

```
*****
*****
*****
*****
*****HTML*****
```

1) What is HTML?

HTML is short for HyperText Markup Language and is the language of the World Wide Web. It is the standard text formatting language used for creating and displaying pages on the Web.

HTML documents are made up of two things: the content and the tags that format it for proper display on pages.

2) What are tags?

Content is placed in between HTML tags in order to properly format it. It makes use of the less than symbol (<) and the greater than symbol (>). A slash symbol is also used as a closing tag. For example:

sample

3) Do all HTML tags come in a pair?

No, there are single HTML tags that do not need a closing tag. Examples are the tag and
 tags.

4) What are some of the common lists that can be used when designing a page?

You can insert any or a combination of the following list types:

- ordered list
- unordered list
- definition list
- menu list
- directory list

Each of this list types makes use of a different tag set to compose

5) How do you insert a comment in HTML?

Comments in HTML begins with “<!--” and ends with “-->”. For example:

```
<!-- A SAMPLE COMMENT -->
```

6) Do all character entities display properly on all systems?

No, there are some character entities that cannot be displayed when the operating system that the browser is running on does not support the characters. When that happens, these characters are displayed as boxes.

7) What is an image map?

Image map lets you link to many different web pages using a single image. You can define shapes in images that you want to make part of an image mapping.

8) What is the advantage of collapsing white space?

White spaces are a blank sequence of space characters, which is treated as a single space character in HTML. Because the browser collapses multiple spaces into a single space, you can indent lines of text without worrying about multiple spaces. This enables you to organize the HTML code into a much more readable format.

HTML5 Interview Questions

HTML5 Interview Questions

9) Can attribute values be set to anything or are there specific values that they accept?

Some attribute values can be set to only predefined values. Other attributes can accept any numerical value that represents the number of pixels for a size.

10) How do you insert a copyright symbol on a browser page?

To insert the copyright symbol, you need to type © or & #169; in an HTML file.

11) How do you create links to sections within the same page?

Links can be created using the <a> tag, with referencing through the use of the number (#) symbol. For example, you can have one line as BACK TO TOP, which would result in the words “BACK TO TOP” appearing on the webpage and links to a bookmark named topmost. You then create a separate tag command like somewhere on the top of the same webpage so that the user will be linked to that spot when he clicked on “BACK TO TOP”.

12) Is there any way to keep list elements straight in an HTML file?

By using indents, you can keep the list elements straight. If you indent each subnested list in further than the parent list that contains it, you can at a glance determine the various lists and the elements that it contains.

13) If you see a web address on a magazine, to which web page does it point?

Every web page on the web can have a separate web address. Most of these addresses are relative to the top-most web page. The published web address that appears within magazines typically points this top-most page. From this top level page, you can access all other pages within the website.

14) What is the use of alternative text in image mapping?

When you use image maps, it can easily become confusing and difficult to determine which hotspots correspond to which links. Using alternative text lets you put a descriptive text on each hotspot link.

15) Do older HTML files work on newer browsers?

Yes, older HTML files are compliant to the HTML standard. Most older files work on the newer browsers, though some features may not work.

16) Does a hyperlink apply to text only?

No, hyperlinks can be used in the text as well as images. That means you can convert an image into a link that will allow users to link to another page when clicked. Surround the image within the `...` tag combinations.

17) If the user's operating system does not support the needed character, how can the symbol be represented?

In cases wherein their operating system does not support a particular character, it is still possible to display that character by showing it as an image instead.

18) How do you change the number type in the middle of a list?

The `` tag includes two attributes – type and value. The type attribute can be used to change the numbering type for any list item. The value attribute can change the number index.

19) What are style sheets?

Style sheets enable you to build consistent, transportable, and well-defined style templates. These templates can be linked to several different web pages, making it easy to maintain and change the look and feel of all the web pages within site.

20) State bullet types available in HTML

With ordered lists, you can select to use some different list types including alphabetical and Roman numerals. The type attribute for unordered lists can be set to disc, square, or circle.

21) How do you create multicolored text in a webpage?

To create text with different colors, use the `...` tags for every character that you want to apply color. You can use this tag combination as many times as needed, surrounding a single character or an entire word.

22) Why are there both numerical and named character entity values?

The numerical values are taken from the ASCII values for the various characters, but these can be difficult to remember.

Because of this, named character entity values were created to make it easier for web page designers to use.

23) Write an HTML table tag sequence that outputs the following:

50 pcs 100 500

10 pcs 5 50

Answer:

```
<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>

24) What is the advantage of grouping several checkboxes together?

Although checkboxes don't affect one another, grouping checkboxes together help to organize them. Checkbox buttons can have their name and do not need to belong to a group. A single web page can have many different groups of checkboxes.

25) What will happen if you overlap sets of tags?

If two sets of HTML tags are overlapped, only the first tag will be recognized. You will find this problem when the text does not display properly on the browser screen.

26) What are applets?

Applets are small programs that can be embedded within web pages to perform some specific functionality, such as computations, animations, and information processing. Applets are written using the Java language.

27) What if there is no text between the tags or if a text was omitted by mistake? Will it affect the display of the HTML file?

If there is no text between the tags, then there is nothing to format. Therefore no formatting will appear. Some tags, especially tags without a closing tag like the tag, do not require any text between them.

28) Is it possible to set specific colors for table borders?

You can specify a border color using style sheets, but the colors for a table that does not use style sheets will be the same as the text color.

29) How do you create a link that will connect to another web page when clicked?

To create hyperlinks, or links that connect to another web page, use the href tag. The general format for this is: text

Replace "site" with the actual page URL that is supposed to be linked to when the text is clicked.

30) What other ways can be used to align images and wrap text?

Tables can be used to position text and images. Another useful way to wrap text around an image is to use style sheets.

31) Can a single text link point to two different web pages?

No. The <a> tag can accept only a single href attribute, and it can point to only a single web page.

32) What is the difference between the directory and menu lists and the unordered list?

The key difference is that the directory and menu lists do not include attributes for changing the bullet style.

33) Can you change the color of bullets?

The bullet color is always the same as that of the first character in the list item. If you surround the and the first character with a set of tags with the color attribute set, the bullet color, and the first character will be a different color from the text.

34) What are the limits of the text field size?

The default size for a text field is around 13 characters. However, if you include the size attribute, you can set the size value to be as low as 1. The maximum size value will be determined by the browser width. If the size attribute is set to 0, the size will be set to the default size of 13 characters.

35) Do <th> tags always need to come at the start of a row or column?

Any <tr> tag can be changed to a <th> tag. This causes the text contained within the <th> tag to be displayed as bold in the browser. Although <th> tags are mainly used for headings, they do not need to be used exclusively for headings.

36) What is the relationship between the border and rule attributes?

Default cell borders, with a thickness of 1 pixel, are automatically added between cells if the border attribute is set to a nonzero value. Likewise, If the border attribute is not included, a default 1-pixel border appears if the rules attribute is added to the <table> tag.

37) What is a marquee?

A marquee allows you to put a scrolling text in a web page. To do this, place whatever text you want to appear scrolling within the <marquee> and </marquee> tags.

38) How do you create text on a webpage that will allow you to send an email when clicked?

To change text into a clickable link to send email, use the mailto command within the href tag. The format is as follows:

`text to be clicked`

39) Are `
` tags the only way to separate sections of text?

No. The `
` tag is only one way to separate lines of text. Other tags, like the `<p>` tag and `<blockquote>` tag, also separate sections of text.

40) Are there instances where the text will appear outside of the browser?

By default, the text is wrapped to appear within the browser window. However, if the text is part of a table cell with a defined width, the text could extend beyond the browser window.

41) How are active links different from normal links?

The default color for normal and active links is blue. Some browsers recognize an active link when the mouse cursor is placed over that link; others recognize active links when the link has the focus. Those that don't have a mouse cursor over that link is considered a normal link.

42) Do style sheets limit the number of new style definitions that can be included within the brackets?

Style sheets do not limit the number of style definitions that can be included within the brackets for a given selector. Every new style definition, however, must be separated from the others by a semicolon symbol.

43) Can I specify fractional weight values such as 670 or 973 for font weight?

Implementation largely depends on the browser, but the standard does not support fractional weight values. Acceptable values must end with two zeroes.

44) What is the hierarchy that is being followed when it comes to style sheets?

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.

45) Can several selectors with class names be grouped together?

You can define several selectors with the same style definition by separating them with commas. This same technique also works for selectors with class names.

46) What happens if you open the external CSS file in a browser?

When you try to open the external CSS file in a browser, the browser cannot open the file, because the file has a different extension. The only way to use an external CSS file is to reference it using `<link/>` tag within another HTML document.

47) How do you make a picture into a background image of a web page?

To do this, place a tag code after the `</head>` tag as follows:

```
<body background = "image.gif">
```

Replace image.gif with the name of your image file. This will take the picture and make it the background image of your web page.

48) What happens if the list-style-type property is used on a non-list element like a paragraph?

If the list-style-type property is used on a non-list element like a paragraph, the property will be ignored and do not affect the paragraph.

49) When is it appropriate to use frames?

Frames can make navigating a site much easier. If the main links to the site are located in a frame that appears at the top or along the edge of the browser, the content for those links can be displayed in the remainder of the browser window.

50) What happens if the number of values in the rows or cols attribute doesn't add up to 100 percent?

The browser sizes the frames relative to the total sum of the values. If the cols attribute is set to 100%, 200% the browser displays two vertical frames with the second being twice as big as the first.

51) Which browsers support HTML5?

The latest versions of Google Chrome, Apple Safari, Mozilla Firefox, and Opera all support most of the HTML5 features.

52) Name two new tags included in the HTML 5

`<Video>` and `<Audio>` are new tags which are included in HTML5 version. They are mainly used as a replacement for Flash, Silverlight, and similar technologies to play multimedia items.

53) Do you know which are two semantic tags are included in HTML5 version?

The <article> and <section> tags are two new tags that are included in HTML5. Articles can be composed of multiple sections that can have multiple articles. An article tag represents a full block of content which is a section of a bigger whole.

54) What is <figure> in HTML5?

This tag represents a piece of self-contained flow content. It is mostly used as a single unit as a reference the main flow of the document.

55) What is the use of Canvas element?

The canvas element helps to build charts, graphs, bypass Photoshop to create 2D images and place them directly into HTML5 code.

56) What are the new FORM elements which are available in HTML5?

The new Form elements in HTML5 offers much better functionality than the earlier versions.

The tags given provided to carry out these functions are:

1) <datalist> – This tag is use to specify a list of options for input controls.

2) <keygen> – This tag represents a key-pair generator field.

3) <output> – It represents the result of any scripting calculation.

57) Tell me two benefits of HTML5 Web Storage

Two main benefits of HTML5 Web Storage:

It can store up to 10 MB data which is certainly more than what cookies have.

Web storage data cannot be transferred with the HTTP request. It helps to increase the performance of the application.

58) What are two types of Web Storage in HTML5?

Two storage types of HTML5 are:

Session Storage: It stores data of current session only. It means that the data stored in session storage clear automatically when the browser is closed.

Local Storage: Local storage is another type of HTML5 Web Storage. In local storage, data is not deleted automatically when the current browser window is closed.

59) What is the Application Cache in HTML5 and why it is used?

The Application Cache concept means that a web application is cached. It can be accessible without the need for internet connection.

Some advantages of Application Cache:

Offline browsing – Web users can also use the application when they are offline.

Speed – Cached resources load quicker

Reduce the server load – The web browser will only download updated resources from the server.

60) Explain five new input types provided by HTML5 for forms?

Following are the important, new data types offered by HTML5:

Date: It allows the user to select a date.

datetime-local: This input type allows the user to select a date and time without time zone.

datetime: This input type allows the user to select a date and time with time zone.

month: It enables the user to select a month and year

email: These input fields used to contain an e-mail address.

61) Which HTML tag is used to display the data in the tabular form?

The HTML table tag is used to display data in tabular form (row * column). It also manages the layout of the page, e.g., header section, navigation bar, body content, footer section. Here is the list of tags used while displaying the data in the tabular form:

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.

62) What are some common lists that are used when designing a page?

There are many common lists which are used to design a page. You can choose any or a combination of the following list types:

Ordered list - The ordered list displays elements in numbered format. It is represented by `` tag.

Unordered list - The unordered list displays elements in bulleted format. It is represented by `` tag.

Definition list - The definition list displays elements in definition form like in dictionary. The `<dl>`, `<dt>` and `<dd>` tags are used to define description list.

62) What is the difference between HTML elements and tags?

HTML elements communicate to the browser to render text. When the elements are enclosed by brackets `<>`, they form HTML tags. Most of the time, tags come in a pair and surround content.

63) 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 `<i>` `</i>` tag is used for italic. Instead of these we use ```` and ```` tags.

11) What is an image map?

Image map facilitates you to link many different web pages using a single image. It is represented by `<map>` tag. You can define shapes in images that you want to make part of an image mapping.

12) How to insert a copyright symbol on a browser page?

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

13) How to create a nested webpage in HTML?

The HTML `iframe` tag is used to display a nested webpage. In other words, it represents a webpage within a webpage. The HTML `<iframe>` tag defines an inline frame.

16) What is a style sheet?

A style sheet is used to build a consistent, transportable, and well-designed style template. You can add these templates on several different web pages.

It describes the look and formatting of a document written in markup language.

20) What is a marquee?

Marquee is used to put the scrolling text on a web page. It scrolls the image or text up, down, left or right automatically. You should put the text which you want to scroll within the `<marquee>.....</marquee>`

21) How many tags can be used to separate a section of texts?

Three tags are used to separate the texts.

`
` tag - Usually `
` tag is used to separate the line of text. It breaks the current line and conveys the flow to the next line

`<p>` tag - The `<p>` tag contains the text in the form of a new paragraph.

`<blockquote>` tag - It is used to define a large quoted section. If you have a large quotation, then put the entire text within `<blockquote>.....</blockquote>` tag.

22) How to make a picture of a background image of a web page?

To make a picture a background image on a web page, you should put the following tag code after the </head> tag.

```
<body background = "image.gif">
```

24) What is the use of a span tag? Give one example.

The span tag is used for following things:

For adding color on text

For adding background on text

Highlight any color text

Example:

```
<p>
```

```
<span style="color:#ffffff;">
```

In this page we use span.

```
</span>
```

```
</p>
```

25) What is the use of an iframe tag?

An iframe is used to display a web page within a web page.

Syntax:

```
<iframe src="URL"></iframe>
```

26) What are the entities in HTML?

The HTML character entities are used as a replacement for reserved characters in HTML. You can also replace characters that are not present on your keyboard by entities. These characters are replaced because some characters are reserved in HTML.

27) Why is a URL encoded in HTML?

An URL is encoded to convert non-ASCII characters into a format that can be used over the Internet because a URL is sent over the Internet by using the ASCII character-set only. If a URL contains characters outside the ASCII set, the URL has to be converted. The non-ASCII characters are replaced with a "%" followed by hexadecimal digits.

28) Does a <!DOCTYPE html> tag is a HTML tag?

No, the <!DOCTYPE html> declaration is not an HTML tag. There are many type of HTML e.g. HTML 4.01 Strict, HTML 4.01 Transitional, HTML 4.01 Frameset, XHTML 1.0 Strict, XHTML 1.0 Transitional, XHTML 1.0 Frameset, XHTML 1.1 etc. So, <!DOCTYPE html> is used to instruct the web browser about the HTML page.

30) What is SVG?

HTML SVG is used to describe the two-dimensional vector and vector/raster graphics. SVG images and their behaviors are defined in XML text files. So as XML files, you can create and edit an SVG image with the text editor. It is mostly used 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>
```

33) Which type of video formats are supported by HTML5?

HTML 5 supports three types of video format:

mp4

WebM

Ogg

34) Is audio tag supported in HTML 5?

Yes. It is used to add sound or music files on the web page. There are three supported file formats for HTML 5 audio tag.

mp3

WAV

Ogg

Let's see the code to play mp3 file using HTML audio tag.

```
<audio controls>
```

```
  <source src="koyal.mp3" type="audio/mpeg">
```

Your browser does not support the html audio tag.

```
</audio>
```

35) What is the difference between progress and meter tag?

The progress tag is used to represent the progress of the task only while the meter tag is used to measure data within a given range. More details.

36) What is the use of figure tag in HTML 5?

The figure tag is used to add a photo in the document on the web page. It is used to handle the group of diagrams, photos, code listing with some embedded content.

```
<p>The Taj Mahal is widely recognized as "the jewel of Muslim art in India and one of the universally admired masterpieces of the world's heritage."</p>
```

```
<figure>
```

```
  
```

```
</figure>
```

39) What is the use of details and summary tag?

The details tag is used to specify some additional details on the web page. It can be viewed or hidden on demand. The summary tag is used with details tag. More details.

40) What is datalist tag?

The HTML 5 datalist tag provides an autocomplete feature on the form element. It facilitates users to choose the predefined options to the users to select data.

```
<label>
```

Enter your favorite cricket player: Press any character


```

<input type="text" id="favCktPlayer" list="CktPlayers">
<datalist id="CktPlayers">
<option value="Sachin Tendulkar">
<option value="Brian Lara">
<option value="Jacques Kallis">
<option value="Ricky Ponting">
<option value="Rahul Dravid">
<option value="Shane Warne">
<option value="Rohit Sharma">
<option value="Donald Bradman">
<option value="Saurav Ganguly ">
<option value="AB diVilliers">
<option value="Mahendra Singh Dhoni">
<option value="Adam Gilchrist">
</datalist>
</label>

```

More details.

41) How are tags migrated from HTML4 to HTML5?

No.	Typical HTML4	Typical HTML5
1)	<div id="header">	<header>
2)	<div id="menu">	<nav>
3)	<div id="content">	<section>
4)	<div id="post">	<article>
5)	<div id="footer">	<footer>

Header and Footer Example

HTML 4 Header and Footer:

```

<div id="header">
  <h1>Monday Times</h1>
</div>

```

.

.

.

```
<div id="footer">
```

```
  <p>&copy; JavaTpoint. All rights reserved.</p>
```

```
</div>
```

HTML 5 Header and Footer:

```
<header>
```

```
  <h1>Monday Times</h1>
```

```
</header>
```

.

.

.

```
<footer>
```

```
  <p>© JavaTpoint. All rights reserved.</p>
```

```
</footer>
```

Menu Example

HTML 4 Menu:

```
<div id="menu">
```

```
  <ul>
```

```
    <li>News</li>
```

```
    <li>Sports</li>
```

```
    <li>Weather</li>
```

```
  </ul>
```

```
</div>
```

HTML 5 Menu:

```
<nav>
```

```
  <ul>
```

```
    <li>News</li>
```

```
<li>Sports</li>
<li>Weather</li>
</ul>
</nav>
```

42) If I do not put <!DOCTYPE html> will HTML 5 work?

No, the browser will not be able to identify that it is an HTML document and HTML 5 tags do not function properly..

43) What is the use of the required attribute in HTML5?

It forces a user to fill text on the text field or text area before submitting the form. It is used for form validation.

Example:

Name: <input type="text" name="name" required>

44) What are the new <input> types for form validation in HTML5?

The new input types for form validation are email, URL, number, tel, and date.

Example:

```
<input type="email">
```

[Interview Tips](#) [Job/HR Interview Questions](#)

[JavaScript Interview Questions](#) [jQuery Interview Questions](#)

[Java Basics Interview Questions](#) [Java OOPs Interview Questions](#)

[Servlet Interview Questions](#) [JSP Interview Questions](#)

[Spring Interview Questions](#) [Hibernate Interview Questions](#)

[PL/SQL Interview Questions](#) [SQL Interview Questions](#)

[Oracle Interview Questions](#) [Android Interview Questions](#)

[SQL Server Interview Questions](#) [MySQL Interview Questions](#)

```
*****
*****

*****
*****

*****JavaScript*****
*****
```

JavaScript Interview Questions for Freshers & Beginners

1. What is JavaScript?

JavaScript, also known as JS, is a popular dynamic programming language used for front-end development. The primary purpose of using JS on web pages is to show things in a dynamic manner.

In front-end development, HTML and CSS are used to show static content on a web page. In scenarios where you need to show dynamic content, animation, interactive maps, etc., the role of JavaScript comes into play. Developers call it the third layer, followed by HTML and CSS.

You probably have seen the use of JS if you have gone through websites where additional content gets added without reloading, changing colours on the pages, etc. The use of JavaScript programming is common in website development, and web application development, as well as in servers, browsers, games, etc.

Over the years, it has become an in-demand language, and an increasing number of people are learning it with reliable web development course online or offline.

2. What is JavaScript Promise?

A Promise in JavaScript is an object or proxy that sets up the connection between producing code and consuming code. It shows whether the asynchronous operation is pending, fulfilled, or rejected. And if it is fulfilled, it will also show the result.

3. What is JavaScript closure?

When you need access to the features of a function that is located outside the inner function, the role of JS closure comes into play. It is such an option that helps even when the function located outside is closed.

4. What is JavaScript array?

In JS, an array is a variable that can store multiple values. In simple terms, you can say that an array has the ability to store a set of more than one item with the name of a single variable.

5. What are the characteristics of an array in JavaScript?

Following are the main characteristics of JavaScript array:

- Resizable
- Zero-indexed
- Ability to build shallow copies
- Feature to store collection of different items or data types

6. What is JavaScript function?

A function in JavaScript is a set of code which is used to do a certain task. It can also be used for calculation. Before using the function, ensure that it is defined somewhere.

The function keyword in JS is used to execute the functions.

7. What is JavaScript object?

An object in JS is like a thing in real life, having some properties, methods, behaviour, etc. For example, a bike is an object with properties like color, brand name, model, etc. The methods of this bike can be a brake, stop, start, drive, etc.

8. What is isNaN() function in JavaScript?

The full form of NaN is Not a Number. So, in simple terms, the isNaN() is a function in JavaScript that is used to check whether a value is a number or not once it is converted to a number.

If the value is Not a Number, it will show true. It should be noted that this function first changes the values to numbers and then checks whether it is NaN or not.

9. What is the difference between client-side and server-side JS?

Such JavaScript interview questions can be asked to both freshers and experienced developers. So, let's know the primary differences between client-side scripting and server-side scripting in JavaScript.

Client-Side Server-Side

Users can see the source code Users can't see the source code

It serves the requests of the users and

shows the content accordingly.

It is used to manage the database and other backend aspects.

Dependent on browser No dependency on browsers

Runs on the web browsers Runs on the web servers

Less data security Strong data security

10. Is JavaScript case-sensitive?

Yes. It is a case-sensitive language.

11. What is JavaScript DOM?

DOM stands for Document Object Model. It is an interface in coding that is used to represent the data and content of a web document. It helps in enabling programs to modify the style, content, and structure of a web document.

For instance, you can call the web page a document. If you want to edit this page in JavaScript, then the role of DOM comes into play.

12. What are operators in JavaScript?

An operator in JS is simply a symbol that is utilized to do specific operations, like addition, subtraction, and more. The operations are done on values and variables, or you can call them operands.

Let's understand it with an example. If you want to add two values, then the "+" operator will be used.

10 + 45: // 55

Here, + is an arithmetic operator, which has added the given values. 10 and 45 are the operands here.

13. What are the different operators in JavaScript?

Below is a list of JavaScript operators with their details:

- Assignment Operators

Used when you need to assign values to variables.

Operator Name

= Assignment operator

+= Addition assignment

-= Subtraction assignment

`*=` Multiplication assignment

`/=` Division assignment

`%=` Remainder assignment

`**=` Exponentiation assignment

- Comparison Operators

Role of these operators is when you want to do a comparison of two values and then have a result. The result will be a boolean value, showing true or false.

Operator Name

`==` Equal to

`!=` Not equal to

`===` Strict equal to

`!==` Strict not equal to

`>` Greater than

`>=` Greater than or equal to

`<` Less than

`<=` Less than or equal to

- Arithmetic Operators

Role of these operators is to do arithmetic calculations.

Operator Name

`+` Addition

`-` Subtraction

`*` Multiplication

`/` Division

`%` Remainder

`++` Increment

`--` Decrement

`**` Exponential

- Logical Operators

Used to do logical operators. After the operation, it shows a boolean value (true/false) as the result.

Operator Name

&& Logical AND

| | Logical OR

! Logical NOT

- String Operators

Used for concatenation of strings.

Operator Meaning

typeof To know the type of the variable

instanceof To know whether an object is an instance of
an object type

- Bitwise Operators

Used when you need to do operations binary numbers.

Operator Name

& Bitwise AND

| Bitwise OR

^ Bitwise XOR

~ Bitwise NOT

>>> Zero-fill right shift

>> Right shift

<< Left Shift

14. What is JavaScript event?

Every action that occurs in code or programming is an event. For instance, the loading of a page is an event. Clicks by the end users on a button are an event. The role of events in JavaScript is to make the program understand that some action has been taken and to react accordingly to that action.

15. What is JavaScript callback?

The callback is a function in JavaScript. Its role is to work as an argument by getting passed to other functions. When it is passed to the other functions, those functions will also have the argument executed.

Since it is passed in functions to call back the arguments in more functions as well, its name is justified. It must be noted that you can use callback in other functions only if they are already executed.

16. How to write hello world in JS?

This is one of the basic JavaScript interview questions for freshers. Writing a hello world program in JS has three different ways, as discussed here:

- `console.log()`

```
// write hello world program in JS  
console.log('Hello World');
```

- `alert()`

```
// write hello world program in JS  
alert("Hello, World!");
```

- `document.write()`

```
// writing hello world program in JS  
document.write('Hello, World!');
```

17. What is BOM in JavaScript?

BOM stands for Browser Object Model. This is one of the most crucial parts of JavaScript in terms of using it on the web part. It helps you to know the functionalities of a browser using objects. As a result, the JS-based projects can interact with the browser.

18. What are comments in JavaScript?

JS comments are simply used to represent a comment or message related to the code. For example, we can use comments to mention details of the code, recommendations, warnings, etc. It should be noted that the JS engine will neglect the comments.

19. What is runtime environment in JavaScript?

The runtime environment is the place to run or execute a program. This environment can access the objects, libraries, etc. and hence runs the code accordingly.

Generally, there are two types of JS environments used by a developer. One is the browser-based runtime environment, and another is the NodeJS-based runtime environment.

The browser-based runtime environment makes use of a JS engine, web APIs, an event loop, and the callback queue.

20. What are the advantages of JavaScript?

Following are the primary advantages or benefits of using JavaScript:

- Fast Performance

Since JS is an interpreted language, you don't have to compile it every single time it runs. As a result, it speeds up the development process. Furthermore, it runs on the client side, where it doesn't depend on the server. This again optimizes the performance of the website or web app by avoiding the need for requesting resources from the web server.

- Easy to Understand and Simple

Even if you are a beginner with no knowledge of coding, you will find this language easy to learn. Opting for an offline or online web development course further makes things effortless for you so that you can become well-versed with the complete website and web application development skills.

- Works Well With Other Languages and Systems

JavaScript is an interoperable language which means it runs well with other languages and web technologies.

- Cross-platform Development

There are several JS frameworks that help you to use JavaScript for cross-platform development. For example, you can use React Native framework to use JS for mobile app development. You can use Electron to implement JS for desktop app development.

- Frameworks and Libraries

The wide range of JavaScript libraries and frameworks make it an ideal choice for programmers and developers. The tasks can be performed with minimum efforts, accelerating the development process. In addition, these tools enable plenty of other benefits, like cross-platform development.

21. What is JavaScript used for?

Here are the primary uses and applications of JavaScript:

- Creating Interactive Web Pages

Using JS, developers can make web pages more interactive and appealing. For example, new features can be added that bring some functionalities to the users and also enhance the overall UX.

Here are some examples of JavaScript in use on web pages.

- Hover colours on buttons
- Carousel slider
- Images zoom in/out

- Countdown timer
- Animations
- Hamburger menu
- Web-based Game Development

You can build games using JavaScript that runs on browsers. For example, you can build puzzles, racing games, role-playing games, etc. A few of the popular games running on JavaScript include Angry Birds, CrossCode, Polycraft, HexGL, etc.

- Web App Development

In addition to robust and highly-interactive websites, you can also develop web applications using JavaScript. For instance, Google Maps is one such web app where JS is used.

- Mobile App Development

Using the wide range of JavaScript frameworks available today, you can also develop mobile games. For example, React Native is one popular JS framework for app development.

- Web Servers

Numerous server-side activities can be managed using JavaScript and relevant frameworks.

- User input validation when a user is about to submit a form
- Show pop-ups and dialog boxes
- Change the look of HTML docs
- Setting up dynamic forms
- Show date and time

22. Who developed JavaScript?

Brendon Eich developed JavaScript.

23. What are the disadvantages of JavaScript?

While preparing for the JavaScript interview questions and answers in 2023, don't forget to know about the main drawbacks or disadvantages of JS.

- Non-restrictive

Compared to other languages, JS is a bit more non-restrictive. Developers need to be attentive so that they can implement everything as required. That's why good web development courses cover these aspects of JavaScript so that the developers know the best practices.

- Tricky Debugging

Since JS runs on the client side in the browser, the debugging options and features are limited.

- Security

Again, the client-side running of the language makes it a bit less secure. Developers need to be aware of these aspects of JavaScript so that they don't leave any space available for attackers.

24. What is the difference between JavaScript and Java?

There are several differences between Java and JavaScript. Whether you are a fresher or an experienced professional, this is among the top JavaScript interview questions for you. Below, we have curated a tabular comparison of Java vs JavaScript so that it becomes easier for you to understand the main differences.

Java	JavaScript
Object-oriented programming language	Object-based scripting language
Can be used for complicated tasks and processes	Can't be used for complicated tasks
Needs code compilation	Text-based code
Independent language	Needs to be used with HTML
Strongly typed programming language.	Loosely typed language. No issues whether data types are declared or not
Need to declare variables before using them in the program.	It's statically-type
High memory consumption	Low memory consumption
Saved as byte code	Saved as source code
For concurrency, it uses threads	For concurrency, it uses events
.java extension used to save programs	.js extension used to save programs
Supports multithreading	Doesn't support multithreading
Objects are based on class	Objects are based on prototype
Need JDK or Java Development Kit to run	

the code

Need text editor to run the code

Primarily used for backend development Can use for both front-end and back-end

25. What was the first name of JavaScript?

During the interview, you can be asked what was JavaScript called when it was developed, or what was the former name of JS. You must be prepared for such JavaScript interview questions and answers.

The original name of JavaScript was Mocha when it was developed. It was later changed to LiveScript and then finally became JavaScript.

26. When was JavaScript invented?

It was developed in 1995.

27. What is JavaScript ES6?

ES6 stands for ECMAScript 6. It is a version of JavaScript which was released in 2015.

Hence, it is also called ES 2015.

There are plenty of new features in JavaScript ES6 compared to the prior versions. These are intended to write better and clean code, while helping developers to get more done with less code.

28. What is a named function in JS?

As the term suggests, a named function is simply a function in JS that has been given a name. This name is given with the use of syntax. The function keyword in JavaScript is used to assign a name to the function or make it a named function. This concept should be in the list of top JavaScript interview questions and answers for freshers.

29. What is an anonymous function?

Unlike named functions, anonymous functions are the ones that do not have any names.

Such functions are declared without the use of identifiers.

There are certain rules for using an anonymous function in JavaScript. First, you can't access it right after creating it. For this, you need to use a variable where the anonymous function is saved.

30. What is the difference between JavaScript and JScript?

You need to know the differences between the two as these sound similar. This can be one of the top JavaScript interview questions for freshers.

JavaScript JScript

A trademark of Oracle Corp. Microsoft owns JScript

No support for active content creation Supports active content creation

Can manage multi-browser compatibility Supports Internet Explorer only

Code runs in browser Code runs only in Internet Explorer

Can't access objects of the browser Can access objects of Internet Explorer

Widely used Not popular as it is supported by Internet Explorer only

JavaScript Interview Questions for Experienced

31. Which are the different states of a JS Promise?

JavaScript Promise can have any of the three states:

- rejected: when the operation fails
- pending: when it's neither rejected nor fulfilled
- fulfilled: when the operation is successful

32. What is the difference between `event.preventDefault()` and `event.stopPropagation()` methods in JavaScript?

Among the list of JavaScript interview questions for 3 years experience or above, keep such concepts and learn about them.

- `preventDefault()` method

It is used when you want to prevent the default action of elements in the browser. You must know that not all actions can be cancelled, and is applicable to only cancellable events.

Syntax:

```
event.preventDefault();
```

- `stopPropagation()` method

It is used to stop the propagation of the parent element so that it can not find access to specified events.

Syntax:

```
event.stopPropagation();
```

33. How many ways are there to create an object in JavaScript?

There are three ways to create a JavaScript object:

Ways Syntax

Object literal object={property1:value1,property2:value2..
...propertyN:valueN}

Using new keyword var objectname=new Object();

Using object constructor var objectname=new Object();

34. Which method can we use in order to find out the character
from a particular index?

For this, we can use the string charAt() method.

For example:

```
var str="WsCube Tech";  
document.writeln(str.charAt(3));
```

35. How can we use an external JavaScript file?

For doing so, we need to add the below-mentioned script tag to the header:

```
<script type="text/javascript" src="wscubetech.js"></script>
```

Here, wscubetech.js is the name of the external JS file we are looking to use.

36. What is JavaScript prototype?

A prototype in JS is one of the most useful objects using which developers can inherit one object's features to another. You can say that a prototype is linked to all the objects and functions in JS. That's what it makes capable of inheriting the features of an object.

37. What is the role of JavaScript callback function?

Since numerous actions in JS are asynchronous, these actions are not capable of holding a program from running before completion. Using the callback function, we can define when to run the code.

38. What is the use of window object in JS?

In JavaScript, the window object shows that a window is open in the browser. This object supports all the popular web browsers and therefore the objects, functions, as well as variables are members of the window object.

Moreover, you can say that the global variables of JS are considered properties, whereas the global functions are considered methods of the window.

39. What is the use of history object in JavaScript?

The use of the JS history object is to find the web URLs that a user has visited in the browser. In addition, it also helps in browsing the last and next pages, as well as specific

pages that a user has visited.

As an important property of the window object, the history object is accessible by:

- window.history
- history

40. How to use JavaScript in HTML?

The primary purpose behind using JavaScript on HTML pages is to make them interactive and dynamic.

For adding JS code to HTML code, the `<script>` tag is used. The JS code that you want to add to the HTML is used within the `<script>` tag, whether you want to place it in the `<head>` or `<body>` section.

This method is recommended for small codes. However, for JS scripts that are heavy, it would be great to opt for a dedicated JavaScript file. It helps you to reduce the overall maintenance of the site, differentiates the JS code and HTML code, and optimizes the loading speed.

41. What are the benefits of using JavaScript comments?

If you have been a developer using JS for quite some time now, then this can be among the top JavaScript interview questions for experienced professionals. You must know that there are a couple of good reasons to use comments in JavaScript.

- Adding information about the program or specific lines of code, helping the users to get the scope or use of that code.
- There are several instances when you need to write a few lines of code on a temporary basis. On such occasions, you can add a comment to the code like “to be removed later on”. And when you want to remove it, the comment helps you easily find that code, instead of searching or reading every line of code.

42. What are the different types of JS comments?

Whether you are a beginner or someone with experience, knowing the different types of comments in JavaScript is important. It's because this can be among the top interview questions on JavaScript for freshers and experienced both.

To answer, start by saying that there are two types of comments.

- Single-line comments

These comments are written with two forward slashes (`//`) before your message of the

comment. You can use single-line comments before and after the statement.

Before statement

```
<html>
<body>
<script>
// This is your single-line comment
document.write("Welcome to JavaScript Interview Questions");
</script>
</body>
</html>
```

After statement

```
<html>
<body>
<script>
var x=15;
var y=30;
var z=x+y;
//adding the value of variables x and y
document.write(c);//add 15 and 30
</script>
</body>
</html>
```

- Multi-line comments

Usually the developers prefer multi-line comments more because these can be used to represent single lines and multiple lines of comments both.

For using it, you need to use a forward slash with an asterisk (at the opening of the comment), and an asterisk with a forward slash (at the closing of the comment).

Example:

```
<html>
<body>
<script>
```



```
/* This is your multi-line comment.  
Browser won't display it */  
document.write("Learn Top JavaScript Interview Questions");  
</script>  
</body>  
</html>
```

43. What is null and undefined in JavaScript?

null in JS is an object which is used to represent that a variable doesn't have any value. This is not available by default. The developer needs to do it intentionally.

Whereas, undefined is a global object in JS which means that the declared variable has not been assigned any value.

44. What is void in JavaScript?

As you might know, the meaning of void is empty. In the case of JS, the void is actually an operator with its role when a function doesn't return any result. The result it will return will be undefined.

45. Why do we use strict mode in JS?

The purpose of using strict mode in JavaScript is to create silent errors. It helps in building an expression named 'use strict' so that we can easily enable the strict mode.

Using it, we can modify the normal JS semantics, such as converting the silent errors to throw errors. As a result, silent errors can be avoided. In addition, the performance of the code can be improved in several cases when we use the strict mode.

46. What is JavaScript hoisting?

It is a technique in JS that helps in using a variable, function, or class before the code execution or declaring these things.

What hoisting in JavaScript does is prioritize the variables, classes, and functions by shifting them to the top level. As a result, you can find their values without code execution. It will not show any errors.

In simple words, the role of JS hoisting is to have the feature to call functions, variables, or classes even if you have not defined them in your code.

47. Which are the best JavaScript frameworks?

The top 10 popular frameworks of JavaScript are:

- ReactJS
- AngularJS
- VueJS
- NodeJS
- BackboneJS
- EmberJS
- Meteor
- Polymer
- Mithril
- Aurelia

48. What is an argument object?

An argument object in JavaScript is a set of multiple arguments stored in the form of arrays.

This is used at the time of invoking the functions.

It is an in-built JS object that can be accessed only inside a function. Also, this function needs to have the values of the arguments being passed.

49. How can you use JS to find the operating system of the client machine?

We can make use of the navigator.appVersion string to find the client OS version.

50. How can you use JS to modify the background color of an HTML document?

We can use the following code to do so:

```
<script type="text/javascript">
document.body.bgColor="blue";
</script>
```

In place of "blue", we can use any colour that we intend to use.

51. What is the best way for exception handling in JavaScript?

JS exception handling is performed using try, catch, finally, and throw keywords. Developers can use try or catch blocks to handle exceptions easily.

52. Explain the role of the debugger keyword in JS.

The debugger keyword in JavaScript is used to halt the program's execution at the point you implement it. It helps you to control the code execution and its flow in a manual manner.

53. What is the syntax of the JavaScript anonymous function?

Here is the syntax for an anonymous function in JavaScript:

```
function() {  
  // Function Body  
}
```

54. How can we create a JavaScript function?

The below syntax can be used for creating a function in JS:

```
function function_name(){  
  //function body  
}
```

55. Is it possible to assign an anonymous function to a variable?

Yes. We can do it.

1. What are the differences between Java and JavaScript?

JavaScript is a client-side scripting language and Java is object Oriented Programming language. Both of them are totally different from each other.

JavaScript: It is a light-weighted programming language ("scripting language") for developing interactive web pages. It can insert dynamic text into the HTML elements. JavaScript is also known as the browser's language.

Java: Java is one of the most popular programming languages. It is an object-oriented programming language and has a virtual machine platform that allows you to create compiled programs that run on nearly every platform. Java promised, "Write Once, Run Anywhere".

2. What are JavaScript Data Types?

There are three major Data types in JavaScript.

Primitive

Numbers

Strings

Boolean

Symbol

Trivial

Undefined

Null

Composite

Objects

Functions

Arrays

3. Which symbol is used for comments in JavaScript?

Comments prevent the execution of statements. Comments are ignored while the compiler executes the code. There are two type of symbols to represent comments in JavaScript:

Double slash: It is known as a single-line comment.

// Single line comment

Slash with Asterisk: It is known as a multi-line comment.

/*

Multi-line comments

...

*/

4. What would be the result of $3+2+"7"$?

Here, 3 and 2 behave like an integer, and "7" behaves like a string. So 3 plus 2 will be 5. Then the output will be $5+"7" = 57$.

5. What is the use of the isNaN function?

The number isNaN function determines whether the passed value is NaN (Not a number) and is of the type "Number". In JavaScript, the value NaN is considered a type of number. It returns true if the argument is not a number, else it returns false.

6. Which is faster in JavaScript and ASP script?

JavaScript is faster compared to ASP Script. JavaScript is a client-side scripting language and does not depend on the server to execute. The ASP script is a server-side scripting language always dependable on the server.

7. What is negative infinity?

The negative infinity is a constant value represents the lowest available value. It means that no other number is lesser than this value. It can be generate using a self-made function or by an arithmetic operation. JavaScript shows the NEGATIVE_INFINITY value as -Infinity.

8. Is it possible to break JavaScript Code into several lines?

Yes, it is possible to break the JavaScript code into several lines in a string statement. It can be broken by using the backslash n '\n'.

For example:

```
console.log("A Online Computer Science Portal\n for Geeks")
```

The code-breaking line is avoid by JavaScript which is not preferable.

```
let gfg= 10, GFG = 5,
```

```
Geeks =
```

```
gfg + GFG;
```

9. Which company developed JavaScript?

Netscape developed JavaScript and was created by Brenden Eich in the year of 1995.

10. What are undeclared and undefined variables?

Undefined: It occurs when a variable is declare but not assign any value. Undefined is not a keyword.

Undeclared: It occurs when we try to access any variable which is not initialize or declare earlier using the var or const keyword. If we use 'typeof' operator to get the value of an undeclare variable, we will face the runtime error with the return value as "undefined". The scope of the undeclare variables is always global.

11. Write a JavaScript code for adding new elements dynamically.

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <title>Document</title>
```

```
</head>
```

```
<body>
```

```
  <button onclick="create()">
```

```
    Click Here!
```

```
</button>
```

```
<script>
```

```
function create() {  
    let geeks = document.createElement('geeks');  
    geeks.textContent = "Geeksforgeeks";  
    geeks.setAttribute('class', 'note');  
    document.body.appendChild(geeks);  
}
```

```
</script>
```

```
</body>
```

```
</html>
```

Output:

12. What are global variables? How are these variables declared, and what are the problems associated with them?

In contrast, global variables are the variables that define outside of functions. These variables have a global scope, so they can be used by any function without passing them to the function as parameters.

Example:

```
let petName = "Rocky"; // Global Variable
```

```
myFunction();
```

```
function myFunction() {  
    console.log("Inside myFunction - Type of petName:", typeof petName);  
    console.log("Inside myFunction - petName:", petName);  
}
```

```
console.log("Outside myFunction - Type of petName:", typeof petName);  
console.log("Outside myFunction - petName:", petName);
```

Output

Inside myFunction - Type of petName: string

Inside myFunction - petName: Rocky

Outside myFunction - Type of petName: string

Outside myFunction - petName: Rocky

It is difficult to debug and test the code that relies on global variables.

13. What do you mean by NULL in JavaScript?

The NULL value represents that no value or no object. It is known as empty value/object.

14. How to delete property-specific values?

The delete keyword deletes the whole property and all the values at once like

```
let gfg={Course: "DSA", Duration:30};  
delete gfg.Course;
```

15. What is a prompt box?

The prompt box is a dialog box with an optional message prompting the user to input some text. It is often used if the user wants to input a value before entering a page. It returns a string containing the text entered by the user, or null.

16. What is the 'this' keyword in JavaScript?

Functions in JavaScript are essential objects. Like objects, it can be assign to variables, pass to other functions, and return from functions. And much like objects, they have their own properties. 'this' stores the current execution context of the JavaScript program. Thus, when it use inside a function, the value of 'this' will change depending on how the function is defined, how it is invoked, and the default execution context.

17. Explain the working of timers in JavaScript. Also elucidate the drawbacks of using the timer, if any.

The timer executes some specific code at a specific time or any small amount of code in repetition to do that you need to use the functions setTimeout, setInterval, and clearInterval. If the JavaScript code

sets the timer to 2 minutes and when the times are up then the page displays an alert message "times up". The setTimeout() method calls a function or evaluates an expression after a specified number of milliseconds.

18. What is the difference between ViewState and SessionState?

ViewState: It is specific to a single page in a session.

SessionState: It is user specific that can access all the data on the web pages.

19. How to submit a form using JavaScript?

You can use document.form[0].submit() method to submit the form in JavaScript.

20. Does JavaScript support automatic type conversion?

Yes, JavaScript supports automatic type conversion.

JavaScript Intermediate Interview Questions

21. What are all the looping structures in JavaScript ?

while loop: A while loop is a control flow statement that allows code to be executed repeatedly based on a given Boolean condition. The while loop can be thought of as a repeating if statement.

for loop: A for loop provides a concise way of writing the loop structure. Unlike a while loop, for statement consumes the initialization, condition and increment/decrement in one line thereby providing a shorter, easy to debug structure of looping.

do while: A do-while loop is similar to while loop with the only difference that it checks the condition after executing the statements, and therefore is an example of Exit Control Loop.

22. How can the style/class of an element be changed?

To change the style/class of an element there are two possible ways. We use document.getElementById method

```
document.getElementById("myText").style.fontSize = "16px;
```

```
document.getElementById("myText").className = "class";
```

23. Explain how to read and write a file using JavaScript?

The readFile() functions is used for reading operation.

```
readFile( Path, Options, Callback)
```

The writeFile() functions is used for writing operation.

```
writeFile( Path, Data, Callback)
```


24. What is called Variable typing in JavaScript ?

The variable typing is the type of variable used to store a number and using that same variable to assign a "string".

```
Geeks = 42;
```

```
Geeks = "GeeksforGeeks";
```

25. How to convert the string of any base to integer in JavaScript?

In JavaScript, `parseInt()` function is used to convert the string to an integer. This function returns an integer of base which is specified in second argument of `parseInt()` function. The `parseInt()` function returns `Nan` (not a number) when the string doesn't contain number.

26. Explain how to detect the operating system on the client machine?

To detect the operating system on the client machine, one can simply use `navigator.appVersion` or `navigator.userAgent` property. The `Navigator appVersion` property is a read-only property and it returns the string that represents the version information of the browser.

27. What are the types of Pop up boxes available in JavaScript?

There are three types of pop boxes available in JavaScript.

Alert

Confirm

Prompt

28. What is the difference between an alert box and a confirmation box?

An alert box will display only one button which is the OK button. It is used to inform the user about the agreement has to agree. But a Confirmation box displays two buttons OK and cancel, where the user can decide to agree or not.

29. What is the disadvantage of using `innerHTML` in JavaScript?

There are lots of disadvantages of using the `innerHTML` in JavaScript as the content will replace everywhere. If you use += like `"innerHTML = innerHTML + 'html'"` still the old content is replaced by HTML. It preserves event handlers attached to any DOM elements.

30. What is the use of `void(0)` ?

The void(0) is used to call another method without refreshing the page during the calling time parameter “zero” will be passed.

For further reading, check out our dedicated article on Intermediate Javascript Interview Questions. Inside, you’ll discover over 20 questions with detailed answers.

JavaScript Interview Questions for Experienced

31. What is the ‘Strict’ mode in JavaScript and how can it be enabled?

Strict Mode is a new feature in ECMAScript 5 that allows you to place a program or a function in a “strict” operating context. This strict context prevents certain actions from being taken and throws more exceptions. The statement “use strict” instructs the browser to use the Strict mode, which is a reduced and safer feature set of JavaScript.

32. How to get the status of a CheckBox?

The DOM Input Checkbox Property is used to set or return the checked status of a checkbox field. This property is used to reflect the HTML Checked attribute.

```
document.getElementById("GFG").checked;
```

If the CheckBox is checked then it returns True.

33. How to explain closures in JavaScript and when to use it?

The closure is created when a child functions to keep the environment of the parent’s scope even after the parent’s function has already executed. The Closure is a locally declared variable related to a function. The closure will provide better control over the code when using them.

// Explanation of closure

```
function foo() {  
    let b = 1;  
    function inner() {  
        return b;  
    }  
    return inner;  
}
```

```
let get_func_inner = foo();
```

```
console.log(get_func_inner());
```

```
console.log(get_func_inner());
```

```
console.log(get_func_inner());
```

Output

1

1

1

34. What is the difference between call() and apply() methods ?

Both methods are used in a different situation

call() Method: It calls the method, taking the owner object as argument. The keyword this refers to the 'owner' of the function or the object it belongs to. We can call a method that can be used on different objects.

apply() Method: The apply() method is used to write methods, which can be used on different objects. It is different from the function call() because it takes arguments as an array.

35. How to target a particular frame from a hyperlink in JavaScript ?

This can be done by using the target attribute in the hyperlink. Like

```
<a href="/geeksforgeeks.htm" target="newframe">New Page</a>
```

36. Write the errors shown in JavaScript?

There are three different types of errors in JavaScript.

Syntax error: A syntax error is an error in the syntax of a sequence of characters or tokens that are intended to be written in a particular programming language.

Logical error: It is the most difficult error to be traced as it is the error on the logical part of the coding or logical error is a bug in a program that causes to operate incorrectly and terminate abnormally.

Runtime Error: A runtime error is an error that occurs during the running of the program, also known as an exception.

37. What is the difference between JavaScript and Jscript?

JavaScript

It is a scripting language developed by Netscape.

It is used to design client and server-side applications.

It is completely independent of Java language.

Jscript

It is a scripting language developed by Microsoft.

It is used to design active online content for the word wide Web.

38. What does `var myArray = [[]];` statement declares?

In JavaScript, this statement is used to declare a two-dimensional array.

39. How many ways an HTML element can be accessed in JavaScript code?

There are four possible ways to access HTML elements in JavaScript which are:

`getElementById()` Method: It is used to get the element by its id name.

`getElementsByClass()` Method: It is used to get all the elements that have the given classname.

`getElementsByTagName()` Method: It is used to get all the elements that have the given tag name.

`querySelector()` Method: This function takes CSS style selector and returns the first selected element.

40. What is the difference between `innerHTML` & `innerText`?

The `innerText` property sets or returns the text content as plain text of the specified node, and all its descendants whereas the `innerHTML` property sets or returns the plain text or HTML contents in the elements. Unlike `innerText`, `innerHTML` lets you work with HTML rich text and doesn't automatically encode and decode text.

41. What is an event bubbling in JavaScript?

Consider a situation an element is present inside another element and both of them handle an event. When an event occurs in bubbling, the innermost element handles the event first, then the outer, and so on.

. Write a JavaScript function to calculate the sum of two numbers.

When managers ask this question, they are looking for the candidate's basic understanding of JavaScript. They assess their understanding of basic syntax along with problem-solving skills. This also helps evaluate the candidate's coding style and attention to detail.

I would take two parameters and the following function can be used to calculate the sum of any 2 numbers that are passed as arguments.

```
function sumOfTwoNumbers(a, b) {  
  
    return a + b;  
  
}
```

Sample Answer

2. Write a JavaScript program to find the maximum number in an array.

A hiring manager asks this question to analyze the candidate's ability to write clear and efficient code. It's crucial for candidates to explain the code step-by-step while demonstrating bug-free code.

```
function findMaxNumber(arr) {  
  
    return Math.max(...arr);  
  
}
```

Sample Answer

3. Write a JavaScript function to check if a given string is a palindrome (reads the same forwards and backwards).

The interviewer is looking for the candidate's familiarity with loop constructs, JavaScript string methods, and other basic JavaScript syntax. They will evaluate the candidate's skills based on the approach used to solve the palindrome problem.

```
function isPalindrome(str) {  
  
    return str === str.split("").reverse().join("");  
  
}
```

Sample Answer

4. Write a JavaScript program to reverse a given string.

Hiring managers are expecting an accurate solution that demonstrates the interviewee's proficiency in JavaScript programming.

```
const reverseString = (str) => str.split("").reverse().join("");
```

5. Write a JavaScript function that takes an array of numbers and returns a new array with only the even numbers.

Interviewers are looking for candidates who can not only clearly explain the solution along with the code, but also show the ability to think logically and articulate their thought processes.

By using the filter method on the array, I can check if each element is even or not by using the modulus operator (%) with 2. The element is even if the result is 0. This can be included in the new array.

```
function filterEvenNumbers(numbers) {  
  
    return numbers.filter(num => num % 2 === 0);  
  
}
```

Sample Answer

6. Write a JavaScript program to calculate the factorial of a given number.

By asking this question, managers aim to assess the candidate's algorithmic thinking and understanding of JavaScript programming. The interviewer expects the candidate to demonstrate their knowledge of the factorial concept.

A factorial number is the product of all positive integers, which are equal to or less than the given number.

```
function factorial(number) {  
  
    if (number === 0 || number === 1) {  
  
        return 1;  
  
    } else {  
  
        return number * factorial(number - 1);  
  
    }  
  
}
```

Sample Answer

7. Write a JavaScript function to check if a given number is prime.

Interviewers can analyze the candidate's knowledge of JavaScript algorithms and mathematical concepts. They expect the candidate to translate a mathematical concept into functional code.

To check if a given number is prime, loop from 2 to the square root of the number. If any integer evenly divides it, the number is not prime.

```
function isPrime(num) {  
  
    if (num <= 1) return false;  
  
    for (let i = 2; i <= Math.sqrt(num); i++) {  
  
        if (num % i === 0) return false;  
  
    }  
  
    return true;  
  
}
```

Sample Answer

8. Write a JavaScript program to find the largest element in a nested array.

When asking this question, interviewers are looking for the candidate's ability to handle nested data structures and apply their knowledge of conditional statements, arrays, and loops. Candidates must apply their knowledge to real-world scenarios.

```
function findLargestElement(nestedArray) {  
  
    let largest = nestedArray[0][0];  
  
    for (let arr of nestedArray) {  
  
        for (let num of arr) {  
  
            if (num > largest) {  
  
                largest = num;  
  
            }  
  
        }  
  
    }  
  
}
```



```
    }  
  
    }  
  
    }  
  
    return largest;  
  
}
```

Sample Answer

9. Write a JavaScript function that returns the Fibonacci sequence up to a given number of terms.

This question helps hiring managers assess the interviewee's understanding of fundamental algorithms in JavaScript. They expect the candidate to consider edge cases and handle errors.

```
function fibonacciSequence(numTerms) {  
  
    if (numTerms <= 0) return [];  
  
    if (numTerms === 1) return [0];  
  
  
    let sequence = [0, 1];  
  
    while (sequence.length < numTerms) {  
  
        let nextNumber = sequence[sequence.length - 1] + sequence[sequence.length - 2];
```

```
sequence.push(nextNumber);

}

return sequence;

}
```

Sample Answer

10. Write a JavaScript program to convert a string to title case (capitalize the first letter of each word).

Interviewers analyze the candidate's ability to break down a problem into manageable steps and demonstrate knowledge of string manipulation, looping, and basic JavaScript functions.

```
function toTitleCase(str) {

    return str.replace(/\b\w/g, l => l.toUpperCase());

}
```

Sample Answer

Advanced JavaScript coding interview questions

Advanced JavaScript coding includes various complex concepts and techniques. Such key concepts are often tested in JavaScript interviews. Some of the concepts are – closure and scope, prototypal inheritance, functional programming, design patterns, memory management, ES6+ features, and many more.

1. Implement a debounce function in JavaScript that limits the frequency of a function's execution when it's called repeatedly within a specified time frame.

Interviewers expect the candidate to showcase their ability to clearly explain the purpose of the debounce function and its usage in scenarios where function calls need to be controlled. They are looking for the person's ability to articulate technical concepts clearly.

By delaying the execution of the debounce function until the specified time frame has passed, the frequency can be limited.

```
function debounce(func, delay) {  
  
  let timer;  
  
  return function() {  
  
    clearTimeout(timer);  
  
    timer = setTimeout(func, delay);  
  
  };  
  
}
```

Sample Answer

2. Write a function that takes an array of objects and a key, and returns a new array sorted based on the values of that key in ascending order.

By asking this question, hiring managers analyze how well the candidate can discuss the sorting algorithm and its time complexity. It's also crucial for candidates to demonstrate their code's robustness.

The following function takes an array of objects and a key to sort the array based on the values in ascending order.

```
function sortByKey(arr, key) {  
  
  return arr.sort((a, b) => a[key] - b[key]);  
}
```

```
}
```

Sample Answer

3. Implement a deep clone function in JavaScript that creates a copy of a nested object or array without any reference to the original.

Hiring managers want to assess the interviewee's skill to handle complex coding tasks and understand the concept of avoiding reference issues while cloning.

By using two methods together and creating a deep clone, I can serialize the object to a JSON string. I would then parse it back into a new object, thereby removing any reference to the original object.

```
function deepClone(obj) {  
  
    return JSON.parse(JSON.stringify(obj));  
  
}
```

Sample Answer

4. Write a recursive function to calculate the factorial of a given number.

Interviewers expect the candidate to write a concise recursive function that handles edge cases. Candidates must show their understanding of how recursion works to avoid infinite loops or stack overflow errors.

```
function factorial(num) {  
  
    if (num <= 1) return 1;  
  
    return num * factorial(num - 1);  
  
}
```

Sample Answer

5. Implement a function that takes two sorted arrays and merges them into a single sorted array without using any built-in sorting functions.

When interviewers ask this question, they seek to assess the knowledge of algorithms and efficiency in handling sorted data. They also look for the ability to think of and execute a correct solution.

I can implement a function that can efficiently merge two sorted arrays.

```
function mergeSortedArrays(arr1, arr2) {  
  
    return [...arr1, ...arr2].sort((a, b) => a - b);  
  
}
```

Sample Answer

6. Write a function that checks if a given string is a palindrome, considering only alphanumeric characters and ignoring case.

Interviewers analyze the interviewee's approach to execute code and demonstrate familiarity with handling case-sensitive and alphanumeric checks, regular expressions, and JavaScript string methods.

```
function isPalindrome(str) {  
  
    const cleanStr = str.replace(/[^a-zA-Z0-9]/g, "").toLowerCase();  
  
    const reversedStr = cleanStr.split("").reverse().join("");  
  
    return cleanStr === reversedStr;  
  
}
```

Sample Answer

7. Create a JavaScript class for a linked list with methods to insert a node at the beginning, end, or at a specific position, and to delete a node from a given position.

By asking this question, interviewers can evaluate how well a candidate can design and implement a class for a linked list while also presenting their problem-solving skills.

I would implement a linked list with methods to insert a node at the beginning, end, and at specific positions. Then, I would delete a node from a given position.

Sample Answer

8. Implement a function that flattens a nested array in JavaScript, converting it into a single-level array.

Managers can gauge the candidate's logical thinking skills and capability to handle complex data structures. Interviewees should demonstrate their knowledge of loops, recursion, and arrays.

```
const flattenArray = (nestedArray) => {  
  
  return nestedArray.flat(Infinity);  
  
};
```

Sample Answer

9. Write a function that determines if two strings are anagrams of each other

When interviewers present this question, they aim to measure how well the candidate can use appropriate string-related methods and identify anagrams accurately.

```
function areAnagrams(str1, str2) {  
  
  return str1.split("").sort().join("") === str2.split("").sort().join("");  
}
```

```
}
```

Sample Answer

10. Create a JavaScript function that returns the Fibonacci sequence up to a given number, utilizing memoization for optimized performance.

Interviewees are expected to show their proficiency in OOP and familiarity with recursion and memoization. They can also determine the candidate's attention to detail in class design and organizing code.

By creating a function that uses an array to store the computed values, a Fibonacci sequence can be generated.

```
function fibonacciWithMemoization(n) {
```

```
    let memo = [0, 1];
```

```
    for (let i = 2; i <= n; i++) {
```

```
        memo[i] = memo[i - 1] + memo[i - 2];
```

```
    }
```

```
    return memo;
```

```
}
```

Sample Answer

Common JavaScript coding interview questions

Some of the common JavaScript coding interview questions typically cover these topics: checking for palindrome, finding missing/largest numbers, object manipulation, removing duplicates, merging, etc.

1. Write a function to check if a given string is a palindrome.

Hiring managers review how well a candidate can handle edge cases while handling case sensitivity, punctuation, and whitespace.

This function takes a string as input to convert it into lowercase and then compares it with its reverse. The string can be deemed a palindrome if the two match.

```
function isPalindrome(str) {  
  
    return str.toLowerCase() === str.toLowerCase().split("").reverse().join("");  
  
}
```

Sample Answer

2. Implement a function to reverse a string without using the built-in reverse() method.

Hiring managers aim to analyze the candidate's knowledge of string manipulation in JavaScript while also measuring their ability to think of alternative solutions.

I would use a for loop to iterate through the characters from the end to the beginning. By appending the character to a new string, it results in the reversed output.

```
function reverseString(str) {  
  
    let reversed = "";  
  
    for (let i = str.length - 1; i >= 0; i--) {  
  
        reversed += str[i];  

```



```
}
```

```
return reversed;
```

```
}
```

Sample Answer

3. Given an array of numbers, write a function to find the largest and smallest numbers in the array.

By presenting the candidates with this question, managers can gauge how well a candidate is familiar with basic JavaScript functions and array manipulation.

I would use the following functions to find the smallest and largest numbers in the array.

```
function findMinMax(arr) {
```

```
    let min = Math.min(...arr);
```

```
    let max = Math.max(...arr);
```

```
    return [min, max];
```

```
}
```

Sample Answer

4. Write a function that takes an array of integers as input and returns a new array with only the unique elements.

Hiring managers can evaluate the candidate's knowledge of JavaScript functions, array handling capabilities, and communicating technical concepts.

```
function getUniqueElements(arr) {  
  
    return Array.from(new Set(arr));  
  
}
```

Sample Answer

5. Implement a function to find the factorial of a given number.

Interviewers can determine the candidate's capability to execute functional code and ability to handle input validation and edge cases. Interviewers also assess the ability to use concise and effective code and provide efficient code implementation.

```
function factorial(number) {  
  
    if (number === 0 || number === 1) return 1;  
  
    return number * factorial(number - 1);  
  
}
```

Sample Answer

6. Write a function that determines if a given number is prime or not.

By asking this question, interviewers can understand how good the candidate is proficient in math operations and JavaScript logic. The interviewee should excute a clean and optimized solution that is efficient.

```
function isPrime(num) {  
  
    if (num <= 1) return false;
```

```
for (let i = 2; i <= Math.sqrt(num); i++) {  
  
    if (num % i === 0) return false;  
  
}  
  
return true;  
  
}
```

Sample Answer

7. Implement a function to find the sum of all the numbers in an array.

Such a question helps understand if the interviewee can manipulate arrays and handle numeric values. This also helps managers assess problem-solving capabilities and ability to pay attention to code efficiency.

I would use the reduce method to implement the following function:

```
function findSum(arr) {  
  
    return arr.reduce((sum, num) => sum + num, 0);  
  
}
```

Sample Answer

8. Given a string, write a function to count the occurrences of each character in the string.

Hiring managers expect the candidate to be familiar with string manipulation and loop constructs. When they ask this question, they can evaluate whether the candidate knows data structures.

```
function countCharacterOccurrences(str) {
```

```
const charCount = {};  
  
for (let char of str) {  
  
    charCount[char] = (charCount[char] || 0) + 1;  
  
}  
  
return charCount;  
  
}
```

Sample Answer

9. Implement a function to remove duplicates from an array.

When interviewers present the candidate with this question, they can gauge the level of understanding a candidate has regarding array methods and different approaches to solve the problem.

The following function removes duplicates from an array by converting it into a Set. This automatically removes duplicates. Next, the function converts the Set back into an array.

```
function removeDuplicates(arr) {  
  
    return Array.from(new Set(arr));  
  
}
```

Sample Answer

10. Write a function that sorts an array of numbers in ascending order.

Interviewees must show their knowledge of bubble sort, merge sort, sorting algorithms, and other approaches. The HR manager aims to measure the capability to execute strong algorithms and handle edge cases.

I can solve this by using JavaScript's built-in sort method.

```
function ascendingSort(numbers) {  
  
    return numbers.sort((a, b) => a - b);  
  
}
```

Sample Answer

Tricky JavaScript coding questions

By asking tricky JavaScript coding questions, managers can assess problem—solving skills, JavaScript concepts, and critical thinking. These go beyond syntax knowledge and require the candidate to think creatively and logically to solve problems.

1. Write a function that reverses the order of words in a sentence without using the built-in reverse() method.

This question not only assesses the creativity of the candidates but also helps hiring managers understand how well a candidate can come up with a clean and understandable solution.

```
function reverseSentence(sentence) {  
  
    const words = sentence.split(' ');  
  
    const reversedWords = words.reverse();  
  
    return reversedWords.join(' ');  
  
}
```

Sample Answer

2. Implement a function that checks if a given string is a palindrome (reads the same forwards and backwards) while ignoring whitespace and punctuation.

Interviewers can gauge the interviewee's capability to handle whitespace and punctuation gracefully while also maintaining the palindrome-checking logic. Candidates must express their knowledge of regular expressions or any other efficient approach.

```
function isPalindrome(str) {  
  
    const cleanedStr = str.replace(/[^\w]/g, "").toLowerCase();  
  
    const reversedStr = cleanedStr.split("").reverse().join("");  
  
    return cleanedStr === reversedStr;  
  
}
```

Sample Answer

3. Write a function that takes an array of integers and returns the largest difference between any two numbers in the array.

Candidates should demonstrate their approach to finding the maximum difference between the array elements to handle edge cases and invalid inputs.

```
function largestDifference(arr) {  
  
    let min = arr[0];  
  
    let maxDiff = 0;  
  
    for (let i = 1; i < arr.length; i++) {
```

```
    if (arr[i] < min) {  
  
        min = arr[i];  
  
    }  
  
    else {  
  
        const diff = arr[i] - min;  
  
        if (diff > maxDiff) {  
  
            maxDiff = diff;  
  
        }  
  
    }  
  
    }  
  
    return maxDiff;  
  
}
```

Sample Answer

4. Implement a function that removes duplicates from an array, keeping only the unique elements.

Interviewers can analyze how well a candidate can effectively communicate code explanations and their familiarity with algorithmic efficiency.

```
function removeDuplicates(arr) {  
  
    return arr.filter((item, index) => arr.indexOf(item) === index);  
  
}
```

Sample Answer

5. Write a function that accepts a number and returns its factorial (e.g., factorial of 5 is 5 x 4 x 3 x 2 x 1).

By presenting this question in the interview, hiring managers can assess the capability of the candidate to handle numeric calculations. They can also determine how well the interviewee can pay attention to handling edge cases, if applicable.

```
function factorial(num) {  
  
    if (num === 0 || num === 1) {  
  
        return 1;  
  
    } else {  
  
        return num * factorial(num - 1);  
  
    }  
  
}
```

Sample Answer

6. Implement a function that flattens a nested array into a single-dimensional array.

Interviewers expect the candidates to demonstrate their ability to work with complex data structures and use appropriate techniques to accomplish tasks.

```
function flattenArray(arr) {  
  
    return arr.flat();  
  
}
```

Sample Answer

7. Write a function that checks if a given string is an anagram of another string (contains the same characters in a different order).

Candidates should showcase how well they can handle complex algorithms and logic. Interviewers are specifically looking for knowledge in string methods, data structures, and loop constructs.

```
function isAnagram(str1, str2) {  
  
    const sortedStr1 = str1.split("").sort().join("");  
  
    const sortedStr2 = str2.split("").sort().join("");  
  
    return sortedStr1 === sortedStr2;  
  
}
```

Sample Answer

8. Implement a function that finds the second smallest element in an array of integers.

Interviewers can measure the candidate's problem-solving skills and their understanding of conditional statements, loops, and arrays.

```
function secondSmallest(arr) {  
  
  const sortedArr = arr.sort((a, b) => a - b);  
  
  return sortedArr[1];  
  
}
```

Sample Answer

9. Write a function that generates a random alphanumeric string of a given length.

By asking this question, interviewers can understand how well a candidate can ensure the function produces a reliable and consistent random output.

```
function generateRandomString(length) {  
  
  const characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';  
  
  let result = "";  
  
  for (let i = 0; i < length; i++) {  
  
    const randomIndex = Math.floor(Math.random() * characters.length);  
  
    result += characters.charAt(randomIndex);  
  
  }  
  
  return result;  
  
}
```

Sample Answer

10. Implement a function that converts a number to its Roman numeral representation.

Hiring managers can gauge a candidate's capability to implement coding solutions and create an efficient algorithm.

```
function toRomanNumeral(number) {  
  
    // Implement your code here  
  
}
```

Sample Answer

JavaScript array coding questions

JavaScript array coding interview questions are technical questions asked to gauge candidates' ability to work with arrays along with their familiarity with fundamental data structures.

1. Write a function that returns the sum of all numbers in an array.

By asking such a question, hiring managers can evaluate whether the candidate would be able to perform common tasks and solve basic coding challenges.

```
function sumArray(arr) {  
  
    return arr.reduce((total, num) => total + num, 0);  
  
}
```

Sample Answer

2. Implement a function that finds the maximum number in an array.

Depending on the candidate's answer, the manager can determine how effectively the interviewee can work with arrays. The managers can also understand the capability to communicate technical solutions.

```
function findMaxNumber(arr) {  
  
    let max = arr[0];  
  
    for (let i = 1; i < arr.length; i++) {  
  
        if (arr[i] > max) {  
  
            max = arr[i];  
  
        }  
  
    }  
  
    return max;  
  
}
```

Sample Answer

3. Write a function that returns a new array containing only the unique elements from an input array.

The hiring manager is specifically looking for candidates who can demonstrate an understanding in data manipulation and JavaScript arrays. Additionally, interviewers evaluate how well the candidate strives for an optimized solution without duplicate elements.

```
function getUniqueElements(inputArray) {  
  
    return [...new Set(inputArray)];  
  
}
```

Sample Answer

4. Implement a function that returns the average value of numbers in an array.

By asking this question, hiring managers can assess the candidate's knowledge of arithmetic operations, array manipulation, and looping.

```
function calculateAverage(numbers) {  
  
    let sum = 0;  
  
    for (let number of numbers) {  
  
        sum += number;  
  
    }  
  
    return sum / numbers.length;  
  
}
```

Sample Answer

5. Write a function that sorts an array of strings in alphabetical order.

When interviewers present this question in the interview, they expect the candidate to be familiar with sorting algorithms and JavaScript array manipulation.

```
function sortStrings(arr) {  
  
    return arr.slice().sort();  
  
}
```

Sample Answer

6. Implement a function that finds the index of a specific element in an array. If the element is not found, the function should return -1.

Interviewers aim to gauge the candidate's proficiency in use of array methods, handling edge cases, and in JavaScript syntax. Candidates must implement the function proper error handling.

```
function findElementIndex(arr, element) {  
  
    const index = arr.indexOf(element);  
  
    return index !== -1 ? index : -1;  
  
}
```

Sample Answer

7. Write a function that removes all falsy values (false, null, 0, "", undefined, and NaN) from an array.

Candidates must showcase communication skills and explain their solutions logically. Interviewers analyze the interviewee's ability to write a function that filters false values from an array.

```
function removeFalsyValues(arr) {  
  
    return arr.filter(Boolean);  
  
}
```

```
}
```

Sample Answer

8. Implement a function that merges two arrays into a single array, alternating elements from each array.

Hiring managers determine a candidate's ability to craft efficient algorithms and knowledge of array manipulation.

```
function mergeArrays(array1, array2) {  
  
    const mergedArray = [];  
  
    const maxLength = Math.max(array1.length, array2.length);  
  
    for (let i = 0; i < maxLength; i++) {  
  
        if (i < array1.length) mergedArray.push(array1[i]);  
  
        if (i < array2.length) mergedArray.push(array2[i]);  
  
    }  
  
    return mergedArray;  
  
}
```

Sample Answer

9. Write a function that finds the second largest number in an array.

Such a question reveals to the interviewers how well the candidate can use loops and array methods, work with them, and utilize logic to find solutions.

```
function findSecondLargest(arr) {  
  
    arr.sort((a, b) => b - a);  
  
    return arr[1];  
  
}
```

Sample Answer

10. Implement a function that groups elements in an array based on a given condition. For example, grouping even and odd numbers into separate arrays.

When interviews ask this question, they aim to evaluate a candidate's understanding of concepts like array methods, conditional statements, and other technical concepts. Candidate should demonstrate good coding style.

```
function groupByCondition(arr, condition) {  
  
    return [  
  
        arr.filter(element => condition(element)),  
  
        arr.filter(element => !condition(element))  
  
    ];  
  
}
```

What is typeof operator?

Entry

JavaScript provides a `typeof` operator that can examine a value and tell you what type it is:

```
var a;  
typeof a; // "undefined"  
  
a = "hello world";  
typeof a; // "string"  
  
a = 42;  
typeof a; // "number"  
  
a = true;  
typeof a; // "boolean"  
  
a = null;  
typeof a; // "object" -- weird, bug  
  
a = undefined;  
typeof a; // "undefined"  
  
a = { b: "c" };  
typeof a; // "object"
```

What is equality in JavaScript ?

Entry

JavaScript has both strict and type-converting comparisons:

Strict comparison (e.g., `===`) checks for value equality without allowing coercion

Abstract comparison (e.g. ==) checks for value equality with coercion allowed

```
var a = "42";
```

```
var b = 42;
```

```
a == b;           // true
```

```
a === b;         // false
```

Some simple equality rules:

If either value (aka side) in comparison could be the true or false value, avoid == and use ===.

If either value in comparison could be of these specific values (0, "", or [] -- empty array), avoid == and use ===.

In all other cases, you're safe to use ==. Not only is it safe, but in many cases, it simplifies your code in a way that improves readability.

What is the object type?

Entry

The object type refers to a compound value where you can set properties (named locations) that each holds their own values of any type.

```
var obj = {  
    a: "hello world", // property  
    b: 42,  
    c: true  
};
```

```
obj.a;           // "hello world", accessed with doted notation
```

```
obj.b;           // 42
```

```
obj.c;           // true
```

```
obj["a"];      // "hello world", accessed with bracket notation
obj["b"];      // 42
obj["c"];      // true
```

Bracket notation is also useful if you want to access a property/key but the name is stored in another variable, such as:

```
var obj = {
    a: "hello world",
    b: 42
};
```

```
var b = "a";
```

```
obj[b];        // "hello world"
obj["b"];      // 42
```

Explain is Scope in JavaScript?

Entry

In JavaScript, each function gets its own scope. The scope is basically a collection of variables as well as the rules for how those variables are accessed by name. Only code inside that function can access that function's scoped variables.

A variable name has to be unique within the same scope. A scope can be nested inside another scope. If one scope is nested inside another, code inside the innermost scope can access variables from either scope.

Explain arrays in JavaScript

Entry

An array is an object that holds values (of any type) not particularly in named properties/keys, but rather in numerically indexed positions:

```
var arr = [
    "hello world",
    42,
```

```

        true
    ];

    arr[0];           // "hello world"
    arr[1];           // 42
    arr[2];           // true
    arr.length;       // 3

```

```

typeof arr;          // "object"

```

Explain what a callback function is and provide a simple example

Junior

A callback the function is a function that is passed to another function as an argument and is executed after some operation has been completed. Below is an example of a simple callback function that logs to the console after some operations have been completed.

```

function modifyArray(arr, callback) {
    // do something to arr here
    arr.push(100);
    // then execute the callback function that was passed
    callback();
}

```

```

var arr = [1, 2, 3, 4, 5];

```

```

modifyArray(arr, function() {
    console.log("array has been modified", arr);
});

```

Given a string, reverse each word in the sentence

Junior

```

var string = "Welcome to this Javascript Guide!";

```

```

// Output becomes !ediuG tpircsavaJ siht ot emocleW

```

```

var reverseEntireSentence = reverseBySeparator(string, "");

// Output becomes emocleW ot siht tpircsavaJ !ediuG

var reverseEachWord = reverseBySeparator(reverseEntireSentence, " ");

function reverseBySeparator(string, separator) {
    return string.split(separator).reverse().join(separator);
}

*****
*****
*****
*****
*****React*****
*****

```

ReactJS Interview Questions and Answers for Freshers

1. What is ReactJS?

ReactJS is an open-source, component-based front-end JavaScript library essential for the view layer of the application. It is supported by Facebook.

ReactJS utilizes a virtual DOM-based structure to fill in the information in HTML DOM. The virtual DOM works quickly, possessing a way that it only modifies single DOM components rather than refreshing the complete DOM every time.

The React application comprises several components, each answerable for outputting a small, reusable part of HTML. Components can be embedded inside other components to permit complex applications to work out of basic building blocks.

A component can also support an internal state. For instance, a TabList component can save a variable compared to the directly open tab.

2. What are the benefits of ReactJS?

The main advantages of ReactJS are as follows:

Easy to learn

ReactJS is not a core but a JS library that takes together objects and targets a specific thing

to complete it effectively. It is the view controller in the MVC structure. Any developer who learns JavaScript can interpret React, master its fundamentals, and develop a fantastic app.

Reusable Components

Reusing components is the primary feature of React JS. Even Facebook has executed React as it supports the reuse of framework components. A developer can begin with numerous components like a checkbox, button, etc.

We develop wrapper components made out of those smaller components. Then, we write higher-level wrapper components. It goes on like that until we have this one root component, and that component is our app.

Virtual DOM

When we are about to build a web application with high customer connection and view updates, similar to the new form-maker on JotForm 4.0, we have to consider the available execution problem.

Updating DOM is generally the bottleneck concerning web execution. React attempts to understand this issue by utilizing virtual DOM and a DOM kept in memory.

Flux and Redux architecture

Facebook developed Flux architecture for its various web applications. It is like React components in its unidirectional flow. This structure has action creators that facilitate making action from method parameters. It also supports a library for these methods.

All these actions are supported together by a focal dispatcher to refresh stores. All views are restored according to the stores. There is likewise Redux, which is an upgraded version of flux architecture.

It has an individual store which is not needed in flux. Redux also allows a feature where middleware can be characterized to capture dispatched actions.

SEO-Friendly

SEO is the pillar of a strong business. A superior ranking is equivalent to more commitment from the person, which results in more income.

In addition, ReactJS generally provides a quicker speed by decreasing the time of page load, which is crucial for SEO.

Developer Tools

The React Developer Tools have been described as Chrome and Firefox dev extensions and

permit us to review the React component order in the virtual DOM.

It also allows us to choose specific components and check and alter their current props and state.

Scope for Code Testing

The applications of ReactJS are simple to set. Moreover, it allows developers to test and debug their programs using native tools quickly.

Performance Enhancement

ReactJS improves execution because of virtual DOM. The DOM is a cross-platform programming API that manages with HTML, XML, or XHTML.

Some developers face the issue when updating the DOM, which hinders the application's performance. ReactJS solved this problem by representing virtual DOM.

The React Virtual DOM exists in memory and is a definition of the internet browser's DOM.

When we compose a React element, we do not correspond precisely to the DOM.

Instead, we are comprising virtual elements that React will pass into the DOM, enabling smoother and quicker performance.

3. What is Create React App?

This is one of the frequently asked React interview questions for freshers. Find the answer below:

Create react app is a React app standard generator developed by Facebook. It supports a development setting configured for usability with minimal setup.

Here are some of the most important things to know about "create react app":

- It involves ES6 and JSX transpilation
- Consists of a Dev server with hot module reloading
- Used for Code linting
- Can contain CSS auto-prefixing
- It can develop the script with JS, CSS, picture packaging, and sourcemaps
- Used for the Jest testing system.

4. What are the features of ReactJS?

There are several features of ReactJS as follows:

Virtual DOM

In React, for each DOM object, there is a corresponding "virtual DOM object." A virtual DOM

object is a definition of a DOM object and makes a virtual replica of the initial DOM.

It is a one-way data-binding; therefore, controlling the virtual DOM is faster than updating the initial DOM since nothing gets drawn onscreen.

JSX

JSX represents JavaScript XML. It is a JavaScript syntax extension. It is an XML or HTML like syntax utilized by ReactJS.

This syntax is handled in JavaScript calls of React framework. It broadens the ES6 so HTML-like content can exist together with JavaScript react code. It is not necessary to utilize JSX, but it is endorsed to use it in ReactJS.

Simplicity

ReactJS uses the JSX file, which makes the application simple to program and understand. We learn that ReactJS is a component-based technique that establishes the code reusable as our need. This creates it simple to use and learn.

One-way Data Binding

One-way data binding defines that data flows only in one direction through the entire application. This provides better control over it. The data is moved from the parent component to the child component through read-only props.

These props cannot be sent back to the parent component. This is how one-way data binding works. Although, the child component can interact with the parent component to update the state via callback functions.

React Native

React Native is a custom performed for React, directly comparable to React DOM on the web. It uses native components rather than web components like React as building blocks. It can start with React Native, and we must know the essential React concepts, like JSX, components, state, and props. It also provides an approach to these platform features, from changing the React program to working on iOS and Android.

5. What are the disadvantages of ReactJS?

The disadvantages of ReactJS are as follows:

Dynamic Technology

One of the disadvantages of ReactJS is that it keeps on changing with time. Therefore, it is a constant learning technique for the developers, who have to understand new methods of

doing things that come with uncertain circumstances.

Average Documentation

It is another disadvantage of constantly updating technologies. There is a lack of suitable documentation because of the fast updating of React technologies.

JSX as a barrier

ReactJS utilizes JSX, which is a syntax extension that enables the combining of HTML and JavaScript. This method is helpful, but some development community members treat JSX as a barrier, especially the new developers. In addition, developers argue about the learning curve complexity of JSX.

View Part

Only UI layers of the app get covered by ReactJS. Therefore, choosing a few other technologies is necessary to get a complete collection of tooling for the project's development.

6. How to install ReactJS?

ReactJS is a JavaScript library included in a single file `react-<version>.js` that can be contained in any HTML page. The developers also generally install the React DOM library `react-dom-<version>.js` along with the main React file:

Example

```
<!DOCTYPE html>

<html>

<head></head>

<body>

<script type="text/javascript" src="/path/to/react.js"></script>
<script type="text/javascript" src="/path/to/react-dom.js"></script>
<script type="text/javascript">

// Use react JavaScript program here or in an individual file

</script>

</body>

</html>
```

To obtain the JavaScript files, go to the <https://reactjs.org/docs/getting-started.html> of the official React documentation.

React also provides JSX syntax. JSX is an extension generated by Facebook that inserts XML syntax into JavaScript. It can use JSX we require to contain the Babel library and change `<script type="text/javascript">` to `<script type="text/babel">` for translating JSX to Javascript program.

```
<!DOCTYPE html>

<html>

<head></head>

<body>

<script type="text/javascript" src="/path/to/react.js"></script>

<script type="text/javascript" src="/path/to/react-dom.js"></script>

<script src="https://npmcdn.com/babel-core@5.8.38/browser.min.js"></script>

<script type="text/babel">

// Use react JSX program here or in an individual file

</script>

</body>

</html>
```

Installing ReactJS via npm

We can also install React using npm by doing the following:

```
npm install --save react react-dom
```

To use React in our JavaScript project, we can do the following:

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

var ReactDOM = require('react-dom');

ReactDOM.render(<App/>, ...);
```

Installing React via Yarn

Facebook released its package manager, Yarn, which can also be used to install React.

After installing Yarn, we are just required to run this command:

```
yarn add react react-dom
```

We can then use React in our project precisely as if we had installed React via npm.

7. What is the difference between ReactJS and AngularJS?

This is among the most common ReactJS interview questions. We have curated a simple tabular comparison to help you understand the differences between ReactJS vs AngularJS.

ReactJS AngularJS

Definition ReactJS is an open-source

JavaScript library. It is

used to develop a user

interface for a single-page

application. It is

answerable only for the

view layer of the

application.

AngularJS is an

open-source JavaScript

framework that can

develop a powerful web

application. It is an MVC

framework that works

with the MVC platform,

where it facilitates

development by giving a

dependable solution.

Developed By Facebook Google

Data Binding ReactJS supports one-way

binding. It provides

singular behavior for your

application. One-way

data-binding defines

some changes we create

to the model that

influence the view, but

not the other way around.

In these methods, the

data only flows in one

direction.

AngularJS uses a two-way data binding, which links the Document Object Model (DOM) values to model data. It defines if a new value is supported in the app for user interaction with the field.

It will appear in the update of both the view and the model.

Simplicity React is not simple, and it takes some time to start a project.

Angular is easy and simple to understand. However, its inherent complexity sometimes confuses.

DOM Usage React uses a virtual DOM.

A virtual DOM is a secure version of the DOM. We can modify any element rapidly without needing to render the whole DOM.

Angular uses the browser's DOM.

Application Structure React is not an MVC-like Framework. It is like a view-based library. React

does not demand its client or developer to use some particular application structure.

Angular is a complete-featured MVC Framework. Angular MVC-based structure always permits the application to split into three associated parts so that they can be simply manipulated.

8. What is the difference between ReactJS and React Native?

Here is the ReactJS and React Native comparison to understand the differences between the two:

React JS	React Native
Definition ReactJS is a JavaScript library responsible for developing a hierarchy of user interface components. It is essential for the rendering of user interface components. It offers support for both the front-end and server-side.	React Native is an open-source JavaScript framework. It facilitates a mobile application for iOS

Android, and Windows. It uses only JavaScript to develop a cross-platform mobile app. React Native is similar to React but uses native components instead of web components as building blocks.

Setup and Running

Process

React.js is a JavaScript library that we can use for web development. When beginning a new project and setting up ReactJS, we must choose Webpack, a bundler. We will then determine which wrapping modules will particularly suit our project.

In React Native, we have everything we require to establish and get started.

It is so simple and quick to release the framework. All it takes is to run one command in our terminal, and we are ready to begin programming our

first-ever React Native app. We will require Xcode (for iOS) or Android Studio (for Android) installed on our system to run the app. It uses an emulator of the focus platform or tests its implementation on our tool.

Animations In Reactjs, animation is possible using CSS, just like typical web development.

In React Native, an animated API is used for persuasive animation across multiple elements of the React Native application.

DOM and Styling Virtual DOM is used to deliver browser code in Reactjs.

In React Native, native APIs are used to deliver elements on mobile.

Capabilities React JS was created to keep Search Engine Optimization (SEO) in mind, in which node is utilized for

rendering on the client-server. Though numerous equivalent tools support this view of the server to rendering, they are accessible to insecure hacks. This is separate from an extensive amount of developer support needed for maintenance.

React Native is exclusively dedicated to developing mobile UI. It defines that separated from being centered on UI, Reacts Native also serves as a JavaScript library, not a framework. Hence the UI built is hugely responsive and adds a smooth feel to our mobile application and quicker load times.

9. Which browsers does ReactJS support?

It supports all the popular and most used web browsers like Google Chrome, Firefox, Mozilla, Microsoft Edge, etc. It doesn't support browsers built without ES5 methods, like Internet Explorer.

10. What is JSX in React?

JSX is a preprocessor phase that inserts XML syntax into JavaScript. We can use React without JSX, yet JSX makes React a lot more classic.

It is just a similar XML. JSX tags contain a tag name, attributes, and children. If an

attribute value is placed within quotes, the value is a string. Alternatively, wrap the value in braces, and the value is the confined JavaScript expression.

JSX gives syntactic sugar to `React.createElement(component, props, ...children)` function.

11. What are the benefits of using JSX in React?

The advantages of JSX are as follows:

- It is always quick as it implements optimization while assembling a program to vanilla JavaScript.
- JSX is additionally type-safe, which defines it as carefully composed, and most of the errors can be found during the compilation of the JSX code to JavaScript.
- It consistently makes writing templates simpler and quicker if we are simple with HTML syntax.

12. What is the difference between DOM and Virtual DOM in ReactJS?

Another common question asked in the ReactJS job interviews is about the comparison between DOM and VDOM. Here is the answer:

DOM

DOM represents the Document Object Model. It is also known as HTML DOM, an abstraction of a structured code known as HTML for web developers. DOM and HTML code are associated, as the elements of HTML are called nodes of DOM.

It describes a structure where users can create, alter, modify documents, and present the content. Therefore while HTML is text, DOM is an in-memory definition of this content.

Virtual DOM (VDOM)

VDOM in ReactJS is a method that syncs the virtual user interface with the real document object model. It is not a dedicated technology but a pattern that helps in handling the events, updating manual DOM, as well as manipulating the attributes.

13. What is the use of keys in ReactJS?

Keys recognize unique virtual DOM elements with their related data driving the UI. It supports React to develop rendering by recycling existing DOM components. Keys

must be unique numbers or strings. Rather than re-rendering with keys, React only re-orders the components. It enhances the application's execution.

14. What is Relay in ReactJS?

Relay is a JavaScript framework supporting an information layer and user-server connection to web applications utilizing the React view layer.

15. What is the difference between createElement and cloneElement?

JSX components are transpiled to React.createElement() functions to make React elements that will be used for the object definition of UI.

Whereas, cloneElement is used to clone an element and pass it new props.

16. How to create components in React?

It is one of the frequently asked React interview questions for freshers. You must know that there are two approaches to creating a component in ReactJS:

Functional Components

It is the simplest method of creating the React component. These pure JavaScript functions obtain the props object as the first argument and return the react component.

```
function Hello({ message }) {  
  return <h1>{`Hello, ${message}`}</h1>  
}
```

Class components

We can use ES6 class to create a component. The function component can be written as:

```
class App extends React.Component {  
  render() {  
    return <h2>{`Hii, ${this.props.message}`}</h2>  
  }  
}
```

17. What is a React State?

States are the soul of React elements. These are the source of information and should be maintained as simply as feasible.

ReactJS states are the objects which decide the component's rendering and behavior. They are mutable, unlike the props, and generate powerful and interactive elements. They are accessed through `this.state()`.

18. What are React Props?

Props are inputs to components. They are read-only components that should be kept immutable. ReactJS Props are used to pass information and methods from a parent component to a child component.

19. What are Default Props?

In ReactJS, `defaultProps` authorize us to set default, or fallback, values for our component props. `defaultProps` are helpful when we call components from various views with fixed props, but in some views, we are required to pass several values.

Example of React Default Props

```
class MyApp extends React.Component {...}  
MyApp.defaultProps = {  
  randomObject: {},  
  ...  
}
```

20. What is the difference between state and props in ReactJS?

Here is the tabular comparison between React state and props:

State	Props
The state is mutable.	Props are immutable.
The state influence data about the components.	Props enable us to pass data from one component to another as an argument.
	They do not allow to generate reusable components.
	They allow to make reusable components.
The State is internal and managed by the component itself.	

Props are external and managed by
whatever renders the component.

The child component cannot access states. The child component can access props.

It can be used for rendering dynamic
changes with the component.

It is used to connect components.

21. What is `setState()` in ReactJS?

The primary method using which we create UI updates to our React applications is through a call to the `setState()` function. This function will execute a shallow merge between the new state that we give and the previous state and trigger a re-render of our component and all decedents.

22. What are the Parameters of `setState()`?

There are two parameters of `setstate()` as follows:

1. Updater: It can be an object with several key-value pairs that should be combined into the state or a function that restores such an object.
2. Call-back (optional): The function implemented after `setState()` is implemented effectively. Because calls to `setState()` are not ensured by React to be atomic, this can be beneficial if we need to implement some action after we are positive that `setState()` has been performed effectively.

24. What is the arrow function in React?

An arrow function is also known as the 'fat arrow function (`=>`).'. It enables binding the context of components properly because auto-binding is not possible by default in ES6. In addition, it creates it easier to work with higher-order components.

Example

1. Without using Arrow functions

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

2. Using Arrow Functions

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

```
}
```

25. What is the context in ReactJS?

The context supports a method to pass information through the component tree without giving props down manually at each level.

For example, authenticated customers, locale preference, and UI theme should be acquired in the application by many components.

```
const {Provider, User} = React.createContext(defaultValue)
```

26. What are React Mixins?

Mixins are an approach to separate elements from having standard functionality. It should not be used and can be replaced with higher-order components or decorators.

27. What is the context.consumer in React?

A consumer is a React element that subscribes to text changes. It needed a function as a child that accepts the current context value as a parameter and returns a react node. The value parameter passed to the function will be similar to the nearest provider's value props for this context above in the tree.

Example

```
<MyContext.Consumer>
```

```
{value => /* render something depend on the context value */}
```

```
</MyContext.Consumer>
```

28. What is React Fiber?

Fiber is the new reconciliation engine or re-executing the basic algorithm in React v16. The objective of React Fiber is to expand its appropriateness for fields like animation, design, gestures, and the ability to pause, abort, or reuse work and assign priority to multiple types of updates; and new concurrency primitives.

29. What is React reconciliation?

When a component's props or state changes, React determines whether an actual DOM update is essential by contrasting the recently returned element with the previously rendered one.

When they are not similar, they will update the DOM. This process is known as reconciliation in ReactJS.

30. What are Pure Components?

Pure components are the easiest and quickest components that can be written. They can restore any component which has a `render()`. These components upgrade the program's simplicity and the application's execution.

Advanced ReactJS Interview Questions for Experienced

If you have worked in this field for some years, then our ReactJS interview questions for senior developer or experienced professionals (1-5 years) are as follows:

1. Why use JSX in React?

There are a number of important reasons behind using JSX:

- It is faster than regular JavaScript because it implements optimization while translating the program to JavaScript.
- JSX can separate innovations by placing markup and logic in separate documents, React uses components that include both.
- It is type-safe, and some bugs can be found at compilation time.
- It makes it simpler to develop templates.

2. Why a web directory can't read JSX?

Web directory cannot read JSX directly because they can only read JavaScript objects, and JSX is not an ordinary JavaScript object. Accordingly, we must change the JSX document into a JavaScript object using transpilers like Babel and then pass it to the browser.

3. What is DOM diffing?

When the elements are rendered twice, Virtual DOM starts checking the changes elements have got. They define the changed component on the page. Several other components don't go through changes. It can decrease the DOM changes due to client activities, and treat DOM doffing. It is generally done to boost the execution of the browser. This is the reason for its capability to execute all the functions rapidly.

4. What is the role of `render()` in React?

Every React component should have a `render()` necessarily. It returns a single React component, which describes the native DOM component. If more than one HTML element should be rendered, then they should be arranged inside one enclosing tag including `<form>`, `<group>`, `<div>`, etc.

This function should remain pure, i.e., it must return a similar outcome each time it is

invoked.

5. What is the difference between elements and components in ReactJS?

Here is the tabular comparison of React elements and components:

Elements	Components
----------	------------

An element is a plain JavaScript object representing the component state, DOM node, and desired properties.	
---	--

A component is the core building structure of the React application. It is a class or function which takes an input and returns a React element.	
--	--

It only holds data about the component type, its properties, and any child elements inside it.	
--	--

It can contain state and props and access the React lifecycle methods.	
--	--

It is immutable. It is mutable.	
---------------------------------	--

Example:

```
const element = React.createElement(  
'div',  
{id: 'login-btn'},
```

Example:

```
function Button ({ onLogin }) {  
  return React.createElement(  
'div',  
{id: 'login-btn', onClick: onLogin},  
'Login'  
)  
'Login'  
}
```

}

6. What are Stateless Functional Components?

In numerous applications, smart components occupy state but render dumb components that receive props and return HTML as JSX. Stateless functional components are substantially more reusable and have a positive execution impact on our application.

They have two main characteristics:

- When rendered, they get an object with all the props passed down.
- They should restore the JSX to be rendered.

7. What are Stateful Components in ReactJS?

If the component's behavior is dependent on the state of the component, it tends to be defined as a stateful component. These stateful components are continually class components and have a state that gets initialized in the constructor.

8. What is the difference between Stateful and Stateless Components?

We have compared the stateful and stateless components below:

Stateful Component	Stateless Component
--------------------	---------------------

It stores info about a component's state change in memory.	
--	--

It evaluates the internal state of the components.	
--	--

It includes information on past, current, and likely future changes in state.	
---	--

It contains no information on history, present, or potential future state changes.	
--	--

It is also called a class component. It is also called a functional component.	
--	--

It can work with all the lifecycle methods of React.	
--	--

It cannot work with any lifecycle method of React.	
--	--

9. When to use the class component over a function component?

If the component requires state or lifecycle methods, then use class components; otherwise, use function components.

From React 16.8, with the inclusion of hooks, we can use state, lifecycle methods, and other characteristics that were only feasible in a class component right in our function component.

10. What is the differences between `setState()` and `replaceState()` methods?

When we use `setState()`, the current and previous states are joined. `replaceState()` throws out the current state and restores it with only what we provide.

`setState()` is used except if we require to delete all previous keys for some reason. We can also set state to false/null in `setState()` rather than using `replaceState()`.

11. What is Prop Drilling?

When developing a React application, a deeply nested element often needs to use data provided by another element that is much higher in the hierarchy. The simplest method is passing a prop from each component to the next in the authority from the source element to the deeply nested element. This is called prop drilling.

12. What is the difference between `componentWillMount()` and `componentDidMount()`?

This is one of the most common React interview questions for senior developer. You must know the differences between the two, as mentioned below:

`componentWillMount()` `componentDidMount()`

This is called only once through the component lifecycle before the element is rendered to the server. It is used to determine props and do any excess logic based on them.

It is called only on the user end. It is generally performed after the initial render when a user has received data

from the server.

It is used to execute state changes before the initial render.

It enables us to implement all kinds of advanced interactions, including state changes.

It is invoked just directly after the mounting has occurred.

Putting the data loading code in this only ensures that data is fetched from the client's end.

Users should avoid using async initialization to minimize side effects or subscriptions in this method.

The data does not store until the initial render is done. Hence, users must set an initial state properly to eradicate undefined state errors.

13. How to bind approaches or event handlers in JSX callbacks?

There are three possible approaches to manage this:

Binding in Constructor

The methods are not constrained by default in JavaScript classes. A similar thing relates to react event handlers represented as class methods. It can generally bind them in the constructor.

```
class Component extends React.Component {  
  constructor(props) {  
    super(props)  
    this.handleClick = this.handleClick.bind(this)  
  }  
  handleClick() {
```

```
// ...
```

```
}
```

```
}
```

Public class fields syntax

If we don't like to use the bind method, then public class fields syntax can be used to bind callbacks accurately.

```
handleClick = () => {
```

```
  console.log('this is:', this)
```

```
}
```

```
<button onClick={this.handleClick}>
```

```
  {'Click me'}
```

```
</button>
```

Arrow functions in callbacks

We can use arrow functions precisely in the callbacks.

```
<button onClick={(event) => this.handleClick(event)}>
```

```
  {'Click me'}
```

```
</button>
```

14. How are Error Boundaries handled in React V15?

React version 15 supports the error boundaries by utilizing the `unstable_handleError` method. It is termed to `componentDidCatch` in React Version 16.

15. What is the aim of `getDerivedStateFromProps()` lifecycle method?

A new static `getDerivedStateFromProps()` lifecycle method is invoked after an element is instantiated as well as before it is re-rendered. It can return an object to refresh the state or nullify that the new props do not need state updates.

```
class MyApp extends React.Component {
```

```
  static getDerivedStateFromProps(props, state) {
```

```
    // ...
```

```
  }
```

```
}
```

16. What is render prop in React?

Render prop is a simple method for sharing programs between components using a prop whose value is a function. The following component uses render prop, which returns a React element.

```
<DataProvider render={data => (  
<h1>{'Hello ${data.target}'}</h1>  
)}>
```

17. What is the lifecycle method of React Component?

Lifecycle methods are used to run programs and interact with our components at various points in the component's life. These methods are based on a component, Mounting, Updating, and Unmounting.

Component Creation

When a React component is created, several functions are called:

- If we are using React.createClass (ES5), 5 user-defined functions are called.
- If we are using class Component extends React.Component (ES6), 3 user-defined functions are called.

getDefaultProps() (ES5 only)

This is the first method called.

Prop values restored by this function will be used as defaults if they are not described when the component is instantiated.

getInitialState() (ES5 only)

This is the second method called.

The return value of getInitialState() describe the initial state of the React component.

The React core will call this function and select the restore value to this.state.

Example

this.state.count will be initialized with the value of this.props.initialCount:

```
getInitialState() {  
  return {  
    count : this.props.initialCount  
  };  
}
```

componentWillMount() (ES5 and ES6)

This is the third method called.

This function can make final changes to the component before being added to the DOM.

```
componentWillMount() {  
  ...  
}
```

render() (ES5 and ES6)

This is the fourth method called.

The render() function must be a pure function of the component's state and props. It returns a single element that defines the component during the rendering process and should either be a description of a native DOM component (e.g., <p/>) or a composite component. If nothing should be performed, it can return invalid or undefined.

This function will be recalled after the component's props or state changes.

```
render() {  
  return (  
    <div>  
      Hello, {this.props.name}!  
    </div>  
  );  
}
```

componentDidMount() (ES5 and ES6)

This is the fifth method called.

The component has been mounted, and we can create the component's DOM nodes, e.g., via refs.

This method should be used for:

- Preparing timers
- Fetching information
- Adding event listeners
- Manipulating DOM component

```
componentDidMount() {
```

...

}

ES6 Syntax

If the component is represented using ES6 class syntax, the functions `getDefaultProps()` and `getInitialState()` cannot be used.

Rather than we declare our `defaultProps` as a static property on the class and declare the state shape and original state in the constructor of our class. These are both set on the class instance at construction time before any other React life cycle function is called.

The following example shows this alternative approach:

```
class MyReactClass extends React.Component {  
  constructor(props){  
    super(props);  
    this.state = {  
      count: this.props.initialCount  
    };  
  }  
  upCount() {  
    this.setState((prevState) => ({  
      count: prevState.count + 1  
    }));  
  }  
  render() {  
    return (  
      <div>  
        Hello, {this.props.name}!<br />  
        You clicked the button {this.state.count} times.<br />  
        <button onClick={this.upCount}>Click here!</button>  
      </div>  
    );  
  }  
}
```

```
}  
  
MyReactClass.defaultProps = {  
  name: 'Bob',  
  initialCount: 0  
};
```

Replacing getDefaultProps()

Default values for the component props are stated by framework the defaultProps property of the class:

```
MyReactClass.defaultProps = {  
  name: 'Bob',  
  initialCount: 0  
};
```

Replacing getInitialState()

The idiomatic way to set up the component's initial state is to set this. state in the constructor:

```
constructor(props){  
  super(props);  
  this.state = {  
    count: this.props.initialCount  
  };  
}
```

Component Removal

This method is known before a component is unmounted from the DOM. This phase includes only one method and is given below.

componentWillUnmount()

This method is invoked directly before a component is destroyed and unmounted permanently. It implements any necessary cleanup related function, including invalidating timers, and event listeners, canceling network requests, or cleaning up the DOM component. If a component instance is unmounted, we cannot mount it again.

Example

```
import React, { Component } from 'react';

export default class SideMenu extends Component {

  constructor(props) {
    super(props);

    this.state = {

      ...

    };

    this.openMenu = this.openMenu.bind(this);
    this.closeMenu = this.closeMenu.bind(this);
  }

  componentDidMount() {
    document.addEventListener("click", this.closeMenu);
  }

  componentWillUnmount() {
    document.removeEventListener("click", this.closeMenu);
  }

  openMenu() {
    ...
  }

  closeMenu() {
    ...
  }

  render() {
    return (
      <div>

        <a

          href = "javascript:void(0)"

          className = "closebtn"

          onClick = {this.closeMenu}

        >

          ×
```



```
</a>
```

```
<div>
```

Some other structure

```
</div>
```

```
</div>
```

```
);
```

```
}
```

```
}
```

Component Update

It is the next stage of the lifecycle of a react component. This stage allows managing user interaction and support interaction with the components hierarchy. In addition, this stage aims to ensure that the component is showing the latest version of itself.

`componentWillReceiveProps(nextProps)`

This is the function called on properties changes. When the component's properties change, it will call this function with the new properties. We can access the old props with `this.props` and the new props with `nextProps`.

With these variables, we can do some comparison operations between old and new props or call functions because of a property change, etc.

`shouldComponentUpdate(nextProps, nextState)`

This is the second function known as properties changes and the first on state changes.

By default, if another component / our component changes a property/state of our component, it will render a new version. Therefore, in this method, this function always returns true.

We can override this function and select more accurately if our component must update or not.

This function is generally used for optimization.

In the method of the function returns false, the update pipeline stops directly.

`componentShouldUpdate(nextProps, nextState){`

`return this.props.name !== nextProps.name ||`

`this.state.count !== nextState.count;`

```
}
```

```
componentWillUpdate(nextProps, nextState)
```

This function works like `componentWillMount()`. Here, we can't modify the component state by invoking this `setState()` method. It will not be known, if `shouldComponentUpdate()` returns false.

```
render()
```

It is invoked to determine `this.props` and `this.state` and return one of the following types: React elements, Arrays and fragments, Booleans or null, String and Number. If `shouldComponentUpdate()` returns false, the code within `render()` will be invoked again to assure that the component shows itself correctly.

```
componentDidUpdate(prevProps, prevState)
```

It is invoked directly after the component updating appears. In this method, we can put some code inside this, which we need to implement once the updating occurs.

This method is not invoked for the basic render.

18. What is PropTypes library in React?

PropTypes is the type-checking addition to React Library, which exports a range of validators to ensure the data component received is valid. As a result, it reduces several bugs and makes components self-documented.

19. How can we renew the State of a component?

We can renew the State of a component using `this.setState()` method. This method does not continually replace the State immediately. Instead, it only include changes to the initial State. It is the main method to update the user interface in response to event handlers and server responses.

20. What is NextJS?

Next.js is a famous and lightweight framework for static and server-rendered applications that work with React. It also gives styling and routing solutions.

21. What are the features of Next.js?

The main features of Next.js are as follows:

- It is used to server-rendered by default.
- It contains programmed code-parting for quicker page loads.
- It is used to clean user-side routing (page-based).

- It can have a Webpack-based dev setting that provides (HMR).
- It can be easy to perform with Express or another Node.js HTTP server.
- It is used customizable with our own Babel and Webpack configurations.

22. What are Error Boundaries?

One of the frequently asked ReactJS interview questions for experienced professionals is about error boundaries. You must know its meaning and role, as explained below.

Error boundaries are the React components that catch the JavaScript errors somewhere in their child component tree and log those errors. As a result, they will show a fallback user interface rather than the component tree that crashed.

A class would be an error boundary if it defines both the lifecycle methods `static getDerivedStateFromError()` or `componentDidCatch()`. The need for `static getDerivedStateFromError()` is for rendering a fallback UI after a bug has been thrown. The need for `componentDidCatch()` is for logging the error data.

4. Explain the concept of the Virtual DOM.

The Virtual DOM (Document Object Model) is a concept implemented in React that provides a programming API that works like a lightweight copy of the actual DOM. This means that whenever a component's state changes, the Virtual DOM gets updated instead of the real DOM. React then efficiently updates the real DOM to match the Virtual DOM, minimizing performance costs and enhancing user experience.

5. Distinguish between a Class component and a Functional component.

Class components are ES6 classes that extend from `'React.Component'` and can hold and manage local state and lifecycle methods. On the other hand, Functional components are simpler and primarily used for rendering UI without handling state or lifecycle methods, although with React Hooks, they are now capable of using both.

6. How do you create a React component?

There are two main ways to create React components:

- Class-based components: These use the class keyword and lifecycle methods to manage state and handle events.
- Functional components: These are simpler functions that return JSX code and can

leverage React Hooks for managing state and side effects.

7. What are Props in React?

Props are read-only properties passed down from parent components to child components.

They act like arguments, providing data to child components without modifying their internal state.

Example:

8. What's the difference between Props and State in React?

- Props: Read-only data passed down from parent to child components. Used for customization without changing internal state.
- State: Internal data managed by a component. State can be changed, making it ideal for keeping track of user inputs, events, and data that changes over time.

9. What does the render() method do in React components?

The render() method is essential in class components. It examines this.props and this.state and returns one of the following: React elements, arrays and fragments, portals, string and numbers, Booleans or null. This output represents what should be displayed on the screen.

10. What are keys in React and why are they important?

Keys are special string attributes that you need to include when creating lists of elements. They help React identify which items have changed, are added, or are removed. Keys should be given to the elements inside the array to give the elements a stable identity, enhancing performance during updates.

11. What is an event in React?

In React, an event is similar to events in plain JavaScript—actions like clicks, form submissions, or key presses. React wraps these events in its own SyntheticEvent wrapper to ensure consistency across different browsers.

12. How do you handle events in React?

Handling events in React is straightforward: you use event handlers. These are functions you write to execute when an event occurs. For example, you might have a button that needs to handle a click event, which you can set up like this: `<button onClick={handleClick}>Click me!</button>`, where handleClick is the function that runs when the button is clicked.

13. What is a stateful component?

A stateful component in React is one that can hold and change state over time. These

components are usually class components but can also be functional components using hooks like `useState`. They are handy when your component needs to remember something or be interactive.

14. What is a stateless component?

Conversely, a stateless component is one that doesn't manage any state. These often serve as presentational components, merely rendering UI elements based on the props they receive. Stateless components can be functional components without any hooks for state management.

15. How do you pass data between components in React?

Passing data between components in React is done through props (short for properties). You pass data from parent components to child components as arguments to the child component in the JSX where it's used.

16. What are controlled components?

In React, a controlled component is one that manages its own state and updates based on user input. For example, form elements like inputs often need to be controlled components, whereas React handles the form data.

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

To update the state of a component in React, you use the `setState` method in class components or the setter function from `useState` in functional components. It's important to remember that state updates may be asynchronous and should not rely on the previous state directly.

18. What is the significance of the `componentDidMount` lifecycle method?

`componentDidMount` is a lifecycle method in class components that is called after the component is rendered for the first time. This is the perfect place to initiate API calls, set timers, or handle any interactions that require the DOM nodes to be present.

19. Explain the purpose of the `useState` hook.

The `useState` hook is a fundamental hook in React for adding state to functional components. It allows you to add and manage state in a component without converting it into a class component.

20. What is the `useEffect` hook and how is it used?

The `useEffect` hook lets you perform side effects in your components. These can be anything from fetching data to directly interacting with the DOM. It can be configured to run after every render or only when certain values change.

Intermediate level- ReactJs Interview Questions and Answers

Now that you're familiar with the basics of React, it's time to move towards more complex concepts. In this section, we'll cover topics like higher-order components, the React lifecycle, and state management, among others. The following set of questions and answers have been carefully curated to provide you with a comprehensive understanding of intermediate-level React concepts.

21. What are higher-order components?

Higher-order components (HOCs) are a powerful pattern used in React to enhance components with additional functionality. An HOC is a function that takes a component and returns a new component. It's useful for reusing code, logic, and bootstrap abstraction in React applications.

22. Explain the lifecycle of a React component.

The lifecycle of a React component can be divided into three phases: mounting, updating, and unmounting. Mounting is when the component is being created and inserted into the DOM.

Updating occurs when a component is re-rendered due to changes in props or state.

Unmounting is the final phase when the component is removed from the DOM. Lifecycle methods like `componentDidMount`, `componentDidUpdate`, and `componentWillUnmount` allow developers to hook into these phases for managing operations appropriately.

23. How can you handle forms in React?

Forms in React can be handled using controlled components where form data is handled by the state within the component. Each state mutation has a corresponding handler function, making it straightforward to modify or validate user input.

24. What is lifting state up in React?

Lifting state up is a common pattern for sharing state between multiple components. It involves moving state to the nearest common ancestor of the components that require it. This way, state can be passed down as props to the components that need it, ensuring consistent data and behavior.

25. How does React implement the re-rendering of components?

React implements re-rendering through its reconciliation algorithm, where it updates the DOM based on the changes in the component's state or props. React efficiently updates only the parts of the DOM that actually changed, rather than re-rendering everything, which enhances performance.

26. What are controlled components?

Controlled components are those where React controls the values of input elements. The input form elements, such as `<input>`, `<textarea>`, and `<select>`, have their values controlled by React's state, and their values change via state, not directly from user input.

27. What are uncontrolled components?

Uncontrolled components work like traditional HTML form inputs, where the forms naturally keep some internal state. In React, uncontrolled components are managed using a ref to get form values from the DOM instead of handling the form state via state.

28. Explain the concept of virtual DOM and how it differs from real DOM.

The virtual DOM is a lightweight copy of the real DOM. It is a concept implemented by React that allows for efficient updates to the UI by minimizing direct manipulations of the real DOM, which can be slow. When a component's state changes, React creates a new virtual DOM and compares it with the previous version. Only the differences are updated in the real DOM.

29. How do you optimize performance in a React application?

Optimizing performance in a React application can involve several strategies, such as using `shouldComponentUpdate` or `React.memo` to prevent unnecessary re-renders, code-splitting to reduce the size of bundles loaded initially, and using lazy loading for components.

30. What is the context API?

The Context API is a way for a React app to effectively produce global variables that can be passed around. This is the alternative to "prop drilling" or moving props from grandparent to child to parent, and so on. Context is often used to share data such as user authentication, themes, or a language preference.

31. How do you use refs in React?

Refs in React are used to get references to a DOM node or an instance of a component in a React Application. Refs are created using `React.createRef()` and attached to React elements via the `ref` attribute.

32. Explain forward refs in React.

Forward refs in React allow you to pass a ref down to a child component. This is particularly useful in higher-order components or when you need the parent component to directly interact with child component DOM nodes.

33. What are synthetic events in React?

Synthetic events in React are wrapper objects around the native event. They combine the behavior of different browser's native events into one API, ensuring that the events behave identically across all browsers.

34. How do you implement error handling in React components?

Error handling in React components can be achieved using error boundaries. An error boundary is a component that catches JavaScript errors in its child component tree, logs those errors, and displays a fallback UI instead of the component tree that crashed.

35. What are portals in React?

Portals provide a first-class way to render children into a DOM node that exists outside the DOM hierarchy of the parent component. This is commonly used for modals, tooltips, and floating menus.

36. How does React Router work?

React Router is a library that enables dynamic routing in a web app. It keeps the UI in sync with the URL, allowing you to handle routing declaratively. It works by changing your application's components depending on the browser's URL, without reloading the page.

37. What is the difference between React Router and traditional routing?

React Router uses client-side routing, where the routing is handled internally by the JavaScript that is loaded on the page, without the need for page reloads. Traditional routing, on the other hand, involves requests to a server and reloading the entire page with new content.

38. How do you implement code-splitting in React?

Code-splitting in React can be implemented using `React.lazy` and `Suspense`. This allows you to split your code into separate chunks which can be loaded on demand. It is particularly useful for improving the initial load time of the application.

Advanced ReactJS Interview Questions for Experienced:

Mastering advanced React concepts is crucial for handling complex projects and architectural challenges. Our "Advanced ReactJS Interview Questions for Experienced" section delves into advanced topics that seasoned developers often encounter. From state management strategies to handling side effects, hooks, and server-side rendering, this section is designed to test and expand your mastery of React.

39. What are the different ways to manage State in a React application?

React offers multiple ways to manage state, each with its own use case. Here are the common

approaches:

- Local state: Managed within a component using `useState` or `useReducer`.
- Global state: Tools like `Redux` or `Context API` help manage state that is accessible by any component in the application.
- Server state: Data fetched from an external server which can be managed via `React Query` or `SWR`.
- URL state: State represented in the URL parameters accessible via `React Router`.

40. How do you handle side effects in React components?

Side effects are operations affecting other components or that involve asynchronous operations. React uses the `useEffect` hook to handle side effects, such as API calls, subscriptions, or manually manipulating the DOM.

Example:

41. Explain the concept of hooks in React. What problems do they solve?

Hooks are functions that let you "hook into" React state and lifecycle features from function components. They allow you to write functional components with the same capabilities as class components, making your code cleaner and easier to maintain.

Problems Solved by Hooks:

- Code Reusability: Share logic across components without resorting to higher-order components (HOCs).
- Component Composition: Build complex UIs by combining simpler functional components.
- State Management: Use the `useState` hook to manage component state within functional components.

42. How would you implement global state management in React without using external libraries?

To manage global state without external libraries, the `Context API` can be utilized effectively. It allows you to share values between components without having to explicitly pass a prop through every level of the tree.

Example:

43. What is React Fiber?

React Fiber is a complete reimplementation of the React core algorithm. It enhances the

suitability of React for areas like animation, layout, and gestures. Its main goal is to enable incremental rendering of the virtual DOM.

44. How do you handle server-side rendering with React?

SSR allows you to render your React application on the server, improving initial page load times and SEO. Libraries like Next.js simplify SSR implementation in React projects.

Example with Next.js:

45. What are the common performance issues in React applications? How do you troubleshoot them?

Performance issues in React often include:

- Unnecessary Re-renders: Optimize components using `shouldComponentUpdate` (class components) or `React.memo` (functional components) to prevent unnecessary re-renders.
- Large Virtual DOM Diffs: Break down complex components into smaller ones to minimize the amount of DOM that needs to be updated.
- Excessive Prop Drilling: Use Context API or state management solutions to avoid passing props through multiple levels of components.

46. How do you secure a React application?

Just like any web application, React applications need to be secured. Here are some key areas to focus on:

- Sanitize User Input: Prevent XSS attacks by sanitizing any user-provided data before displaying it on the UI.
- Secure API Communication: Use HTTPS for API communication to encrypt data transmission.
- Implement Authentication and Authorization: Control user access to specific features and data based on their roles.

47. What are the pros and cons of using Redux?

Redux is a popular state management library, but it's not always necessary. Here's a quick breakdown:

Pros:

- Centralized State: Keeps all application state in one place, making it easier to manage and reason about.

- Predictable Updates: Makes application flow more predictable and easier to debug.
- Large Community and Ecosystem: Plenty of resources and tools available for working with Redux.

Cons:

- Complexity: Setting up and managing Redux can add complexity to smaller applications.
- Boilerplate Code: Requires writing additional code for actions, reducers, and middleware.

48. How do you integrate TypeScript with React?

TypeScript can be integrated by creating React components with TypeScript. This adds static type checking, enhancing the reliability and maintainability of the application.

Example:

49. Explain the main principles of Redux.

Redux follows three fundamental principles:

- Single source of truth: The state of your entire application is stored in one object tree.
- State is read-only: The only way to change the state is to emit an action.
- Changes are made with pure functions: Reducers are pure functions that take the previous state and an action to compute the next state.

50. How do you handle asynchronous actions in Redux?

Asynchronous actions in Redux are handled by middleware like Redux Thunk or Redux Saga.

These allow you to write action creators that return a function instead of an action.

Example with Redux Thunk:

51. What is React Suspense and how do you use it?

React Suspense lets you specify the loading indicator in case some components in the tree below it are not yet ready to render. It's used for code splitting and lazy loading components.

Example:

52. How do you test React components?

Popular testing libraries for React include:

- React Testing Library: A lightweight library focused on testing components in isolation from implementation details.
- Jest: A popular testing framework that can be used to write unit and integration tests for React components.

Example:

53. What is the use of static type checking in React?

Static type checking helps identify potential errors in your code before runtime. TypeScript, as mentioned earlier, is a popular choice for adding static typing to React projects. It improves code readability, maintainability, and helps catch errors early in the development process.

54. Explain the role of immutability in React.

Immutability is a core concept in React, especially when working with state and props. It helps prevent unexpected mutations and enables optimized performance with pure components.

```
*****
*****

*****
*****

*****MySQL*****
*****
```

1. What is MySQL?

This is going to be the most asked MySQL interview questions for all experience-level professionals. You must have a clear idea about the basics first.

MySQL is the world's second most popular and most commonly used open-source relational database management system (RDBMS). It is based on a structured query language that Oracle supports. In addition, MySQL supports many operating systems, the most well-known of which are Windows, Linux, and UNIX.

Although the MySQL database can be used to create a wide range of applications, it is exclusively utilized for web applications. Therefore, it is a critical component of the open-source Lamp project.

MySQL was once owned by a for-profit company, MySQL AB, before being purchased by Sun Microsystems, which was later purchased by Oracle, which now controls MySQL.

2. What is LAMP Stack?

The LAMP is a web development tech stack. It comprises Linux (operating system), Apache (web server), MySQL (database), and PHP (programming language).

3. MySQL database is written in which language?

MySQL is written in C and C++ languages.

This is one of the most common MySQL interview questions for freshers, as they should know the base of the database they are using.

4. What are the features of MySQL database?

It has the following features that you must know if you are preparing for the MySQL interview questions and answers:

- Flexible structure
- High performance
- Manageable and easy to use
- Replication and high availability
- Security and storage management
- Drivers
- Graphical Tools
- MySQL Enterprise Monitor
- MySQL Enterprise Security
- JSON Support
- Replication & High-Availability
- Manageability and Ease of Use
- OLTP and Transactions
- Geo-Spatial Support

5. What are the differences between MySQL and SQL?

You must know the differences between the two because this can be asked among the interview questions on MySQL.

SQL stands for Standard Query Language. It is used to interface with databases such as MySQL.

MySQL is a database that stores and protects numerous sorts of data. A PHP script is necessary to store and retrieve values from the database.

SQL is a programming language, but MySQL is a database.

SQL is used to create database management systems, whereas MySQL allows data handling, storage, deletion, and modification.

6. What are the differences between a database and a table?

There is a significant difference between a database and a table. The distinctions are as follows:

- Tables represent data division in a database, whereas a database represents a set of data and tables.
- Tables are used to group data about one another to produce a dataset. This dataset will be incorporated into the database. The data saved in any form in the table is a part of the database, but the opposite is not valid.

7. Why use the MySQL database server?

The core reason to choose the MYSQL server is that it is open source and can be used by developers and small businesses at no cost. There are several more reasons to use it that you must know as part of your preparation for MySQL technical interview questions and answers.

- The MySQL server is free and open source.
- MySQL's community is vast and supportive. Thus any issues with MySQL are fixed as soon as feasible.
- MySQL has been available on the market for a while, and relatively stable versions are available. All bugs found in prior releases have been continuously removed, and each upgrade provides a more stable version.
- The MySQL database server is speedy, dependable, and simple to use. The software is simple to use and customize. On the internet, MySQL software is freely downloadable.

8. What are the various tables available in MySQL?

There are various tables in MySQL that can be used. However, MyISAM is the default database engine in MySQL. There are five different types of tables available:

- MyISAM
- Heap
- Merge
- INNODB
- ISAM

9. How to check the MySQL version?

If you have been using this database system for a while, then you must know the process

because it is also one of the most frequently asked and top MySQL interview questions.

On Linux, we can check the MySQL version with the following command:

```
mysql -v
```

When we use MySQL in Windows, we should open the MySQL command-line tool, which provides the version information without any flags. Use the following statement to learn more about the server information:

```
SHOW VARIABLES LIKE "%version%";
```

It will provide the following output:

We may see further version information about the installed MySQL software in this output, such as the versions of InnoDB, protocol, and SSL library.

10. How to create columns in MySQL?

A column is a set of cells in a table that holds one value for each row. For adding columns to an existing table in MySQL, you should use the ALTER TABLE command.

11. How to delete a table in MySQL database?

The Drop Table command in MySQL can be used to delete a table. This permanently deletes a table's whole data set from the database, including structure and definition.

You must know that precautions are essential while deleting a table. It is because we are unable to retrieve the table in MySQL after using this command.

```
DROP TABLE table_name;
```

12. How to add a foreign key constraint in MySQL database?

To link two or more tables together, utilize a foreign key. To join the two tables, it matches the primary key field of another table. It enables us to form a parent-child bond with the tables.

There are two ways to add a foreign key to a table:

- Using the CREATE TABLE Statement
- Using the ALTER TABLE Statement

The syntax for defining a foreign key using the CREATE TABLE OR ALTER TABLE statement is as follows:

```
[CONSTRAINT constraint_name]
```

```
FOREIGN KEY [foreign_key_name] (col_name, ...)
```

```
REFERENCES parent_tbl_name (col_name,...)
```

You must practice these things before going to the job interview, so that you are well prepared for such MySQL technical interview questions and answers.

13. How to change the MySQL root password?

We can change the MySQL root password by typing the following statement into a new notepad file:

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'NewPassword';
```

Open the Command Prompt tool and go to the MySQL directory. Please copy the following folder, paste it into the DOS command, and press the Enter key.

```
C:\Users\wscube> CD C:\Program Files\MySQL\MySQL Server\bin
```

After this, use the below statement to update or reset the password:

```
mysqld --init-file=C:\\mysql-notepadfile.txt
```

Finally, we can use this new password to log into the MySQL server as root. To ensure the password change, delete the C:mysql-init.txt file after launching the MySQL server.

14. How can we create a database in MySQL Workbench?

For this, we must first run the program and log in with the username and password.

Next, navigate to the Navigator tab and select the Schema menu. Finally, choose Create Schema from the Schema menu, or click the database icon, as seen in the following screen.

A new popup box displays where we must enter all of our information. After inputting the information, click the Apply button, followed by the Finish button, to finish the database construction.

15. How can we create a table in MySQL Workbench?

This is yet another important thing that you must know and have in your list of MySQL interview questions and answers for experienced professionals.

- Launch MySQL Workbench and navigate to the Navigation tab, then click on the Schema menu, which displays all previously created databases.
- Double-click on any database to open it. It will display the sub-menus from which we must select the Tables option.
- Select the Tables sub-menu, right-click it, and then choose the Create Table option.
- We can also construct a table by clicking the create a new table icon.
- It will launch a new popup page where we must enter all the necessary information to construct a table. Enter the table name and then column information here.

- After inputting the information, click the Apply button, followed by the Finish button, to complete the table creation.

16. How to change a table's name in MySQL?

To change the name of the table, we must modify or rename the table name. To change the name of one or more tables in the current database, use the rename command, and use the following syntax:

```
mysql> RENAME existing_table TO new_table;
```

For changing the names of multiple tables, you can go with this syntax:

```
RENAME TABLE existing_tab1 TO new_tab1,  
existing_tab2 TO new_tab2, existing_tab3 TO new_tab3;
```

17. How to change the name of a database in MySQL?

We may sometimes need to update or rename the database. To do so, we must first create a new database on the MySQL server.

MySQL includes the mysqldump shell command. Using it, you can create a dumped copy of the database, as well as import the data to the new database.

```
mysqldump -u username -p "password" -R existingdatabasename >  
existingdatabasename.sql
```

The following command can be used for data importing from existing to new database:

```
mysql -u username -p"password" newdatabaseame < existingdatabasename.sql
```

This is how you can rename a MySQL database. You must be prepared for this type of MySQL interview questions for 3 years experience or below.

18. How to import a MySQL database?

Importing a database in MySQL is moving data from one location to another. However, it is a highly beneficial tool for backing up essential data or transferring data across sites.

For example, we have a database of contacts that must be kept in a secure location. So we must export it to a safe area, and if it is lost from its original site, we can restore it using import options.

We can import a database into MySQL in two ways:

- Command Line Tool
- MySQL Workbench

19. How can we change the column name in MySQL

database?

We kept one of the column names erroneously while generating a table. In MySQL, we must use the ALTER TABLE and Alter commands in tandem to change or rename an existing column name.

The syntax for renaming a column in MySQL is as follows:

```
ALTER TABLE table_name
```

```
CHANGE COLUMN existing_column_name new_column_name column_definition
```

```
[FIRST|AFTER existing_column];
```

20. How to delete columns in MySQL database?

Using the ALTER TABLE command, we can remove, drop, or delete one or more columns from an existing table:

```
ALTER TABLE table_name DROP COLUMN columnname_A, columnname_B....;
```

21. How to insert data into MySQL table?

Using the INSERT command, we can insert data into a MySQL table. For example, we can use this expression to insert a single or several rows into a table. The basic syntax for inserting a record into a table is as follows:

```
INSERT INTO table_name (fieldA, fieldB,...fieldN )
```

```
VALUES (valueA, valueB,...valueN);
```

For inserting multiple rows into a table, use the below syntax:

```
INSERT INTO table(fieldA, fieldB,...fieldN)
```

```
VALUES
```

```
(valueA, value B, ...),
```

```
(valueA, valueB, ...),
```

```
...
```

```
(valueB, valueB, ...);
```

22. How to delete a row in a table in MySQL?

Using the DELETE command within the database, a row can be deleted. The generic syntax of the DELETE command is as follows:

```
DELETE FROM table_name WHERE Condition_specified;
```

It should be noted that if the WHERE clause is not included with the syntax, this statement will delete every record in the table.

23. How to join two tables in MySQL?

We can join tables in MySQL using the JOIN clause. MySQL supports a variety of JOIN clauses. These clauses relate to many tables and return only records that have the same value and property in all of them.

The four most straightforward techniques to combine two or more tables in MySQL are as follows:

- Inner Join
- Left Join
- Right Join
- Cross Join

24. How to update a table in MySQL?

We can use the UPDATE statement, which comes with the SET and WHERE clauses, to update existing records in a table.

The SET clause modifies the values of the columns supplied. The WHERE clause, which specifies the criteria, is optional. This statement can also be used to alter the values in one or more columns of a single row or multiple rows simultaneously.

The following is a generic syntax for the UPDATE statement used to edit data in a MySQL table:

```
UPDATE table_name
```

```
SET field1=new-value1, field2=new-value2, ...
```

```
[WHERE Clause]
```

25. What is MySQL Workbench?

MySQL Workbench is a user interface for MySQL (GUI) applications for accessing and managing MySQL databases. Oracle created and maintained it, offering SQL creation, data migration, and complete administrative tools for server configuration, user management, backup, and so on.

In addition, this Server Administration can be used to generate new physical data models, E-R diagrams, and SQL development. It works with all major operating systems. MySQL Server versions 5.6 and higher include support for it.

It is primarily available in three editions, as listed below:

- Community Edition (Open Source, GPL)

- Standard Edition (Commercial)
- Enterprise Edition (Commercial)

26. How to remove the primary key from MySQL?

We can do it using the ALTER TABLE statement. The syntax for this is:

```
ALTER TABLE table_name DROP PRIMARY KEY;
```

This is among the crucial interview questions on MySQL that is a bit technical, but you must know the correct answer.

27. What is a Stored Procedure in MySQL?

A stored procedure in MySQL is a group of SQL statements saved in the database. SQL queries such as INSERT, UPDATE, DELETE, and so on can be included in the stored procedure.

A procedure allows us to reuse the same code by running a single line. It saves the data dictionary in the database.

28. How to run a stored procedure in MySQL?

Using the CALL query, we may execute a stored procedure in MySQL. This query accepts the name of the stored procedure and any parameters that must pass to it. The basic syntax for running a stored procedure is as follows:

```
CALL stored_procedure_name (argument_list);
```

Let's understand it with this example:

```
CALL Product_Pricing (@pricelow, @pricehigh);
```

Here, Product_Pricing is a stored procedure that calculates and returns the lowest and maximum product prices.

29. How to create a view in MySQL?

A view is a database object that derives its values from the base table. It is a virtual table produced by connecting one or more tables in a query. It functions similarly to the base table but contains no data. If the underlying table changes, the changes are mirrored in the View.

The general syntax for creating a VIEW in MySQL is as follows:

```
CREATE [OR REPLACE] VIEW view_name AS
```

```
SELECT columns
```

```
FROM tables
```

```
[WHERE conditions];
```

30. How to create a MySQL trigger?

A MySQL trigger is a code that performs a certain action in a database automatically invoked if certain events occur on a table or View in the database.

For example, it can be run whenever records are added to a table, or any columns are modified.

In MySQL, we can create a trigger with the following syntax:

```
CREATE TRIGGER trigger_name
[before | after]
{insert | update | delete}
ON table_name [FOR EACH ROW]
BEGIN
--variable declarations
--trigger code
END;
```

31. How to clear the console screen in MySQL?

Before version 8, it was impossible to clear the screen using MySQL in Windows. At the time, the only way to remove the screen was to exit the MySQL command-line tool and reopen MySQL.

After MySQL version 8, we can use the following command to clear the command line screen:

```
mysql> SYSTEM CLS;
```

32. How to create a new MySQL user?

In MySQL, a USER is a record in the USER-TABLE. It holds the login details, account privileges, and host information for the MySQL account so that the database can be managed.

A new user account can be created with the help of MySQL Establish User statement. It provides recent reports with authentication, SSL/TLS, resource limit, role, and password management characteristics.

The fundamental syntax for creating a new user in MySQL is as follows:

```
CREATE USER 'username'@'host' IDENTIFIED WITH authentication_plugin BY 'password';
```

33. How to check for USERS in MySQL?

To manage a database in MySQL, we need to see a list of all user accounts on a database server. The command below is used to get a list of all database server users:

```
mysql> SELECT USER FROM mysql.user;
```

34. How to insert a date into a table in MySQL database?

To add a date to a MySQL table, we can use the INSERT statement. MySQL supports date data types, including DATE, TIMESTAMP, DATETIME, and YEAR. The date format in MySQL is YYYY-MM-DD by default.

The following is the fundamental syntax for inserting a date into a MySQL table:

```
INSERT INTO name_of_table (name_of_column, column_date) VALUES ('DATE: Manual Date', '2022-07-22');
```

Use the below statement to insert mm/dd/yyyy format

```
INSERT INTO name_of_table VALUES (STR_TO_DATE(date_value, format_specifier));
```

35. How to find the database size in MySQL?

To have information about the tables and databases, we can query the information_schema.tables table in MySQL. It will return data length, index length, collation, creation time, etc.

Using the syntax below, we can determine the size of the database on the server:

```
SELECT table_schema AS "Database",  
ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS "Size (MB)"  
FROM information_schema.TABLES  
GROUP BY table_schema;
```

36. How does MySQL indexing work?

Indexing is the process of converting an unsorted list into an ordered list. It aids in increasing query efficiency when searching tables in MySQL. The indexing functions similarly to a book index.

For example, we have a book and wish to learn about search. Go through each page without indexing until the desired topic is located. It contains a collection of keywords that may be used to find the topic stated on pages. Then we may go directly to those pages without having to wade through all of them.

37. Who owns MySQL?

MySQL is the most widely used free and open-source database software under the GNU

General Public License. MySQL AB, a Swedish firm, initially owned and sponsored it. It is presently owned by Sun Microsystems (formerly Oracle Corporation), which manages and improves the database.

38. In MySQL, how to view the database?

Viewing or listing the accessible databases is a regular operation while working with the MySQL server. For example, using the following command, we can inspect all of the databases on the MySQL server host:

```
mysql> SHOW DATABASES;
```

39. How to enable auto increment in MySQL?

Auto Increment is a constraint that creates a unique number automatically when inserting a new entry into the table. It is typically used for a table's primary key field.

For example, in MySQL, we can use the ALTER TABLE query to set the value of an AUTO_INCREMENT column as follows:

```
ALTER TABLE table_name AUTO_INCREMENT = value;
```

40. What are the differences between TRUNCATE and DELETE in MySQL?

- DELETE is a DML command, and TRUNCATE is a DDL command.
- The Where command cannot be used with TRUNCATE QL, although it can be used with DELETE.
- TRUNCATE cannot be used in conjunction with indexed views, although DELETE may.
- To erase data from a table, use the DELETE command. It just deletes the rows of data from the table, but truncating is a dangerous operation that should be used with caution because it permanently deletes every row from a table.

41. How many triggers can be used in MySQL?

In the MySQL database, only six triggers are permitted to be used.

- Before Insert
- After Insert
- Before Update
- After Update
- Before Delete

- After Delete

42. What is a heap table?

HEAP tables are tables that are stored in memory. When creating a heap table in MySQL, you must define the TYPE as HEAP. These are frequently referred to as memory tables. They are used for temporary high-speed storage. They don't support BLOB or TEXT fields.

43. What are BLOB and TEXT in MySQL?

A huge binary object is referred to as a BLOB. It is used to store varying amounts of info. The BLOB comes in four varieties.

- TINYBLOB
- BLOB
- MEDIUMBLOB
- LONGBLOB

The only distinction between these is the maximum length of data they can carry.

The BLOB TEXT is case-insensitive. The values of TEXT are non-binary strings (character strings). Character sets and values are saved and compared depending on the character set collation.

TEXT is classified into four categories.

- TINYTEXT
- TEXT
- MEDIUMTEXT
- LONGTEXT

44. What is a MySQL trigger?

A trigger is a sequence of instructions that run in response to certain occurrences.

45. What is the difference between a heap table and a temporary table?

Heap tables:

- They are present in memory and are utilized for temporary high-speed storage. They don't support BLOB or TEXT fields.
- Heap tables do not support AUTOINCREMENT.
- Indexes should not be NULL.

Temporary tables:

They are used to store temporary data. However, it is sometimes advantageous to keep transitory data. Therefore, the temporary table is erased when the client session ends.

Main differences:

All clients share the heap tables. However, the temporary tables are not.

Heap tables are another storage engine, whereas temporary tables require a particular privilege (create a temporary table).

46. What's the difference between FLOAT and DOUBLE?

FLOAT saves floating-point integers with up to 8 places of precision and allocates 4 bytes.

DOUBLE, on the other hand, keeps floating-point integers with accuracy up to 18 places and allocates 8 bytes.

47. What are the disadvantages of MySQL?

If you have been working on this database for some years now, then this is going to be one of the top MySQL interview questions and answers for experienced professionals.

Here are some drawbacks of MySQL:

- For large-scale databases, MySQL is inefficient.
- Versions below 5.0 do not support the COMMIT and STORED PROCEDURES capabilities.
- Transactions are not processed efficiently.
- MySQL's functionality is heavily reliant on third-party extensions.
- Development is not a community-driven process.

48. What are the differences between CHAR and

VARCHAR?

Here are the main differences between CHAR and VARCHAR:

- In terms of storage and retrieval, CHAR and VARCHAR differ.
- The length of a CHAR column is constant, whereas the size of a VARCHAR column is flexible.
- The maximum number of characters CHAR data types can carry is 255, but VARCHAR data types can hold up to 4000.
- VARCHAR is 50 percent slower than CHAR.
- VARCHAR employs dynamic memory allocation, whereas CHAR uses static memory allocation.

49. In MySQL, how to retrieve the current date?

Use the following syntax to get the current date:

```
SELECT CURRENT_DATE();
```

50. What is the default MySQL port number?

MySQL's default port is 3306.

51. What is REGEXP in MySQL?

REGEXP is a regular expression pattern match. A regular expression is a vital tool for specifying a pattern for a complex search.

It is essentially a customized text string used to describe a search pattern. To further comprehend it, consider a circumstance in which you search for .txt files in the file manager to list all text files. .*\.txt is the regex equivalent of .txt.

52. How many columns can be added to an index?

A typical table can have a maximum of 16 indexed columns.

53. What is the difference between the functions NOW()

and CURRENT DATE() in MySQL?

The NOW() command displays the current year, month, and date with hours, minutes, and seconds.

On the other hand, the CURRENT DATE() only indicates the current year, month, and date.

As a beginner, you must be ready to face such MySQL interview questions and answers for freshers.

54. What query is used to display the top 20 rows?

It is one of the top interview questions on MySQL queries that can be asked to experienced professionals.

Use this syntax:

```
select column1, column2, ... from table_name LIMIT n
```

Here, put n=20 so that it will show the top 20 rows from your MySQL database table.

55. What is MySQL SAVEPOINT statement?

A savepoint is a defined point in any transaction.

SAVEPOINT is a MySQL statement used to create a named transaction savepoint with the name of the identifier.

56. What is SQLyog?

SQLyog is the most popular administrative GUI tool. It is the most widely used MySQL development and administration tool. It combines the functionality of MySQL administrator, phpMyadmin, and other programs.

57. How to back up a MySQL database?

This is among the frequently asked MySQL interview questions for 3 years experience. It can also be put forward even if you have more experience.

phpMyAdmin makes data backup simple. First, you can choose which database to back up by clicking the database name in the left-hand navigation bar.

Then, select the export button and ensure that all the tables you wish to back up are highlighted. Then, under export, select the desired option and save the output.

58. What is the use of ENUMs in MySQL?

It is a data type in MySQL. By creating ENUMs, we allow the end-user to supply accurate input. If the user offers information not part of the ENUM-defined data, the query will not execute, and an error message stating "The incorrect Query" will be presented.

For example, imagine we wish to take the user's gender as an input. Therefore we define ENUM('male,' 'female,' 'other'), and hence any text other than these three results in error.

ENUMs are used to restrict the values that can be inserted into the table:

For example:

```
CREATE TABLE months (month ENUM 'January', 'February', 'March'); INSERT months  
VALUES ('April').
```

59. What benefits does MyISAM have over InnoDB?

MyISAM takes a cautious approach to disk space management, storing each MyISAM table in its file that can be compressed further if necessary.

InnoDB, on the other hand, keeps the tables in the tablespace. As a result, it is difficult to optimize it further.

60. What is the difference between mysql_connect and mysql_pconnect?

It's yet another technical interview question on MySQL. Here is the short answer.

Mysql_connect() creates a new database connection, whereas mysql_pconnect() creates a permanent database connection.

It specifies that mysql_pconnect() does not open the database each time the page is loaded.

61. What is the function `mysql_close()` used for?

It is used to cancel a connection started by `mysql_connect()`.

62. What is a MySQL data directory?

The MySQL data directory is where MySQL saves its data. This data directory's subdirectories each represent a MySQL database. The information controlled by MySQL = server `mysqld` is saved in the data directory by default.

63. How to find the location of the MySQL data directory?

In Windows, the default location of the MySQL data directory is `C:\mysql\data` or `C:\Program Files\MySQL\MySQL Server 5.0\data`.

64. What is the "i-am-a-dummy" flag in MySQL used for?

The "i-am-a-dummy" signal in MySQL causes the MySQL engine to reject `UPDATE` and `DELETE` operations unless the `WHERE` clause is present.

65. What is Access Control List in MySQL?

An Access Control List is a set of permissions associated with a particular item. MySQL caches the Access Control Lists in memory, and anytime a user attempts to authenticate or execute a command; it checks the permissions required for the object. If the permissions are available, the execution succeeds.

66. What is InnoDB?

InnoDB is a SQL storage database. InnoDB also supports ACID transactions and includes support for foreign keys. Initially held by InnobaseOY, it currently belongs to Oracle Corporation, which bought it in 2005.

67. How to activate batch mode in MySQL?

In MySQL, we use the following command to enable batch mode:

```
mysql;
```

```
mysql mysql.out;
```

68. What are the drivers in MySQL?

The following are the drivers available in MySQL:

- PHP Driver
- JDBC Driver
- ODBC Driver
- C WRAPPER
- PYTHON Driver
- PERL Driver
- RUBY Driver
- CAP11PHP Driver
- Ado.net5.mxz

69. What are DDL, DML, and DCL?

SQL instructions are broadly classified into DDL, DML, and DCL.

- Data Definition Language (DDL) is responsible for all database schemas and describes how data should be stored in the database. DDL includes commands such as CreateTABLE and ALTER TABLE.
- Data Manipulative Language (DML) is concerned with data operations and manipulations. DML commands include Insert, Select, and so on.
- Data Control Languages (DCL) are associated with permits and grants. In short, these define the authorization to access any part of the database.