

1. What is HTML and what is its purpose?
- HTML stands for Hyper Text markup language.
  - It is a standard markup language for web page creation.

- It allows the creation and structure of sections, paragraph & links using HTML elements such as tags & attributes.
- The basic purpose of HTML is to create and structure the content of web pages interactively.

## 2. Difference b/w HTML & xHTML

HTML	xHTML
1. HTML is an SGML-based language. That defines the standard for generalizing mark up language for documents.	1. It is an XML based language which defines it manipulates & process the data using XML technologies.
2. Not a case-sensitive language.	2. It is case-sensitive language.
3. It doesn't require a closing tag for an element or even "/" symbol which indicates the closing of a tag.	3. Empty elements also be closed, there must be "/" symbol at the end of empty element all elements & attributes must be in lowercase since it is case-sensitive.
4. Tags & attributes can be specified in lowercase or uppercase since it is not case sensitive.	4. Elements & attributes must be in lowercase since it is case-sensitive.
5. The declaration of <!DOCTYPE> is not necessary. Only it requires minimum of <html><head><title><body>	5. <!DOCTYPE> is necessary followed by <html><head><title><body>

6. They do not need

6. Nested elements do not require to be closed in the order in which they are opened.

7. It is not mandatory to put quotes while using the attributes in HTML.

8. webpage can be displayed even if the HTML document has some errors in it.

9. filename extension is .html or .htm

9. filename extension can also be .xhtml, .xhtml or .xml or .htm

3. What are the new features introduced in HTML5

→ semantic elements:

such as <header>, <footer>, <aside>, <nav>,

<section>, <article>, <main>, etc.

→ These elements will give you more clear and meaningful structure to web documents.

→ improves accessibility & search engine optimization.

2. multimedia support:

HTML5 gives a built-in support for

multimedia elements without the need for

external plugins. The <video> & <audio>

elements allow for easy embedding of

videos, audio content in web pages.

### 3. Figure and FigCaption

HTML 5 allows to use a `<figure>` element to markup a photo in a document & `<figcaption>` element to define a caption for photo. `<figcaption>` tag defines a caption for a `<figure>` element.

- \* It can be provides a container for a content that is equivalent to `<figure>` else you can add a `<caption>` for two/more image, a block of code, or other content.

`<figure>`

```
<img alt="...png" />
<figcaption> quicks for gelie </figcaption>
</figure>
```

### 4. Drag & Drop:

HTML 5 introduced native support for drag & drop functionality within web app, making it more intuitive for user to interact with content.

### 5. Local Storage:

HTML 5 introduced the concept of local & session storage APIs which provide a way to store data on a client-side. This enable web applications to store & retrieve data locally which will improve their performance, reducing dependency on server-side storage.

6. Geo location API: HTML 5 introduced geolocation API, allowing web applications to access the user's location information with their permission. This enables location-based services & applications that tailor the content based on the user's geographical location.

7. form elements: input, button, select, text, checkbox, radio

8. progress indicator: progress element

9. canvas: used to draw a graphics on a web page using JavaScript

(1) How do you include "Comments" in HTML?

<!-- comment -->

(2) Difference b/w <div> & <span> tag?

<div>

<span>

→ used to create larger blocks of content

\* The '<div>' tag is a block-level element.

\* Typically used to group & contain other HTML elements providing a way to create sections/division inside a web page.

\* used for layout purposes. Developers can apply CSS styles such as positioning, margins, padding, background color.

→ used to create smaller sections of content

\* The '<span>' tag is an inline element.

\* applying grouping & applying a style for smaller section of text or other inline elements.

\* within a larger block of content

\* CSS styling can be applied when you want to style for a specific portion of text / to inline formatting / styling.

\* It will cause a line break before & after the element causing it to occupy its own horizontal space on webpage.

It does not create a line break before / after the element. It only applies the targeted positioning of content without altering the overall layout.

Q. What are semantic elements and why are they important?

→ Semantic elements clearly describes its meaning to both the browser & the developer like `<header>`, `<footer>`, `<form>`, `<table>`, `<article>`, `<section>`, `<nav>`, but not `<div>`, `<span>`.

`<main>` Represents the main content area of a webpage.

`<article>`: Represents a self-contained & independent piece of content.

Importance:

- \* Improves accessibility

- \* Search engine optimization

- \* Enhances code readability & maintainability

- \* Contributes to a better user experience

- \* Promotes consistency & standardization in web development.

Q. What is the purpose of `<header>`, `<nav>`, `<section>`, `<footer>` tags in HTML 5?

→

`<header>` - container section of at the beginning of a webpage. It contains logos, nav menus, page title, branding & any other elements that provide a top-level structure / identification of the page.

2. <nav> It is used to define a section of a website that contains a navigation links, represents a navigation menu / navigation bar of webpage.

3. <section>  
→ It defines section in a document.  
→ It represents logically / thematically related grouping of content.

4. <footer> represents the footer section of a webpage typically contains information such as copyright notices, legal disclaimers, contact information links to related resources, other content that is placed at the bottom of the page.

Q. How do you create a hyperlink in HTML?  
→ Using anchor tag i.e. <a> and href as its attribute which contains a link.

<a href = "www.google.com" > Google </a>

Q. What is difference between <ol> & <ul> elements?

- \* <ul> -> Unordered list
- \* <ol> -> Ordered list

\* Items in the list will be marked with the bullet by default, you can also customize by

<ol>

- \* ordered list
- \* list is created in a particular / specific order where each item is numbered / listed by default.
- \* You can customize by type attribute which supports value "1", "a", "i", "u", "g" etc

Q1. Q1. How do you embed an image in HTML?

- By using `<img>` tag which has `src` or `alt` attribute where you can add a direct URL of an image, or you can add a bound image by copying the path where it is located, and image must be in `png` or `jpg` format for better performance.
- and you can also add `alt="alt"` attribute giving an alternative text also you can give height & width.

Q2. Difference b/w `<strong>` & `<em>` tag.

<code>&lt;strong&gt;</code>	<code>&lt;em&gt;</code>
→ used to highlight the text that has strong importance or significance.	→ highlights the words that need additional attention to convey a different tone to emph.
* <code>strong</code> makes a texts to bold text.	* <code>em</code> makes a texts to italic text.

Q3. How do you create a table?

- To create a table user `<table>`, `<tr>`, `<th>` & `<td>` tag.
- `<table>` tag → container tag for entire table.
- `<tr>` → represents a row & contains a `<td>` / `<th>`
- `<th>` name `<th>` → header cell used for 1<sup>st</sup> row for column headings.
- `<td>` tag, `<td>` → used for regular data in a table i.e. standard table data cell.
- `<tr>` → `<td>` moni `<td>`
- `<td>` 28 `<td>`
- `<tr> <table>`

13. What is the purpose of the `<form>` tag in HTML?

How do you create a form?

→ `<form>` tag: It is used to create forms.

It is used to create an interactive forms on web pages.

Purpose:

It will take the data from the user and

will store it to a server for processing.

→ Handling for client-side scripting.

Creation of a form:

\* `<form>` method = "GET" action = "Submit-form".

→ If we want to submit a form then we have to use `POST`.

→ A form has an attribute called method i.e. get/post.

Get will just the get the information (post means)

it will send it to a server.

→ Within the `<form>` tag it uses other form controls.

Such as `<input>`, `<select>`, `<textarea>` etc.

`<button>`. These controls allow user to

enter / select data.

→ `<label for="name">`, `<name>` & `<label for="name">`

`<input type="text" id="name" name="name">`

`<input type="submit">` (Identify `<button>`)

`</form>` (End of the form tag)

Each input field has attributes such as 'type',

'id', specifying the type of input expected.

'id' for identifying the field.

'name' to associate the input with a name when the form is submitted.

(Name label)

\* When the form is submitted, the data entered by the user is sent to the specified URL in the 'action' attribute using the HTTP method (specified in the method's attribute (e.g. GET & post)).

Q. What are some new input types introduced in HTML5?

→ 1. 'color': Allows user to pick select a color using a color picker.

`<input type="color" name="colorpicker">`

2. 'date': provides a date input field with a date picker for selecting a date.

3. 'email': Validates that the input value is an email address.

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

4. 'number': Restricts input to numeric values & provides controls for increment & decrement value.

5. 'range': creates a slider control for selecting a value within a specified range.

6. 'search': Renders an input field with a search icon / custom search controls.

7. 'tel': specifies an input field for entering a telephone number.

8. 'url': validates the input value is a URL.

9. 'datetime', 'datetime-local', 'time', 'week': These input types handle specific date & time formats such as full date & time, local date & time, time only & week.

number respectively.

15. How do you include audio & video content in HTML?

→ we can use the `<audio>` & `<video>` tag to include audio & video elements respectively.

Audio :-

```
<audio src="audio-file.mp3" controls></audio>
```

In the above example the 'controls' attribute adds playback controls to the audio player.

Video :-

```
<video src="1.mp4" controls></video>
```

16. What is the purpose of the `<iframe>` tag & how is it used?

→ Purpose of `<iframe>` tag :-  
The tag used to embed the other document into the html page.

Ex:-

```
<iframe src="https://eight.html" title="video player">  
width="400" height="300", frameborder="0">  
</iframe>
```

17. How do you choose css styles to html elements?

→ we can add css styles to html element by following methods

1. inline elements css

2. internal css

3. external css

1. inline css:- The styling can be done in the element itself using style=" " keyword.

`<p style="color: Black; background-color: Green">`  
It defines `<p>`.

2. Internal css:- The styling for HTML can be done in the HTML document itself but in the head element by using `<style>` tag.

Ex:-  
`<head> <style>`  
`color: red;`  
`</style>`

3. External css:- The styling can be done in the separate CSS file by including adding a link to that CSS file in the HTML document.

In the external file styling can be done same as in Internal CSS but `<style>` tag will not be included.

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

Q. What is the role of the alt attribute in `<img>` tag?  
 → Alt attribute provides information for an image if a image cannot be loaded by a screen reader or used to access the page.

19.

How do you create a numbered list with custom numbering styles in HTML?

→

We can use the "type" attribute to specify a custom numbering style for an ordered list in HTML.

Sol:

To create an numbered list with Roman numerals, You can use the following code.

list type="I">> list item 1 </li> &

<li> Item 2 </li>

<li> Item 3 </li>

</ol>

20.

What is the difference b/w `<script>` & `<script defer>`?

→

The "async" & "defer" attributes in the `<script>` tag are used to control when a JS file is executed.

→ The "async" attribute tells the browser to load & execute the script asynchronously, which means that the script is not downloaded.

In this background of executed as soon as it's ready, without waiting for the rest of page to load.

→ The "defer" attribute tells the browser to load the script in the background but to wait until the rest of the page has finished loading before executing this script.

Q1. What is responsive web design, & why is it important?

→ Responsive web design is an approach to web design that aims to create websites that are optimized for all devices, including desktops, laptops, tablets, & smartphones.

→ It ensures that websites are easy to use & navigate on any device, regardless of the screen size or resolution.

Responsive web design is important because it allows websites to adapt to the user's device, rather than forcing the user to adapt to the website. This results in a better user experience, which can lead to increased engagement, higher conversion rates, better search engine rankings.

Additionally, responsive design can save time & money, since it eliminates the need to create separate websites / apps for different devices.

Q2. How do you make a website responsive using CSS?

→ To make a website responsive using CSS, you can use media queries to apply different styles based on the size of the screen.

Ex:- `/* Default styles */`  
`body { font-size: 16px; }`

`/* Large screens */`  
`body { font-size: 18px; }`

1<sup>o</sup> styles for screens smaller than 768px

@media (max-width: 767px) {

body {

font-size: 14px;

1<sup>o</sup> styles for screens b/w 768px & 992px

@media (min-width: 768px) and (max-width: 991px)

body {

font-size: 16px;

1<sup>o</sup> styles for screens larger than 992px

@media (min-width: 992px) {

body {

font-size: 18px;

Q. what is media query in CSS, & how is it used for responsive design?

→

Media Query is a technique used to apply diff. styles to a web page based on characteristics of the device / screen being used to view it.

→ media queries allow u to target specific devices / screen sizes & apply diff. styles based on those characteristics.

Ex:-

@media screen and (max-width: 768px) {

body {

font-size: 14px;

}

The above style can be applied to screens that are smaller than 768px wide. This allows you to create styles that are optimized for smaller screens, such as smartphones & tablets.

Q. Explain the difference between a fluid layout and a fixed layout in terms of responsiveness.

Fluid layout      fixed layout

It is a type of layout where it uses a fixed units, that uses relative units like pixels, to define such as percentages, to define the width of elements on a web page. This means to define the width of elements on a web page. This means on a web page. A fixed layout does not allow the width of elements on a web page. This means to adjust to different screen sizes, which can be problematic for responsive design. It can be flexible & adjust to different sizes, making it an ideal choice for responsive design.

Fluid layout are often used for designs that are used in conjunction with media queries to create a responsive design that adapts to different devices. They are not suited for responsive design.

Q. How do you make images responsive in CSS?

- To make images responsive in CSS, you can use the 'max-width' property & set it to 100%.
- This will ensure that the image scales down proportionally to fit the width of its container, while maintaining its aspect ratio.

Ex:-

```
img {  
    max-width: 100%;  
    height: auto;
```

26. what are the breakpoints in responsive design, how are they determined?

→ A breakpoint in a responsive design is the "point" at which the website's content & design will adapt in a certain way to provide the best possible user experience.

→ Breakpoints are determined by the designer / developer based on the needs of the website & the devices that it will be viewed on.

→ Typically, breakpoints are chosen based on common screen sizes / device types, such as smartphones, tablets, laptops & desktops.

→ Determined by designer / developer based on different factors, including content of the site, layout needs of site & needs of the user.

Once the breakpoints have been determined, developer will use media queries in CSS to apply different styles to site at each breakpoint, to provide best user experience & ensuring that site is accessible to as many users as possible.

27. How can you hide elements on specific screen sizes using CSS?

→ By using media queries of 'display' property.

Ex:

```
@media(max-width: 600px) {
```

element to hide {

```
display: none;
```

}

y.

28 what is the purpose of the max-width property in responsive CSS?

- The max-width property in responsive CSS is used to set the max width of an element.
- \* It's often used in combination with the width property to create flexible layouts that adapts to dif screen sizes.

: Ex:-

```
img {  
    width: 100%;  
    max-width: 500px;  
}
```

width is set to 100% which means it will expand to fill the width of its container. However max-width is set to 500px. which means the image will never be wider than 500px even if the container is wider.

29 How do you create a responsive navigation menu using CSS?

- By using media query & display property.

1st Default styles for the navigation menu.

```
nav {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
}
```

```
nav ul {  
    list-style: none;  
    padding: 0;  
}
```

```
li {  
    margin: 0;  
    padding: 0;  
}
```

```
display: flex;
```

## nav li {

margin: 0 10px;

4

nav a { text-decoration: none; color: black; }

text-decoration: none; color: black;

color: black; !important; color: black;

3. margin: 0 10px; border-bottom: 1px solid black;

1. media query for screens smaller than 600px

@media (max-width: 600px) {

nav {

flex-direction: column;

list-style-type: none; padding: 0; margin: 0;

nav ul { list-style-type: none; padding: 0; margin: 0; }

flex-direction: column; align-items: center;

align-items: center; justify-content: space-around;

display: none; transform: rotate(-90deg);

3

nav::after {

margin: 10px 0; border-radius: 50%; width: 10px; height: 10px; background-color: black; border: 1px solid black; position: absolute; right: -10px; top: -10px; border-top-left-radius: 0; border-bottom-left-radius: 0; border-top-right-radius: 50%; border-bottom-right-radius: 50%;

1. show the navigation menu when the Hamburger icon is clicked

#menu-toggle:checked + ul {

display: flex; align-items: center; justify-content: space-between; width: fit-content; margin: auto; font-size: 1.2em; font-weight: bold; font-family: inherit; border: 1px solid black; padding: 5px; border-radius: 10px; transition: all 0.3s ease-in-out; background-color: white; position: relative; z-index: 1; background-color: white; border: 1px solid black; padding: 5px; border-radius: 10px; transition: all 0.3s ease-in-out; background-color: white; position: relative; z-index: 1;

4

5

To show the navigation menu on smaller screen we're using a checkbox input with an ID of 'menu-toggle' & a label with a 'for' attribute that matches the ID when the label is clicked, & toggle seu checked attribute

of the checkbox. We're using the `<label>` + `<input>` selector to target the "ul" element that immediately follows the checkbox. When the checkbox is checked, we're setting the display of the "ul" element to `flex`, which shows the navigation menu.

30. Explain the concept of mobile-first design & how it's related to responsive CSS.

→ Mobile-first design: when we start by writing our CSS for mobile devices, then use media queries to add in styling for larger screen sizes.

In general, that means media queries use a `min-width`.

mobile-first design is closely related to responsive CSS, which is the practice of using CSS to create a website or application that adapts to all screen sizes. By designing for mobile first, you can create a design that is optimized for small screens & touch interactions. You can then use CSS to add styles & layout rules that enhance the design on larger screens.

31. What is CSS flexbox, and what problem does it solve?

→ CSS flexbox is a layout module that provides a more efficient way to create flexible, responsive layout without using `float`. It allows to create a flexible container & specify how the child elements should be arranged within that container.

- It solves the problem of creating responsive layout that can adapt to all screen sizes.
- Flex items
- Explain the `flex` box flex containing flex item.
- A flex container is an element that has the 'display:flex' @ 'display:inline-block' property applied to it.
- flex container is an element that establishes a new flex formatting context. Then contains defines the layout rules for the item.
- The flex item is a direct child of a flex container that participates in the layout of that container. The items are arranged to align within the container according to those axes.
- How do you create a flex container in CSS?
- We need to apply 'display:flex' @ 'display:inline-flex' property to an element. This element becomes the flex container as all of its direct children become flex items.
- `Ex:-`
- `<div class="flex-container">`
- `<div class="flex-item"> item1 </div>`
- `<div class="flex-item"> item2 </div>`
- `<div class="flex-item"> item3 </div>`
- `</div>`
- Flex-containers
- display:flex;
- display: inline-block;

34.

what are the main properties used to control the layout in flexbox?

→

'display': This property is used to create a flex container by setting its value to 'flex' or 'inline-flex'.

'flex-direction': 'row' → 'row-reverse', 'column', 'flex-direction': 'column-reverse'.

'justify-content': 'flex-start', 'flex-end', 'center', 'space-around', 'space-between', 'space-around'.

'align-items': 'flex-start', 'flex-end', 'center', 'baseline' @ 'stretch'.

'align-content': 'flex-start', 'flex-end', 'center', 'space-between', 'space-around', 'stretch'.

6. 'flex-wrap': wrap, (items wrap to the next line)

'no-wrap': (items stay on one line)

'wrap-reverse': (items wrap to the next line but in reverse order)

7. 'flex-flow': This property is a shorthand for 'flex-direction' & 'flex-wrap'.

ex: 'flex-flow: row wrap;

28. Explain the difference between `justify-content`, `justify-items`, `align-items`, `flex-direction` properties in flexbox.

→ `justify-content`: Aligns flex items along the main axis of the container. The main axis is horizontal, e.g. in a column-oriented container. The cross axis is vertical.

`align-items`: Aligns flex items along the cross axis in a row-oriented container. The main axis is horizontal.

`flex-direction`: This property is used to control the horizontal alignment of flex items.

The `property` determines how much a flex item should grow relative to the other flex items within the same flex container. It takes a `units` value that represent the proportion available span that the flex item should take up.

→ `flex-shrink`: The property determines how much a flex item should shrink relative to the other flex items within the same flex container.

The `value` here should be less than `flex-grow`. It takes a `units` value that represents the proportion of the available space that the flex item should give up when there isn't enough space.

→ `flex-basis`: The property determines the initial size of a flex item before any available space is distributed among the flex items.

29. How do you align flex items horizontally vertically within a flex container?

By using '`justify-content`' & '`align-items`' properties.

flex-item: 0<sup>th</sup>-child(2)  
order:-1;

卷之三

What are flexbox breakpoints & how can they be used for responsive design?  
flexbox breakpoints are specific points in the width of a web page where the layout of the page changes in response to the size of the screen as window.

use own defined media query in our CSS  
that apply diff styles to our flexbox  
layout based on the size of the viewer  
media queries allow us to define diff  
style for diff screen size devices

flex-containers (flexbox):

- display: flex;
- flex-wrap: wrap;

② media screen & c. min-width: 768px

```
    .flex-container {  
      display: flex;  
      justify-content: space-around;  
    }  
  
    .flex-item {  
      width: 150px;  
      height: 100px;  
      background-color: #ccc;  
      margin: 10px;  
    }  
  
    @media screen (min-width: 1024px) {  
      .flex-item {  
        flex-direction: column;  
        align-items: center;  
      }  
    }
```

and the people of the city in general.

Q3. If we want to uniquely identify an element on a web page, it is a way to

so that it

Line  
Page

classmate

What are HTML

→ HTML attributes are

that can be added to the properties. They provide additional information about the elements, such as its appearance, behavior,

Q7. Explain the difference between global attributes & document specific attributes.

→ \* Global attributes are attributes that can be used with any element.

provide general information about the element and can be used to search for it.

The element using e.g. C#.

Element specific attributes are attributes that can only be used with specific HTML element.

~~so 'very'~~ can only be used with a noun.  
So attributes can used with 'long' & 'short'.

1.2. **Document** **format** **is** **standard** **&** **clear**

→ We can add attributes to an HTML element.

Attribute core mesh is up of a name & value.

~~22:~~ ~~the~~ ~~difference~~ ~~between~~ ~~the~~ ~~two~~ ~~elements~~ ~~is~~ ~~the~~ ~~sum~~ ~~of~~ ~~the~~ ~~two~~ ~~elements~~

Because 'id' attribute is used as a selector in CSS/JS to target a specific element on the page.

What is 'style' class attribute of 'id' attribute us.

→ class attribute is used to group similar elements together so that they can be styled @ once.

In this case, we can use 'class' on the same

'class' name on multiple elements by also specifying 'multiple' 'class' name to single element.

`<h1 class="head">` This is another heading like

`<h2 class="head">`

`<h3 class="head">`

`<h4 class="head">`

`<h5 class="head">`

`<h6 class="head">`

`<h7 class="head">`

`<h8 class="head">`

`<h9 class="head">`

`<h10 class="head">`

`<h11 class="head">`

`<h12 class="head">`

`<h13 class="head">`

`<h14 class="head">`

`<h15 class="head">`

`<h16 class="head">`

Q7. How do you add alternative text to an image using the alt attribute?

→ The alt attribute .. you can include the alternative text in the <img> tag for the image. The alt attribute provides a description of the image for user who are unable to see the image.

Q8. What is the purpose of the target attribute in HTML links & what are the possible values?

→ The 'target' attribute in HTML links is used to specify where the linked document will be displayed when the user clicks on the link.

→ possible values:

1. '\_self': The linked document will be displayed in the same frame @ window on the current

document.

2. '\_blank': The linked document will be displayed in a new window/tab.

3. '\_parent': The linked document will be displayed in the parent frame or window of the current frame/window.

4. '\_top': The linked document will be displayed in the full body of the window, replacing any frames that may be present.

Q9. How do you use the src attribute to embed an external resource such as an image/video in the HTML

resource such as an image/video in the HTML, we can use the src attribute

to embed an external resource such as an image/video in the appropriate HTML element.

ex: ``

out: ``

Ex: `<video src="https://www.example.com/videos/example">`

50. What is the purpose of the `disabled` attribute and how is it used in HTML form elements?

→ The 'disabled' attribute in HTML is used to disable a form element so that it cannot be interacted with by the user.

Code:

```
<label for="username">Username:</label>  
<input type="text" id="username" name="username" disabled>
```

51. Is there any relation b/w Java & Javascript

→ Both are two different programming languages that have different syntax and are used for different purposes.

- Java is a general-purpose programming language that is used to create desktop & mobile, web application.
- JS is a client-side scripting language that is used for creating interactive web pages & web applications.

52. Is Javascript a compiled / interpreted language?

→ JS is an interpreted language. That means JS code is executed directly by the browser's runtime environment without the need for a compilation step.

53. Is JavaScript a case-sensitive language?

→ Yes, that means variable names, function names & other identifiers in JS are distinguished by their capitalization.

Ex: Variable Name 'myVariable' is not same as myvariable.

54. what is node.js?

Node.js is an open-source, cross-platform JS

runtime environment that allows developers to run JS code outside of a web browser.

55. what is hoisting?

→ It is a JS mechanism where variable & function declaration are moved to the top of their scope before code is executed.

56. what is the difference b/w let and var?

→ → 'let' is block-scoped. That means variable declared with 'let' are only accessible within the block in which they are declared in, which can be a smaller scope than a function. 'let' variables cannot be redeclared within their scope, but they can be reassigned.

→ "var" is <sup>(global)</sup> function-scoped, which means that variable declared with 'var' are accessible within the function. They are declared <sup>(global)</sup> outside the function if they are declared in the global scope. 'var' variable can also be redeclared & reassigned within their scope.

57. what is the diff b/w undeclared & undefined variable.

→ \* Undeclared Variable is a variable that has not been declared in any scope. (Variable does not exist.)

\* undefined variable is a variable that has been declared but has not been assigned a value. Trying to access an undefined variable will not result in an error, but will return the value "undefined".

Ex:- let x;  
In main body console.log(x); Output: undefined;

58.

What is Scope in JS? refer to the area of use.

→ Scope in JS refers to the area of use.

Where a variable @ function is accessible.

→ Global scope:-

It refers to variable function that are accessible anywhere in code. Thus variable can be declared outside of any function and can be accessed from anywhere in the code.

```
let x=1;
function addOne(x){
```

x++;

return x+1;

}

addOne();

console.log(x);

Output:-

2

Variable

addOne()

return x+1;

function addOne(x){

x++;

return x+1;

}

addOne();

Output:-

1

function addOne(x){

x++;

return x+1;

}

addOne();

Output:-

2

→ Local Scope :- It refers to variable function that are only accessible from within a certain function or block. These variables are declared

inside a function block & can be accessed from within a function block.

or:-

```
function addOne(x){
```

x++;

return x+1;

}

addOne();

Output:-

1

→ Function addOne() :-

```
function addOne(x){
```

x++;

return x+1;

}

addOne();

Output:-

2

→ console.log(x);

```
function addOne(x){
```

x++;

return x+1;

}

addOne();

Output:-

1

59. What are reserved words? Can we use reserved words as identifier.

→ Reserved words are words that have a special meaning in JavaScript and cannot be used as variable name, function name

or any other identifiers in code.

→ No. If we try to use a reserved word as a identifier we will get a Syntax error.

Ex:- let var1;

var1

Output:-

Syntax Error

→ Why do you need 'strict' mode? How do you declare strict mode?

→ Strict mode is a feature in JS that allows to place code in a 'strict' operating context which can help to write more securely optimized code.

Benefits:-

→ Preventing the use of undeclared variables.

→ Preventing the use of variable functions within duplicated name.

→ To enable 'strict mode' in JS code we can add the string "use strict" at the beginning of a script @ function. This will cause certain actions that would otherwise be allowed (such as using undeclared variables) to throw errors, making it easier to catch bugs in the code.

→ Using "use strict" at the beginning of a script

→ Undeclared variables will throw errors, making it easier to catch bugs in the code.

→ Using "use strict" at the beginning of a script

→ Undeclared variables will throw errors, making it easier to catch bugs in the code.

→ Using "use strict" at the beginning of a script

→ Undeclared variables will throw errors, making it easier to catch bugs in the code.

→ Using "use strict" at the beginning of a script

→ Undeclared variables will throw errors, making it easier to catch bugs in the code.

→ Using "use strict" at the beginning of a script

→ Undeclared variables will throw errors, making it easier to catch bugs in the code.

61. what are global variables?

→ Global variables are the variables which can be declared or defined outside of all the functions. Those variables can be accessible through the program irrespective of the braces. Generally var is the keyword used to declare a variable globally in Javascript.

62. what are the problems with global variables?

→ Global variable can cause a number of problems in a program.

Ex: they can be easily overwritten or modified accidentally, leading to unexpected behaviour.

They can also make it difficult to track down bugs & can lead to naming conflicts if multiple variable with the same name are used across diff parts of a program.

63. what is NaN property?

→ NaN stands for "not a number" and is a value that is returned when a mathematical operation or a function fails to produce a valid number.

Ex:- If we divide 10 by 0

Dividing by zero or taking the square root

of a -ve number will result in NaN.

64. what is the purpose of delete operator?

→ The delete operator is used to remove a property from an object. It can be used to remove a specific key-value pair from an

object / to remove the method / property from an object's prototype. The delete operator can also be used to free up memory that is no longer needed.

Ques. what is the difference between null and undefined

→ null & undefined are both special values.

-> JavaScript has indicates the absence of a value whereas null are used to slightly different

to undefined in case to indicate that a variable has been declared but has not been assigned a value. Eg. indicates that a value is missing but not been set yet.

\* null on the other hand is used to indicate that a variable has been explicitly assigned a value of null. Eg. used to indicate intentional absence of a value.

Ques. what are bitwise operator available in JavaScript

→ 6 Bitwise operator available in JavaScript

(i) AND (&&)

(ii) OR (|)

(iii) XOR (^)

(iv) NOT (~)

(v) leftShift (ll)

(vi) RightShift (rr)

Ques. I declare let and const variables

→ No, you cannot redeclare let & const variables.

Attempting to do so will result in Syntax error

Ques. Does const variable makes the value immutable.

→ while const variables themselves are not immutable, the values they are assigned to cannot be changed. This means that you cannot assign a new value to a const variable once it has been initialized.

Ques. what is ES6? list down some of the features of ES6.

→ ES6 is also known as ECMAScript 2015, major update to the JavaScript language that was released in 2015. Some of the features introduced in ES6 include let & const declarations, arrow functions, template literals, classes in modules.

Ques. what are the possible ways to create objects in JavaScript

→ There possible ways to create objects in JS includes object literals, constructor functions, & Object.create() method, you can also create objects using class syntax in ES6.

Ques. what is the difference between slice & splice.

→ The slice() method returns a new array containing a copy of elements from the original array.

\* The splice() method, can be used to add / delete elements from an array in place.

Ques. what is the difference between

a. == and === operators

b. = and =

c. % = and =

→ a. The == operator compares only the values for equality but performs type coercion if the types of the two values are different.

\* The === operator compares both value and datatype. dont perform type coercion.

b. \* The `=` operator is used to assign a value to a variable.

c. The `==` operator is used to compare two values for equality.

c. The `%` operator is used to perform the remaindering division operation on a variable.

\* The `=>` operator is used to assign a value to a variable.

13. what is a regular order function?

→ It is a function that takes one or more functions as arguments and/or returns a function itself, its result `(@)`. Function which will return other function.

14. what is the currying function?

→ Currying is a technique in functional programming, where a function that takes multiple arguments is transformed into a series of functions that take one argument each. This allows the resulting function to be composed & reused more easily.

15. what are arrow functions?

→ Arrow function is a shorthand syntax for creating functions in JavaScript. They are similar to traditional function expression, but have a more concise syntax & do not bind their own this value (while working with arrays & objects).

16. what is a spread operator?

→ The spread operator is a syntax for spreading the contents of an iterable (such as array or object) into a list of arguments / elements. It is denoted by (...) dots & used in function calls, array literals,

17.

what is a rest parameter?

→ It is a syntax for representing an indefinite number of arguments as an array. It is denoted by (...) followed by parameter name, must be the last parameter in a function definition. The rest parameter allows a function to accept any number of arguments, which can be useful when working with variable-length argument lists.

18. what happens if you do not use rest parameter as a last argument?

→ If you do not use the rest parameter as the last argument in a function definition, you will receive a syntax error. The rest parameter must always be the last parameter in a function definition, as it represents all the remaining arguments passed to the function.

19. what are regular expressions patterns?

→ Regular expression patterns are a language of character sets, defined in regex. They are used to match and manipulate text in strings. They can be used to perform tasks such as decomposing, replacing & validating text.

Regular expression patterns, are defined using special syntax & can be used with various string methods in JavaScript.

Q. What is a regular expression?

- It is also called as regex / regexpr
- It is a pattern used to match & manipulate text in strings. It is a powerful tool for working with text data; & can be used for tasks such as searching, replacing & validating text.

Q. How do you search a string for a pattern?

- We use 'search()' method to search for a pattern in string. The 'search()' method returns the index of first occurrence of pattern in string or -1 if the pattern not found.

Ex. `String.search(pattern)`

Q. What is the purpose of switch case?

- Switch Statement is a control structure in JS that focus you to execute different blocks of code based on off condition.

Q. What are the conventions to be followed for the usage of switch case?

- Use identification to show the structure of switch statement. Indent each case block to make it clear which code is executed for each case (use "===" instead of "=" to compare the expression with value)
- Use 'break' statement to exit from each case in switch
- Use a 'default' case to handle unexpected values & use constant variables by case value rather than properties. This make your code more readable & easier to understand.

Ex: what are primitive data type  
Data type which are immutable, meaning they cannot be changed once they are created.  
Ex: numbers, strings, boolean, null & undefined.

Ex: what are different way to access object properties.

- 1. use dot notation, which involves using a period followed by the name of the property.

If you have an object called "person" with a property called "name", you can access it by person.name.

2. Bracket notation: which involves using square brackets property "name" of the object person["name"]

Ex: what are function parameters null.

Ex: function parameters are optional. If a function is called with fewer arguments than it expects, the missing arguments are set to undefined.

2. Function parameters can have default value if a parameter is not provided. The default value is used instead.

3. It can be destructured. This allow you to extract values from objects or arrays, we sum as parameters.

4. Function parameters can be rest parameter.

5. It can have type annotation to allow you to specify the expected type of parameter which can help catch errors at runtime.

6. It can have default values by type annotation.

Ex: Different ways which create infinite loop.  
→ T in a loop that never terminates.

(1). Using while loop with condition that is always true.

```
while(true){}
```

(2). or for loop with no exit condition.

```
for(;;){}
```

(3). using recursion without a base case.

```
infiniteLoop();
```

(4). using a do while loop with condition that always true.

```
do{}
```

```
while(true);
```

Ex: what are template literals?

Template literals are a way to use strings in multiline string.

Ex: `(`` ${}`` ${}`` ${}`` ${})``

placeholder

Ex: what are default values in destructuring assignment?

It is a way to extract values from objects, arrays and assign them to variables you can also provide default values for variables in case the value being destructured is undefined @null.

Ex: `const [name = 'John', age = 20] = ['Eric'];`

console.log(name);

1.3.6

console.log(age);

1.3.6

Q10) How do you swap variables in destructuring assignment?

→ Swapping variables using destructuring assignment can be useful when you want to avoid assigning a temporary variable to store one of the values while swapping the value of two variables.

Ex: `let a = 1,`

`b = 2,`

`[a, b] = [b, a];`

`console.log(a); // 2`

`console.log(b); // 1`

Q11) Is that possible to use expressions in switch cases?

→ Yes, it's possible to use expressions in switch cases, the expression is evaluated once & compared to each case statement until a match is found.

Q12) what are the differences b/w for...in statements, for...of:

→ It will loop through the elements of the sequence, you can access the elements using for...of.

For...in: we can access the position / index of the element.

for...of:

- Q13) Difference b/w arguments - object and rest parameter  
 → The arguments object is an array-like object that contains the values of all the arguments passed to a function.
- \* Rest parameter is a syntax for representing an indefinite number of arguments as an array.
  - \* Difference b/w spread operator & rest operator  
 → Both use the same symbol (3 dots), but diff in ways spread operator is used to spread the elements of an array / an iterable object into a new array / function.
  - \* Rest operators is used to represent an indefinite number of arguments as an array.
- Q14) Explain all the array methods, what are the outputs & whether the method modifies the original array  
 → Array methods in JavaScript includes:
  - \* push(), pop(), shift(), unshift(), slice(), splice(), concat(), join(), find(), lastIndexOf(), forEach(), map(), filter(), reduce & more.
  - \* The some() array methods returns will modify original array. Such as push(), pop(), shift(), unshift(), splice().
  - \* And others will not modify the original array.

method	Def & output	modifi array
(1) <u>push</u> :	adds one/more elements to the end of an arry returns new length of the array.	yes
2. <u>pop()</u>	removes last ele of arry	yes
3. <u>shift</u> :	removes 1 <sup>st</sup> elem of an arry array returns that ele.	yes
4. <u>unshift</u> :	adds 1/more elements to begin of arry array returns new length of the array.	yes
5. <u>concat</u> :	combines 2/more array into a new array & doesn't modify original array returns the new array.	returns new array
6. <u>sliced</u> :	returns a new array that contains a copy of the arry from begin to ending index.	so - 11 -
7. <u>splice</u> :	Changes the contents of an array by removing / replacing existing elements & adding new ele. It returns an array containing the removed elements.	yes & return an array with the removed elements if any.

8. - `indexOf()` returns the first index at which a given element can be found in an array. (or) -1 if not present.
9. `forEach()` executes a provided func once for each array element. it does not return anything.
10. `map()` creates a new array with the results of calling a provided func on every elements.
11. `filter()` creates a new array with all elements that pass the test implemented by the provided func.
12. `reduce()`: applies a func against an accumulator & each element in the array to reduce it to a single value.