a web page creation approach that uses flexible images, flexible layouts, and CSS media queries. This design approach aims to build web pages that detect the orientation and screen size of the visitors so that the layout can be changed accordingly.

10. Tell us about the general CSS nomenclature.

In CSS, the styling commands are written in value and property fashion. CSS includes a system terminator-semicolon. The entire style is wrapped in curly braces and attached to the selector. This creates a style sheet that can be applied to an HTML page.

11. What are the limitations of CSS?

1. CSS cannot always assure compatibility with every browser; you need to be cautious while choosing the style selector.
2. The parent selector tag is not available, thus you can’t select the parent selector tag.
3. Some selectors can lead to cross-browser issues due to their less browser-friendly behavior.
4. We cannot request a webpage through CSS.

12. How to include CSS in the webpage?

1. With the help of a link tag, you can include an external style sheet file as a CSS file into your HTML file.

2. You can add CSS styles included within your HTML page and write it in the stand-alone stylesheet form of CSS.
3. CSS can be included directly in the HTML tag by adding an inline style to HTML elements.
4. One can import an external stylesheet file as a new CSS file by using the @import rule.

13. What are the different types of Selectors in CSS?

Universal Selector, Element type Selector, ID selector, Class selector, Descendant combinatory, Child Combinator, General Sibling Combinator, Adjacent sibling combinator, and Attribute selector.

14. What is a CSS Preprocessor? What are Sass, Less, and Stylus? Why do people use them?

CSS preprocessor is a tool used to enhance the basic functionality and let us use the complex logical syntax such as variables, functions, mixins, and code nesting within vanilla CSS scripts themselves.

1. Sass (Syntactically Awesome Style Sheets) uses .sass extension. It is used for indentation; it doesn't use semicolons or curly brackets.
2. Less (Leener Stylesheets) uses .less extension. It is easy to add to any JavaScript Project by using NPM or less.js file. Here, @ is used to define the variables.
3. Stylus provides great flexibility in writing syntax. It is able to use native CSS as well as the exclusion of brackets, colons, and semicolons. There is no need to use @ or $ to define the variables.

People use SASS, LESS, and Stylus in order to extend the basic functionality of vanilla CSS.

15. What is VH/VW (viewport height/ viewport width) in CSS?

VH and VW are CSS units used to measure viewport height and viewport width respectively in percentage form in the responsive design techniques. E.g. If the height of the browser is 1000px, then VH is 1/100 of the height of the viewport that is 1000px*(1/100) = 10px, which is the height of the browser. The same applies to VW (viewport width).

16. Difference between reset vs normalize CSS? How do they differ?

1. Reset CSS is used to remove all built-in styles in the browser such as paddings, margins, and font sizes, and can be reset using all the elements.
2. Normalize CSS is used to make all built-in styles in the browser consistent and correct bugs as per varying browsers.

17. What is the difference between inline, inline-block, and block?

1. Block Elements are and <p>. They usually start on a new line and can take space for an entire row or width.
2. Inline elements are <a>, , , and <img> tags. They don't start on a new line. However, they appear on the same line as the content and tags beside them.
3. Inline block elements have padding and margins and set height and width values. Though, they are similar to inline elements.

18. Is it important to test the webpage in different browsers?

Yes, it is the most crucial thing or the most important trial to do when you design a webpage for the first time and make changes to it. Testing your website periodically in different browsers will help you make every webpage compatible with it as browsers have been going through many updates.

19. What are Pseudo classes?

Pseudo-classes are the type of pseudo-elements that don’t exist in a normal document tree. It allows selecting the regular elements under certain conditions especially when we try to hover over the link; the anchor tags are :link, :visited, :hover, :active, :focus

In this example, the color will be red on the anchor tag when it’s hovered.

/* mouse over link */

a:hover {

color: #FFOOFF;

}

20. How do you specify units in the CSS? What are the different ways to do it?

There are mainly four different units in the CSS that are px, em, pt, and percentage (%).

1. Px (Pixel) is used for fine-grained control and alignment and not cascade. To get it sharp, we can use 1px or multiple of px.
2. Em is used to maintain relative size and responsive fonts. 1em = 16px having also the same font size. It is advisable to set the font size to 10px in common practice.
3. Pt (point) is a fixed-size unit that is used in print. 1pt = 1/72 inch.
4. Percentage (%) is used to set the font size with respect to the font size of the body. Thus, it is necessary to set the reasonable font size of the body.

21. Does margin-top or margin-bottom have an effect on inline elements?

No, mMargin-top or margin-bottom does not have an effect on the inline elements.

22. What property is used for changing the font face?

The font-family property is used for changing the font face that is applied to the element in the DOM.

23. What are the differences between adaptive design and responsive design?

- **Adaptive Design:**
- Main focus is to develop a website in multiple fixed layout sizes.
- Offers good control over the design to develop variation in screens.
- It is very time-consuming and takes a lot of effort to build the best possible adaptive design as examining it will need to go for many options with respect to the realities of the end user.

- There are six standard screen sizes for the appropriate layouts; they are 320px, 480px, 760px, 960px, 1200px, 1600px to design.
- **Responsive Design**
- Main focus is to show content with respect to browser space.
- Offers less control over the design.
- It takes less time to build the design and there is no screen size issue irrespective of content.
- It uses CSS media queries to design the screen layouts with respect to specific devices and property changes in the screen.

24. How are the CSS selectors matched against the elements by the browser?

Initially, there is a filtration of elements in the DOM via browsers with respect to key selectors that are traversed until we get parents' elements to determine the matches. Then the browser works on finding all the span elements present in the DOM and traverses them to parent elements to check whether they are matched to paragraph p elements. At last, when the browser finds all matches as parents, the matching process will be stopped and there will be black color applied to the content.

25. How is the border-box different from the content box?

Border-box consists of properties such as content, padding, and the border with respect to height and width. However, Content-box is a default value property used for the box-sizing as well as it helps to add border and padding to give proper height and width to the box without having a border and padding properties.

26. How is opacity specified in CSS3?

Opacity is the measure of content transparency. It is measured in the range from 0 to 1. Value 1 means the content is completely opaque. It is not supportable in the internet browser. Also, the 60% of opacity is applicable in the div section where we need to apply the filter property (polyfill) to make it completely opaque.

27. What is cascading in CSS?

Cascading is defined as the process of style declaration and its weight that will help the browser in selecting the styling rules with respect to time.

28. When does DOM reflow occur?

DOM reflow occurs when you insert, move, update, remove, and animate the elements in the DOM as well as when you modify content on the page and change style.

29. Different Box Sizing Property?

Content-box, Padding-box and Border-box

30. How to center align a div inside another div?

A div inside another div A is center aligned with the help of aligning div property to content via HTML script and CSS to the element in the DOM.

31. What is the grid system?

The CSS grid system is a type of powerful layout of 2 dimensional systems with respect to columns and rows.

32. What are the different ways to hide the element using CSS?

display: none, visibility: hidden, position: absolute

33. What does the: root pseudo-class refer to?

The: root selector pseudo-class refers to the CSS selector level 3. It helps to target the highest-level parent element present in the DOM.

34. What does Accessibility (a11y) mean?

Accessibility is to make the system accessible in such a manner that the website should have the text-to-speech capability, for people with physical disabilities, designed with the help of software or hardware combinations.

35. How do I restore the default value of a property?

keyword “initial” is used to restore the default value of a property.

36. Difference between CSS grid vs flexbox?0

1. CSS Grid Layout is a two-dimensional system along with rows and columns. It is used for large-sized layouts.
2. Flexbox is a Grid layout with a one-dimensional system either within a row or a column. It is used for the components of an application.

37. How does Calc work?

Calc can be used to specify the result of the mathematical operation of two or more elements. For example to specify the width elements by the addition of two or more elements, we can write as

```
.foo {

Width: calc(100px+50px)

}
```

38. What do CSS Custom properties variables mean?

CSS Custom properties variables are defined for CSS variables as well as cascading variables with specific values that can be reused.

39. What is the difference between CSS variables and pre-processor (SASS, LESS, Stylus) variables?

1. CSS variables don’t need a pre-processor. It is cascaded, accessed, and manipulated in JavaScript.
2. Preprocessor variables don’t cascade.

40. What does * { box-sizing: border-box; } do? What are its advantages?

Box-sizing: border-box is used to provide the inner dimension for the elements in the document by providing padding and border with respect to the height and width of the content.

41. What does !important mean in CSS?

The style “!important” in the CSS has the highest precedence. Also, the cascaded property will be overridden with it.

42. What is progressive rendering? How do you implement progressive rendering on the website? What are its advantages?

Progressive rendering is the process of improving the performance of a webpage with respect to loading time in order to display the content speedily. The use of loading the lazy loading of the image with the help of Intersection Observer API via viewport.

43. Does style1.css have to be downloaded and parsed before style2.css can be fetched?

No. The CSS file will be downloaded via browser in order to appear on the HTML page.

44. How to determine if the browser supports a certain feature?

@support tag in the CSS is used to scan and determine whether the browser supports a certain feature or not.

45. How does absolute positioning work?

Absolute positioning is used to place the element which is then removed from the HTML document from the normal workflow without creating any space for the element in the HTML page layout. The element can be positioned respectively to the closest positioned ancestor; otherwise, if the ancestor is not found, the element is placed with respect to the initial container box. The values provided to the top, right, left and bottom determine the final position of the element.

46. Does this property work overflow: hidden?

Overflow: the hidden property is used to specify the content's clipping. We need to add scrollbars to the content area at the time of specified container size exceeding the content limit where the content gets enclosed. This makes the content invisible via clipping; also the overflow value will be hidden.

Advanced CSS Interview Questions

Here are some advanced CSS interview questions!

1. When should you use translate () instead of absolute positioning?

Translate is a CSS transform value. On changing opacity or transform, browser reflow or repaint is not triggered. Transform requires the browser to create a GPU layer for elements but using the CPU changes absolutes positioning properties. Translate () is more efficient and results in shorter paint times. On using translate (), element occupies original space, unlike in changing absolute positioning.

2. Name different ways to position some aspects in CSS.

The positioning property specifies the positioning method type. The five different position values are fixed, static, absolute, sticky, and relative. The elements are positioned using top, left, right, and bottom properties. These properties need to be set first, and they work depending on position value.

3. What are mixins?

A mixin is similar to a function block of code returning a single value—mixin output lines of Sass code that directly compiles into CSS styles. At the same time, the function returns a value that becomes the value for a CSS property or something that can be passed to another mixin.

4. How can you optimize the webpage for prints?

Identify and control 'content areas' of the website. A website generally has a footer, header, sidebar, navbar, and main content area. Most of the work is done by controlling the content area. So, conquer print media without changing the website's integrity by using page breaks, creating a stylesheet for print, size your page for print, and avoid unnecessary HTML tables.

5. What is meant by CSS working under the hood?

When a browser displays a document, it combines style information and document content. The document is processed in two stages:

- Conversion of HTML and CSS into Document Object Model
- DOM displays contents of browser

6. Differentiate between the use of ID selector and class selector.

ID Selector:

<style>

{

text-align: right;

color: blue;

}

</style>

CSS class Selector:

<style>

.right {

text-align: right;

color: blue;

}

</style>

7. Tell us about CSS float property.

The float property of CSS positions an image to the right or left as needed, including text wrapping around it. All properties of elements used before it remain unchanged.

8. What do you understand by pseudo-elements?

Pseudo-elements provide special effects to some selectors. CSS finds use in applying styles in HTML markups. If additional markup or style is not feasible for a document, the pseudo-elements help by allowing extra markup without interfering with the original document.

9. Differentiate between logical and physical tags.

Logical tags mainly focus on content and are older as compared to physical ones. Logical ones do not find much usage in presentation and terms of aesthetics. At the same time, physical ones find application in presentation.

10. How media types in CSS work?

The four types of media properties are print, speech, and screen. Example of using print-media type:

```
@media print {

h2 {

background-color: blue;

 }

}
```

Frequently Asked CSS Interview Questions

In this last section, we look at the most frequently asked CSS interview questions!

1. Tell us something about CSS3.

CSS3 is divided into modules and is supported by almost every browser. Many graphics-related characteristics are introduced in CSS3 like box-shadow, Border-radius, and flexbox. A user can create precise multiple background images using properties like background-position, background-repeat, and background-image styles.

2. How is a CSS selector used?

With a CSS selector, we can choose the content we want to style to bridge between HTML files and style sheets. CSS selector syntax is "select" HTML elements created on their class, id, type, etc.

3. What are CSS image scripts?

A group of images placed into one image is a CSS image script. It can reduce load time and project multiple images into a single web page.

4. Explain CSS specificity.

CSS specificity is a rank or score that decides style declaration to be used to an element. ID selectors have high specificity, while universal selector * has low specificity. The four CSS categories that authorize the selector's specificity level are IDs, inline style, elements/pseudo-elements, and classes and attributes.

5. Define gradients in CSS.

A property of CSS that allows displaying smooth transformation between two or more specified colors. The types of gradients are linear and radial.

6. What are the properties of flexbox?

The properties of flexbox are flex-direction, wrap, flow, content, and align-items, and content.

7. Tell us about the use of the CSS Box Model.

The CSS Box model is a box binding HTML element that includes padding, border, margin, and the actual content. With the box model, we get the authority to add a border all around elements and define space between elements.

8. What are the position states in CSS?

The four-position states in CSS are relative, static, absolute, and fixed. The default position state is static.

9. Differentiate between absolute and relative in CSS.

The main difference is that relative is used for the same tag in CSS. If we write right:20 px, then padding shifts 20 px in the right. Whereas absolute is relative to the non-static parent, i.e., if we write right:20 px, the result will be 20 px far from the right edge of the parent element.

10. What is common between class and ID?

Both class and ID are used in HTML to assign a value from CSS. The ID is used as an element, whereas the class is used as a block.

Top Bootstrap Interview Questions and Answers

There is a pretty good chance of a web development job interview to feature Bootstrap-based questions. So, you need to keep up your Bootstrap preparation to grab the role. To help you with the cause, here are top Bootstrap interview questions with answers:

Question: What do you mean by the Bootstrap Grid System?

Answer: The Bootstrap Grid System is a responsive, mobile-first system that scales up to 12 columns as per the increase in the device or viewport size. The system features predefined classes for easy layout options and powerful mix-ins for generating effectively semantic layouts.

Question: Please explain Normalize in Bootstrap.

Answer: For establishing cross-browser consistency, Bootstrap makes use of Normalize. It is a small CSS file capable of offering better cross-browser consistency in the default styling of HTML elements. Also, Normalize.css is an HTML5-ready and modern alternative to CSS resets.

Question: Can you enumerate the various lists supported by Bootstrap?

Answer: Following are the three types of lists supported by Bootstrap:

- **Definition Lists –** This type of list allows each list item to have both the <dt> and the <dd> elements. The <dt> denotes definition term, which is the term or phrase being defined. The <dd> element contains the definition for the <dt> element.
- **Ordered Lists –** This type of list follows some kind of sequential order. Also, it is prefaced by numbers.
- **Unordered Lists –** Traditionally styled with bullets, an unordered list doesn't follow any particular order. Using the .list-unstyled class allows removing the bullets styling from the unordered list. For placing all list items on a single line, the .list-inline class can be used.

Question: What do you mean by Glyphicons? How do you use them?

Answer: Glyphicons are icon fonts that are used in web projects. Although Glyphicons Halflings aren't free and require licensing in general, they are available free of cost for Bootstrap projects. Add the following code anywhere you wish to use the Glyphicons:

```
<span class = “glyphicon glyphicon-search”></span>
```

Note: - For proper padding, it is advised to leave a space between the icon and the text.

Question: Could you explain how to use the Dropdown plugin in Bootstrap?

Answer: There are three ways of toggling the dropdown plugin's hidden content in Bootstrap:

- **With data attributes** – Add data-toggle = “dropdown” to some button or link to toggle a dropdown. For example,

```
<div class = "dropdown">
```

```
<a data-toggle = "dropdown" href = "#">Dropdown trigger</a>

<ul class = "dropdown-menu" role = "menu" aria-labelledby = "dLabel">

...

</ul>

</div>
```

- **With JavaScript –** Following method is used for calling the dropdown toggle via JS:

```
$('.dropdown-toggle').dropdown()
```

- Using data-target attribute in place of href="#" – If the web browser isn't enabling JavaScript, then it is better to keep links intact. For this, the data-target attribute is preferred over href="#". For example,

```
<div class = "dropdown">

<a id = "dLabel" role = "button" data-toggle = "dropdown" data-target = "#" href = "/somepage.html">

Dropdown

<span class = "caret"></span>

</a>

<ul class = "dropdown-menu" role = "menu" aria-labelledby = "dLabel">

...

</ul>

</div>
```

Question: Please provide an explanation of input groups in Bootstrap.

Answer: Input groups are simply extended Form Controls in Bootstrap. One can easily append and prepend buttons or text to the text-based inputs using the input groups. Appending and prepending content to an input field allows adding common elements to the user input.

You can append or prepend elements to a .form-control by:

- Wrapping it in a element with the class .input-group
- With the same element, put the extra content inside a element with the .input-group-addon class
- Finally, place the element before or after the <input> element as required

Question: How will you create a tabbed, pills, and vertical pills navigation menu in Bootstrap?

Answer:

1. **For creating a tabbed navigation menu**
    1. Start with a basic unordered list with the .nav base class
    2. Now, add the .nav-tabs class
2. **For creating a pills navigation menu**
    1. Start with a basic unordered list with the .nav base class
    2. Now, add the .nav-pills class
3. **For creating a vertical pills navigation menu**
    1. Stack the pills vertically using the .nav-stacked class
    2. Now, add the .nav and .nav-pills classes

Question: What do you understand by the Bootstrap navbar? How will you create one?

Answer: One of the most prominent features of Bootstrap, a navbar is a responsive 'meta' component that serves as a navigation header for an application or website. There can be several navbars in an application or website.

In mobile views, a navbar collapses and becomes horizontal when the available viewport width increases. The navbar includes styling for basic navigation and site names. Here is how to create a navbar in Bootstrap:

- Add the .navbar and .navbar-default classes to the <nav> tag
- Next, add role = "navigation" to the <nav> element
- Now, add .navbar-header, a header class, to the element. For making the text slightly larger, include an <a> element with the navbar-brand class
- Add an unordered list with .nav and .navbar-nav classes for adding links to the Bootstrap navbar

Question: Can you explain Bootstrap breadcrumb?

Answer: A Bootstrap breadcrumb is a great way to display hierarchy-based information for a website. Simply, it is an unordered list with a .breadcrumb class. CSS automatically adds the separator for a Bootstrap breadcrumb.

In blogs, the breadcrumb can display categories, publishing dates, or tags. It indicates the present page's location within a navigational hierarchy.

Question: How do you create and customize thumbnails in Bootstrap?

Answer: For creating thumbnails using Bootstrap, add a <a> tag with .thumbnail class around an image. It will add four pixels of padding as well as a gray border. When hovered, an animated glow outlines the image.

You can add any type of HTML content, such as buttons, headings, or paragraphs, into thumbnails. This is how to customize thumbnails using Bootstrap:

- Change the <a> tag with .thumbnail class to a tag
- Add anything that you need inside the tag. You can use the default span-based naming convention for sizing

Note: - If you wish to group multiple images then place them in an unordered list. Each list item will be floated to the left.

Question: Please explain Bootstrap alerts. Also, tell how you will create a Bootstrap Dismissal Alert.

Answer: Used for styling messages to the user, Bootstrap Alerts provide contextual feedback messages for typical user actions. To create a Bootstrap Dismissal Alert:

- Create a wrapper and add a .alert class and one of the 4 contextual classes, namely .alert-danger, .alert-info, .alert-success, and .alert-warning, for adding a basic alert
- Add the optional .alert-dismissable class to the
- Next, add a close button
- Finally, use the <button> element with the data-dismiss = “alert” data attribute

Question: Can you explain how to create an alternate and a striped progress bar using Bootstrap?

Answer: For creating a progress bar with various styles:

- Add a with a .progress class
- Inside the , add an empty with a .progress-bar class and a progress-bar-* class (* can be danger, info, success, or warning)
- Lastly, add a style attribute with the width expressed as a percentage. E.g., style = “80%”

For creating a striped progress bar:

- Add a with .progress and .progress-striped classes
- Inside the , add an empty with a .progress-bar class and progress-bar-* class, where * can be any of the danger, info, success, or warning
- Now, add a style attribute with the width being expressed as a percentage, E.g., style = “70%”

Question: What do you understand by Bootstrap media objects?

Answer: Bootstrap media objects are abstract object styles for building different types of components, such as blog comments and Tweets, which feature either a left-aligned or right-aligned image and textual content.

The main aim of a Bootstrap media object is to make code required for developing blocks of information incredibly smaller. In order to achieve easy extensibility, lightweight markup, and other desirable aspects, classes are applied to some of the simple markups.

Question: What purposes do the .media and .useful classes serve in Bootstrap?

Answer: The .media class allows a media object, such as audio, images, or video, to float to the left or right of a content block. For adding article lists or comment threads to an unordered list, we use the .useful class.

Question: Do you know what Bootstrap panels are? Also, explain how to create a Bootstrap panel with a heading.

Answer: Bootstrap panel components are used for putting your DOM component in a box. To get a basic panel, simply add .panel and .panel-default classes to the element. There are two ways of adding panel heading to a Bootstrap panel:

1. Use any of the <h1>, <h2>, <h3>, <h4>, <h5>, or <h6> tags with a .panel-title class (Adds a pre-styled heading)
2. Use the .panel-heading class (Adds a heading container to the panel)

Question: Can you explain the purpose of the Scrollspy plugin?

Answer: Using the Scrollspy plugin in Bootstrap allows you to target certain sections of the page based on the scroll position. Thereafter, you can add .active classes, based on the scroll position, to the Bootstrap navbar.

Question: Please enumerate the various contextual classes available for styling the panels in Bootstrap.

Answer: Various contextual classes used in Bootstrap for styling the panels, i.e. making them more meaningful, are:

- .panel-danger
- .panel-info
- .panel-primary
- .panel-success
- .panel-warning

Question: Is it possible to put a table within the Bootstrap panel?

Answer: Yes, it is possible to put a table within a Bootstrap panel. Use the .table class within a panel to get a non-bordered table within the same.

Note: - If there is an element containing .panel-body class then we need to add an extra border to the top of the table for clearly defined separation. In case there is no element containing the aforementioned class then the component moves from the panel header to the table without any issues.

Question: What do you mean by Bootstrap well?

Answer: In order to make the content appear sunken or adding an inset effect to a webpage, we use the Bootstrap well. In actuality, it is a container in .

In order to create a Bootstrap well, simply wrap the entire content that you want to appear in the Bootstrap well with a containing the .well class.

Question: Why do we use the affix plugin in Bootstrap?

Answer: We use the affix plugin in Bootstrap for affixing a to some certain location on a webpage. The plugin also allows toggling pinning on and off for the affixed . Social icons are the most popular example of using the affix plugin in Bootstrap.

Note: - The affixed starts from a particular location on the webpage and scrolls with it. However, after a certain mark, it will be locked in place, thus stopping scrolling with the rest of the webpage.

Question: What is the need of Bootstrap?

Answer: Bootstrap acts as a front-end framework that supports the development of various kinds of web applications such as HTML, JS, and CSS.

Question: What is the benefit of Bootstrap?

Answer: It helps in developing highly responsive, quick, and easy to use layout. The main advantage of Bootstrap is observed while developing mobile applications that have design templates. These templates help in creating UI, including alert tab, dropdown, forms, etc.

Question: What are the key components of Bootstrap? Explain?

Answer: The key components of Bootstrap include the following:

1. Scaffolding: It supports the grid system, styles, and backgrounds of various kinds.
2. CSS: It offers CSS files for developing web applications.
3. Customize: It supports the development of customizing the framework.
4. JS Plugins: It consists of JS and JQuery plugins required in the web designing and application process.

Question: What is a Bootstrap container?

Answer: A Bootstrap container consists of HTML code that can be placed for making them a highly responsive and fast web application.

Question: Define types of layout in Bootstrap? Explain?

Answer: There are two types of layout in Bootstrap include fixed design and fluid layout. The fluid layout helps in creating a full-width screen that adjusts with the browser size. The fixed layout, on the other hand, cannot adjust itself as per the browser screen and has a fixed layout of 940 px.

Question: Can we display code in Bootstrap? How?

Answer: Yes, we can display code in Bootstrap. It can be displayed using tags such as <code> tag and <pre> tag. The <code> tag can help in displaying the code inline while the <pre> tag can help in displaying code with different lines.

Question: Define Bootstrap alerts?

Answer: Bootstrap alerts are meant to create alert messages by adding styles and making them look attractive enough to grab the attention of the viewer.

Question: Define Bootstrap thumbnails?

Answer: Bootstrap thumbnails help using the layout images, text, videos, and other features in a grid system. This way, we can add several thumbnails through tags around the image.

Question: What will happen if we add many tags with the class of thumbnails around the image?

Answer: This will result in creating four pixels of padding and also develop a grey border.

Question: What are the labels in Bootstrap?

Answer: The Labels in Bootstrap are used for offering tips, counts, and other information that could help in marking up the web page.

Question: What is the benefit of Jumbotron in Bootstrap?

Answer: Jumbotron in Bootstrap can be used for increasing the size of heading and increase margins on the landing page.

Question: What is normalized in Bootstrap?

Answer: Normalized in Bootstrap is a small CSS file. It is used to create cross-browser consistency.

Question: Define panels in Bootstrap?

Answer: Panels in Bootstrap are different components that can be put into the DOM component in a box with the purpose of retrieving basic panel using the class.panel as the element.

Question: What is a grid system in Bootstrap?

Answer: A grid system in Bootstrap helps in making up to 12 columns across a webpage. This way, it helps in placing the contents of the webpage in different columns, which can be used as headings or subheadings.

Question: Are there grid classes in Bootstrap?

Answer: Yes, there are four grid classes in the Bootstrap. These include xs, sm, md, and lg.

Question: What is the function of the xs grid class?

Answer: The xs grid class is used for screens of mobile phones, which are lesser than 786 px wide.

Question: What is the function of the sm grid class?

Answer: The sm grid class is used for the wide tablet screens, which are more than 786 px wide.

Question: What is the function of the md grid class?

Answer: The md grid class is used for small laptops screen, which is either equal or greater than 992 px wide.

Question: What is the function of the lg grid class?

Answer: The lg grid class is used for larger screens of laptops and desktops, which are equal or even greater than 1200 px wide.

Question: What is a global style? Are they used in Bootstrap?

Answer: Global style stands for the standardized version of font size, line height, and other features that are used and acceptable on the international platform. They are used in Bootstrap for Bootstrap Default Typography.

Question: What is the global style for Bootstrap Default Typography?

Answer: The global style for Bootstrap Default Typography includes the following default:

1. The font size of 14 px
2. The line-height of 1.428.
3. The font style of Helvetica and Arial with sans-serif fallback

All the above-stated styles, size, and height are required to be used in the body as well as in paragraphs.

Question: Is there any pre-requisite for run Bootstrap properly?

Answer: Yes, there is a pre-requisite of JQuery, which is required to run the Bootstrap properly. It acts as the only JavaScript plugins in Bootstrap.

Question: Is there are a difference between Foundation and Bootstrap?

Answer: Yes, there is a difference between Foundation and Bootstrap. The Foundation supports the SASS processors and is mostly used for mobile UI designing purposes. On the other hand, Bootstrap supports the designing of both mobiles as well as desktop web portals using lesser preprocessors.

Question: Are you aware of the transition plugin in Bootstrap? Why is it used?

Answer: Yes, I am aware of the transition plugin in Boostrap. This is used to provide simple transition effects. These include fading or even sliding in modals.

Question: What is an Affix plugin, and how it helps Bootstrap?

Answer: Affix plugin is used as a social icon on a web page. It allows the to get attached to a particular location of the page. It will start at a location, but when the page hits, it moves away and get a block with the in place. This way, it helps in scrolling the rest of the webpage.

Question: What is the scrollspy plugin in Bootstrap?

Answer: Scrollspy plugin in Bootstrap supports auto-updating nav plugin options to apply. This helps the fetching section of the web page using a scroll position. It can be done using the .active class, which can be added in the navbar based scroll position.

Question: Can we create a Bootstrap panel using heading?

Answer: Yes, we can create a Bootstrap panel using a heading. It can be done in two ways.

- **First way:** We can make use of .panel-heading class with the purpose of adding heading containers in the panel.
- **Second way:** We can use heading tag for instance <h1>,<h2>,<h3>,<h4>,<h5> and <h6>with the .panel-title class. This will help in giving more styles to the headings as per our choice.

Question: What is a Jumbotron in Bootstrap?

Answer: A Jumbotron in Bootstrap is used for increasing the size of the heading and applying margins for the landing page contents. It can be created using the container in the with the class of .jumbotron.

Question: Define lead body copy in Bootstrap?

Answer: A lead body copy in Bootstrap is used for the addition of various ascents to the paragraph. It can be applied using the class="lead" and will help in enlarging the font size as well as increasing height.

Question: Do you know how a navbar works in the Bootstrap?

Answer: Yes, Navbar works with a navigation header for the application or the website. It acts as a vital feature that helps in creating a highly responsive meta component in web applications. In addition to that, Navbar never collapses in the mobile view and could become horizontal as well as vertical as the available viewport width increases due to the tilting of the mobile phones.

JavaScript Interview Questions for Freshers

Here are some basic JavaScript interview questions and answers for you to prepare during your interviews.

1. What do you understand about JavaScript?

Fig: JavaScript Logo

JavaScript is a popular web scripting language and is used for client-side and server-side development. The JavaScript code can be inserted into HTML pages that can be understood and executed by web browsers while also supporting object-oriented programming abilities.

2. What’s the difference between JavaScript and Java?

JavaScript	Java

JavaScript is an object-oriented scripting language.	Java is an object-oriented programming language.
JavaScript applications are meant to run inside a web browser.	Java applications are generally made for use in operating systems and virtual machines.
JavaScript does not need compilation before running the application code.	Java source code needs a compiler before it can be ready to run in realtime.

3. What are the various data types that exist in JavaScript?

These are the different types of data that JavaScript supports:

- Boolean - For true and false values
- Null - For empty or unknown values
- Undefined - For variables that are only declared and not defined or initialized
- Number - For integer and floating-point numbers
- String - For characters and alphanumeric values
- Object - For collections or complex values
- Symbols - For unique identifiers for objects

4. What are the features of JavaScript?

These are the features of JavaScript:

- Lightweight, interpreted programming language
- Cross-platform compatible
- Open-source
- Object-oriented
- Integration with other backend and frontend technologies
- Used especially for the development of network-based applications

5. What are the advantages of JavaScript over other web technologies?

These are the advantages of JavaScript:

Enhanced Interaction

JavaScript adds interaction to otherwise static web pages and makes them react to users' inputs.

Quick Feedback

There is no need for a web page to reload when running JavaScript. For example, form input validation.

Rich User Interface

JavaScript helps in making the UI of web applications look and feel much better.

Frameworks

JavaScript has countless frameworks and libraries that are extensively used for developing web applications and games of all kinds.

6. How do you create an object in JavaScript?

Since JavaScript is essentially an object-oriented scripting language, it supports and encourages the usage of objects while developing web applications.

```
const student = {
    name: 'John',
    age: 17
}
```

7. How do you create an array in JavaScript?

Here is a very simple way of creating arrays in JavaScript using the array literal:

```
var a = [];
var b = [‘a’, ‘b’, ‘c’, ‘d’, ‘e’];
```

8. What are some of the built-in methods in JavaScript?

Built-in Method	Values
Date()	Returns the present date and time

concat()	Joins two strings and returns the new string
push()	Adds an item to an array
pop()	Removes and also returns the last element of an array
round()	Rounds of the value to the nearest integer and then returns it
length()	Returns the length of a string

9. What are the scopes of a variable in JavaScript?

The scope of a variable implies where the variable has been declared or defined in a JavaScript program. There are two scopes of a variable:

Global Scope

Global variables, having global scope are available everywhere in a JavaScript code.

Local Scope

Local variables are accessible only within a function in which they are defined.

10. What is the ‘this’ keyword in JavaScript?

The ‘this’ keyword in JavaScript refers to the currently calling object. It is commonly used in constructors to assign values to object properties.

11. What are the conventions of naming a variable in JavaScript?

Following are the naming conventions for a variable in JavaScript:

- Variable names cannot be similar to that of reserved keywords. For example, var, let, const, etc.
- Variable names cannot begin with a numeric value. They must only begin with a letter or an underscore character.
- Variable names are case-sensitive.

12. What is Callback in JavaScript?

In JavaScript, functions are objects and therefore, functions can take other functions as arguments and can also be returned by other functions.

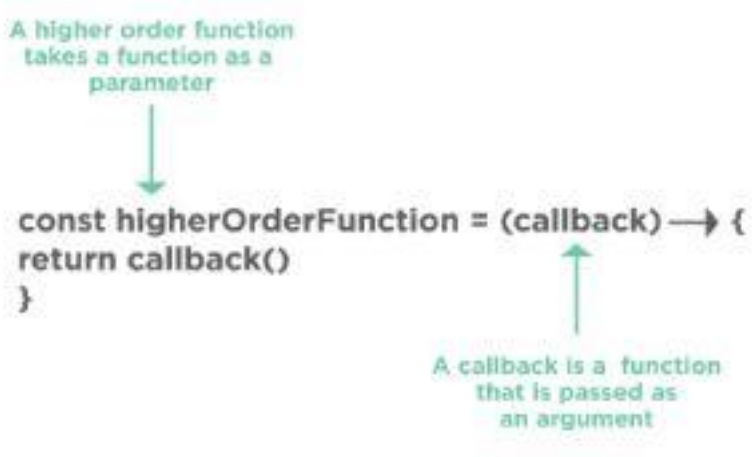


Fig: Callback function

A callback is a JavaScript function that is passed to another function as an argument or a parameter. This function is to be executed whenever the function that it is passed to gets executed.

13. How do you debug a JavaScript code?

All modern web browsers like Chrome, Firefox, etc. have an inbuilt debugger that can be accessed anytime by pressing the relevant key, usually the F12 key. There are several features available to users in the debugging tools.

We can also debug a JavaScript code inside a code editor that we use to develop a JavaScript application—for example, Visual Studio Code, Atom, Sublime Text, etc.

14. What is the difference between Function declaration and Function expression?

Function declaration	Function expression
Declared as a separate statement within the main JavaScript code	Created inside an expression or some other construct
Can be called before the function is defined	Created when the execution point reaches it; can be used only after that

Offers better code readability and better code organization	Used when there is a need for a conditional declaration of a function

Example:
 	
function abc() {
 	
return 5;
 	
}
	Example:
	
	var a = function abc() {
	
	return 5;
	
	}

15. What are the ways of adding JavaScript code in an HTML file?

There are primarily two ways of embedding JavaScript code:

- We can write JavaScript code within the script tag in the same HTML file; this is suitable when we need just a few lines of scripting within a web page.
- We can import a JavaScript source file into an HTML document; this adds all scripting capabilities to a web page without cluttering the code.

Intermediate JavaScript Interview Questions and Answers

Here are some intermediate level JavaScript interview questions and answers for you to prepare during your interviews.

16. What do you understand about cookies?

Fig: Browser cookies

A cookie is generally a small data that is sent from a website and stored on the user's machine by a web browser that was used to access the website. Cookies are used to remember information for later use and also to record the browsing activity on a website.

17. How would you create a cookie?

The simplest way of creating a cookie using JavaScript is as below:

document.cookie = "key1 = value1; key2 = value2; expires = date";

18. How would you read a cookie?

Reading a cookie using JavaScript is also very simple. We can use the document.cookie string that contains the cookies that we just created using that string.

The document.cookie string keeps a list of name-value pairs separated by semicolons, where 'name' is the name of the cookie, and 'value' is its value. We can also use the split() method to break the cookie value into keys and values.

19. How would you delete a cookie?

To delete a cookie, we can just set an expiration date and time. Specifying the correct path of the cookie that we want to delete is a good practice since some browsers won't allow the deletion of cookies unless there is a clear path that tells which cookie to delete from the user's machine.

```
function delete_cookie(name) {

  document.cookie = name + "=; Path=/; Expires=Thu, 01 Jan 1970 00:00:01 GMT;";

}
```

20. What's the difference between let and var?

Both let and var are used for variable and method declarations in JavaScript. So there isn't much of a difference between these two besides that while var keyword is scoped by function, the let keyword is scoped by a block.

21. What are Closures in JavaScript?

Closures provide a better, and concise way of writing JavaScript code for the developers and programmers. Closures are created whenever a variable that is defined outside the current scope is accessed within the current scope.

```
function hello(name) {

  var message = "hello " + name;

  return function hello() {

    console.log(message);

  };

}
```

```
//generate closure

var helloWorld = hello("World");

//use closure

helloWorld();
```

22. What are the arrow functions in JavaScript?

Arrow functions are a short and concise way of writing functions in JavaScript. The general syntax of an arrow function is as below:

```
const helloWorld = () => {

  console.log("hello world!");

};
```

23. What are the different ways an HTML element can be accessed in a JavaScript code?

Here are the ways an HTML element can be accessed in a JavaScript code:

- getElementByClass('classname'): Gets all the HTML elements that have the specified classname.
- getElementById('idname'): Gets an HTML element by its ID name.
- getElementbyTagName('tagname'): Gets all the HTML elements that have the specified tagname.
- querySelector(): Takes CSS style selector and returns the first selected HTML element.

24. What are the ways of defining a variable in JavaScript?

There are three ways of defining a variable in JavaScript:

Var

This is used to declare a variable and the value can be changed at a later time within the JavaScript code.

Const

We can also use this to declare/define a variable but the value, as the name implies, is constant throughout the JavaScript program and cannot be modified at a later time.

Let

This mostly implies that the values can be changed at a later time within the JavaScript code.

25. What are Imports and Exports in JavaScript?

Imports and exports help in writing modular code for our JavaScript applications. With the help of imports and exports, we can split a JavaScript code into multiple files in a project. This greatly simplifies the application source code and encourages code readability.

calc.js

```
export const sqrt = Math.sqrt;

export function square(x) {

  return x * x;

}

export function diag(x, y) {

  return sqrt(square(x) + square(y));

}
```

This file exports two functions that calculate the squares and diagonal of the input respectively.

main.js

```
import { square, diag } from "calc";

console.log(square(4)); // 16

console.log(diag(4, 3)); // 5
```

Therefore, here we import those functions and pass input to those functions to calculate square and diagonal.

26. What is the difference between Document and Window in JavaScript?

Document	Window
The document comes under the windows object and can also be considered as its property.	Window in JavaScript is a global object that holds the structure like variables, functions, location, history, etc.

27. What are some of the JavaScript frameworks and their uses?

JavaScript has a collection of many frameworks that aim towards fulfilling the different aspects of the web application development process. Some of the prominent frameworks are:

- React - Frontend development of a web application
- Angular - Frontend development of a web application
- Node - Backend or server-side development of a web application

28. What is the difference between Undefined and Undeclared in JavaScript?

Undefined	Undeclared
Undefined means a variable has been declared but a value has not yet been assigned to that variable.	Variables that are not declared or that do not exist in a program or application.

29. What is the difference between Undefined and Null in JavaScript?

Undefined	Null
Undefined means a variable has been declared but a value has not yet been assigned to that variable.	Null is an assignment value that we can assign to any variable that is meant to contain no value.

30. What is the difference between Session storage and Local storage?

Session storage	Local storage
The data stored in session storage gets expired or deleted when a page session ends.	Websites store some data in local machine to reduce loading time; this data does not get deleted at the end of a browsing session.

31. What are the various data types that exist in JavaScript?

Javascript consists of two data types, primitive data types, and non-primitive data types.

- Primitive Data types: These data types are used to store a single value. Following are the sub-data types in the Primitive data type.
- Boolean Data Types: It stores true and false values.

Example:

```
var a = 3;

var b =  4;

var c = 3;

(a == b) // returns false

(a == c) //returns true
```

- Null data Types: It stores either empty or unknown values.

Example:

var z = null;

- Undefined data Types: It stores variables that are only declared, but not defined or initialized.

Example:

var a; // a is undefined

var b = undefined; // we can also set the value of b variable as undefined

- Number Data Types: It stores integer as well as floating-point numbers.

Example:

var x = 4; //without decimal

var y = 5.6; //with decimal

- String data Types: It stores characters and alphanumeric values.

Example:

var str = "Raja Ram Mohan"; //using double quotes

var str2 = 'Raja Rani'; //using single quotes

- Symbols Data Types: It stores unique identifiers for objects.

Example:

var symbol1 = Symbol('symbol');

- BigInt Data Types: It stores the Number data types that are large integers and are above the limitations of number data types.

Example:

```
var bigInteger =  2345678901234567890123456789012345678901234567890;
```

- Non-Primitive Data Types

Non-Primitive data types are used to store multiple as well as complex values.

Example:

```
// Collection of data in key-value pairs

var obj1 = {

   x:  43,

   y:  "Hello world!",

   z: function(){

      return this.x;

   }

}

// Data collection with an ordered list

var array1 = [5, "Hello", true, 4.1];
```

32. What is the ‘this’ keyword in JavaScript?

The Keyword ‘this’ in JavaScript is used to call the current object as a constructor to assign values to object properties.

33. What is the difference between Call and Apply? (explain in detail with examples)

- Call

Call uses arguments separately.

Example:

```
function sayHello()

{

  return "Hello " + this.name;

}

var obj = {name: "Sandy"};

sayHello.call(obj);

// Returns "Hello Sandy"
```

- Apply

Apply uses an argument as an array.

Example:

```
function saySomething(message)

{

  return this.name + " is " + message;

}
```

```
var person4 = {name:  "John"};

saySomething.apply(person4, ["awesome"]);
```

34. What are the scopes of a variable in JavaScript?

The scope of variables in JavaScript is used to determine the accessibility of variables and functions at various parts of one's code. There are three types of scopes of a variable, global scope, function scope, block scope

- Global Scope: It is used to access the variables and functions from anywhere inside the code.

Example:

```
var globalVariable = "Hello world";

function sendMessage(){

  return globalVariable; // globalVariable is accessible as it's written in global space

}

function sendMessage2(){

  return sendMessage(); // sendMessage function is accessible as it's written in global space

}

sendMessage2();  // Returns "Hello world"
```

- Function scope: It is used to declare the function and variables inside the function itself and not outside.

Example:

```
function awesomeFunction()
```

```
{
  var a = 3;
  var multiplyBy3 = function()
{
    console.log(a*3); // Can access variable "a" since a and multiplyBy3 both are written inside the same function
  }
}
console.log(a); // a is written in local scope and can't be accessed outside and throws a reference error
multiplyBy3(); // MultiplyBy3 is written in local scope thus it throws a reference error
```

- Block Scope: It uses let and const to declare the variables.

Example:

```
{
  let x = 45;
}
console.log(x); // Gives reference error since x cannot be accessed outside of the block
for(let i=0; i<2; i++){
  // do something
```

```
}

console.log(i); // Gives reference error since i cannot be accessed outside of the for loop block
```

35. What are the arrow functions in JavaScript?

Arrow functions are used to write functions with a short and concise syntax. Also, it does not require a function keyword for declaration. An arrow function can be omitted with curly braces { } when we have one line of code.

syntax of an arrow function:

```
const helloWorld = () => {

  console.log("hello world!");

};
```

Example:

```
// Traditional Function Expression

var add = function(a,b)

{

  return a + b;

}

// Arrow Function Expression

var arrowAdd = (a,b) => a + b;
```

36. Explain Hoisting in javascript. (with examples)

Hoisting in javascript is the default process behavior of moving declaration of all the variables and functions on top of the scope where scope can be either local or global.

Example 1:

```
hoistedFunction();  // " Hi There! " is an output that is declared as function even after it is called

function hoistedFunction(){

  console.log(" Hi There! ");

}
```

Example 2:

```
hoistedVariable = 5;

console.log(hoistedVariable); // outputs 5 though the variable is declared after it is initialized

var hoistedVariable;
```

37. Difference between “ == “ and “ === “ operators (with examples)

1. “==” operator is a comparison operator that used to compare the values
2. “===” operator is also a comparison operator that is used to compare the values as well as types.

Example:

```
var x = 3;

var y = "3";
```

(x == y) // it returns true as the value of both x and y is the same

(x === y) // it returns false as the typeof x is "number" and typeof y is "string"

38. Difference between var and let keyword

- Keyword “var”
- In JavaScript programming, the “var” keyword has been used from the very initial stages of JavaScript.
- We can perform functions with the help of the keyword “var’ by accessing various variables.
- Keyword “let”
- The Keyword “let” was added later in ECMAScript 2015 in JavaScript Programming.
- Variable declaration is very limited with the help of the “let” keyword that is declared in Block. Also, it might result in a ReferenceError as the variable was declared in the “temporal dead zone” at the beginning of the block.

39. Implicit Type Coercion in javascript (in detail with examples)

When the value of one data type is automatically converted into another data type, it is called Implicit type coercion in javascript.

- String coercion

Example:

var x = 4;

var y = "4";

x + y // Returns "44"

- Boolean coercion

Example:

```
var a = 0;

var b = 32;

if(a) { console.log(a) }   // This code will run inside the block as the value of x is 0(Falsy)

if(b) { console.log(b) }    // This code will run inside the block as the value of y is 32 (Truthy)
```

40. Is javascript a statically typed or a dynamically typed language?

Yes, JavaScript is a dynamically typed language and not statically.

41. NaN property in JavaScript

NaN property in JavaScript is the “Not-a-Number” value that is not a legal number.

42. Passed by value and passed by reference

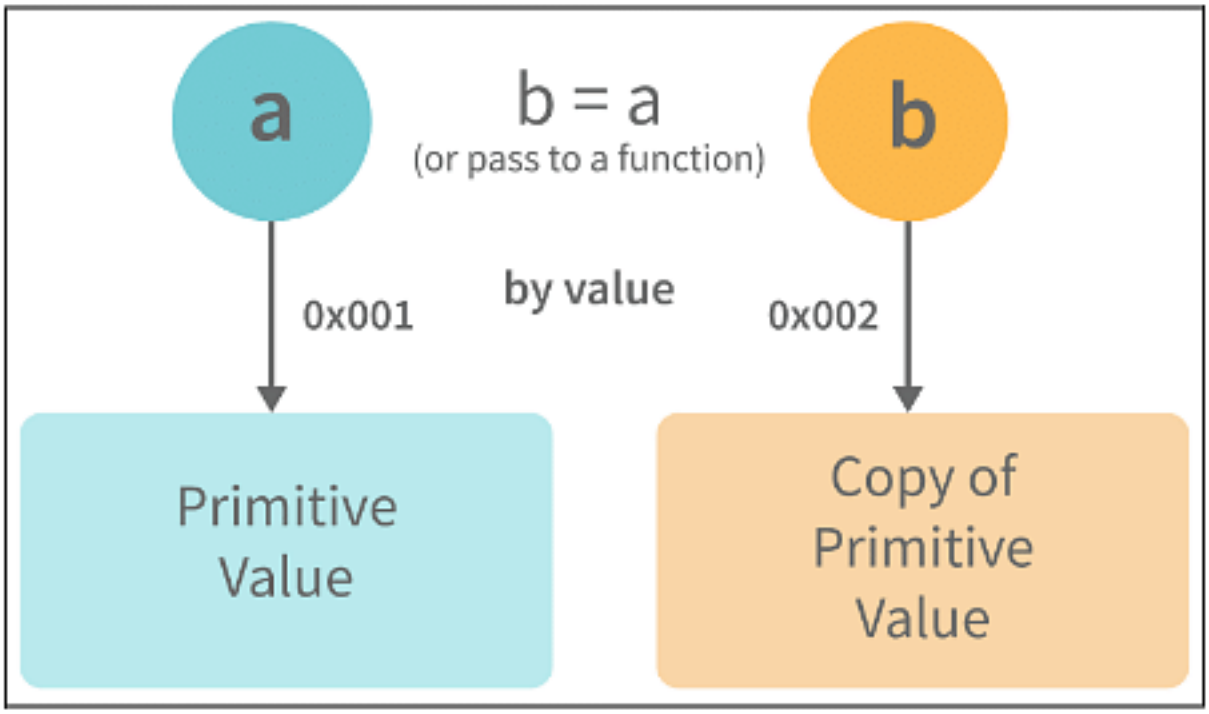
- Passed By Values Are Primitive Data Types.

Consider the following example:

Here, the a=432 is a primitive data type i.e. a number type that has an assigned value by the operator. When the var b=a code gets executed, the value of ‘var a’ returns a new address for ‘var b’ by allocating a new space in the memory, so that ‘var b’ will be operated at a new location.

Example:

```
var a = 432;

var b = a;
```



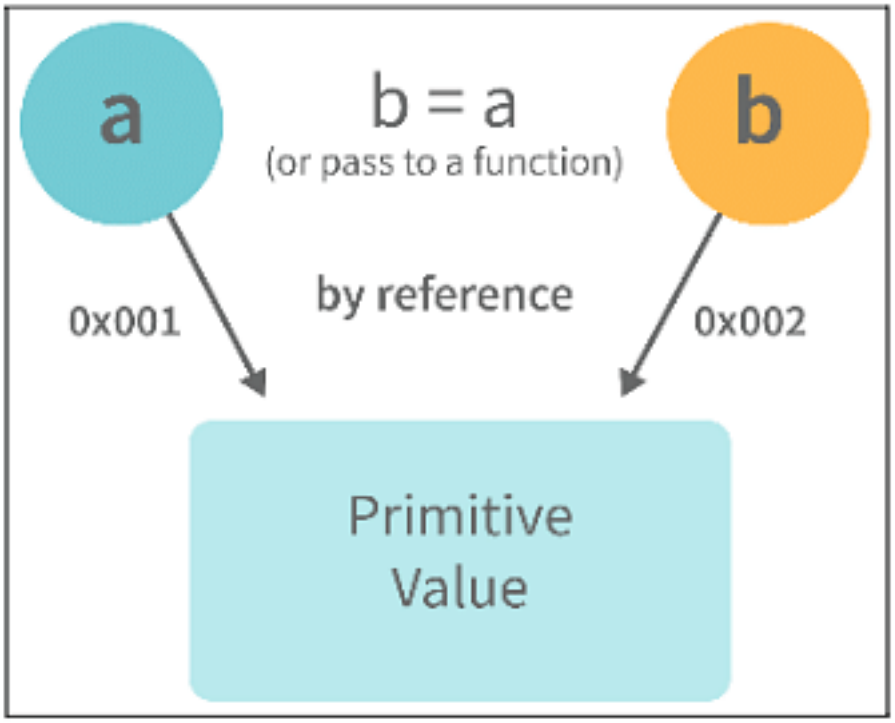
- Passed by References Are Non-primitive Data Types.

Consider the following example:

The reference of the 1st variable object i.e. ‘var obj’ is passed through the location of another variable i.e. ‘var obj2’ with the help of an assigned operator.

Example:

```
var obj = { name: "Raj", surname: "Sharma" };

var obj2 = obj;
```



43. Immediately Invoked Function in JavaScript

An Immediately Invoked Function also abbreviated as IIFE or IIFY runs as soon as it is defined. To run the function, it needs to be invoked otherwise the declaration of the function is returned.

Syntax

```
(function()

{

  // Do something;

})

();
```

44. Characteristics of javascript strict-mode

- Strict mode does not allow duplicate arguments and global variables.

- One cannot use JavaScript keywords as a parameter or function name in strict mode.
- All browsers support strict mode.
- Strict mode can be defined at the start of the script with the help of the keyword ‘use strict’.

45. Higher Order Functions (with examples)

Higher-order functions are the functions that take functions as arguments and return them by operating on other functions

Example:

```
function higherOrder(fn)
 {
  fn();
}
higherOrder(function() { console.log("Hello world") });
```

46. Self Invoking Functions

Self Invoking Functions is an automatically invoked function expression followed by (), where it does not need to be requested. Nevertheless, the declaration of the function is not able to be invoked by itself.

47. difference between exec () and test () methods

- exec()
- It is an expression method in JavaScript that is used to search a string with a specific pattern.
- Once it has been found, the pattern will be returned directly, otherwise, it returns an “empty” result.

- test ()
- It is an expression method in JavaScript that is also used to search a string with a specific pattern or text.
- Once it has been found, the pattern will return the Boolean value 'true', else it returns ‘false’.

48. currying in JavaScript (with examples)

In JavaScript, when a function of an argument is transformed into functions of one or more arguments is called Currying.

Example:

```
function add (a) {
  return function(b){
    return a + b;
  }
}
add(3)(4)
```

49. Advantages of using External JavaScript

- External Javascript allows web designers and developers to collaborate on HTML and javascript files.
- It also enables you to reuse the code.
- External javascript makes Code readability simple.

50. What are object prototypes?

Following are the different object prototypes in javascript that are used to inherit particular properties and methods from the Object.prototype.

1. Date objects are used to inherit properties from the Date prototype
2. Math objects are used to inherit properties from the Math prototype
3. Array objects are used to inherit properties from the Array prototype.

51. Types of errors in javascript

Javascript has two types of errors, Syntax error, and Logical error.

52. What is memoization?

In JavaScript, when we want to cache the return value of a function concerning its parameters, it is called memoization. It is used to speed up the application especially in case of complex, time consuming functions.

53. Recursion in a programming language

Recursion is a technique in a programming language that is used to iterate over an operation whereas a function calls itself repeatedly until we get the result.

54. Use of a constructor function (with examples)

Constructor functions are used to create single objects or multiple objects with similar properties and methods.

Example:

function Person(name,age,gender)

```
{
  this.name = name;
  this.age = age;
  this.gender = gender;
}
var person1 = new Person("Vivek", 76, "male");
console.log(person1);
var person2 = new Person("Courtney", 34, "female");
console.log(person2);
```

55. Which method is used to retrieve a character from a certain index?

We can retrieve a character from a certain index with the help of charAt() function method.

56. What is BOM?

BOM is the Browser Object Model where users can interact with browsers that is a window, an initial object of the browser. The window object consists of a document, history, screen, navigator, location, and other attributes. Nevertheless, the window's function can be called directly as well as by referencing the window.

57. Difference between client-side and server-side

- Client-side JavaScript

- Client-side JavaScript is made up of fundamental language and predefined objects that perform JavaScript in a browser.
- Also, it is automatically included in the HTML pages where the browser understands the script.
- Server-side Javascript
- Server-side JavaScript is quite similar to Client-side javascript.
- Server-side JavaScript can be executed on a server.
- The server-side JavaScript is deployed once the server processing is done.

58. What is the prototype design pattern?

The Prototype design Pattern is also known as a property or prototype pattern that is used to produce different objects as well as prototypes that are replicated from a template with a specific value.

59. Differences between declaring variables using var, let and const.

var	let	const
There is a global scope as well as a function scope.	There is neither a global scope nor a function scope.	There is neither a global scope nor a function scope.
1. There is no block scope.	There is no block scope.	There is no block scope.

It can be reassigned.	cIt cannot be reassigned.	It cannot be reassigned.

Example 1: Using ‘var’ and ‘let’ variable

```
var variable1 = 31;

let variable2 = 89;

function catchValues()

{

  console.log(variable1);

  console.log(variable2);

// Both the variables are accessible from anywhere as their declaration is in the global scope

}

window.variable1; // Returns the value 31

window.variable2; // Returns undefined
```

Example 2: Using ‘const’ variable

```
const x = {name:"Vijay"};

x = {address: "Mumbai"}; // Throws an error

x.name = "Radha"; // No error is thrown

const y = 31;
```

y = 44; // Throws an error

60. Rest parameter and spread operator

- Rest Parameter(...)
- Rest parameter is used to declare the function with improved handling of parameters.
- Rest parameter syntax can be used to create functions to perform functions on the variable number of arguments.
- It also helps to convert any number of arguments into an array as well as helps in extracting some or all parts of the arguments.
- Spread Operator(...)
- In a function call, we use the spread operator.
- It's also to spread one or more arguments that are expected in a function call.
- The spread operator is used to take an array or an object and spread them.

61. Promises in JavaScript

Promises in JavaScript have four different states. They are as follows:

Pending	Fulfilled	Rejected	Settled
Pending is an initial state of promise. It is the initial state of promise where it is in the pending state	It is the state where the promise has been fulfilled that assures that the async operation is done.	It is the state where the promise is rejected and the async	It is the state where the promise is rejected or fulfilled.

that neither is fulfilled nor rejected.		operation has failed.	

Example:

```
function sumOfThreeElements(...elements)

{

  return new Promise((resolve,reject)=>{

    if(elements.length > 3 )

{

      reject("Just 3 elements or less are allowed");

    }

    else

{

      let sum = 0;

      let i = 0;

      while(i < elements.length)

{

        sum += elements[i];

        i++;
```

```
    }
    resolve("Sum has been calculated: "+sum);
  }
 })
}
```

62. Classes in JavaScript

classes are syntactic sugars for constructor functions mentioned in the ES6 version of JavaScript. Classes are not hoisted-like Functions and can’t be used before it is declared. Also, it can inherit properties and methods from other classes with the help of extended keywords. If the strict mode (‘use strict’) is not followed, an error will be shown.

63. What are generator functions?

Generator functions are declared with a special class of functions and keywords using function*. It does not execute the code, however, it returns a generator object and handles the execution.

64. What is WeakSet?

WeakSet is a collection of unique and ordered elements that contain only objects which are referenced weakly.

65. What is the use of callbacks?

- A callback function is used to send input into another function and is performed inside another function.
- It also ensures that a particular code does not run until another code has completed its execution.

66. What is a WeakMap?

Weakmap is referred to as an object having keys and values, if the object is without reference, it is collected as garbage.

67. What is Object Destructuring? (with examples)

Object destructuring is a method to extract elements from an array or an object.

Example 1: Array Destructuring

```
const arr = [1, 2, 3];

const first = arr[0];

const second = arr[1];

const third = arr[2];
```

Example 2: Object Destructuring

```
const arr = [1, 2, 3];

const [first,second,third,fourth] = arr;

console.log(first); // Outputs 1

console.log(second); // Outputs 2

console.log(third); // Outputs 3
```

68. Prototypal vs Classical Inheritance

- Prototypal Inheritance

- Prototypal inheritance allows any object to be cloned via an object linking method and it serves as a template for those other objects, whether they extend the parent object or not.
- Classical Inheritance
- Classical inheritance is a class that inherits from the other remaining classes.

69. What is a Temporal Dead Zone?

Temporal Dead Zone is a behavior that occurs with variables declared using let and const keywords before they are initialized.

70. JavaScript Design Patterns

When we build JavaScript browser applications, there might be chances to occur errors where JavaScript approaches it in a repetitive manner. This repetitive approach pattern is called JavaScript design patterns. JavaScript design patterns consist of Creational Design Pattern, Structural Design Pattern, and Behavioral Design patterns.

71. Difference between Async/Await and Generators

- Async/Await
- Async-await functions are executed sequentially one after another in an easier way.
- Async/Await function might throw an error when the value is returned.
- Generators
- Generator functions are executed with one output at a time by the generator's yield by yield.
- The 'value: X, done: Boolean' is the output result of the Generator function.

72. Primitive data types

The primitive data types are capable of displaying one value at a time. It consists of Boolean, Undefined, Null, Number, and String data types.

73. Role of deferred scripts

The Deferred scripts are used for the HTML parser to finish before executing it.

74. What is Lexical Scoping?

Lexical Scoping in JavaScript can be performed when the internal state of the JavaScript function object consists of the function's code as well as references concerning the current scope chain.

75. What is this [[[]]]?

This '[[[]]]' is a three-dimensional array.

76. Are Java and JavaScript the same?

Yes, Java and JavaScript are the same.

77. How to detect the OS of the client machine using JavaScript?

The OS on the client machine can be detected with the help of navigator.appVersion string

78. Requirement of debugging in JavaScript

- To debug the code, we can use web browsers such as Google Chrome, and Mozilla Firefox.
- We can debug in JavaScript with the help of two methods, console.log() and debugger keyword.

79. What are the pop-up boxes available in JavaScript?

Pop-up boxes available in JavaScript are Alert Box, Confirm Box, and Prompt Box.

Advanced JS Interview Questions and Answers

Here are some advanced level JavaScript interview questions and answers for you to prepare during your interviews.

80. How do you empty an array in JavaScript?

There are a few ways in which we can empty an array in JavaScript:

- By assigning array length to 0:

```
var arr = [1, 2, 3, 4];

arr.length = 0;
```

- By assigning an empty array:

```
var arr = [1, 2, 3, 4];

arr = [];
```

- By popping the elements of the array:

```
var arr = [1, 2, 3, 4];

while (arr.length > 0) {

  arr.pop();

}
```

- By using the splice array function:

```
var arr = [1, 2, 3, 4];

arr.splice(0, arr.length);
```

81. What is the difference between Event Capturing and Event Bubbling?

Event Capturing	Event Bubbling
This process starts with capturing the event of the outermost element and then propagating it to the innermost element.	This process starts with capturing the event of the innermost element and then propagating it to the outermost element.

82. What is the Strict mode in JavaScript?

Strict mode in JavaScript introduces more stringent error-checking in a JavaScript code.

- While in Strict mode, all variables have to be declared explicitly, values cannot be assigned to a read-only property, etc.
- We can enable strict mode by adding ‘use strict’ at the beginning of a JavaScript code, or within a certain segment of code.

83. What would be the output of the below JavaScript code?

```
var a = 10;

if (function abc(){})

{

a += typeof abc;
```

```
}

console.log(a);
```

The output of this JavaScript code will be 10undefined. The if condition statement in the code evaluates using eval. Hence, eval(function abc(){}) will return function abc(){}.

Inside the if statement, executing typeof abc returns undefined because the if statement code executes at run time while the statement inside the if the condition is being evaluated.

84. Can you write a JavaScript code for adding new elements in a dynamic manner?

```
<script type="text/javascript">

function addNode() {

    var newP = document.createElement("p");

    var textNode = document.createTextNode(" This is a new text node");

    newP.appendChild(textNode); document.getElementById("firstP").appendChild(newP);

}

</script>
```

85. What is the difference between Call and Apply?

Call	Apply
In the call() method, arguments are provided individually along with a ‘this’ value.	In the apply() method, arguments are provided in the form of an array along with a ‘this’ value.

86. What will be the output of the following code?

```
var Bar = Function Foo()

{

return 11;

};

typeof Foo();
```

The output would be a reference error since a function definition can only have a single reference variable as its name.

87. What will be the output of the following code?

```
var Student = {

  college: "abc",

};
```

```
var stud1 = Object.create(Student);

delete stud1.college;

console.log(stud1.company);
```

This is essentially a simple example of object-oriented programming. Therefore, the output will be ‘abc’ as we are accessing the property of the student object.

88. How do you remove duplicates from a JavaScript array?

There are two ways in which we can remove duplicates from a JavaScript array:

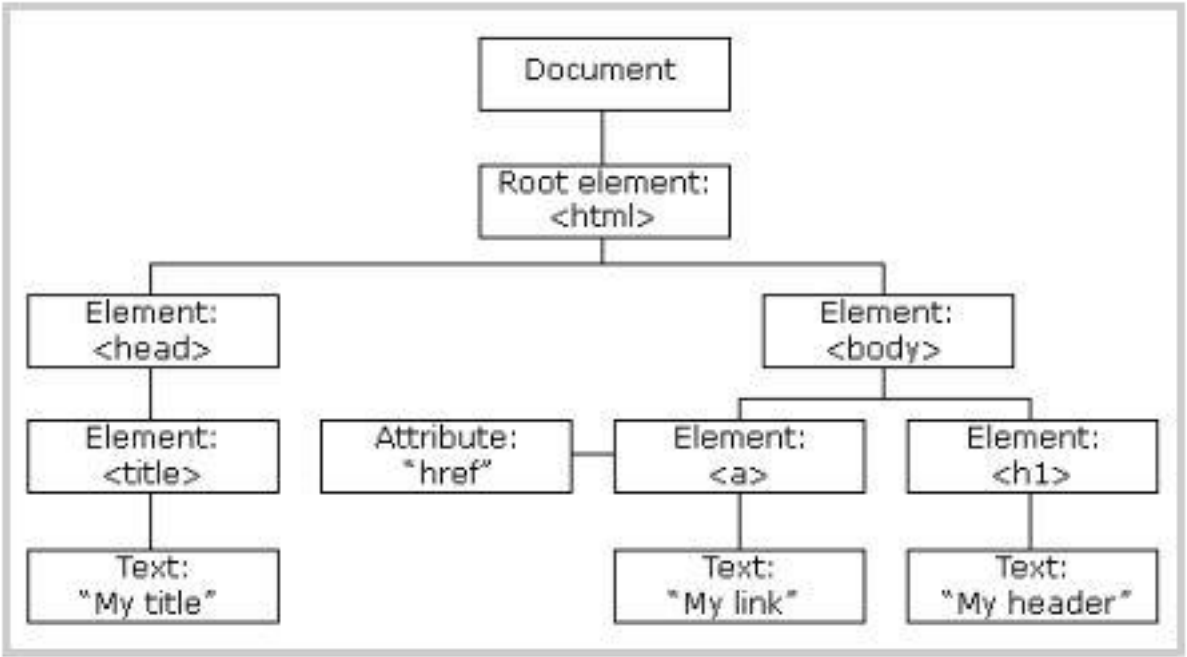
By Using the Filter Method

To call the filter() method, three arguments are required. These are namely array, current element, and index of the current element.

By Using the For Loop

An empty array is used for storing all the repeating elements.

81. Can you draw a simple JavaScript DOM (Document Object Model)?



As you prepare for your upcoming job interview, we hope that these JavaScript Interview Questions and answers have provided more insight into what types of questions you are likely to be asked.

Introduction to MongoDB

When dealing with data, there are two types of data as we know – (i) structured data and (ii) unstructured data. Structured data is usually stored in a tabular form whereas unstructured data is not. To manage huge sets of unstructured data like log or IoT data, a NoSQL database is used.

What is MongoDB ?

- MongoDB is an open-source NoSQL database written in C++ language. It uses JSON-like documents with optional schemas.
- It provides easy scalability and is a cross-platform, document-oriented database.
- MongoDB works on the concept of Collection and Document.
- It combines the ability to scale out with features such as secondary indexes, range queries, sorting, aggregations, and geospatial indexes.
- MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License (SSPL).

MongoDB Basic Interview Questions

1. What are some of the advantages of MongoDB?

Some advantages of MongoDB are as follows:

- MongoDB supports field, range-based, string pattern matching type queries. for searching the data in the database
- MongoDB support primary and secondary index on any fields
- MongoDB basically uses JavaScript objects in place of procedures
- MongoDB uses a dynamic database schema
- MongoDB is very easy to scale up or down
- MongoDB has inbuilt support for data partitioning (Sharding).

2. What is a Document in MongoDB?

A Document in MongoDB is an ordered set of keys with associated values. It is represented by a map, hash, or dictionary. In JavaScript, documents are represented as objects:

```
{"greeting" : "Hello world!"}
```

Complex documents will contain multiple key/value pairs:

```
{"greeting" : "Hello world!", "views" : 3}
```

3. What is a Collection in MongoDB?

A collection in MongoDB is a group of documents. If a document is the MongoDB analog of a row in a relational database, then a collection can be thought of as the analog to a table.

Documents within a single collection can have any number of different “shapes.”, i.e. collections have dynamic schemas.
For example, both of the following documents could be stored in a single collection:

```
{"greeting" : "Hello world!", "views": 3}
{"signoff": "Good bye"}
```

You can download a PDF version of Mongodb Interview Questions.

Download PDF

---

4. What are Databases in MongoDB?

MongoDB groups collections into databases. MongoDB can host several databases, each grouping together collections.
Some reserved database names are as follows:
admin
local
config

5. What is the Mongo Shell?

It is a JavaScript shell that allows interaction with a MongoDB instance from the command line. With that one can perform administrative functions, inspecting an instance, or exploring MongoDB.

To start the shell, run the mongo executable:

```
$ mongod
$ mongo
MongoDB shell version: 4.2.0
connecting to: test
>
```

The shell is a full-featured JavaScript interpreter, capable of running arbitrary JavaScript programs. Let’s see how basic math works on this:

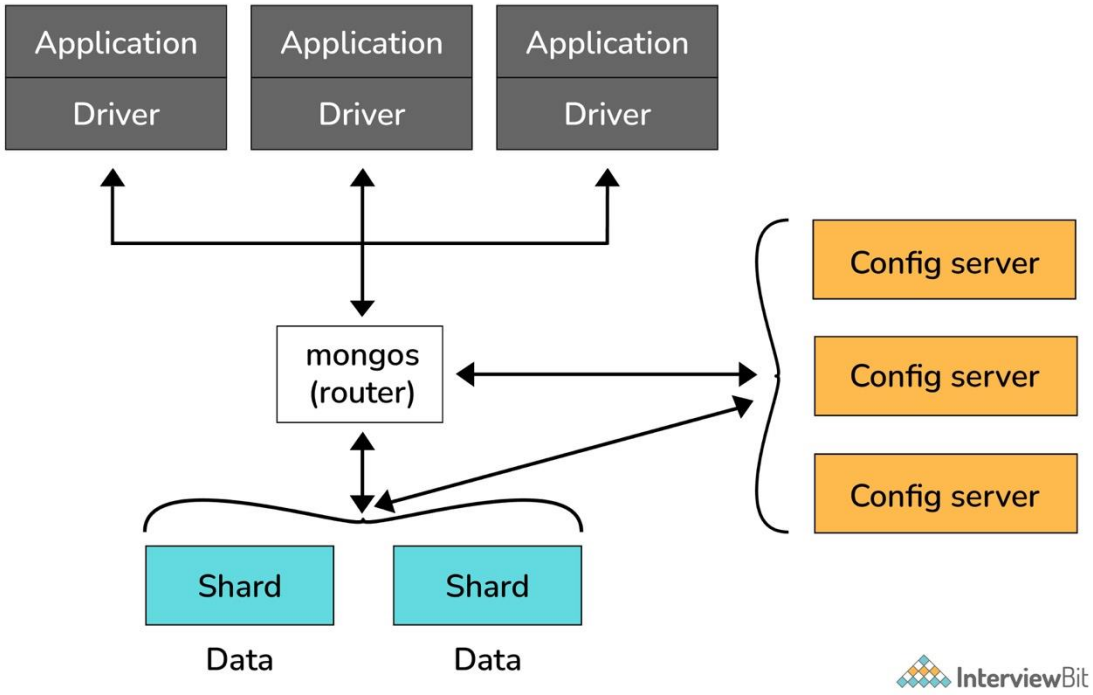
```
> x = 100;
200
> x / 5;
20
```

6. How does Scale-Out occur in MongoDB?

The document-oriented data model of MongoDB makes it easier to split data across multiple servers. Balancing and loading data across a cluster is done by MongoDB. It then redistributes documents automatically.

The mongos acts as a query router, providing an interface between client applications and the sharded cluster.

Config servers store metadata and configuration settings for the cluster. MongoDB uses the config servers to manage distributed locks. Each sharded cluster must have its own config servers.



7. What are some features of MongoDB?

- **Indexing:** It supports generic secondary indexes and provides unique, compound, geospatial, and full-text indexing capabilities as well.
- **Aggregation:** It provides an aggregation framework based on the concept of data processing pipelines.
- **Special collection and index types:** It supports time-to-live (TTL) collections for data that should expire at a certain time
- **File storage:** It supports an easy-to-use protocol for storing large files and file metadata.
- **Sharding:** Sharding is the process of splitting data up across machines.

8. How to add data in MongoDB?

The basic method for adding data to MongoDB is "inserts". To insert a single document, use the collection's `insertOne` method:

```
> db.books.insertOne({"title" : "Start With Why"})
```

For inserting multiple documents into a collection, we use `insertMany`. This method enables passing an array of documents to the database.

9. How do you Update a Document?

Once a document is stored in the database, it can be changed using one of several update methods: `updateOne`, `updateMany`, and `replaceOne`.
`updateOne` and `updateMany` each takes a filter document as their first parameter and a modifier document, which describes changes to make, as the second parameter. `replaceOne` also takes a filter as the first parameter, but as the second parameter `replaceOne` expects a document with which it will replace the document matching the filter.

For example, in order to replace a document:

```
{
   "_id" : ObjectId("4b2b9f67a1f631733d917a7a"),
   "name" : "alice",
   "friends" : 24,
   "enemies" : 2
}
```

10. How do you Delete a Document?

The CRUD API in MongoDB provides `deleteOne` and `deleteMany` for this purpose. Both of these methods take a filter document as their first parameter. The filter specifies a set of criteria to match against in removing documents.

For example:

```
> db.books.deleteOne({"_id" : 3})
```

11. How to perform queries in MongoDB?

The `find` method is used to perform queries in MongoDB. Querying returns a subset of documents in a collection, from no documents at all to the entire collection. Which documents get returned is determined by the first argument to `find`, which is a document specifying the query criteria.

Example:

```
> db.users.find({"age" : 24})
```

12. What are the data types in MongoDB?

MongoDB supports a wide range of data types as values in documents. Documents in MongoDB are similar to objects in JavaScript. Along with JSON's essential key/value–pair nature, MongoDB adds support for a number of additional data types. The common data types in MongoDB are:

- Null
  `{"x" : null}`
- Boolean
  `{"x" : true}`
- Number
  `{"x" : 4}`
- String
  `{"x" : "foobar"}`
- Date
  `{"x" : new Date()}`
- Regular expression
  `{"x" : /foobar/i}`
- Array
  `{"x" : ["a", "b", "c"]}`
- Embedded document
  `{"x" : {"foo" : "bar"}}`
- Object ID
  `{"x" : ObjectId()}`
- Binary Data
  Binary data is a string of arbitrary bytes.
- Code
  `{"x" : function() { /* ... */ }}`

13. When to use MongoDB?

You should use MongoDB when you are building internet and business applications that need to evolve quickly and scale elegantly. MongoDB is popular with developers of all kinds who are building scalable applications using agile methodologies.
MongoDB is a great choice if one needs to:

- Support a rapid iterative development.
- Scale to high levels of read and write traffic - MongoDB supports horizontal scaling through Sharding, distributing data across several machines, and facilitating high throughput operations with large sets of data.
- Scale your data repository to a massive size.
- Evolve the type of deployment as the business changes.
- Store, manage and search data with text, geospatial, or time-series dimensions.

MongoDB Intermediate Interview Questions

14. How is Querying done in MongoDB?

The `find` method is used to perform queries in MongoDB. Querying returns a subset of documents in a collection, from no documents at all to the entire collection. Which documents get returned is determined by the first argument to find, which is a document specifying the query criteria.

For example: If we have a string we want to match, such as a "username" key with the value "alice", we use that key/value pair instead:

```
> db.users.find({"username" : "alice"})
```

15. Explain the term “Indexing” in MongoDB.

In MongoDB, indexes help in efficiently resolving queries. What an Index does is that it stores a small part of the data set in a form that is easy to traverse. The index stores the value of the specific field or set of fields, ordered by the value of the field as specified in the index.
MongoDB’s indexes work almost identically to typical relational database indexes.

Indexes look at an ordered list with references to the content. These in turn allow MongoDB to query orders of magnitude faster. To create an index, use the `createIndex` collection method.

For example:

```
> db.users.find({"username": "user101"}).explain("executionStats")
```

Here, `executionStats` mode helps us understand the effect of using an index to satisfy queries.

16. What are Geospatial Indexes in MongoDB?

MongoDB has two types of geospatial indexes: 2dsphere and 2d. 2dsphere indexes work with spherical geometries that model the surface of the earth based on the WGS84 datum. This datum model the surface of the earth as an oblate spheroid, meaning that there is some flattening at the poles. Distance calculations using 2sphere indexes, therefore, take the shape of the earth into account and provide a more accurate treatment of distance between, for example, two cities, than do 2d indexes. Use 2d indexes for points stored on a two-dimensional plane.

2dsphere allows you to specify geometries for points, lines, and polygons in the GeoJSON format. A point is given by a two-element array, representing [longitude, latitude]:

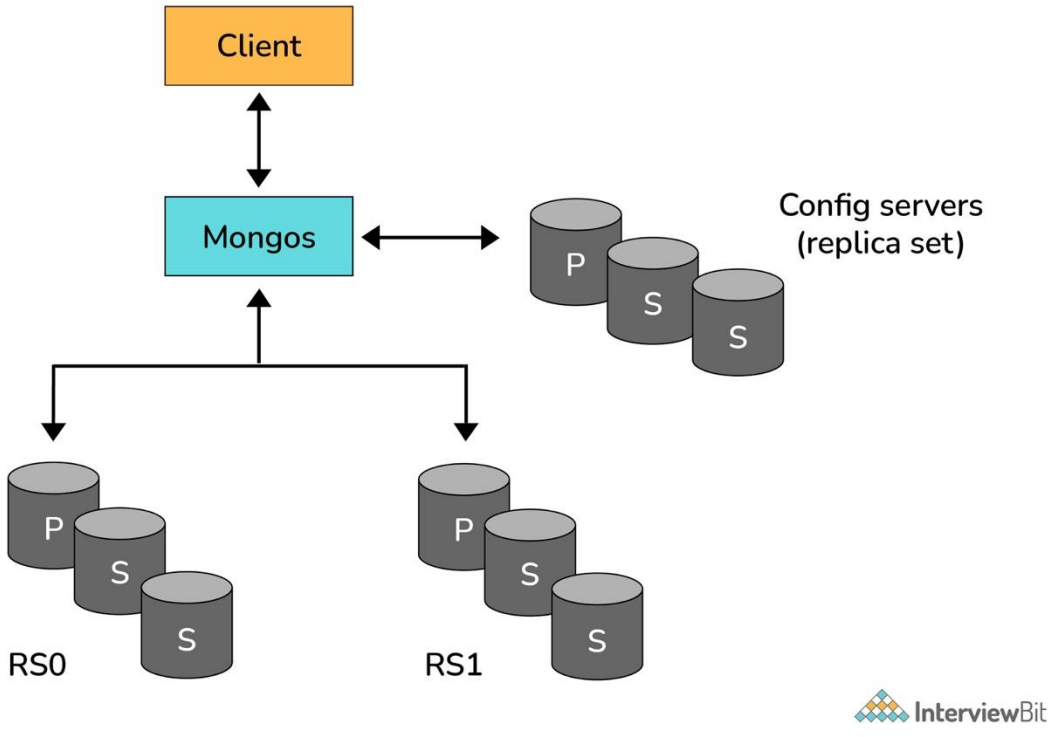
```
{
   "name" : "New York City",
   "loc" : {
       "type" : "Point",
       "coordinates" : [50, 2]
   }
}
```

A line is given by an array of points:

```
{
   "name" : "Hudson River",
   "loc" : {
       "type" : "LineString",
       "coordinates" : [[0,1], [0,2], [1,2]]
   }
}
```

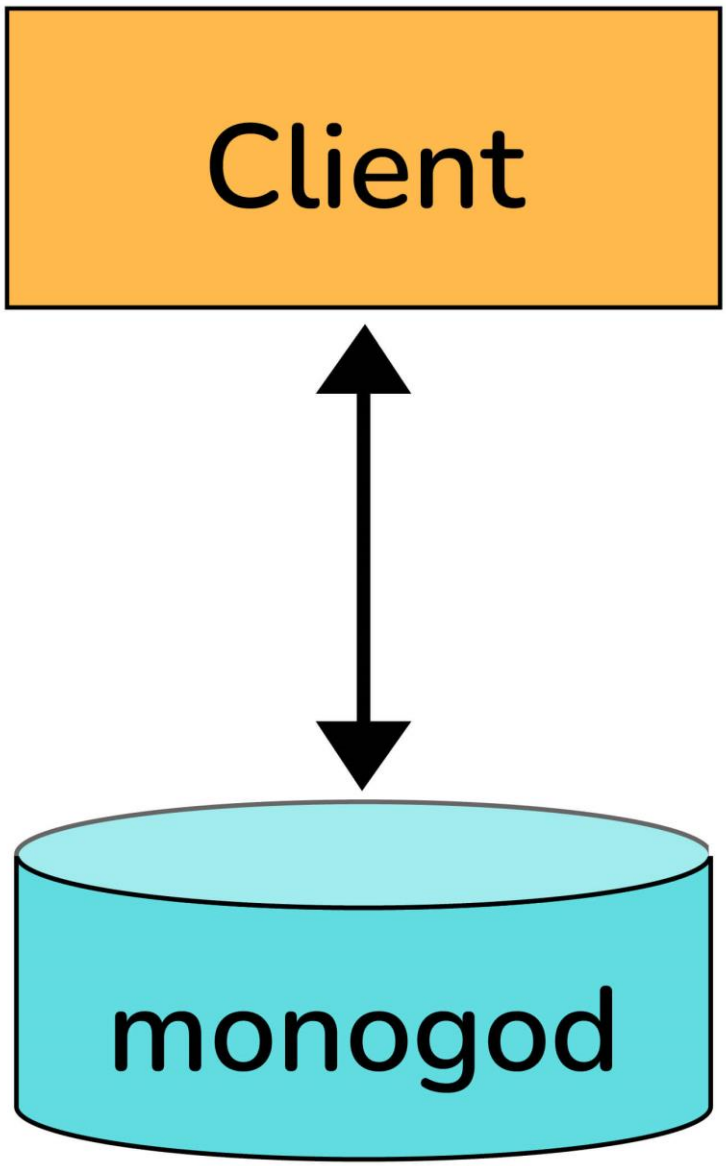
17. Explain the process of Sharding.

Sharding is the process of splitting data up across machines. We also use the term "partitioning" sometimes to describe this concept. We can store more data and handle more load without requiring larger or more powerful machines, by putting a subset of data on each machine.
In the figure below, RS0 and RS1 are shards. MongoDB's sharding allows you to create a cluster of many machines (shards) and break up a collection across them, putting a subset of data on each shard. This allows your application to grow beyond the resource limits of a standalone server or replica set.



Sharded Client

Connection



Non Sharded Client Connection

18. Explain the SET Modifier in MongoDB?

If the value of a field does not yet exist, the "$set" sets the value. This can be useful for updating schemas or adding user-defined keys.

Example:

```
> db.users.findOne()
{
```

```
    "_id" : ObjectId("4b253b067525f35f94b60a31"),
    "name" : "alice",
    "age" : 23,
    "sex" : "female",
    "location" : "India"
}
```

To add a field to this, we use "$set":

```
> db.users.updateOne({"_id" :
ObjectId("4b253b067525f35f94b60a31")},
... {"$set" : {"favorite book" : "Start with Why"}})
```

MongoDB Advanced Interview Questions

19. What do you mean by Transactions?

A transaction is a logical unit of processing in a database that includes one or more database operations, which can be read or write operations. Transactions provide a useful feature in MongoDB to ensure consistency.

MongoDB provides two APIs to use transactions.

- **Core API:** It is a similar syntax to relational databases (e.g., start_transaction and commit_transaction)
- **Call-back API:** This is the recommended approach to using transactions. It starts a transaction, executes the specified operations, and commits (or aborts on the error). It also automatically incorporates error handling logic for "TransientTransactionError" and"UnknownTransactionCommitResult".

20. What are MongoDB Charts?

MongoDB Charts is a new, integrated tool in MongoDB for data visualization.

MongoDB Charts offers the best way to create visualizations using data from a MongoDB database.
It allows users to perform quick data representation from a database without writing code in a programming language such as Java or Python.

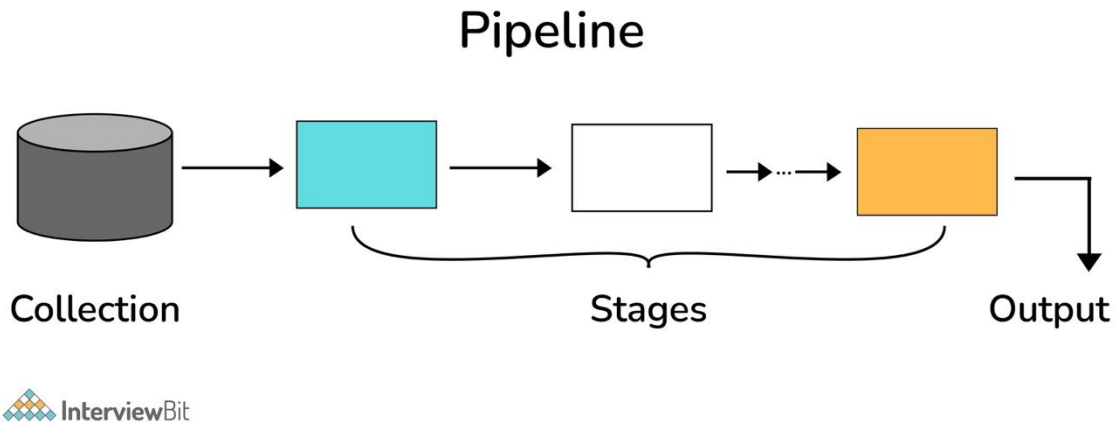
The two different implementations of MongoDB Charts are:

- MongoDB Charts PaaS (Platform as a Service)
- MongoDB Charts Server

21. What is the Aggregation Framework in MongoDB?

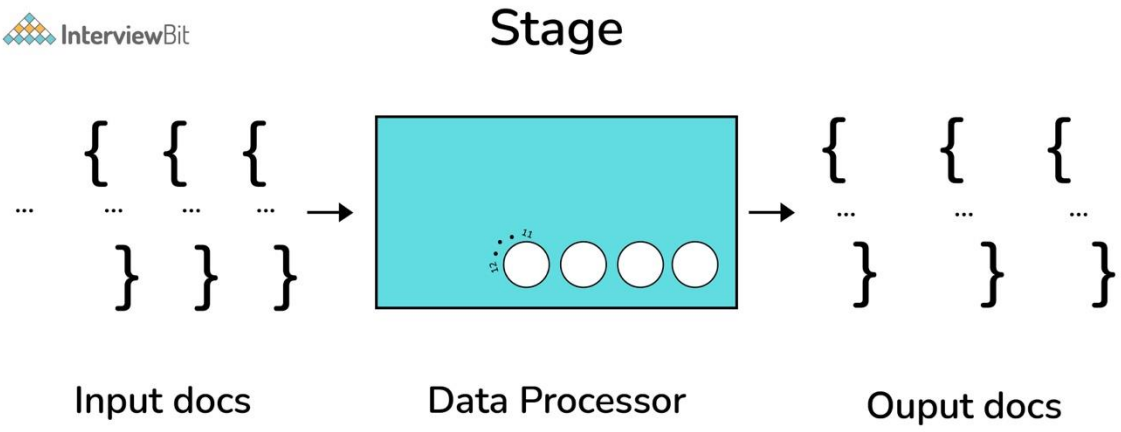
- The aggregation framework is a set of analytics tools within MongoDB that allow you to do analytics on documents in one or more collections.

- The aggregation framework is based on the concept of a pipeline. With an aggregation pipeline, we take input from a MongoDB collection and pass the documents from that collection through one or more stages, each of which performs a different operation on its inputs (See figure below). Each stage takes as input whatever the stage before it produced as output. The inputs and outputs for all stages are documents—a stream of documents.



22. Explain the concept of pipeline in the MongoDB aggregation framework.

An individual stage of an aggregation pipeline is a data processing unit. It takes in a stream of input documents one at a time, processes each document one at a time, and produces an output stream of documents one at a time (see figure below).



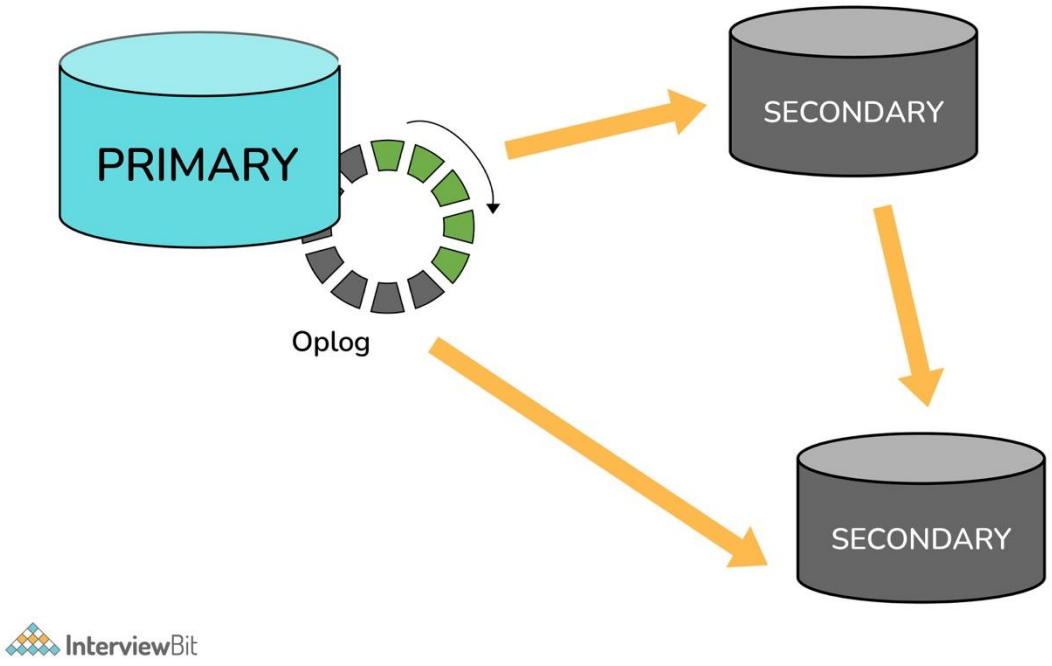
23. What is a Replica Set in MongoDB?

To keep identical copies of your data on multiple servers, we use replication. It is recommended for all production deployments. Use replication to keep your application running and your data safe, even if something happens to one or more of your servers.

Such replication can be created by a replica set with MongoDB. A replica set is a group of servers with one primary, the server taking writes, and multiple secondaries, servers that keep copies of the primary's data. If the primary crashes, the secondaries can elect a new primary from amongst themselves.

24. Explain the Replication Architecture in MongoDB.

The following diagram depicts the architecture diagram of a simple replica set cluster with only three server nodes – one primary node and two secondary nodes:



- In the preceding model, the PRIMARY database is the only active replica set member that receives write operations from database clients. The PRIMARY database saves data changes in the Oplog. Changes saved in the Oplog are sequential—that is, saved in the order that they are received and executed.
- The SECONDARY database is querying the PRIMARY database for new changes in the Oplog. If there are any changes, then Oplog entries are copied from PRIMARY to SECONDARY as soon as they are created on the PRIMARY node.
- Then, the SECONDARY database applies changes from the Oplog to its own datafiles. Oplog entries are applied in the same order they were inserted in the log. As a result, datafiles on SECONDARY are kept in sync with changes on PRIMARY.
- Usually, SECONDARY databases copy data changes directly from PRIMARY. Sometimes a SECONDARY database can replicate data from another SECONDARY. This type of replication is called Chained Replication because it is a two-step replication process. Chained replication is useful in certain replication topologies, and it is enabled by default in MongoDB.

25. What are some utilities for backup and restore in MongoDB?

The mongo shell does not include functions for exporting, importing, backup, or restore. However, MongoDB has created methods for accomplishing this, so that no scripting work or complex GUIs are needed. For this, several utility scripts are provided that can be used to get data in or out of the database in bulk. These utility scripts are:

- mongoimport
- mongoexport
- mongodump
- mongorestore

26. Conclusion

MongoDB is a powerful, flexible, and scalable general-purpose database. It combines the ability to scale out with features such as secondary indexes, range queries, sorting, aggregations, and geospatial indexes.
Thus, in conclusion, MongoDB is:

- Supports Indexing
- Designed to scale
- Rich with Features
- High Performance
- Load Balancing
- Supports sharding

Although MongoDB is powerful, incorporating many features from relational systems, it is not intended to do everything that a relational database does. For some functionality, the database server offloads processing and logic to the client-side (handled either by the drivers or by a user's application code). Its maintenance of this streamlined design is one of the reasons MongoDB can achieve such high performance.

Here are few References to understand MongoDB in-depth:

---

1. Compare MongoDB with Cassandra.

Learn about a brief comparison between MongoDB and Cassandra using this table.

Criteria	MongoDB	Cassandra
Data Model	Document	Bigtable like

Database scalability	Read	Write
Querying of data	Multi-indexed	Using Key or Scan

To have a detailed comparison between MongoDB and Cassandra, check out Cassandra Versus MongoDB!

2. What makes MongoDB the best?

MongoDB is considered to be the best NoSQL database because of its following features:

- Document-oriented (DO)
- High performance (HP)
- High availability (HA)
- Easy scalability
- Rich query language

3. How to do transactions/locking in MongoDB?

MongoDB does not use conventional locking with reduction as it is planned to be light, high-speed, and knowable in its presentation. It can be considered as parallel to the MySQL MyISAM auto entrust sculpt. With the simplest business sustain, performance is enhanced, particularly in a structure with numerous servers.

4. When and to what extent does data get extended to multi-slice?

MongoDB scrap stands on a collection. So, an album of all substances is kept in a lump or mass. Only when there is an additional time slot, there will be more than a few slice data achievement choices, but when there is more than one lump, data gets extended to a lot of slices and it can be extended to 64 MB.

5. Compare MongoDB with Couchbase and CouchbaseDB.

Although MongoDB, Couchbase, and Couchbase DB are common in many ways, still they are different in the case of necessities for the execution of the model, crossing points, storage, duplications, etc.

6. When do we use a namespace in MongoDB?

During the sequencing of the names of the database and the collection, the namespace is used.

7. If you remove an object attribute, is it deleted from the database?

Yes, it is deleted. Hence, it is better to eliminate the attribute and then save the object again.

8. How can we move an old file into the moveChunk directory?

Once the functions are done, the old files are converted to backup files and moved to the **moveChunk** directory at the time of balancing the slices.

9. Explain the situation when an index does not fit into RAM.

When an index is too huge to fit into RAM, then MongoDB reads the index, which is faster than reading RAM because the indexes easily fit into RAM if the server has got RAM for indexes, along with the remaining set.

10. How does MongoDB provide consistency?

MongoDB uses the **reader–writer locks**, allowing simultaneous readers to access any supply like a database or a collection but always offering private access to single writes.

11. Why is MongoDB not chosen for a 32-bit system?

Mongo DB is not considered as a 32-bit system because for running the 32-bit MongoDB, with the server, information and indexes require 2 GB. That is why it is not used in 32-bit devices.

12. How does Journaling work in MongoDB?

Write operations are saved in memory while journaling is going on. The on-disk journal files are really dependable for the reason that the journal writes are habitual. Inside **dbPath**, a journal subdirectory is designed by MongoDB.

13. How can you isolate the cursors from intervening with the write operations?

The **snapshot()** method is used to isolate the cursors from intervening with writes. This method negotiates the index and makes sure that each query comes to any article only once.

14. Define MongoDB.

It is a document-oriented database that is used for high availability, easy scalability, and high performance. It supports the dynamic schema design.

15. Explain the replica set.

It is a group of mongo instances that maintains the same dataset. Replica sets provide redundancy and high availability and are the basis for all production deployments.

Also, Checkout this interesting blog on Redis Vs MongoDB!

16. What are the key features of MongoDB?

There are three main features of MongoDB:

- Automatic scaling
- High performance

- High availability

17. What is CRUD?

MongoDB provides CRUD operations:

- **C**reate
- **R**ead
- **U**pdate
- **D**elete

Click here to learn more about MongoDB CRUD Operations in our MongoDB Tutorial!

18. What is Sharding?

In MongoDB, sharding means to store data on multiple machines.

19. What is Aggregation in MongoDB?

In MongoDB, aggregations are operations that process data records and return computed results.

20. Define Namespace in MongoDB.

It is the concatenation of the collection name and the name of the database.

Intermediate Interview Questions

21. Which syntax is used to create a Collection in MongoDB?

We can create a collection in MongoDB using the following syntax:

```
db.createCollection(name,options)
```

22. Which syntax is used to drop a Collection in MongoDB?

We can use the following syntax to drop a collection in MongoDB:

```
db.collection.drop()
```

23. Explain Replication.

Replication is the process of synchronizing data across multiple servers.

24. What is the use of an Index in MongoDB?

In MongoDB, indexes provide high-performance read operations for frequently used queries.

If you have any doubts or queries related to MongoDB, get them clarified from MongoDB Experts on our MongoDB Community!

25. Which command is used for inserting a document in MongoDB?

The following command is used for inserting a document in MongoDB:

```
database.collection.insert (document)
```

26. What is the use of GridFS in MongoDB?

GridFS is used for storing and retrieving large files, such as audio, image, and video files.

27. What is the use of Journaling in MongoDB?

Journaling is used for safe backups in MongoDB.

28. Which command is used to see a connection?

We can use the following command to see the connection:

```
db_adminCommand (“connPoolStats”)
```

29. Define the primary Replica set.

The primary replica set accepts all write operations from clients.

To have a detailed comparison between Firebase and MongoDB, check out Firebase Vs MongoDB!

30. Define the secondary Replica sets.

The secondaries replicate the primary replica set's oplog and apply the operations to their datasets such that the secondaries' datasets reflect the primary's dataset.

31. What is the use of Profiler?

Profiler is used to show the performance characteristics of every operation against the database.

32. What type of data is stored by MongoDB?

MongoDB stores data in the form of documents, which are JSON-like field and value pairs.

33. What is the purpose of Replication?

Replication provides redundancy, and it increases data availability.

34. What are Embedded documents?

Embedded documents capture relationships between data by storing related data in a single document structure.

35. Define the application-level Encryption.

The application-level encryption provides encryption on a per-field or per-document basis within the application layer.

36. What is Storage Encryption?

Storage encryption encrypts all MongoDB data on storage or on the operating system to ensure that only authorized processes can access the protected data.

37. Which method is used to create an index?

The createIndex() method is used to create an index.

38. What is Replica set oplog?

The oplog records all operations that modify the data in the replica set.

39. What is Vertical Scaling?

Vertical scaling adds more CPU and storage resources to increase capacity.

40. Define Horizontal Scaling.

Horizontal scaling divides the dataset and distributes data over multiple servers, or shards.

Advanced Interview Questions

41. What are the components of the Sharded cluster?

The sharded cluster has the following components:

- Shards
- Query routers
- Config servers

42. Which command is used to create a database?

To create a database, we can use the **Database_Name** command.

43. Which command is used to drop a database?

The **db.dropDatabse()** command is used to drop a database.

44. What is the use of the pretty() method?

The pretty() method is used to show the results in a formatted way.

45. Which method is used to remove a document from a collection?

The remove() method is used to remove a document from a collection.

46. Define MongoDB Projection.

Projection is used to select only the necessary data. It does not select the whole data of a document.

47. What is the use of the limit() method?

The limit() method is used to limit the records in the database.

48. What is the syntax of the limit() method?

The syntax of the limit() method is as follows:

```
>db.COLLECTION_NAME.find().limit(NUMBER)
```

49. What is the syntax of the sort() method?

In MongoDB, the following syntax is used for sorting documents:

```
>db.COLLECTION_NAME.find().sort({KEY:1})
```

50. Which command is used to create a backup of the database?

The mongodump command is used to create a backup of the database.

51. What is a Collection in MongoDB?

In MongoDB, a collection is a group of MongoDB documents.

52. What is the use of the db command?

The db command gives the name of the currently selected database.

53. Which method is used to update documents into a collection?

The update() and save() methods are used to update documents into a collection.

54. What is the syntax of the skip() method?

The syntax of the skip() methopd is as follows:

```
>db.COLLECTION_NAME.find().limit(NUMBER).skip(NUMBER)
```

55. Which command is used to restore the backup?

The mongorestore command is used to restore the backup.

56. What is the use of the dot notation in MongoDB?

MongoDB uses the dot notation to access the elements of an array and the fields of an embedded document.

57. Define Auditing.

Auditing provides administrators with the ability to verify that the implemented security policies are controlling the activity in the system.

58. Define the Aggregation pipeline.

The aggregation pipeline is a framework for performing aggregation tasks. The pipeline is used to transform documents into aggregated results.

59. Define MapReduce.

MapReduce is a generic multi-phase data aggregation modality that is used for processing quantities of data.

60. What is Splitting in MongoDB?

Splitting is a background process that is used to keep chunks from growing too large.

61. Which language is used to write for MongoDB?

C++ is used for writing and implementing MongoDB.

62. In which format does MongoDB store data?

MongoDB uses collections to store data rather than tables.

63. What is the use of the save() method?

The save() method is used to replace the existing document with a new document.

64. What is MongoDB?

MongoDB (from humongous) is a cross-platform document-oriented database. Classified as a NoSQL database, MongoDB eschews the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas (MongoDB calls the format 'BSON'), making the integration of data in certain types of applications easier and faster. Released under a combination of the GNU Affero General Public License and the Apache License, MongoDB is open-source.

MongoDB was first developed by the software company 10gen (now, MongoDB Inc.) in October 2007 as a component of a planned platform as a service product. Then, the company shifted to an open-source development model in 2009, with 10gen offering commercial support and other services. Since then, MongoDB has been adopted as backend software by a number of major websites and services, including Craigslist, eBay, Foursquare, SourceForge, Viacom,

and the New York Times, among others. Currently, MongoDB is the most popular NoSQL database system.

For a better understanding of MongoDB, refer to this What is MongoDB? blog.

65. What is the use of MongoDB?

MongoDB is a relational database management system (RDBMS) replacement for web applications. So, when we have something close to RDBMS, MongoDB could be of good use.

It gives us the additional partition tolerance, which RDMBS doesn't offer, but it has problems with availability. Nonetheless, if we want more scalability, MongoDB would be the right choice for us. It's suitable for real-time analytics and high-speed logging, and it's highly scalable as well. Craigslist uses MongoDB for archived posts.

66. What do you understand by NoSQL databases? Is MongoDB a NoSQL database? Explain.

Presently, the Internet is loaded with big data, big users, and so on that are becoming more complex day by day. NoSQL is the answer to all these problems; it is not a traditional database management system, not even a relational database management system (RDBMS).

NoSQL stands for 'Not only SQL', and it is a type of database that can handle and sort all types of unstructured, messy, and complicated data. It is just a new way to think about databases.

Yes, MongoDB is a NoSQL database.

67. What type of a DBMS is MongoDB?

MongoDB is a document-oriented DBMS.

68. What is the difference between MongoDB and MySQL?

Although both MongoDB and MySQL are free and open-source databases, there is a lot of difference between them in terms of data representation, relationships, transaction, querying

data, schema design and definition, performance speed, normalization, and many more. To compare MySQL with MongoDB is like a comparison between relational and non-relational databases.

To have a detailed comparison between Firebase and MongoDB, check out MongoDB vs SQL!

69. What is the use of MongoDB?

- MongoDB is typically used as the primary data store for operational applications with real-time requirements (i.e., low latency, high availability, etc.). MongoDB is generally a good fit for 60–80 percent of the applications we build today. MongoDB is easy to operate and scale in the ways that are hard if not impossible with relational databases.
- MongoDB excels in many use cases where the relational databases aren't a good fit, like applications with unstructured, semi-structured, and polymorphic data, as well as those with large scalability requirements or multi-datacenter deployments.
- MongoDB may not be a good fit for some applications. For example, applications that require complex transactions (e.g., a double-entry bookkeeping system) and scan-oriented applications that access large subsets of the data mostly may not be a good fit for MongoDB. Also, MongoDB is not a drop-in replacement for legacy applications built around the relational data model and SQL.
- Some common use cases of MongoDB include mobile apps, product catalogs, real-time personalization, content management, and applications delivering a single view across multiple systems.

70. What kind of a database is MongoDB?

MongoDB is a document-oriented DBMS. We can think of it as MySQL but with JSON-like objects comprising the data model, rather than RDBMS tables. Significantly, MongoDB supports neither joins nor transactions. However, it features secondary indexes, an expressive query language, atomic writes on a per-document level, and fully-consistent reads. Operationally, MongoDB offers the master–slave replication with automated failover and built-in horizontal scaling via automated range-based partitioning.

To learn more about MongoDB, check out Intellipaat's MongoDB Tutorial!

71. Which language is MongoDB written in?

MongoDB is implemented in C++. However, drivers and client libraries are typically written in their own respective languages. Although, some drivers use C extensions for better performance.

Get a detailed comparison between Dynamodb and Mongodb. Read our blog on Dynamodb Vs Mongodb

72. What are the limitations of the 32-bit versions of MongoDB?

MongoDB uses memory-mapped files. When running a 32-bit build of MongoDB, the total storage size for the server, including data and indexes, is 2 GB. For this reason, we do not deploy MongoDB to production on 32-bit machines.

If we're running a 64-bit build of MongoDB, there's virtually no limit to the storage size. For production deployments, 64-bit builds and operating systems are strongly recommended.

73. While creating a schema in MongoDB, what are the points need to be taken into consideration?

While creating a schema in MongoDB, the points need to be taken care of are as follows:

- Design our schema according to the user requirements
- Combine objects into one document if we want to use them together; otherwise, separate them
- Do joins while on write, and not when it is on read
- For most frequent use cases, optimize the schema
- Do complex aggregation in the schema

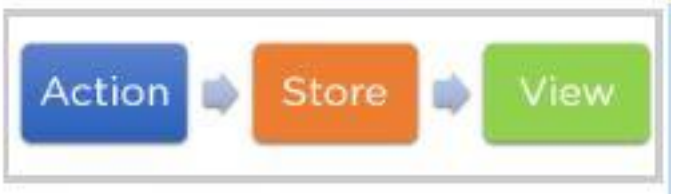
ReactJS

Basic Level - ReactJS Interview Questions

Here are some React Interview Questions on basic concepts.

1. What are the features of React?

	JSX: JSX is a syntax extension to JavaScript. It is used with React to describe what the user interface should look like. By using JSX, we can write HTML structures in the same file that contains JavaScript code.
	Components: Components are the building blocks of any React application, and a single app usually consists of multiple components. It splits the user interface into independent, reusable parts that can be processed separately.

	Virtual DOM: React keeps a lightweight representation of the real DOM in the memory, and that is known as the virtual DOM. When the state of an object changes, virtual DOM changes only that object in the real DOM, rather than updating all the objects.
	One-way data-binding: React's one-way data binding keeps everything modular and fast. A unidirectional data flow means that when designing a React app, you often nest child components within parent components.
	High performance: React updates only those components that have changed, rather than updating all the components at once. This results in much faster web applications.

2. What is JSX?

JSX is a syntax extension of JavaScript. It is used with React to describe what the user interface should look like. By using JSX, we can write HTML structures in the same file that contains JavaScript code.

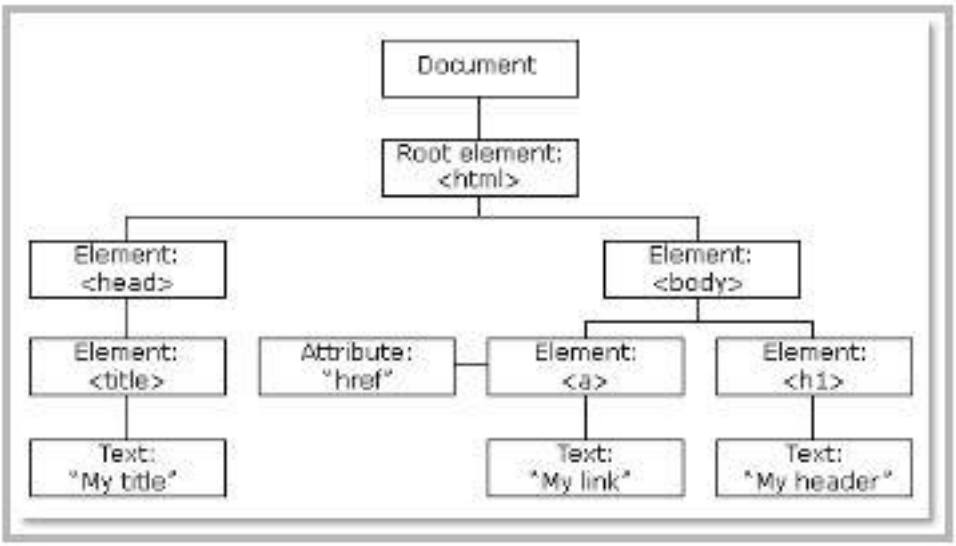
```
render() {
    return (
        <div>
            <h1> This is a JSX code</h1>
        </div>
    );
}
```

3. Can web browsers read JSX directly?

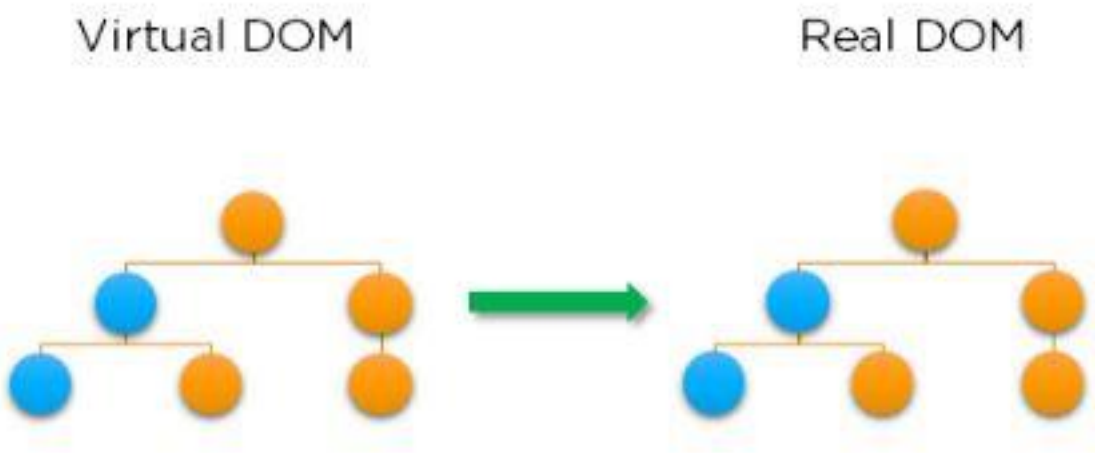
- Web browsers cannot read JSX directly. This is because they are built to only read regular JS objects and JSX is not a regular JavaScript object
- For a web browser to read a JSX file, the file needs to be transformed into a regular JavaScript object. For this, we use Babel

4. What is the virtual DOM?

DOM stands for Document Object Model. The DOM represents an HTML document with a logical tree structure. Each branch of the tree ends in a node, and each node contains objects.



React keeps a lightweight representation of the real DOM in the memory, and that is known as the virtual DOM. When the state of an object changes, the virtual DOM changes only that object in the real DOM, rather than updating all the objects. The following are some of the most frequently asked react interview questions.



5. Why use React instead of other frameworks, like Angular?

Easy creation of dynamic applications: React makes it easier to create dynamic web applications because it provides less coding and provides more functionality,

	whereas, with JavaScript applications, code tends to get complex very quickly.
	Improved performance: React uses virtual DOM, which makes web applications perform faster. Virtual DOM compares its previous state and updates only those components in the real DOM, whose states have changed, rather than updating all the components — like conventional web applications.
	Reusable components: Components are the building blocks of any React application, and a single app usually consists of multiple components. These components have their own logic and controls, and they can be reused through the application, which, in turn, dramatically reduces the development time of an application.

	Unidirectional data flow: React follows a unidirectional data flow. This means that when designing a React app, we often nest child components within parent components. And since the data flows in a single direction, it becomes easier to debug errors and know where the problem occurs in an application at the moment.
	Dedicated tools for easy debugging: Facebook has released a chrome extension that we can use to debug React applications. This makes the process of debugging React to web applications faster and easier.

6. What is the difference between the ES6 and ES5 standards?

This is one of the most frequently asked react interview questions.

These are the few instances where ES6 syntax has changed from ES5 syntax:

- **Components and Function**

```
// ES5
var MyComponent = React.createClass({
   render: function() {
      return(
       <h3>Hello Simplilearn</h3>
      );
   }
});

// ES6
class MyComponent extends React.Component {
   render() {
      return(
       <h3>Hello Simplilearn</h3>
      );
   }
}
```

- exports vs export

```
sqoop export --connect
jdbc:mysql://localhost/retail_db -username
root --password cloudera --table dept --
export-dir /user/cloudera/departments
```

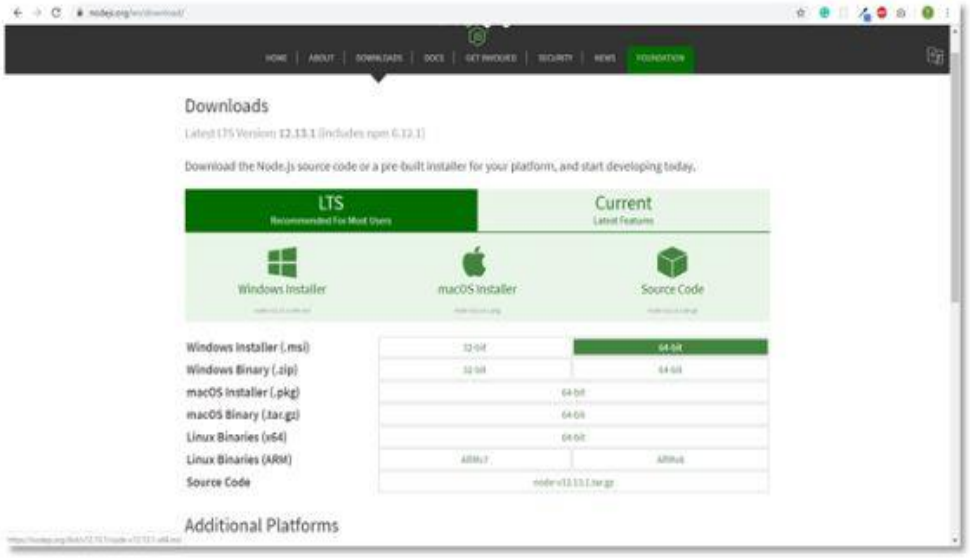
- require vs import

```
// ES5
var React = require('react');

// ES6
import React from 'react';
```

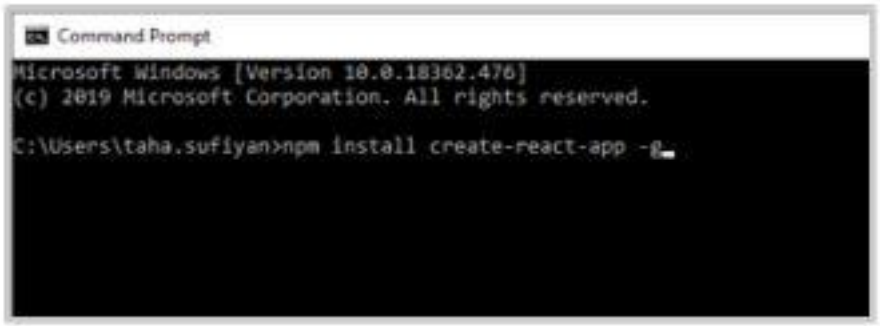
7. How do you create a React app?

These are the steps for creating a React app:

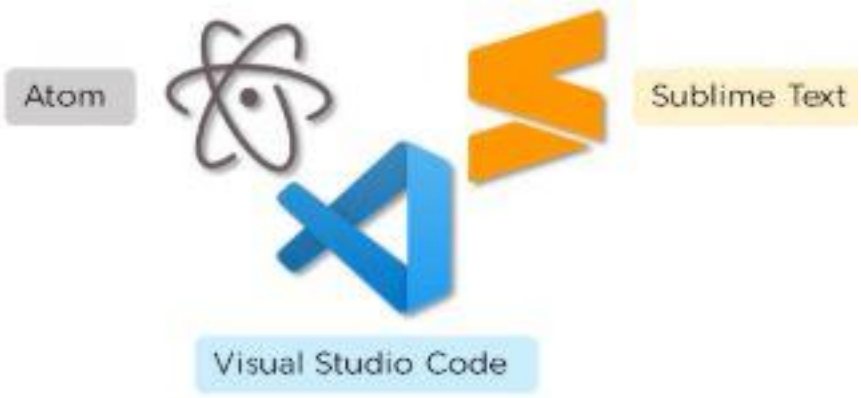
- Install NodeJS on the computer because we need npm to install the React library. Npm is the node package manager that contains many JavaScript libraries, including React.



- Install the create-react-app package using the command prompt or terminal.



- Install a text editor of your choice, like VS Code or Sublime Text.



We have put together a set of Node.js interview questions in case you would like to explore them.Please note, This is one of the most frequently asked react interview questions.

8. What is an event in React?

An event is an action that a user or system may trigger, such as pressing a key, a mouse click, etc.

- React events are named using camelCase, rather than lowercase in HTML.
- With JSX, you pass a function as the event handler, rather than a string in HTML.

```
<Button onPress={lightItUp} />
```

9. How do you create an event in React?

A React event can be created by doing the following:

```
class Simple extends React.Component {
    work() {
        alert("Good Work!");
    }
    render() {
        return (
            <button onClick={this.work}>Do some work!</button>
        );
    }
}
```

10. What are synthetic events in React?

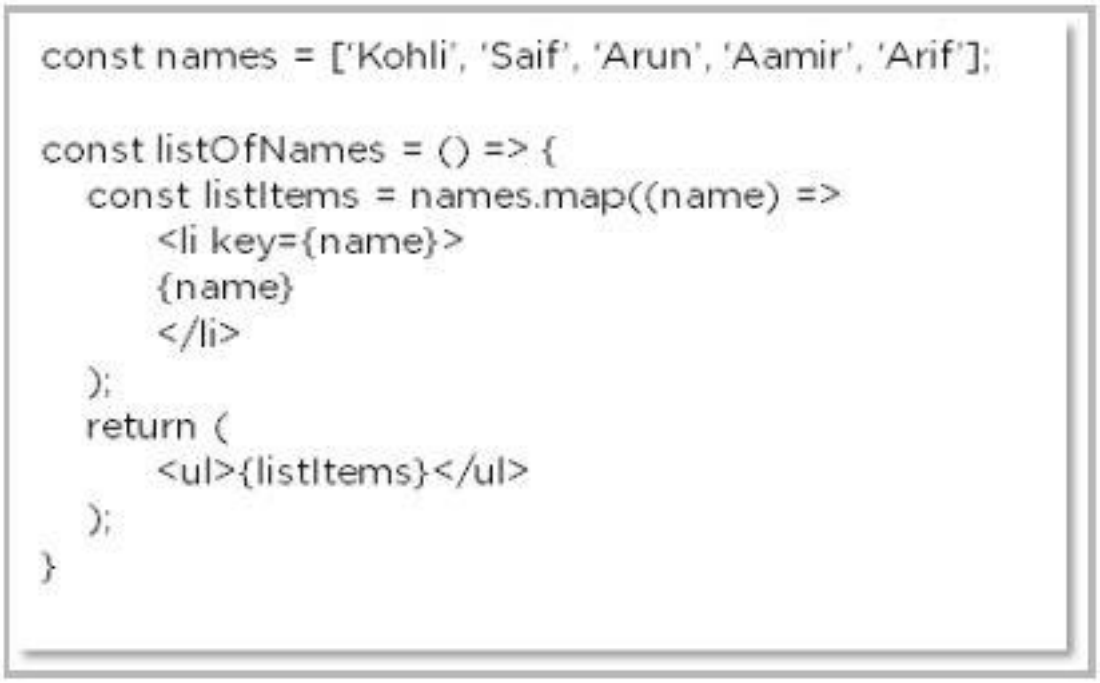
- Synthetic events combine the response of different browser's native events into one API, ensuring that the events are consistent across different browsers.
- The application is consistent regardless of the browser it is running in. Here, preventDefault is a synthetic event.

```
function ActionLink() {
    function handleClick(e) {
        e.preventDefault();
        console.log('You just clicked a Link.');
    }
    return (
        <a href="#" onClick={handleClick}>
            Click_Me
        </a>
    );
}
```

11. Explain how lists work in React

- We create lists in React as we do in regular JavaScript. Lists display data in an ordered format

- The traversal of lists is done using the map() function



```
const names = ['Kohli', 'Saif', 'Arun', 'Aamir', 'Arif'];

const listOfNames = () => {
   const listItems = names.map((name) =>
       <li key={name}>
       {name}
       </li>
   );
   return (
       <ul>{listItems}</ul>
   );
}
```

12. Why is there a need for using keys in Lists?

Keys are very important in lists for the following reasons:

- A key is a unique identifier and it is used to identify which items have changed, been updated or deleted from the lists
- It also helps to determine which components need to be re-rendered instead of re-rendering all the components every time. Therefore, it increases performance, as only the updated components are re-rendered

13. What are forms in React?

React employs forms to enable users to interact with web applications.

- Using forms, users can interact with the application and enter the required information whenever needed. Form contain certain elements, such as text fields, buttons, checkboxes, radio buttons, etc

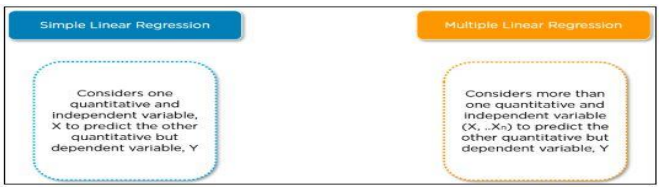
  Forms are used for many different tasks such as user authentication, searching, filtering, indexing, etc

14. How do you create forms in React?

We create forms in React by doing the following:

```
class NameForm extends React.Component {
 this.state = {value: ''};

 handleChange(event) {
  this.setState({value:  event.target.value});
 }

 handleSubmit(event) {
  alert('A name was entered: ' + this.state.value);
  event.preventDefault();
 }

 render() {
  return (
   <form onSubmit={this.handleSubmit.bind(this)}>
    <label>
     Name:
     <input type="text"  value={this.state.value}
onChange={this.handleChange.bind(this)} />
    </label>
    <input type="submit"  value="Submit" />
   </form>
  );
 }
}
```

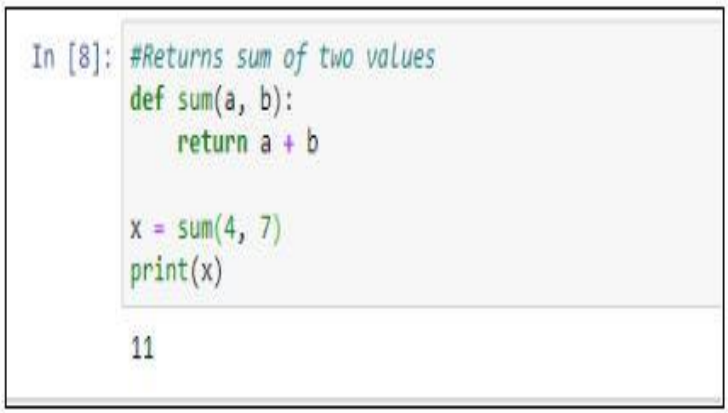
The above code will yield an input field with the label Name and a submit button. It will also alert the user when the submit button is pressed.



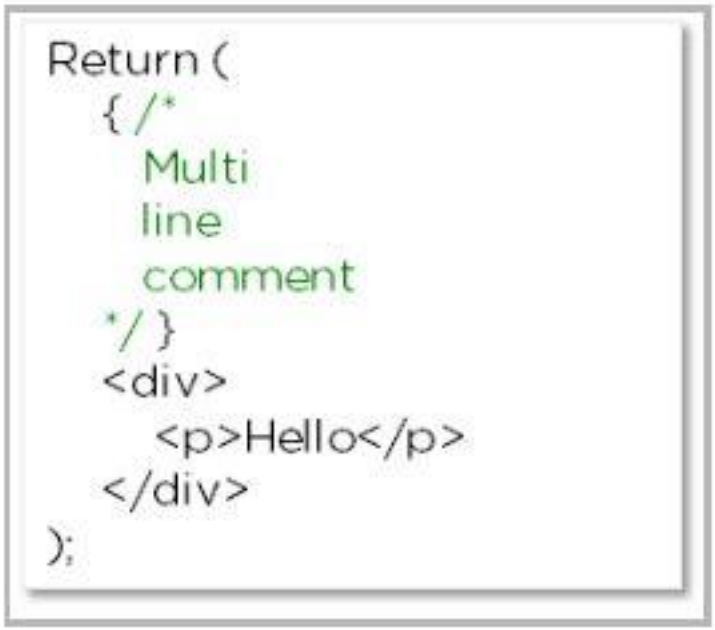
15. How do you write comments in React?

There are basically two ways in which we can write comments:

- Single-line comments



- Multi-line comments



16. What is an arrow function and how is it used in React?

- An arrow function is a short way of writing a function to React.
- It is unnecessary to bind ‘this’ inside the constructor when using an arrow function. This prevents bugs caused by the use of ‘this’ in React callbacks.

Without Arrow function

```
render() {
  return(
    <MyInput onChange={this.handleChange.bind(this) } />
  );
}
```

With Arrow function

```
render() {
  return(
    <MyInput onChange={ (e) => this.handleOnChange(e) } />
  );
}
```

17. How is React different from React Native?

	React	React Native
Release	2013	2015

Platform	Web	Mobile – Android, iOS
HTML	Yes	No
CSS	Yes	No
Prerequisites	JavaScript, HTML, CSS	React.js

18. How is React different from Angular?

	Angular	React
Author	Google	Facebook
Architecture	Complete MVC	View layer of MVC

DOM	Real DOM	Virtual DOM
Data-Binding	Bi-directional	Uni-directional
Rendering	Client-Side	Server-Side
Performance	Comparatively slow	Faster due to Virtual DOM

In case you have any doubts about these Basic React interview questions and answers, please leave your questions in the comment section below.

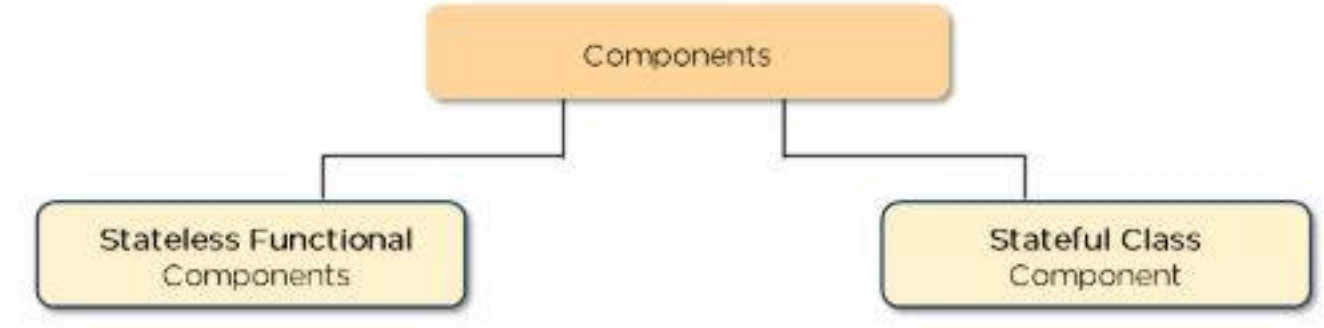
ReactJS Interview Questions on Components

Here are some React Interview Questions on components.

19. What are the components in React?

Components are the building blocks of any React application, and a single app usually consists of multiple components. A component is essentially a piece of the user interface. It splits the user interface into independent, reusable parts that can be processed separately.

There are two types of components in React:



- Functional Components: These types of components have no state of their own and only contain render methods, and therefore are also called stateless components. They may derive data from other components as props (properties).

```
function Greeting(props) {
  return <h1>Welcome to {props.name}</h1>;
}
```

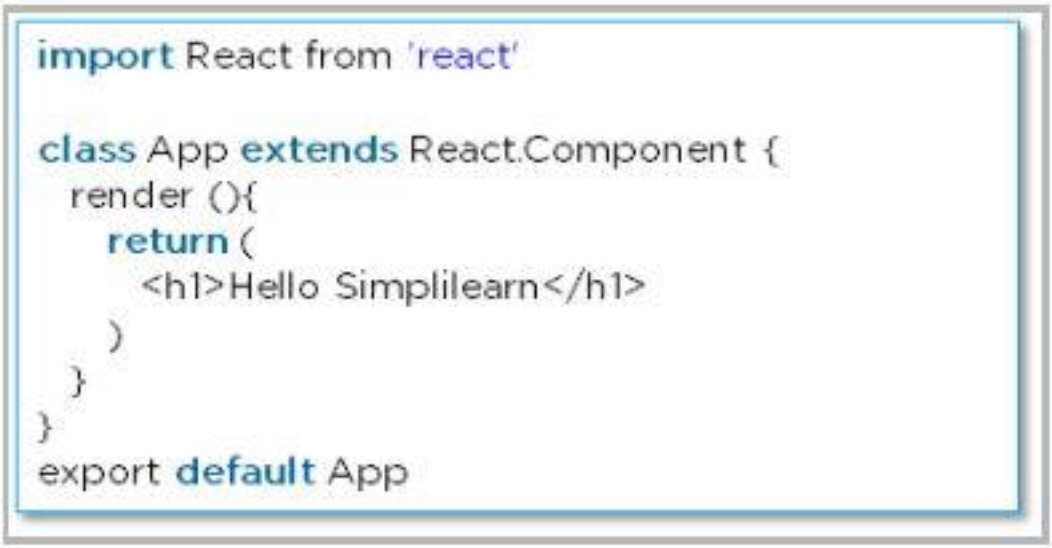
- Class Components: These types of components can hold and manage their own state and have a separate render method to return JSX on the screen. They are also called Stateful components as they can have a state.

```
class Greeting extends React.Component {
  render() {
    return <h1>Welcome to {this.props.name}</h1>;
  }
```

```
}
```

20. What is the use of render() in React?

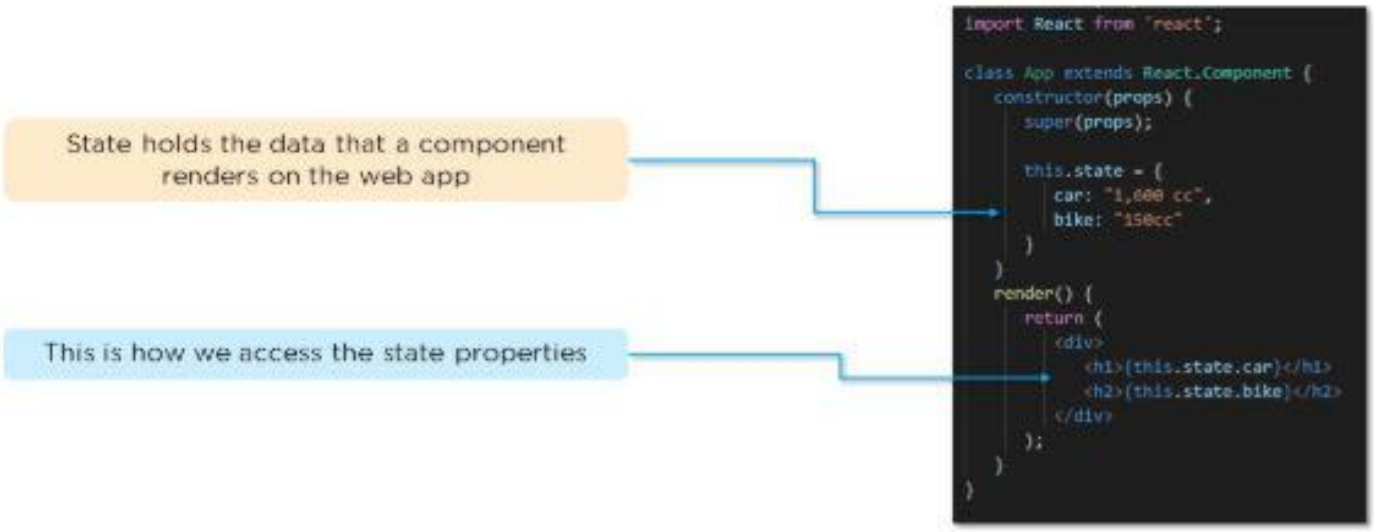
- It is required for each component to have a render() function. This function returns the HTML, which is to be displayed in the component.
- If you need to render more than one element, all of the elements must be inside one parent tag like , <form>.



21. What is a state in React?

- The state is a built-in React object that is used to contain data or information about the component. The state in a component can change over time, and whenever it changes, the component re-renders.
- The change in state can happen as a response to user action or system-generated events. It determines the behavior of the component and how it will render.

22. How do you implement state in React?



23. How do you update the state of a component?

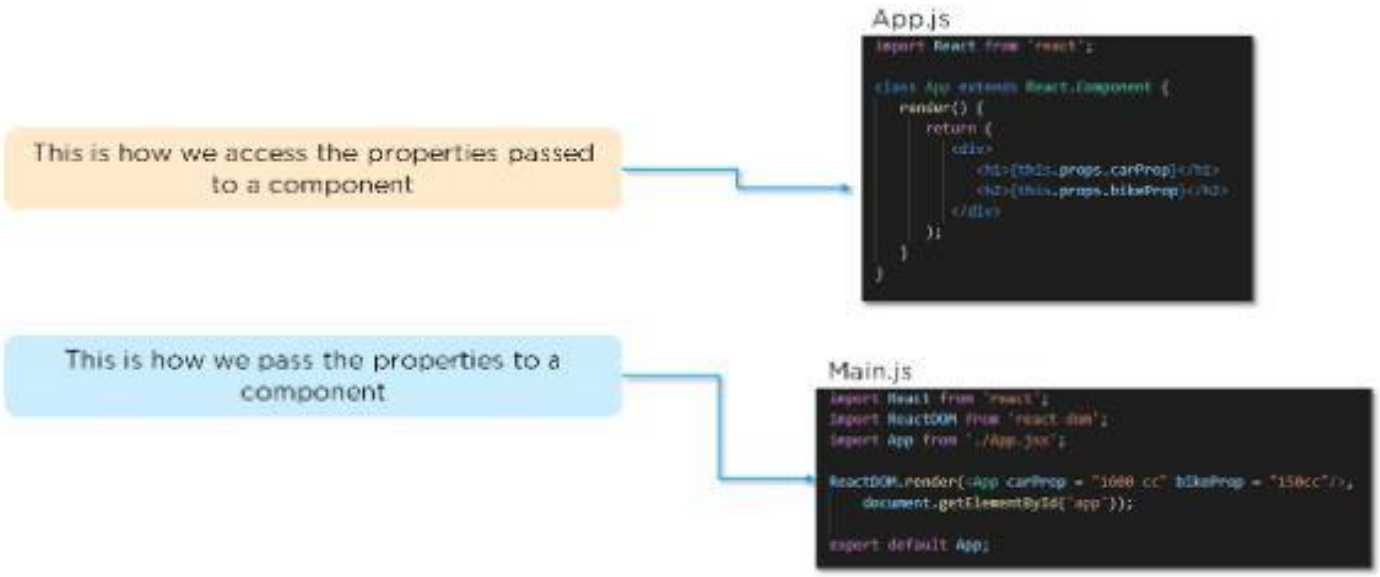
We can update the state of a component by using the built-in 'setState()' method:

```
class App extends React.Component {
  constructor() {
    super();
    this.state = {
      message: "Welcome to Simplilearn"
    };
    this.buttonPress = this.buttonPress.bind(this);
  }
  buttonPress() {
     this.setState({
       message:"The best place to learn"
     });
  }
  render() {
    return (
      <div>
        <h1>{this.state.msg}</h1>
        <button onClick = {this.buttonPress}>Click Me!</button>
      </div>
    );
  }
}
```

24. What are props in React?

- Props are short for Properties. It is a React built-in object that stores the value of attributes of a tag and works similarly to HTML attributes.
- Props provide a way to pass data from one component to another component. Props are passed to the component in the same way as arguments are passed in a function.

25. How do you pass props between components?



26. What are the differences between state and props?

	State	Props
Use	Holds information about the components	Allows to pass data from one component to other components as an argument

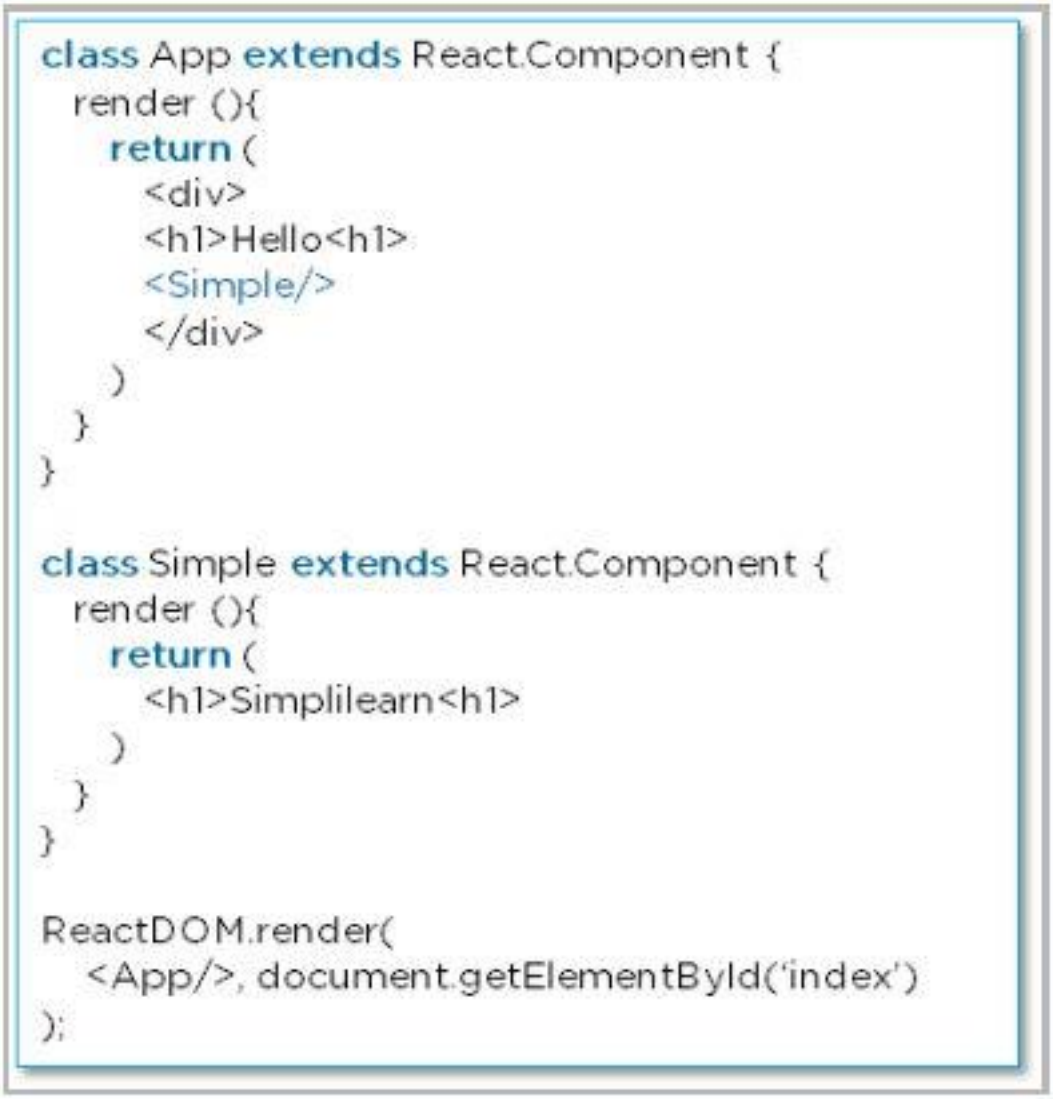
Mutability	Is mutable	Are immutable
Read-Only	Can be changed	Are read-only
Child components	Child components cannot access	Child component can access
Stateless components	Cannot have state	Can have props

27. What is a higher-order component in React?

A higher-order component acts as a container for other components. This helps to keep components simple and enables re-usability. They are generally used when multiple components have to use a common logic.

28. How can you embed two or more components into one?

We can embed two or more components into one using this method:



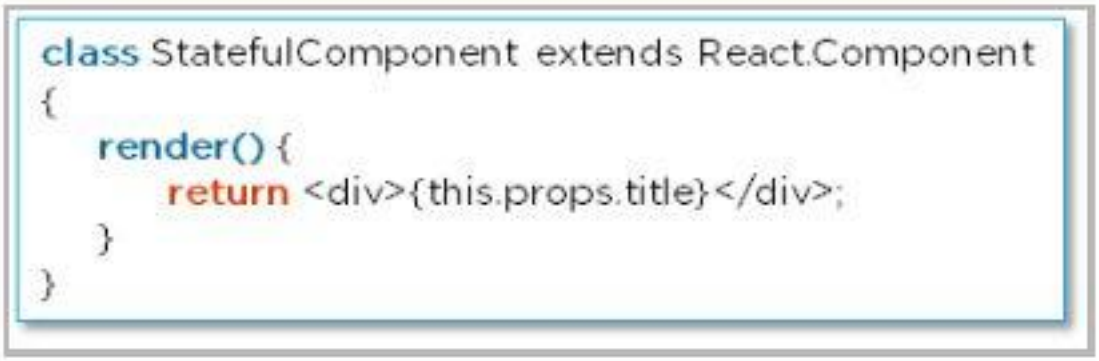
```
class App extends React.Component {
  render (){
    return (
      <div>
      <h1>Hello<h1>
      <Simple/>
      </div>
    )
  }
}

class Simple extends React.Component {
  render (){
    return (
      <h1>Simplilearn<h1>
    )
  }
}

ReactDOM.render(
  <App/>, document.getElementById('index')
);
```

29. What are the differences between class and functional components?

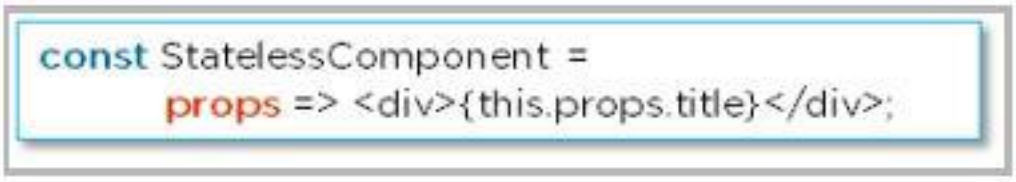
	Class Components	Functional Components
State	Can hold or manage state	Cannot hold or manage state

Simplicity	Complex as compared to the stateless component	Simple and easy to understand
Lifecycle methods	Can work with all lifecycle methods	Does not work with any lifecycle method
Reusability	Can be reused	Cannot be reused

- Class components example:



- Functional components example:



30. Explain the lifecycle methods of components.

- getInitialState(): This is executed before the creation of the component.

- componentDidMount(): Is executed when the component gets rendered and placed on the DOM.
- shouldComponentUpdate(): Is invoked when a component determines changes to the DOM and returns a “true” or “false” value based on certain conditions.
- componentDidUpdate(): Is invoked immediately after rendering takes place.
- componentWillUnmount(): Is invoked immediately before a component is destroyed and unmounted permanently.

So far, if you have any doubts about the above React interview questions and answers, please ask your questions in the section below.

ReactJS Redux Interview Questions

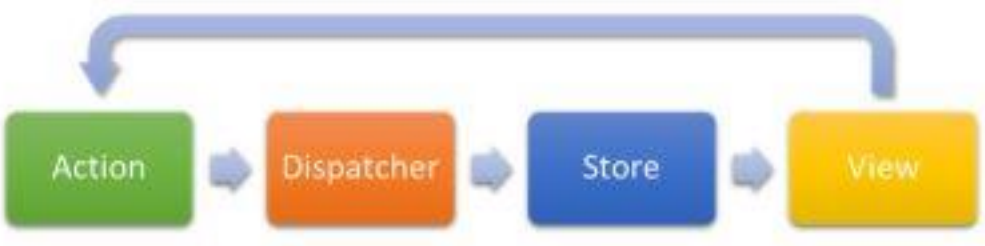
Here are some ReactJS Interview Questions on the ReactJS Redux concept.

31. What is Redux?

Redux is an open-source, JavaScript library used to manage the application state. React uses Redux to build the user interface. It is a predictable state container for JavaScript applications and is used for the entire application’s state management.

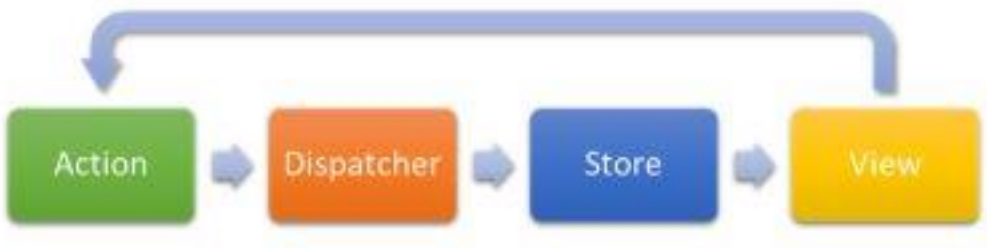
32. What are the components of Redux?

- Store: Holds the state of the application.
- Action: The source information for the store.
- Reducer: Specifies how the application's state changes in response to actions sent to the store.

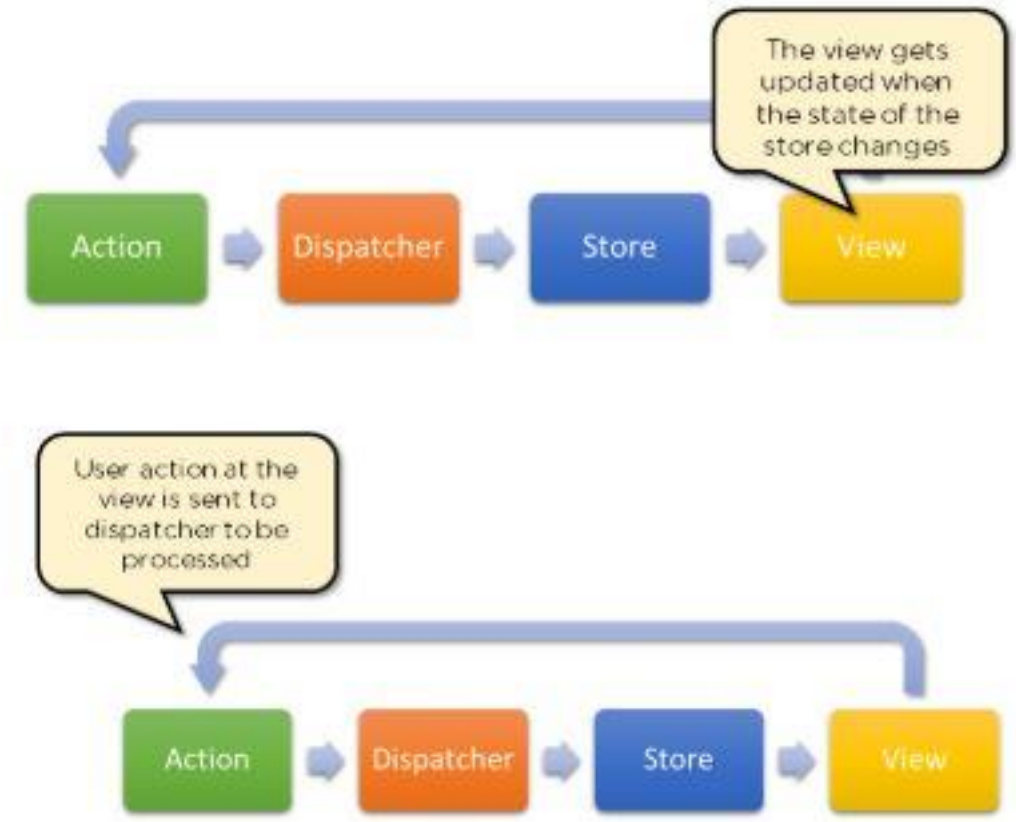


33. What is the Flux?

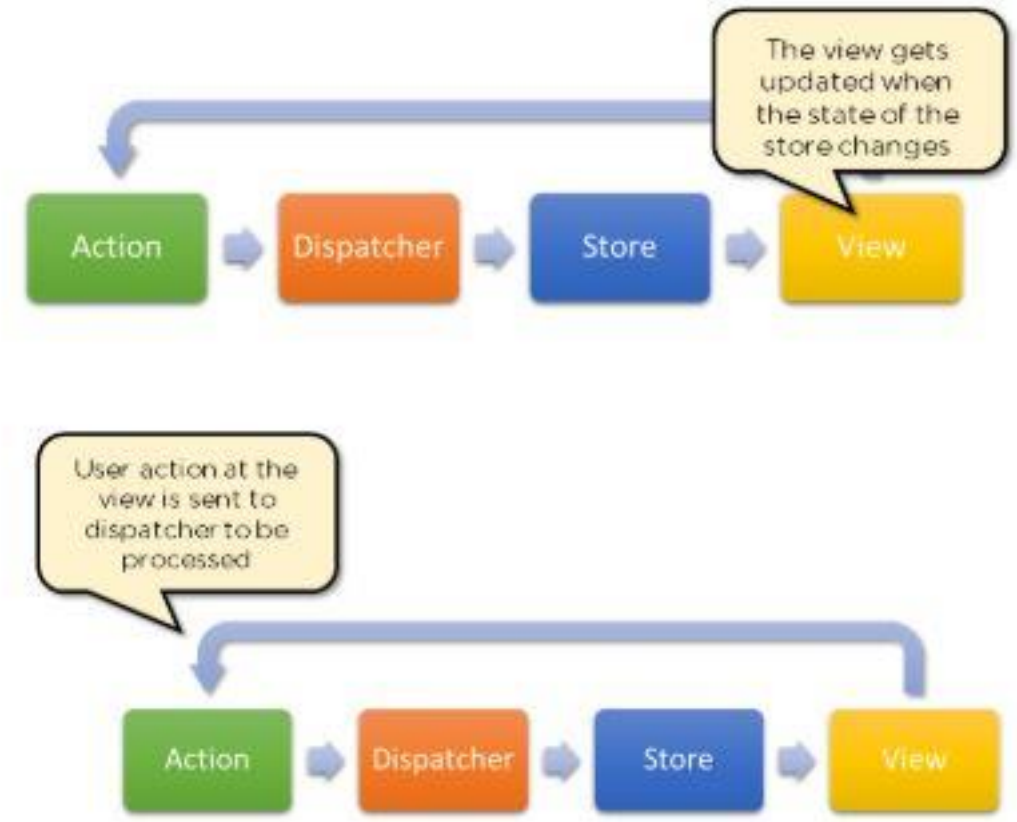
- Flux is the application architecture that Facebook uses for building web applications. It is a method of handling complex data inside a client-side application and manages how data flows in a React application.



- There is a single source of data (the store) and triggering certain actions is the only way way to update them.The actions call the dispatcher, and then the store is triggered and updated with their own data accordingly.



- When a dispatch has been triggered, and the store updates, it will emit a change event that the views can rerender accordingly.



34. How is Redux different from Flux?

SN	Redux	Flux
1.	Redux is an open-source JavaScript library used to manage application State	Flux is an architecture and not a framework or library
2.	Store’s state is immutable	Store’s state is mutable

3.	Can only have a single-store	Can have multiple stores
4.	Uses the concept of reducer	Uses the concept of the dispatcher

So far, if you have any doubts about these React interview questions and answers, please leave your questions in the section below.

ReactJS Router Questions

Here are some ReactJS Interview Questions on React Router concepts.

35. What is React Router?

React Router is a routing library built on top of React, which is used to create routes in a React application. This is one of the most frequently asked react interview questions.

36. Why do we need to React Router?

- It maintains consistent structure and behavior and is used to develop single-page web applications.
- Enables multiple views in a single application by defining multiple routes in the React application.

37. How is React routing different from conventional routing?

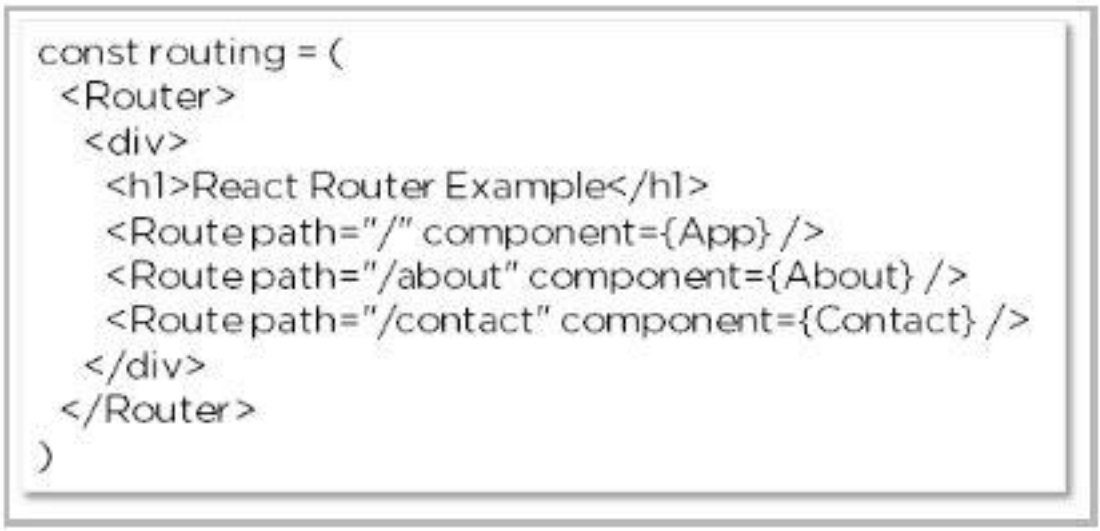
SN	React Routing	Conventional routing

1.	Single HTML page	Each view is a new HTML file
2.	The user navigates multiple views in the same file	The user navigates multiple files for each view
3.	The page does not refresh since it is a single file	The page refreshes every time user navigates
4.	Improved performance	Slower performance

38. How do you implement React routing?

We can implement routing in our React application using this method:

Considering we have the components App, About, and Contact in our application:



Hope you have no doubts about this ReactJS interview questions article, in case of any difficulty, please leave your problems in the section below.

ReactJS Styling Questions

Here are some ReactJS Interview Questions on Styling concept ReactJS.

39. How do you style React components?

There are several ways in which we can style React components:

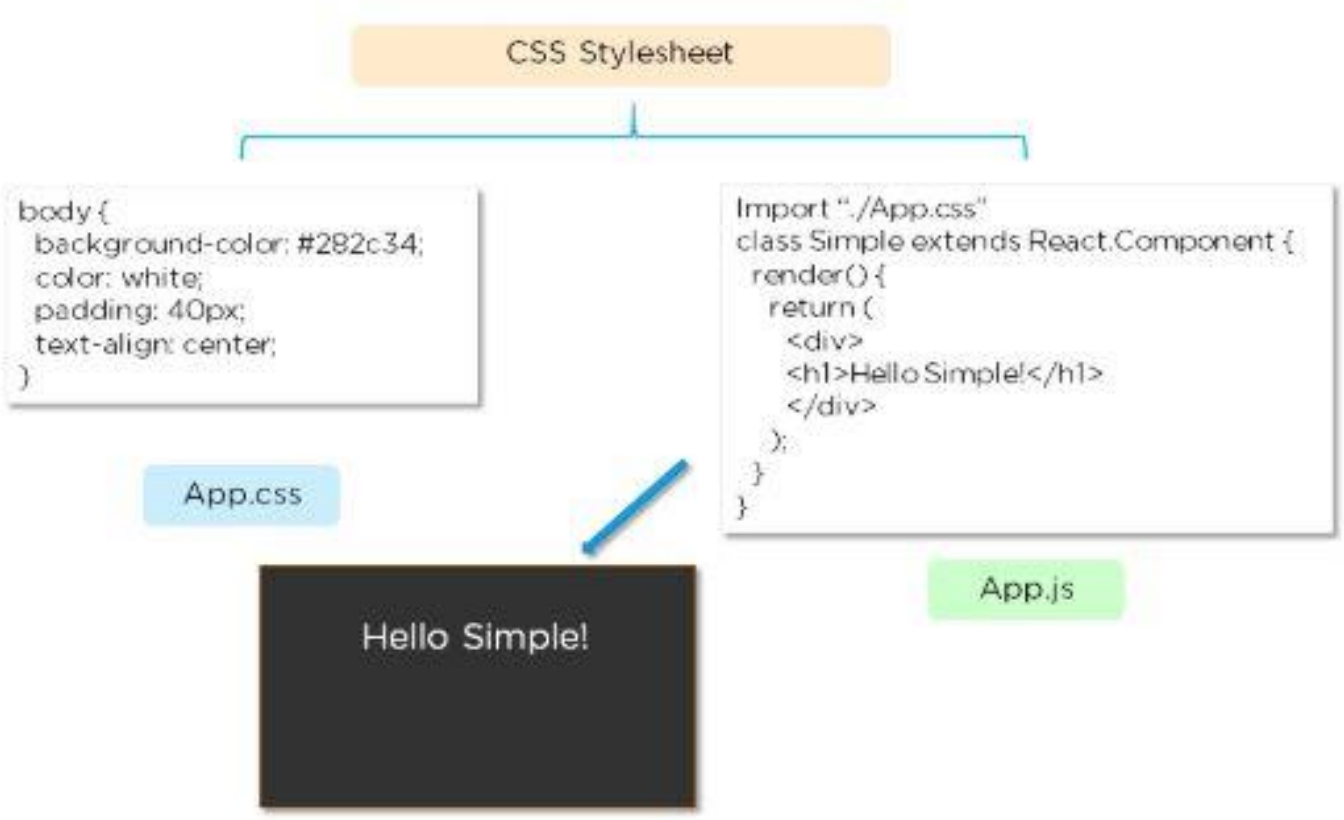
- Inline Styling

```
class Simple extends React.Component {
  render() {
    return (
      <div>
      <h1 style={{color: "blue"}}>Hello Simple!</h1>
      </div>
    );
  }
}
```

- JavaScript Object

```
class Simple extends React.Component {
 render() {
   const simpleStyle = {
     color: "white",
     backgroundColor: "Green",
     margin: "8px",
     fontFamily: "Open Sans"
   };
   return (
     <div>
     <h1 style={simpleStyle}>Hello Simple!</h1>
     </div>
   );
 }
}
```

- CSS Stylesheet



40. Explain the use of CSS modules in React.

- The CSS module file is created with the .module.css extension
- The CSS inside a module file is available only for the component that imported it, so there are no naming conflicts while styling the components.

These are all the basic to advanced ReactJS interview questions that are frequently asked in interviews. We hope these ReactJS interview questions will be helpful in clearing your interview round. All the best for your upcoming job interview!

Node.js Interview Questions and Answers For Freshers

This section will provide you with the Basic Node.js interview questions which will primarily help freshers.

1. What is Node.js? Where can you use it?

Node.js is an open-source, cross-platform JavaScript runtime environment and library to run web applications outside the client's browser. It is used to create server-side web applications.

Node.js is perfect for data-intensive applications as it uses an asynchronous, event-driven model. You can use I/O intensive web applications like video streaming sites. You can also use it for developing: Real-time web applications, Network applications, General-purpose applications, and Distributed systems.

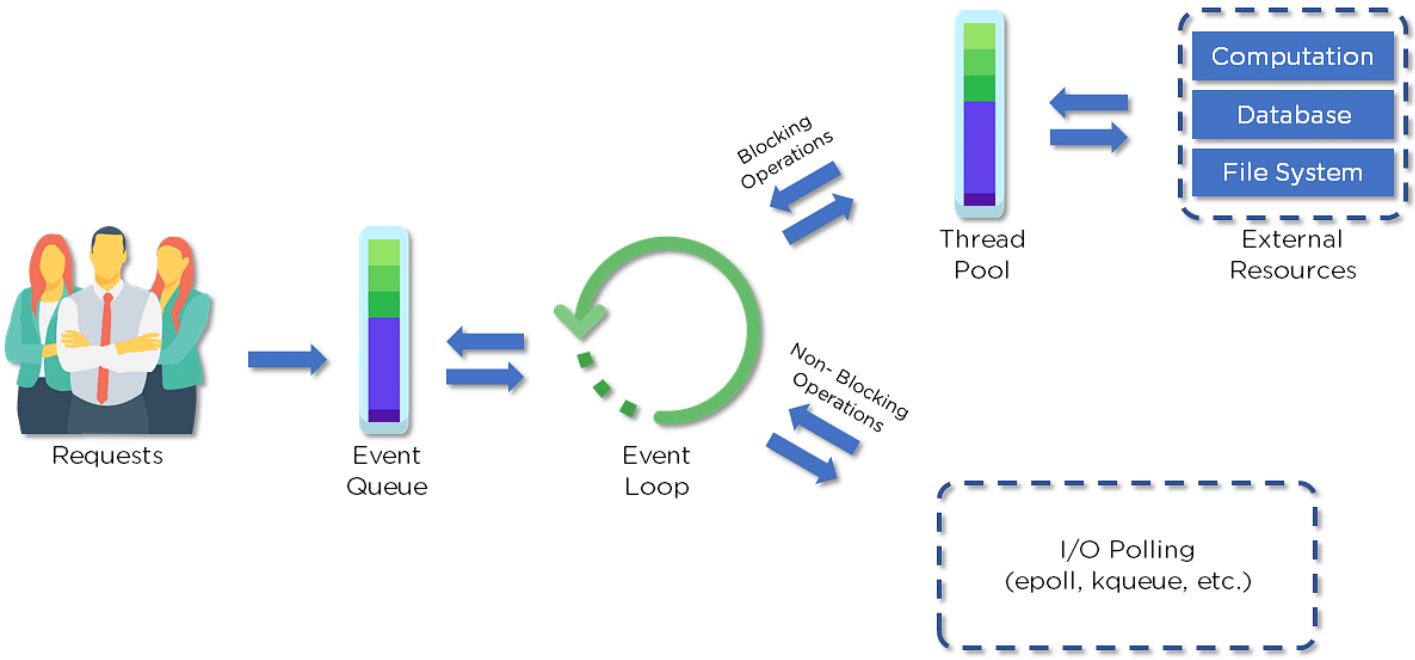
2. Why use Node.js?

Node.js makes building scalable network programs easy. Some of its advantages include:

- It is generally fast
- It rarely blocks
- It offers a unified programming language and data type
- Everything is asynchronous
- It yields great concurrency

3. How does Node.js work?

A web server using Node.js typically has a workflow that is quite similar to the diagram illustrated below. Let's explore this flow of operations in detail.



- Clients send requests to the webserver to interact with the web application. Requests can be non-blocking or blocking:
- Querying for data
- Deleting data
- Updating the data
- Node.js retrieves the incoming requests and adds those to the Event Queue
- The requests are then passed one-by-one through the Event Loop. It checks if the requests are simple enough not to require any external resources
- The Event Loop processes simple requests (non-blocking operations), such as I/O Polling, and returns the responses to the corresponding clients

A single thread from the Thread Pool is assigned to a single complex request. This thread is responsible for completing a particular blocking request by accessing external resources, such as computation, database, file system, etc.

Once the task is carried out completely, the response is sent to the Event Loop that sends that response back to the client.

4. Why is Node.js Single-threaded?

Node.js is single-threaded for async processing. By doing async processing on a single-thread under typical web loads, more performance and scalability can be achieved instead of the typical thread-based implementation.

5. If Node.js is single-threaded, then how does it handle concurrency?

The Multi-Threaded Request/Response Stateless Model is not followed by the Node JS Platform, and it adheres to the Single-Threaded Event Loop Model. The Node JS Processing paradigm is heavily influenced by the JavaScript Event-based model and the JavaScript callback system. As a result, Node.js can easily manage more concurrent client requests. The event loop is the processing model's beating heart in Node.js.

6. Explain callback in Node.js.

A callback function is called after a given task. It allows other code to be run in the meantime and prevents any blocking. Being an asynchronous platform, Node.js heavily relies on callback. All APIs of Node are written to support callbacks.

7. What are the advantages of using promises instead of callbacks?

- The control flow of asynchronous logic is more specified and structured.
- The coupling is low.
- We've built-in error handling.
- Improved readability.

8. How would you define the term I/O?

- The term I/O is used to describe any program, operation, or device that transfers data to or from a medium and to or from another medium
- Every transfer is an output from one medium and an input into another. The medium can be a physical device, network, or files within a system

9. How is Node.js most frequently used?

Node.js is widely used in the following applications:

1. Real-time chats
2. Internet of Things
3. Complex SPAs (Single-Page Applications)
4. Real-time collaboration tools
5. Streaming applications
6. Microservices architecture

10. Explain the difference between frontend and backend development?

Front-end	Back-end
Frontend refers to the client-side of an application	Backend refers to the server-side of an application

It is the part of a web application that users can see and interact with	It constitutes everything that happens behind the scenes
It typically includes everything that attributes to the visual aspects of a web application	It generally includes a web server that communicates with a database to serve requests
HTML, CSS, JavaScript, AngularJS, and ReactJS are some of the essentials of frontend development	Java, PHP, Python, and Node.js are some of the backend development technologies

11. What is NPM?

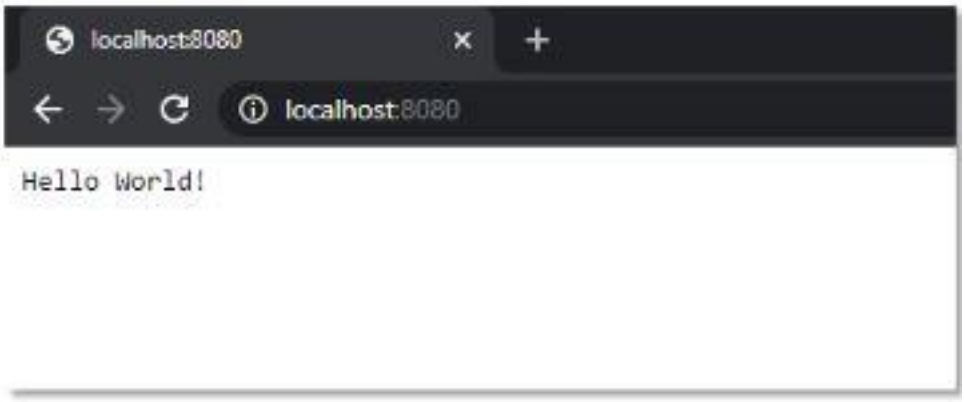
NPM stands for Node Package Manager, responsible for managing all the packages and modules for Node.js.

Node Package Manager provides two main functionalities:

- Provides online repositories for node.js packages/modules, which are searchable on search.nodejs.org
- Provides command-line utility to install Node.js packages and also manages Node.js versions and dependencies

12. What are the modules in Node.js?

Modules are like JavaScript libraries that can be used in a Node.js application to include a set of functions. To include a module in a Node.js application, use the require() function with the parentheses containing the module's name.



Node.js has many modules to provide the basic functionality needed for a web application. Some of them include:

Core Modules	Description
HTTP	Includes classes, methods, and events to create a Node.js HTTP server
util	Includes utility functions useful for developers
fs	Includes events, classes, and methods to deal with file I/O operations

url	Includes methods for URL parsing
query string	Includes methods to work with query string
stream	Includes methods to handle streaming data
zlib	Includes methods to compress or decompress files

13. What is the purpose of the module .Exports?

In Node.js, a module encapsulates all related codes into a single unit of code that can be parsed by moving all relevant functions into a single file. You may export a module with the module and export the function, which lets it be imported into another file with a needed keyword.

14. Why is Node.js preferred over other backend technologies like Java and PHP?

Some of the reasons why Node.js is preferred include:

- Node.js is very fast
- Node Package Manager has over 50,000 bundles available at the developer's disposal
- Perfect for data-intensive, real-time web applications, as Node.js never waits for an API to return data

- Better synchronization of code between server and client due to same code base
- Easy for web developers to start using Node.js in their projects as it is a JavaScript library

15. What is the difference between Angular and Node.js?

Angular	Node.js
It is a frontend development framework	It is a server-side environment
It is written in TypeScript	It is written in C, C++ languages
Used for building single-page, client-side web applications	Used for building fast and scalable server-side networking applications
Splits a web application into MVC components	Generates database queries

Also Read: What is Angular?

16. Which database is more popularly used with Node.js?

MongoDB is the most common database used with Node.js. It is a NoSQL, cross-platform, document-oriented database that provides high performance, high availability, and easy scalability.

17. What are some of the most commonly used libraries in Node.js?

There are two commonly used libraries in Node.js:

- ExpressJS - Express is a flexible Node.js web application framework that provides a wide set of features to develop web and mobile applications.
- Mongoose - Mongoose is also a Node.js web application framework that makes it easy to connect an application to a database.

18. What are the pros and cons of Node.js?

Node.js Pros	Node.js Cons
Fast processing and an event-based model	Not suitable for heavy computational tasks
Uses JavaScript, which is well-known amongst developers	Using callback is complex since you end up with several nested callbacks
Node Package Manager has over 50,000 packages that provide the functionality to an application	Dealing with relational databases is not a good option for Node.js

Best suited for streaming huge amounts of data and I/O intensive operations	Since Node.js is single-threaded, CPU intensive tasks are not its strong suit

19. What is the command used to import external libraries?

The “require” command is used for importing external libraries. For example - “var http=require (“HTTP”).” This will load the HTTP library and the single exported object through the HTTP variable.

Now that we have covered some of the important beginner-level Node.js interview questions let us look at some of the intermediate-level Node.js interview questions.

Node.js Interview Questions and Answers For Intermediate Level

20. What does event-driven programming mean?

An event-driven programming approach uses events to trigger various functions. An event can be anything, such as typing a key or clicking a mouse button. A call-back function is already registered with the element executes whenever an event is triggered.

21. What is an Event Loop in Node.js?

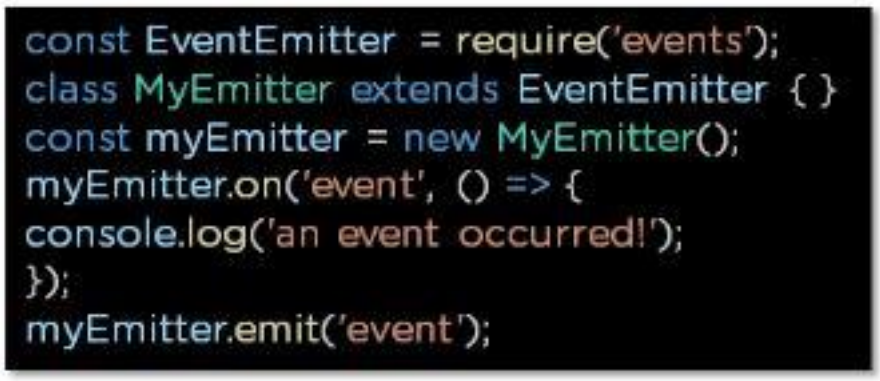
Event loops handle asynchronous callbacks in Node.js. It is the foundation of the non-blocking input/output in Node.js, making it one of the most important environmental features.

22. Differentiate between process.nextTick() and setImmediate()?

The distinction between method and product. This is accomplished through the use of nextTick() and setImmediate(). next Tick() postpones the execution of action until the next pass around the event loop, or it simply calls the callback function once the event loop's current execution is complete, whereas setImmediate() executes a callback on the next cycle of the event loop and returns control to the event loop for any I/O operations.

23. What is an EventEmitter in Node.js?

- EventEmitter is a class that holds all the objects that can emit events
- Whenever an object from the EventEmitter class throws an event, all attached functions are called upon synchronously



24. What are the two types of API functions in Node.js?

The two types of API functions in Node.js are:

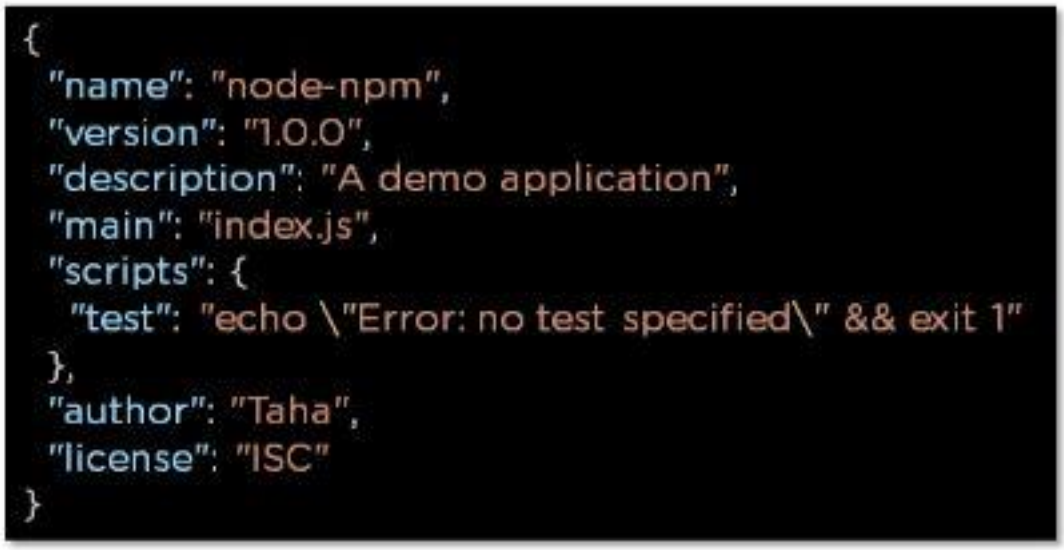
- Asynchronous, non-blocking functions
- Synchronous, blocking functions

25. What is the package.json file?

The package.json file is the heart of a Node.js system. This file holds the metadata for a particular project. The package.json file is found in the root directory of any Node application or module

This is what a package.json file looks like immediately after creating a Node.js project using the command: npm init

You can edit the parameters when you create a Node.js project.



26. How would you use a URL module in Node.js?

The URL module in Node.js provides various utilities for URL resolution and parsing. It is a built-in module that helps split up the web address into a readable format.

27. What is the Express.js package?

Express is a flexible Node.js web application framework that provides a wide set of features to develop both web and mobile applications

28. How do you create a simple Express.js application?

- The request object represents the HTTP request and has properties for the request query string, parameters, body, HTTP headers, and so on
- The response object represents the HTTP response that an Express app sends when it receives an HTTP request

29. What are streams in Node.js?

Streams are objects that enable you to read data or write data continuously.

There are four types of streams:

Readable – Used for reading operations

Writable − Used for write operations

Duplex − Can be used for both reading and write operations

Transform − A type of duplex stream where the output is computed based on input

30. How do you install, update, and delete a dependency?

31. How do you create a simple server in Node.js that returns Hello World?

- Import the HTTP module
- Use createServer function with a callback function using request and response as parameters.
- Type “hello world."
- Set the server to listen to port 8080 and assign an IP address

32. Explain asynchronous and non-blocking APIs in Node.js.

- All Node.js library APIs are asynchronous, which means they are also non-blocking

- A Node.js-based server never waits for an API to return data. Instead, it moves to the next API after calling it, and a notification mechanism from a Node.js event responds to the server for the previous API call

33. How do we implement async in Node.js?

As shown below, the async code asks the JavaScript engine running the code to wait for the request.get() function to complete before moving on to the next line for execution.

34. What is a callback function in Node.js?

A callback is a function called after a given task. This prevents any blocking and enables other code to run in the meantime.

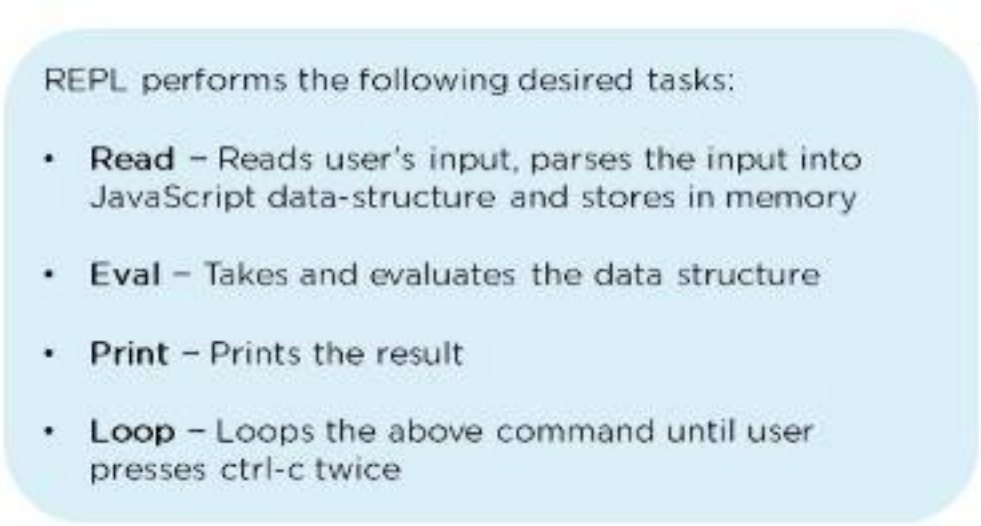
In the last section, we will now cover some of the advanced-level Node.js interview questions.

Node.js Interview Questions and Answers For Experienced Professionals

This section will provide you with the Advanced Node.js interview questions which will primarily help experienced professionals.

35. What is REPL in Node.js?

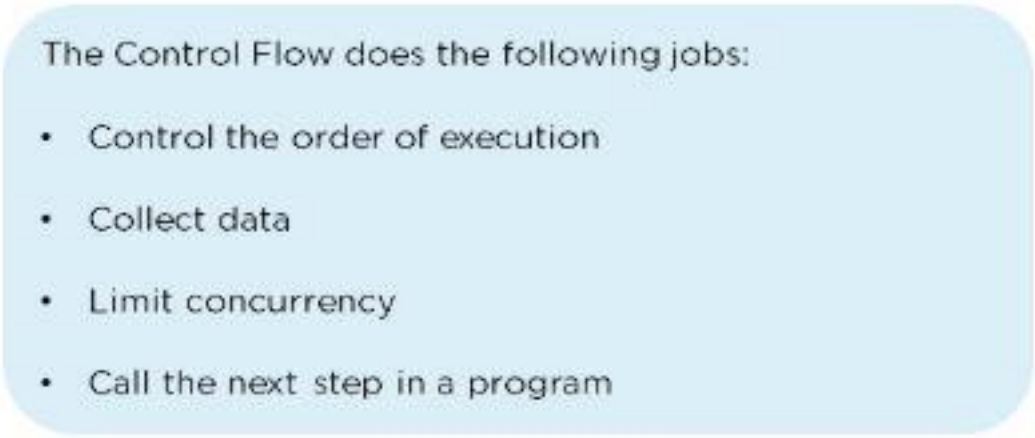
REPL stands for Read Eval Print Loop, and it represents a computer environment. It’s similar to a Windows console or Unix/Linux shell in which a command is entered. Then, the system responds with an output



36. What is the control flow function?

The control flow function is a piece of code that runs in between several asynchronous function calls.

37. How does control flow manage the function calls?



38. What is the difference between fork() and spawn() methods in Node.js?

fork()	spawn()

child_process.fork(modulePath[, args][, options])	child_process.spawn(command[, args][, options])
fork() is a particular case of spawn() that generates a new instance of a V8 engine.	Spawn() launches a new process with the available set of commands.
Multiple workers run on a single node code base for multiple tasks.	This method doesn't generate a new V8 instance, and only a single copy of the node module is active on the processor.

39. What is the buffer class in Node.js?

Buffer class stores raw data similar to an array of integers but corresponds to a raw memory allocation outside the V8 heap. Buffer class is used because pure JavaScript is not compatible with binary data.

40. What is piping in Node.js?

Piping is a mechanism used to connect the output of one stream to another stream. It is normally used to retrieve data from one stream and pass output to another stream.

41. What are some of the flags used in the read/write operations in files?

- **r** – Open file for reading. An exception occurs if the file does not exist.
- **r+** – Open file for reading and writing. An exception occurs if the file does not exist.
- **w** – Open file for writing. The file is created (if it does not exist) or truncated (if it exists).
- **w+** – Open file for reading and writing. The file is created (if it does not exist) or truncated (if it exists).
- **a** – Open file for appending. The file is created if it does not exist.
- **a+** – Open file for reading and appending. The file is created if it does not exist.

42. How do you open a file in Node.js?

This is the syntax for opening a file in Node.js

```
fs.open(path, flags[, mode], callback)
```

43. What is callback hell?

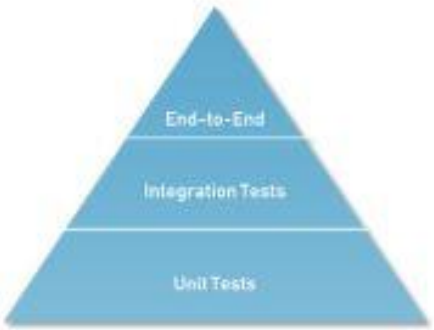
- Callback hell, also known as the pyramid of doom, is the result of intensively nested, unreadable, and unmanageable callbacks, which in turn makes the code harder to read and debug
- improper implementation of the asynchronous logic causes callback hell

44. What is a reactor pattern in Node.js?

A reactor pattern is a concept of non-blocking I/O operations. This pattern provides a handler that is associated with each I/O operation. As soon as an I/O request is generated, it is then submitted to a demultiplexer

45. What is a test pyramid in Node.js?

- A test pyramid is a figure which explains the proportion of unit tests, integrations tests, and end-to-end tests that are required for the proper development of a project
- The components of a test pyramid are given below:
  - **Unit Tests:** They test the individual units of code in isolation
  - **Integrations Tests:** They test the integration among dissimilar units
  - **End-to-End (E2E) Tests:** They test the whole system, from the User Interface to the data store, and back.



46. For Node.js, why does Google use the V8 engine?

The V8 engine, developed by Google, is open-source and written in C++. Google Chrome makes use of this engine. V8, unlike the other engines, is also utilized for the popular Node.js runtime. V8 was initially intended to improve the speed of JavaScript execution within web browsers. Instead of employing an interpreter, V8 converts JavaScript code into more efficient machine code to increase performance. It turns JavaScript code into machine code during execution by utilizing a JIT (Just-In-Time) compiler, as do many current JavaScript engines such as SpiderMonkey or Rhino (Mozilla).

47. Describe Node.js exit codes.

Exit codes are a set of specific codes which are used for finishing a specific process. Given below are some of the exit codes used in Node.js:

- Uncaught fatal exception
- Unused
- Fatal Error
- Internal Exception handler Run-time failure
- Internal JavaScript Evaluation Failure

48. Explain the concept of middleware in Node.js.

Middleware is a function that receives the request and response objects. Most tasks that the middleware functions perform are:

- Execute any code
- Update or modify the request and the response objects
- Finish the request-response cycle
- Invoke the next middleware in the stack

49. What are the different types of HTTP requests?

HTTP defines a set of request methods used to perform desired actions. The request methods include:

GET: Used to retrieve the data

POST: Generally used to make a change in state or reactions on the server

HEAD: Similar to the GET method, but asks for the response without the response body

DELETE: Used to delete the predetermined resource

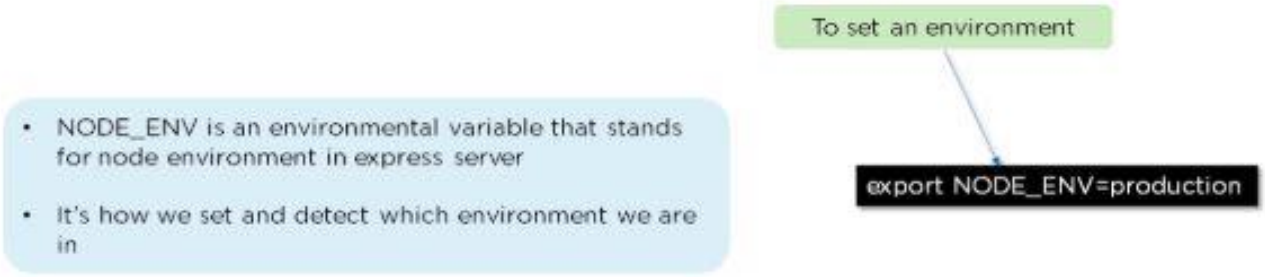
50. How would you connect a MongoDB database to Node.js?

To create a database in MongoDB:

- Start by creating a MongoClient object
- Specify a connection URL with the correct IP address and the name of the database you want to create

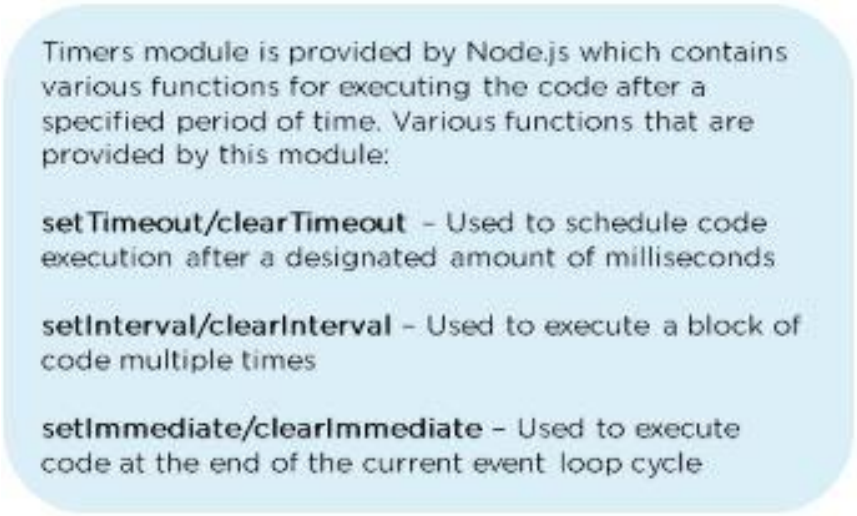
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/mydb";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  console.log("Database created!");
  db.close();
});
```

51. What is the purpose of NODE_ENV?



52. List the various Node.js timing features.

As you prepare for your upcoming job interview, we hope that this comprehensive guide has provided more insight into what types of questions you'll be asked.



53. What is WASI, and why is it being introduced?

The WASI class implements the WASI system called API and extra convenience methods for interacting with WASI-based applications. Every WASI instance represents a unique sandbox environment. Each WASI instance must specify its command-line parameters, environment variables, and sandbox directory structure for security reasons.

Conclusion

I believe that these Node.js interview questions would help you understand what kind of questions may be asked to you in an interview, and by going through these Node.js interview questions, you can prepare and crack your next interview in one go.

Express JS Interview Questions and Answers for Freshers

1. How do you install an express application generator for scaffolding?

Express application generator is used for quickly creating an application skeleton. The given command is used for installing the express application generator.

```
npm install express-generator –g
express myApp
```

It will create the project "myApp" with some files. Then we install all the dependencies stated in package.json using the given command:

```
cd myApp
npm install
```

2. How do you install a yeoman for scaffolding?

Generators are used by yeoman for scaffolding the applications. And we can use the following command to install yeoman.

```
npm install –g yeoman
```

3. Mention the arguments that are available in an Express JS route handler function.

The arguments that are available in the route handler function of Express JS are given below:

- **Res** - It is the response object.
- **Req** - It is the request object
- **Next (optional)** - This argument is used for passing the management to any of the above-given route handlers.

4. Mention the ways of debugging on Linux as well as Windows.

Debugging on Windows can be done as follows:

```
set DEBUG = express:*
node app.js
```

And debugging on Linux can be done as follows:

```
DEBUG = express:*
node app.js
```

5. List the built-in middleware functions provided by Express.

Express JS provides the following built-in middleware functions:

i. **Static**: We use it for serving static assets like images, HTML files, etc.

ii. **JSON**: This is available in Express 4.16.0+. And we use it for passing the incoming requests with JSON payloads.

iii. **URL encoded**: This is also available in Express 4.16.0+. And we use it for passing the incoming requests with URL-encoded payloads.

6. Mention some third-party middleware provided by Express JS.

Some of the many third-party middlewares that the Express JS provides are:

- Cookie-parser
- Body-parser
- Cors
- Mongoose
- Express-validator
- Sequelize.

7. When is application-level Middleware used?

We use the application-level Middleware for binding the app object with the help of the app.use() method. It can be applied on all routes. The syntax is given below:

```
// This Middleware executes for each route.
App.use(function (req, res, next) {
  console.log('Current Time:', Date.now())
  next()
```

```
})
```

8. Tell us about Router-level Middleware and Built-in Middleware.

Router-level Middleware - We use the router-level Middleware for binding with a particular instance of Express.Router().

Built-in Middleware - The version 4.x of Express introduced the built-in Middleware. The dependency on connecting gets removed by use of this Middleware.

9. Mention some of the databases with which Express JS is integrated.

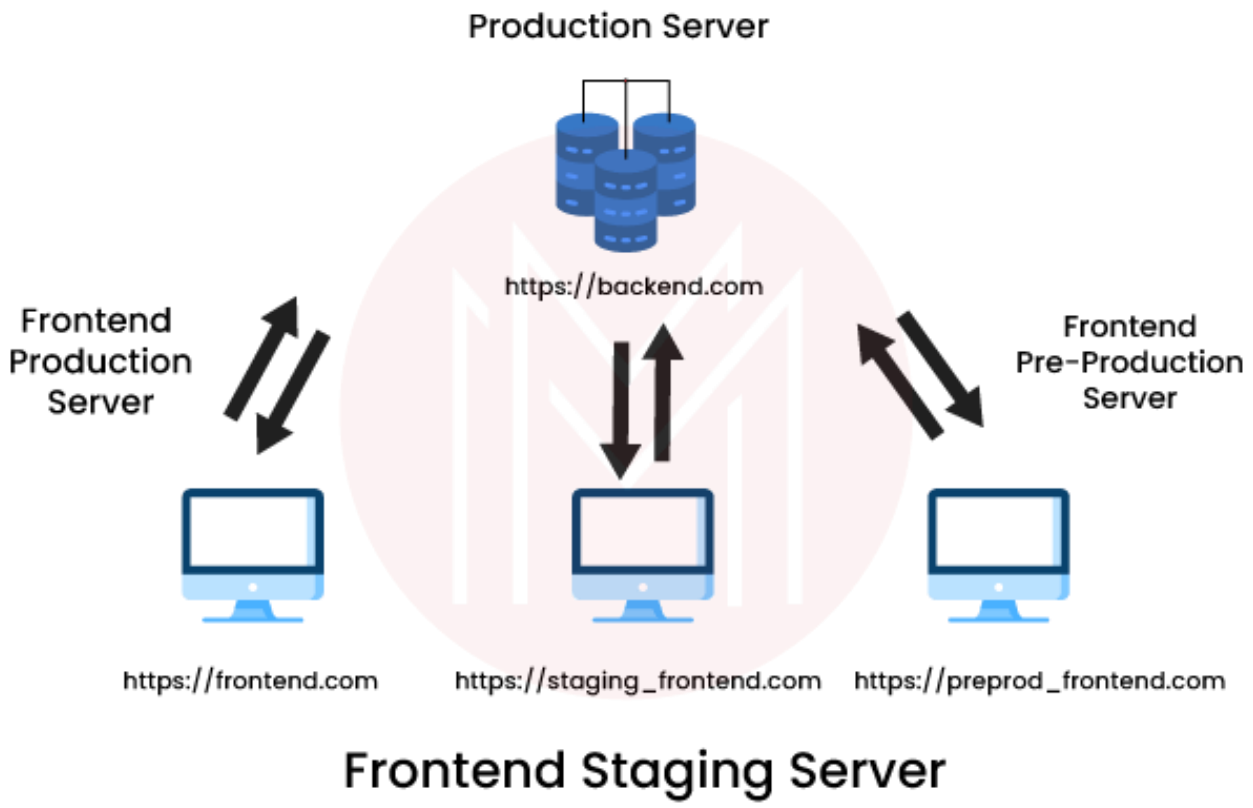
A myriad of NoSQL and RDBMS databases are supported by Express JS, such as:

- MySQL
- MongoDB
- PostgreSQL
- SQLite
- Oracle
- SQLite.

10. What is meant by CORS in Express JS? And what are the ways by which it can be achieved?

CORS is the acronym for Cross-origin resource sharing. We can request the restricted resources from another server or domain using this mechanism. And we can do this mainly in the following three ways:

i. Express cors module
ii. Res.header() (or res.set()): Multiple headers can be set using this way.
iii. Res.setHeader(): Only a single header can be set in this way.



11. What ways are provided by Express JS to configure the properties?

Express JS provides us with two ways for configuring the properties, which are given below:

- With process.ENV
- With require.JS.

12. How are the properties configured with process.ENV?

The properties are configured by the given steps:

- We create a file within the project folder and name it ".env."
- All other properties are let to be separate within the ".env" file.
- We can employ any of the properties in server.js.

13. How are the properties configured with require.JS?

The properties are configured by the given steps:

- We create a file within the config folder of the project folder and name it "config.json."
- All the config properties are present there within the config.json file.

14. How can the Express JS application be structured?

There is no specific answer to this question. The dimensions of our application and hence the concerned team define the solution in different situations. The express logic in the Routes and alternative applications can board as many files as we want in any directory structure. The given examples can be read for further inspiration:

- Route map
- Route listings
- MVC vogue controllers.

15. How is the plain HTML rendered?

We don’t need to render HTML with the function- res.render(). Instead, we can use the res.sendFile() function if we have a specific file. And we can use the Express.static() middleware function if we serve several assets from a directory.

Express JS Interview Questions For Experienced

16. Write the code for "Hello world" using Express.

Create a new file by the name- index.js and type the following commands:

```
var express = require (‘express’);
var app = express ();

app.get (‘/’, function (req, res){
       res.send (“Hello world”);
});

app.listen (3000);
```

Now go to the terminal after saving it and type:

```
nodemon index.js
```

17. What are the most used HTTP methods in Express JS?

The following HTTP methods are the most used ones:

- **GET** - A specified resource's representation is requested by the GET method. These requests can only retrieve data.

- **POST** - Posting of the data enclosed in the request as a new entity is done using the POST method. The entity is identified by the URI.
- **PUT** - Modification in the existing entity is done with the data enclosed in the request identified by the URI.
- **DELETE** - The request for deleting the specified source is made by the DELETE method.

18. How can the cookies be manipulated using 'Response.cookie()'?

We use the "res.cookie('username', 'Flavio')" command is used for manipulating. But it accepts a third parameter containing various options as specified below:

```
res.cookie ('username', 'Flavio', { domain: 'flaviocopes.com', path: '/administrator', secure: true })
res.cookie ('username', 'Flavio' , { expires: new Date(Date.now() + 90000), httpOnly: true})
```

19. When does a Cross-Origin resource get failed in Express JS?

A cross-Origin can fail in the following scenarios-

- If it's to a different domain
- If it's to a different port
- If it's to a different subdomain
- If it's to a different protocol.

20. How can you use a Pug template engine inside Express?

We will first install it using the given command:

```
npm install pug
```

Then we will set it as following when initializing the Express app:

```
const express = require ('express')
const app = express()
app.set ('view engine', 'pug')
```

21. What do you mean by the sanitizing input process?

People can always enter weird things via the client-side code. They use the tools to POST things directly to our endpoints. For this, the Express provides various sanitizing methods to prevent these happenings.

22. Mention some methods for sanitizing.

Consider the following sanitizing methods:

- Trim() will trim the characters at the beginning as well as the ending of a string.
- Escape() will replace ‘, “, <, >, &, / with the corresponding HTML entities.
- NormalizeEmail() will canonicalize an email address.
- Blacklist() will remove the characters appearing on the blacklist.

23. Give an example of HTML form code allowing user to upload a file.

Consider the given example of HTML form code that allows a user to upload file.

```
<form method = “POST” action = “/submit-form”>
  <input type = “file” name = “document” />
  <input type = “submit” />
</form>
```

24. What are the methods that you can call when the Formidable.File objects arise giving the information about the uploaded file?

We can call the following methods in such cases-

- **File.name**- the name of the file
- **File.path**- the path to which the file is written
- **File.size**- the size of the file in bytes
- **File. type**- the file's MIME-type.

25. What steps will you follow to set up HTTP for Express with the help of Let's Encrypt and Certbot?

We will follow the given steps to set up HTTP:

1. Installing certbot
2. Generating the SSL certificate by Certbot
3. Allowing Express to serve the static files
4. Confirming the domain
5. Obtaining the certificate
6. Setting up the renewal.

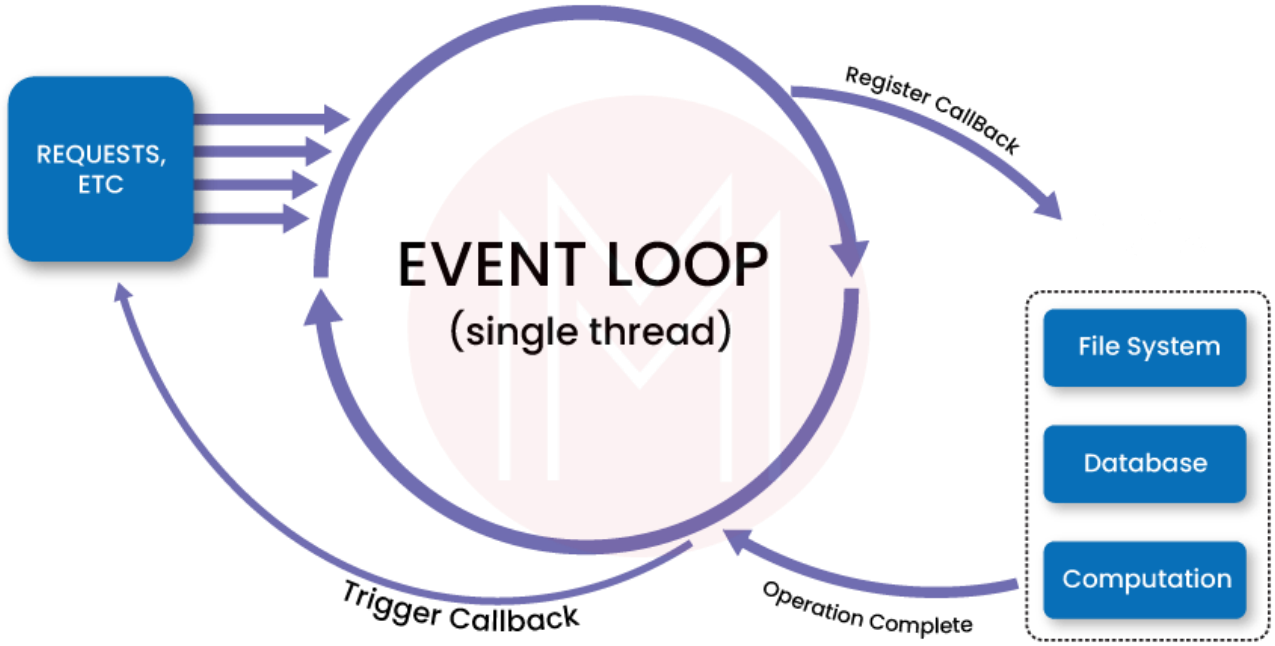
26. What options are available in the terminal command for generating a skeleton Express JS app?

The options available are given below:

- --sessions or –s for adding session report
- --hogan or –H for adding Hogan.js engine support
- --ejs or –e for adding EJS engine support
- --css <engine> or –c <engine> for adding style sheet support
- -jshtml or –J for adding JSHTML engine support
- --force or –f for forcing app generation on the directory which is non-empty.

27. What is meant by an event-loop in Node JS?

The event-loop manages the async content using a listener and queue. The main thread sends the async function to a different thread whenever it requires to be executed. Alongside, v8 is allowed to execute the main code. The event loop has different stages including pending callbacks, timers, check, poll, close callbacks, etc. with different FIFO queues.

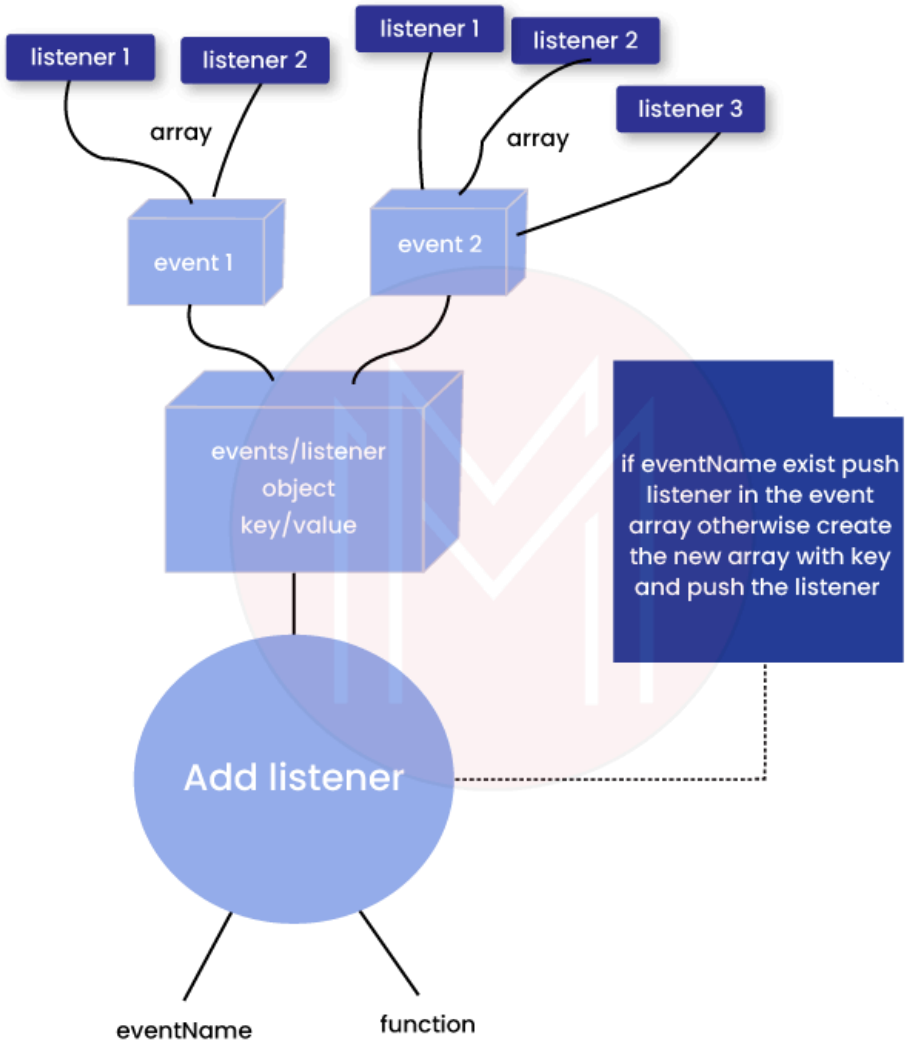


28. Why should the Express server and app be separated?

The server initializes the Middleware, routes, and other application logic. On the other hand, the app contains all the business logic that the server-initiated routes will serve. This enables the encapsulation of the business logic from the application logic for smooth functioning.

29. What is meant by an Event emitter in Node JS?

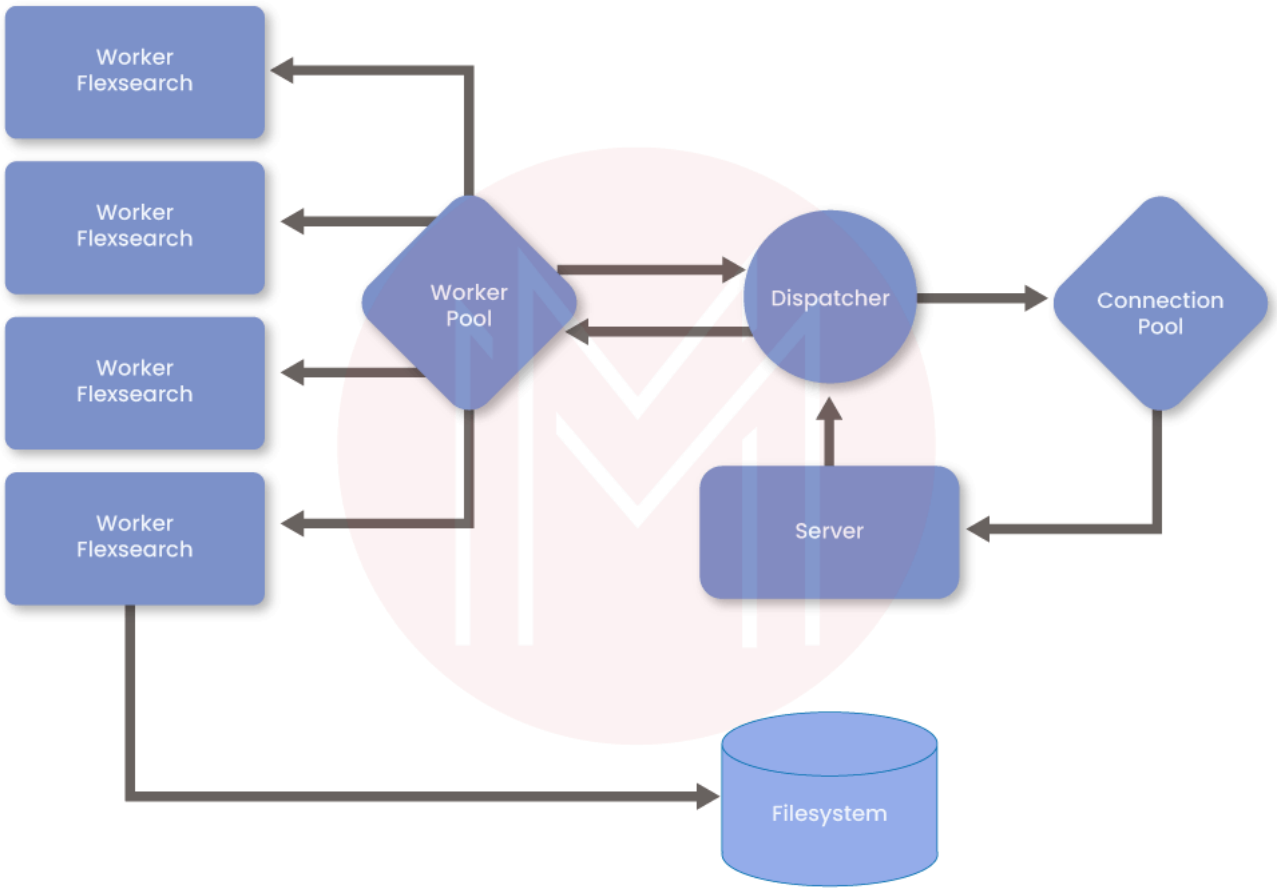
It's a class of Node JS capable of emitting events. We do this by attaching the named events emitted by the object by using the function- eventEmitter.on().



30. Differentiate between worker threads and clusters in Node JS.

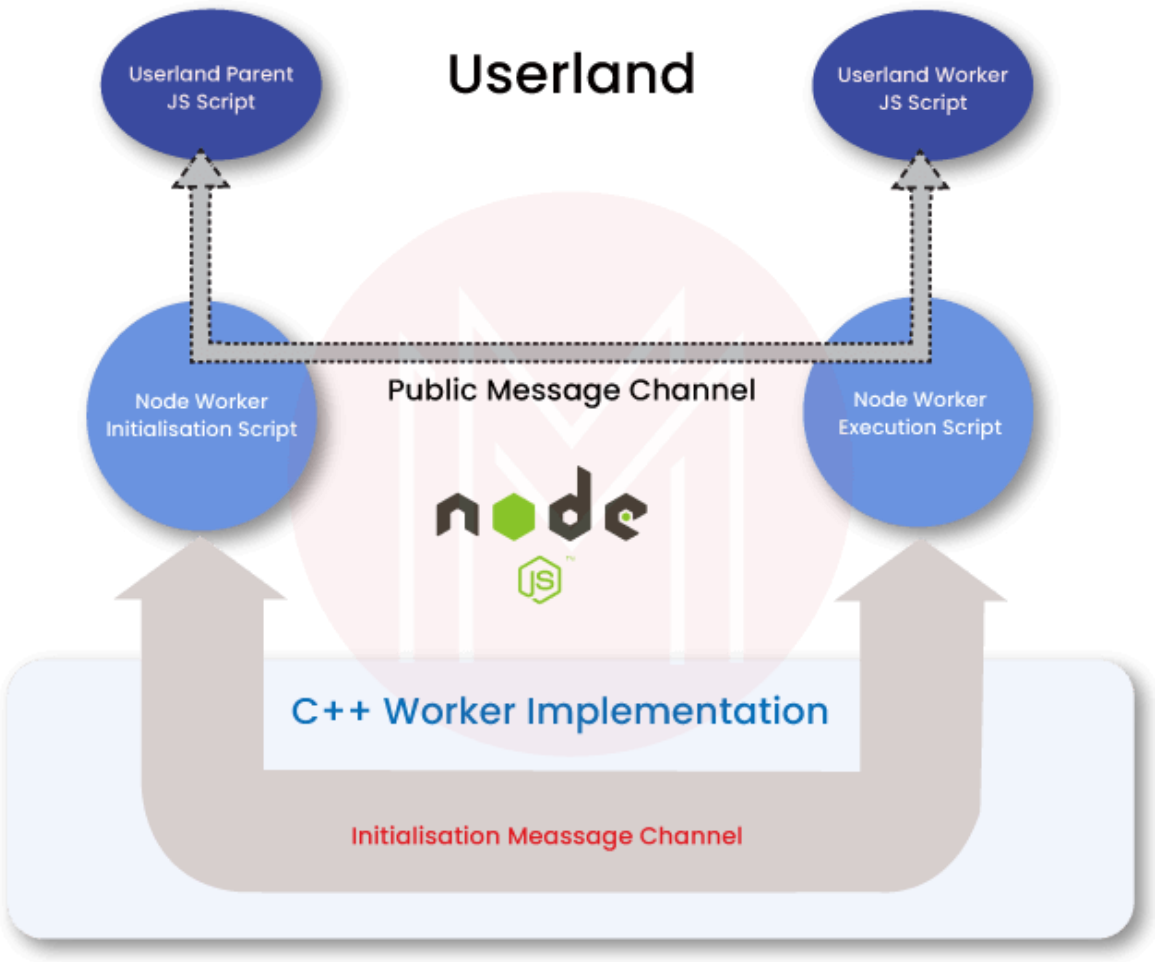
Cluster:

- Each CPU has one process with an IPC to communicate.
- Clusters help when multiple servers are required to accept HTTP requests through a single port.
- The processes have separate memory because of being spanned in different CPUs leading to memory issues.



Worker Threads:

- Only a single process is there with multiple threads.
- Each Node contains one Node having most APIs accessible.
- The memory is shared with other threads.
- We can use this for CPU-intensive tasks.



Let's go through the top 10 Frequently Asked Express JS Interview Questions and Answers (FAQs)

Most Common Express JS FAQs

1. List down major companies that use Express JS.

The following major companies use Express JS-

- Twitter
- Stack
- Accenture
- Intuit
- Bepro Company
- Trustpilot
- BlaBlaCar.

2. What are the popular alternatives to Express JS?

React JS, Meteor, Mean, Flask, Catalyst, Django, Apache Flex, and Laravel are some of the popular alternatives to Express JS.

3. Enlist some distinct features of Express JS.

Some distinct features of Express JS are given below-

- We can design single-page, multi-page, and hybrid web apps as well as APIs with Express JS.
- A routing table is defined for performing HTTP operations.
- Middleware can be set up for responding to RESTful/HTTP requests.
- The MVC-like structure enables organizing the web apps into MVC architecture.
- HTML pages can be dynamically rendered on the basis of passing arguments to templates.
- NoSQL, as well as RDBMS databases, are supported by it.
- High performance is delivered due to its super-fast I/O. the performance is adequate because of the thin layer prepared by it.
- Routing gets easy by its robust API.
- It is single-threaded as well as asynchronous.

4. Which major tools integrate with Express JS?

The following popular tools integrate with Express JS:

- Sentry
- Node JS
- Datadog
- Mean
- Nodemon
- Bugsnag
- LoopBack
- Sails JS.

5. Is Express JS a back-end framework or a front-end framework?

Express JS is a back-end framework built on JavaScript. It is the MEAN stack's back-end component. Here, 'M' refers to MongoDB, and it manages the database. 'E' refers to Express, and it manages the back-end. 'A' refers to AngularJS, and it handles the back-end. And 'N' refers to Node.

6. Differentiate between Node JS and Express JS.

Node JS is an open-source platform on which the JavaScript code is executed outside of a browser. It is used by several companies, including Uber, Walmart, Netflix, etc. It is a platform acting as a web server and not a programming language or framework. On the other hand, Express JS is a framework built on Node JS.

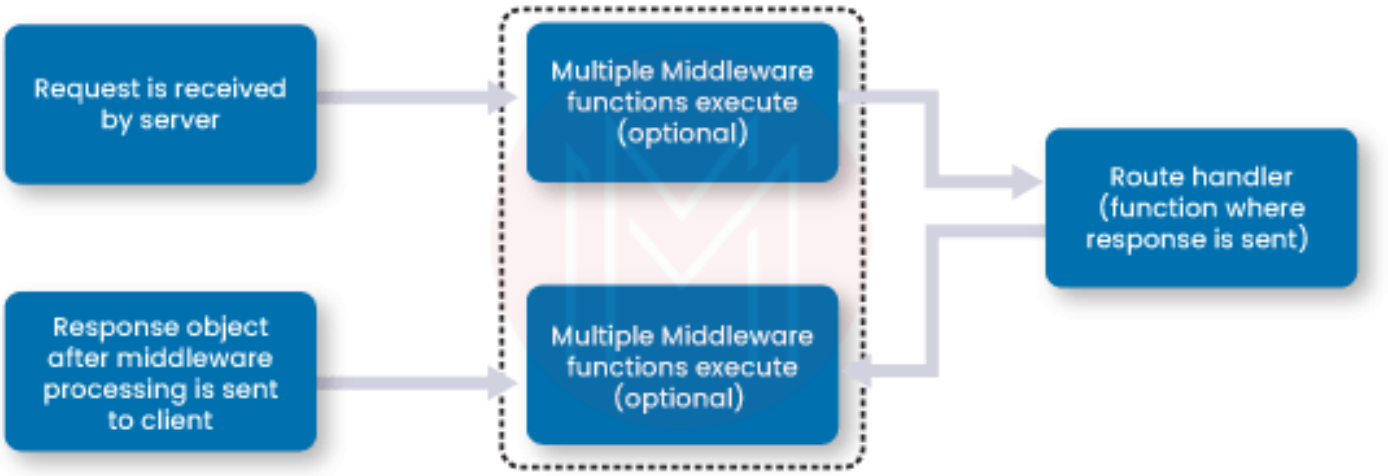
7. Which template engines are supported by Express?

All template engines following the locals, path, and callback signatures are supported by Express JS.

8. What is Middleware, and what are its functions?

Middleware is the function that we invoke before the final request process through the express routing layer. Its functions are given below:

1. Any code like setting headers, validation, etc., can be executed.
2. Changes can be made to the response (res) and request (req) objects.
3. The request-response cycle can also be ended by Middleware.
4. The next middleware function can be called in the stack for proceeding and processing the final request.



9. List the main types of Middleware.

The main types of Middleware are given below:

- Router-level Middleware
- Application-level Middleware
- Built-in Middleware
- Error-handling Middleware
- Third-party Middleware.

10. What is meant by Scaffolding in Express JS? What are the ways to achieve this?

The process of creating the structure of the application is referred to as scaffolding. The two ways of achieving it are given below:

- Express application generator
- Yeoman.

Conclusion

Node JS along with Express JS, allows building a complex and high-level web application from a single line of code easily. That's why it is one of the most sought-after skills in the industry today. And this was a comprehensive list of Express JS Interview Questions that are often asked in interviews. We hope that it will be beneficial for your scheduled interview in the near future.