Interview Questions

1) What is HTML and what is its Purpose.

→ HTML stands for Hyper Text Markup Language. It is the standard markup language for web page creation. It describes the structure of web page.

It designs the basic layout and formatting of web pages. HTML is made up of elements or tags and attributes which works together to identify the document parts and tell the browser how to display them.

2) Difference between HTML and XHTML

→

HTML	XHTML
• HTML stands for Hyper Text markup language	• It stands for Extensible Hyper text markup language
• The format is document file format.	• The format is a markup language
• It is case insensitive	• It is case sensitive
• Doctype is not necessary to write at the top	• Doctype is very necessary to write at the top of the file
• It is not necessary to close the tags in the order they are opened.	• It is necessary to close the tags in order they are opened
• While using attribute it is not necessary to mention quotes Ex: 	• While using attributes it is mandatory to mention quotes. Ex: <br id="br1">
• File name Extension used are .html, .htm.	• File Name extension are .xhtml, .xht, .xml.

3) What are the new features introduced in HTML5 ?

→ ① Video & Audio

↳ embed any audio and video in the webpage

Ex:- <video controls preload>
<source src="...." type="video/mp4" />
 .....
</video>

② Svg

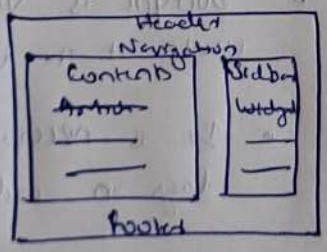
The svg element is used for the scalable vector graphics in html5, we can create any kind of graphical animation with svg tag, such as circle, square, rectangle etc.

Ex: <svg width="100" height="100">
<circle cx="50" cy="50" r="40" stroke="green"
stroke-width="4" fill="yellow" />
</svg>

o/p => 

③ Semantic Tags

Semantic Tags are very useful to make human and search engine friendly websites and apps.



Semantic Layout Structure

a) article

To distribute any independent content in a webpage you can use article tag.

<article>
<h1> ... </h1>
 .....
</article>

b) nav

nav element is used for the part of a internet site that links to different pages of the website.

Ex: <nav>

 HTML

 CSS

"

"

</nav>

c) header

header element can used to institution collectively information such as brand logo, navigation object & search box

Ex: <header>

<img src = "Company-logo.png" >

<nav>

  Login

  SignUp

</nav>

</header>

d) footer

footer is a tag used to define the footer of a document or a section. Such as contain information in footer. It can have one or more footer.

< footer>

 ...

 Contact element : ...

</footer>

e) section

Section tag is used to divide a document or in a page or sections. for ex:- headers, footers, latest news etc

```
<article>
    <section>
        <h1> </h1>
        <p> </p>
    </section>
    <section>
        <p> </p>
    </section>
</article>
```

f) aside

most of the web page have columns, sidebars are present.

Ex.
```
<aside>
    <h1> </h1>
    <p> </p>
</aside>
```

** It can also be divided into multiple sections **

g) main

main is a tag which is used to contain main content of the page and it contains unique content. More than one (1) main tag is not accepted in a document. It does not include navigation bar, header and footers.

Ex.
```
<main>
    <h1> .... </h1>
    <p> .... </p>
    <article>
        <h2> .... </h2>
        <p> ... </p>
    </article>
</main>
```

h) figure and figcaption

It has become semantically possible to insert an image in a page with its caption

```
<figure>
    <img src="..." alt="Arrow A" />
    <figcaption>
        <p> .... </p>
    </figcaption>
</figure>
```

i) address

```
<address>
    Written by <a href="...@example.com"> ABC </a> <br>
    Visit us at: <br>
    Example.com <br>
</address>
```

j) details

Additional details can be given which will be visible when user clicks on a small triangle shape

```
<details>
    <summary> Epcot Center </summary>
    <p> ... </p>
</details>
```

k) Summary

```
<details>
    <summary> Epcot Center </summary>
    <p> ... </p>
</details>
```

l) time

```
<p> Opens from <time> 10:00 </time> to <time> 21:00 </time>
    every weekday. </p>

<p> I have a date on <time datetime="2008-02-10 23:00">
    Valentines day </time>. </p>
```

(4) HTML5 APIs

a) Canvas API

Used to draw Images.

Ex:-

```
<canvas id = "myCanvas"> </canvas>
<script>
    var canvas = document.getElementById ("myCanvas");
    var ctx = canvas.getContext ("2d")
    ctx.fillStyle = "#FF0000";
    ctx.fillRect (0, 0, 80, 80);
</script>
```

O/P 

b) geolocation API

It is used to locate a user's position.

Ex.

```
<p> . . . . </p>
<button onclick = "getLocation()"> Try It </button>
<p id = "demo"> </p>
<script>
    var x = document.getElementById ("demo")
    function getLocation() {
        if (navigator.geolocation) {
            navigator.geolocation.getCurrentPosition (showPosition);
        }
        else {
            x.innerHTML = "Geolocation is not supported by browser.";
        }
    }
    function showPosition (position) {
        x.innerHTML = "Latitude:" + position.coords.latitude +
        "<br>Longitude:" + position.coords.longitude;
    }
</script>
```

4) drag / drop API

In HTML any element can be dragged and dropped.

Ex: /* CSS */

```
#div1{
    width: 350px;
    height: 70px;
    padding: 10px;
    border: 1px solid #aaa;
}
/* JavaScript */
function allowDrop(ev){
    ev.preventDefault();
}
function drag(ev){
    ev.preventDefault();
    var data = ev.dataTransfer.getData("text");
    ev.target.appendChild(document.getElementById(data));
}
/* HTML */
<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)">
</div>
<img id="drag1" src="admec_logo.jpg" draggable="true"
    ondragstart="drag(event)" width="336" height="69" >
```

(5) New types for input tags

Input is an attribute which is an old attribute but in HTML, it is reintroduced with a new values like email, month, number, range, Scan etc.

a) Email Attribute

When in a form we write the type as "email", by itself gets an instruction from the code to write an email. This was not possible in earlier browsers

```
<form action = " " method = "get">
    <label for = "email"> Email : </label>
    <input id = "email" name = "email" type = "email" />
    <button type = "submit"> Submit form </button>
</form>
```

⑥ Content Editable

It is an attribute which is used to permit the user to edit the content.

```
<blockquote contenteditable = "true">
    <p> ..... </p>
</blockquote>

<cite contenteditable = "true"> -- Write your own name </cite>
```

⑦ Progress Bar

This tag is used to show progress of an task during execution of that. Progress ~~bar~~ tag can be used with conjunction of Javascript. It is like progress bar.

```
<progress value = "22" max = "100"> </progress>
```

(8) Placeholder in form

It gives hint to what to be inserted on a particular field of the form.

<input text = "text" placeholder = "Enter your Name" />

(9) Required Attribute.

It helps to gather information mandatorily. Without this we cannot submit the form.

<input type = "text" placeholder = "Enter name" required />

(10) Preload Videos

It specifies the way to upload the video along with loading of the page. Value need to added auto / metadata / none

<video controls preload = "none">

<source src = "ABC.mp4" type = "video/mp4">

Your browser supports any video tag

</video>

(11) Regular Expressions

We can add particular pattern as an input.

[a-z], [A-Z], [a-zA-Z], [0-9], [a-zA-Z0-9] & so on.

(12) Accessibility:

HTML5 provides best of accessibility features to a website. This has made use of website simplified. Such as in forms validation is best example of providing accessibility. Labels should be properly inserted.

(13) Inline Elements

To keep code up to mark, semantically these inline elements help a lot.

- mark - It highlights content that is marked in some or other way
- time - This helps in adding current time as well as date to the page
- meter - It helps in indicating how much space in storage in storage disk is there.
- progress bar - It helps in knowing progress of task.

(14) Dynamic Page Support

Nowadays there is need of dynamic as well as interactive websites rather than static websites. Here are common features that provide dynamic touch to website:-

copy — (1)

(15) Cryptographic Nonces

In newer version of HTML, we can add cryptographic nonces to all styles as well as scripts, we can use nonce script and style tag. This nonce tag basically generates a random number for one time use-only.
So it is generated each time when page refreshes. It prevents the scripts to website.

```
<script nonce = 'asdfghjkl'>
    // Some javascript
</script>
```

(16) Reverse links

It basically allows the user again back link, and another tags.
It explains relationship when current document as well as linked document but in other direction.

(17) Zero width Images

This feature is helpful when there is no need to show them to users such as while loading image files as otherwise it takes more space. It is advisable to use zero width images with alt tag that is

```
<img src = '...jpg' width = "0" height = "0" alt = "" >
```

4) How do you include comments in HTML?

→ Comment in HTML starts with <!-- and ends with -->

5) Explain difference between and tags

div

- The tag is a block level element
- It is best to attach it to a section of a web page
- It accepts align attribute
- This tag should be used to wrap a section, for highlighting that section

```
<body>
  <div> div tag </div>
  ..
  ..
</body>
```

Span

- The tag is an inline element
- It is best to attach a CSS to a small section of line in a web page
- It doesn't accept align attribute
- This tag should be used to wrap any specific word that you want to highlight in your webpage

```
<body>
  <p> <span style = background color : red >
    ABC </span> ---- </p>
</body>
```

6) What are Semantic elements in HTML 5 and why are they important?

A Semantic elements clearly describes its meaning to both the browser and the developer.

Article, aside, header, footer, details, figcaption, figure, main, mark, nav, section, summary, time

It is important to give correct interpretation of the content

7) What is the purpose of <header>, <nav>, <section> and <footer> tags in HTML5?

→ • <header>

The header element represents a container for introductory content or a set of navigation links. A header element typically contains one or more heading (h1-h6), logo or icon, and authorship information.

Note: You can have several <header> elements in one HTML document.

```
<header>
    <h1> .... </h1>
    <h2> .... </h2>
</header>
```

• <footer>

The <footer> element defines a footer for a document or section. It typically contains copyright information, contact details, sitemap, back to its top links, and related documents.

You can have several <footer> elements in one Document.

```
<footer>
    <p> .... </p>
    <p> <a href = "mailto.g@example.com"> hg@gmail.com </a> </p>
</footer>
```

• <nav>

The <nav> elements defines a set of navigation links.

```
<nav>
    <a href = "/html/"> HTML </a> |
    <a href = "/css/"> CSS </a> |
</nav>
```

• <Section>

The section element defines a section in a document. A semantic element can be used for chapters, Introduction, News items, Contact inform

```
<section>
    <p> .... </p>
    <h1> .... </h1>
</section>
<section>
    <h2> .... </h2>
    <p> .... </p>
</section>
```

> How do you create a hyperlink in a HTML?

TO create a hyper link in HTML we use <a> and tags, which are tags used to define the links. The <a> tag indicates where the hyperlink starts and tag indicates where it ends. Add the URL for link in .

Note: <a>... tags should be used in body only.

```
<!DOCTYPE Html>
<html>
<head>
...
</head>
<body>
   ....
   <a href = 'url'> text </a>
   ....
</body>
</html>
```

What is the difference between <ol> and <ul> elements?

Ordered list	Unordered list
• It is used to create an ordered list of items.	• It is used to create an unordered list of items
• The list items will be marked with numbers by default.	• The list items will be marked with bullets by default.

```
<ol>
  <li>...</li>
  <li>...</li>
</ol>
```

```
<ul>
  <li>...</li>
  ..
</ul>
```

10) How do you embed a Image in a html?

→ The <img> tag is used to embed a image in a web page.

<img src = "cars-logo.jpg" alt = "Cars" >

11) Explain the difference between and tags?

→ as representing "strong importance to its contents". While is used to change the meaning of a sentence as spoken emphasis does ("I love carrots" vs. "I love carrots"), is used to give portions of a sentence added importance. Both and can be nested to increase relative degree of importance. (- tags shows as italic)

12) How do you create a table in HTML?

→ HTML Table allow web developers to arrange data into rows & colu...

<tr> - table row <td> - table data <th> - table header

<th> - Define a header cell in table

<td> - Defines a cell in table

<tr> - Defines a row in a table

Eg.-

```
<table>
  <tr>
    <th> ... </th>
    <th> ... </th>
  </tr>
  <tr>
    <td> Email </td>
    <td> ... </td>
  </tr>
</table>
```

What is the purpose of <form> tag in HTML and how do you create a form?

The <form> tag is used to collect information that visitor provide.

The <form> tag is used to create a HTML form for user input.

```
<form action="/action.php" method="get">
    <label for="fname">First Name:</label>
    <input type="text" id="fname" name="fname"><br><br>
    <input type="checkbox" name="v1" value="Bike">
    <label for="v1"> I have a bike</label><br>
    <input type="radio" id="html" name="fav_lang" value="HTML">
    <label for="html">HTML</label>
    <input type="submit" value="Submit">
</form>
```

What are some new input types introduced in HTML?

Email address field

When this type is used, the user is required to type a valid email address into the field otherwise it throws an error.

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

Search field

Search fields are intended to be used to create a search boxes on pages and apps. This type of field is set by using value search for type attribute.

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

Phone number field

A special field for filling in phone number can be created using tel as value of type attribute.

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

Url field

A special type of field or entering a URL's can be created using value url for type attribute

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

Numeric field

Controls for entering number can be created with <input> type of number

```
<input type="number" name="age" id="age" min="1" max="10" step="2" />
```

Slider controls

A slidebar is created using <input> with a type attribute set to value range. Therefore, they are used to pick a number.

```
<input
    type="range"
    name="price"
    id="price"
    min="5000"
    max="500000"
    step="100"
    value="250000" />
<Output class="price-output" for="price"></Output>
```

Date and time pickers

datetime-local – creates a widget to display and pick up a date and time with no specific time zone.

```
<input type="datetime-local" name="datetime" id="datetime" />
```

month → select month with a year.

```
<input type="month" name="month" id="month" />
```

time → widget to display and pickup a time.

```
<input type="time" id="time" name="time" />
```

week → display week number & year.

```
<input type="week" id="week" name="week" />
```

Constraining date/time values

<input type="date" name="ABC" min="2013-06-01" max="2013-08-31" step="7" id="myABC" />

Color picker Controls

A color control can be created using <input> element with its type attribute

<input type="color" name="color" id="color" />

How do you include audio and video content in HTML?

<audio>
<source src="file_name" type="audio/file_type">
</audio>

Attributes of audio tag

autoplay – when is page is loaded it autoplays
controls – it displays audio controls
loop – it will start audio again when finished
muted – when page is loaded it is automatically muted
preload – Audio will be loaded when page is ready.
src – url – It specifies the URL of audio file

Ex: <audio src="./ten.mp3" controls></audio>

<video>
<source src="file_name" type="video_file_type">
</video>

Ex. <video controls>
<source src="....mp3">
</video>

11) What is purpose of <iframe> tag and how it is used?

→ The iframe in HTML stands for inline frame. An iframe is used to embed another document within its current HTML document.

Syntax:-

<iframe src="url" title="description"></iframe>

Ex- <!DOCTYPE html>
<html>
<body>
<h1>...</h1>
<h2>...</h2>
...
<iframe src="" height="300" width="400"
style="border:none">
</iframe>
</body>
</html>

12) How do you added CSS style to HTML element?

→ It can be added in 3 ways:

- Inline - by using style attribute in html element
- Internal - by using <style> element in <head> section
- External - by using a <link> element to link external CSS

Inline

<h1 style="color:blue;">A blue Heading</h1>

Internal

<style>
...
</style>

External

<head>
<link rel="stylesheet" href="style.css">
</head>

) What is the role of alt attributes in <img> tags ?

The alt attributes provides alternate information for an image if a user for some reason cannot view it (because of slow connection, an error in src attribute, or if the user uses a screen reader.

```
<img src = "smyl.jpg" alt = "Smiley face" >
```

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

```
<!DOCTYPE html>
<html>
<head>
    <title> .... </title>
    <style>
        ol{
            list style : none;
        } counter-reset : num;

        ol li{
            counter-increment : num;
        }
        ol li::before{
            content : counter(num)". ";
            color:green;
        }
    </style>
    </head>
    <body>
        <ol>
            <li> .... </li>
            :
            :
        </ol>
    </body>
</html>
```

20) What is the difference between <script async> and <script defer>?

<script async>	<script defer>
- Loads with low priority without rendering blocking.	- Loads with low priority without blocking.
- It executes immediately when loaded.	- It doesnot executes immediately when loaded
- It doesn't executes just before the DomContentLoaded event.	- It executes just before the DomContentLoaded event.
- Executes in no partial order.	- Doesn't executes in no partial order
- Doesn't executes in sequence	- Executes in sequence

21) What is responsive Web Design and why it is important?

→ It is an approach to web design that aims to make web pages render well on a variety of devices and window or screen sizes.

It is important to increase reach of customers and clients on smaller devices like tablets and Smartphones. It offers an optimized browsing experience.

22) How do you make a website responsive using CSS?

→ Media query is provided by CSS to achieve the concept of responsiveness. This is a way to conditionally apply CSS rules.

```
@ media (query){
    // CSS rules
}
```

@ media rule indicates media query using different styles can be applied based on media types, screen sizes and orientation. Media queries can be used to specify certain media types such as speech, screen, print and certain media features such as min-width,

```
@ media screen and (max-width : 768 px) {
    .content {
        width : 100%.
    }
}
```

What is media query in CSS and how is it used for responsive design?

Media queries are a key part of responsive web design, as they allow you to create different layouts depending on size of viewport, but they can also be used to detect other things about environment your site is running on.

```
@ media only screen and (max-width : 600 px) {
    body {
        background-color : light blue;
    }
}
```

Explain the difference between fluid layout and fixed layout in terms of responsiveness?

Fluid layout - This uses the bootstrap.container-fluid class of layout.

This layout uses proportional values such as measuring units for a block of content, images, or any item. Used for creating an element that is 100% wider and covers all screen widths. Fluid layout continously resizes as you change the width of your browser by any amount, leaving no extra empty space on sides. Hence it is named "fluid layout"

Fixed layout - This uses the bootstrap.container class for layout. The fixed-layout has specific pixel width values the changes its width value with the help of media queries. It provides a responsive fixed-width container. Fixed layout resizes in chunks at several certain width as pixels values are specified.

25) How do you make images responsive in CSS?

→ When you upload an image to your website, it has a default width and height. You can change them both with CSS.

To make image responsive, you need to give a new value to its width property. Then height will be adjusted automatically

```
//html
<img src="nature.jpg" alt="Nature" class="responsive">

//css
① .responsive {
      width: 100%;
      height: auto;
  }

② .responsive {
      max-width: 100%;
      height: auto;
  }

③ .responsive {
      width: 100%;
      max-width: 400px;
      height: auto;
  }
```

26) What are breakpoints in responsive design, and how they are determined?

→ CSS breakpoints are the points added in the code, the website content responds to these points depending on the device screen size.

7) How can you hide elements on specific screen sizes using CSS?

→ To hide an element in a responsive layout, we need to use CSS display property set to its "none" value along with @media rule. The content of second <p> element having a "hidden mobile" class be hidden on devices smaller than 767 px.

Ex. For hiding an element on extra small devices

```
<!DOCTYPE html>
<html>
  <head>
  <meta name="viewport" content="width=device-width, initial-scale=1,
        maximum-scale=1, user-scalable=no" />
<style>
  @media (max-width: 767 px) {
      .hidden mobile {
          display: none;
      }
  }
</style>
</head>
<body>
    <h1> ... </h1>
    <p> .... </p>
    <p class="hidden-mobile"> .... </p>
    <p> .... </p>
</body>
</html>
```

Ex: To hide an element on small devices.

```
@media (max-width: 576 px){
    .element {
        display: none;
    }
}
```

```
<p class="class">
```

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

→ The max-width property defines the maximum width of an element. If the content is larger that maximum width, it will automatically change height of element. If content is smaller, the property has no effect.

Syntax:

max-width : none | Length | initial | inherit | '%' ;

Ex:-

```
p.ex1 {
    max-width: 50%;
}
```

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

→

```
<link rel="stylesheet" href="ABC.css">

<div class="top nav" id="myTopNav">
    <a href="#home" class="active">Home</a>
    <a href="#news">News</a>
    <a href="#contact">Contact</a>
    <a href="#about">About</a>
    <a href="javascript:void(0);" class="icon" onclick="myFunction()">
        <i class="fa fa-bars"></i>
    </a>
</div>
```

```
.topnav {
    background-color: #333;
    overflow: hidden;
}

.topnav a {
    float: left;
    display: block;
    color: #f2f2f2;
    text-align: center;
    padding: 14px 16px;
    font-size: 17px;
}

.topnav a:hover {
    background-color: #ddd;
    color: black;
}

.topnav a.active {
    background-color: #04AA6D;
    color: white;
}

.topnav .icon {
    display: none;
}
```

1) Explain the concept of mobile-first design and how it relates to responsive CSS?

→ Mobile-first design is process of planning and developing a website keeping in mind the mobile users first. Then we can scale it upto desktops and other devices.

A mobile responsive website is a website that is capable of adapting its content based on the device it is being viewed on.

2) What is CSS Flexbox and what problem does it solve?

Flexbox is a one-dimensional layout method for arranging items in rows or columns. Items flex (expand) to fill additional space or shrink to fit in smaller spaces.

It has the ability to fill the extra spaces without the need to use Javascripts.

3) Explain the difference between flex container & flex items?

A flex container in an HTML element whose display property's value is flex or inline-flex. Flex items are the direct children of a flex container.

CSS Flex Container (Parent)

```
.flex-container {
    display: flex;
    flex-direction: column;
                    column-reverse;
                    row;
                    row-reverse;
```

Properties
flex-direction:
flex-wrap:
flex-flow:
justify-content:
align-items:
align-contents:

Flex Items

Child Items/Elements

The direct child elements of a flex container automatically become flexible (flex) items.

Flex item properties are:-

- order
- flex-grow
- flex-shrink
- flex-basis
- flex
- align-self

32) How do you create a flex container in CSS?

→ It is an HTML element whose display property's value is flex or inline-flex.

① flex-direction – it defines the direction the container wants to stack the flex items

```
.flex-container {
    display: flex;
    flex-direction: column;
}
```

② flex-wrap – It specifies whether the flex items should wrap or

```
.flex-container {
    display: flex;
    flex-wrap: wrap;
}
```

③ flex-flow – It is property of setting both the flex-direction and flex-wrap properties

```
.flex-container {
    display: flex;
    flex-flow: row wrap;
}
```

④ justify-content – It is used to align the flex items.

```
.flex-container {
    display: flex;
    justify-content: center;
}
```

⑤ align-items – it is the property used to align the flex items.

```
.flex-container {
    display: flex;
    height: 200 px;
    align-items: center;
}
```

3) What are the main properties used to control the layout in flex box?

flex-direction => Specifies the direction of flexible items inside a flex container.

flex-flow => A shorthand property for flex-direction and flex-wrap.

flex-wrap => Specifies whether the flex items should wrap or not, if there is not enough room for them on one flex line

justify-content => It is used to align the flex line

align-items => It is the property used to align the flex items

5) What is the purpose of flex-grow, flex-shrink, and flex-basis properties?

```
<style>
#main {
    width: 350 px;
    height: 100 px;
    border: 1px solid blue;
} display: flex;

#main div {
    flex-grow: 1;
    flex-shrink: 1;
    flex-basis: 100 px;
}
</style>
```

```
<body>
<div id = "main">
<div style = background-color: coral; "></div>
" " : lightblue; "></div>
" " : lavender; "></div>
" " : pink; "></div>
</div>
</body>
```

① Flex shrink: It specifies how item will shrink relative to rest of flexible inside same container. Let the second flex-item shrinks 3 times more than the rest.

```
div:nth-of-type(2){
    flex-shrink: 3
}
```
— it is included in style in html.

② flex-grow :- It specifies how much the item will grow relative to the rest of flexible items inside the container.

Note :- If the element is not a flexible item, the flexible-grow property has no effect.

<style>, added from ①.

```
# main div: nth-of-type(1){ flex-grow: 1; }
# main div: nth-of-type(2){ flex-grow: 3; }
# main div: nth-of-type(3){ flex-grow: 3; }
```

③ flex-basis :- This property specifies the initial length of a flexible item.

Note: If the element is not a flexible, then flex-basis not works

<style added from ②

```
# main div: nth-of-type(2){
    flex-basis: 100 px
}
```
— Set the initial length of second flex-item to 100 pixels.

6) How do you specify the direction of flex items with a flex container?

→ flex-direction property defines in which direction the container wants to stack the flex items

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
    display: flex;
    flex-direction: column; or row, column-reverse, row-reverse
    background-color: blue;
}

.flex-container > div {
    background-color: white;
    width: 100 px;
    margin: 10 px;
    text-align: center;
    line-height: 75 px;
    font-size: 30 px;
}
</style>
</head>
<body>
    <div class="flex-container">
        <div> 1 </div>
        <div> 2 </div>
        <div> 3 </div>
    </div>
</body>
</html>
```

33) How do you align flex items horizontally & vertically within a flex container.

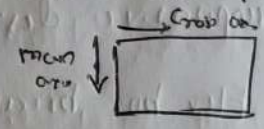
→ We can align flex items horizontally & vertically within a flex-container using flex direction property.

```
<style>
  .flex-container {
      display: flex
      flex-direction: column; & row
      background-color : blue
  }

  .flex-container > div {
      background color : white;
      width : 100 px;
      margin : 10px;
      text-align : center;
      line-height : 75 px;
      font-size : 30 px;
  }
</style>
<body>
  <div class = "flex-container">
      <div> 1 </div>
      <div> 2 </div>
      <div> 3 </div>
  </div>
</body>
```

38) Explain difference between justify content & align items properties in flexbox?

→ Justify Content



- Controls alignment of all items on the main axis
- flex-start → align elements horizontally left
  flex-end → align elements horizontally right
  center → align element at center
  space-between → Distribute element horizontally evenly across entire width
  space-around → Distribute element horizontally evenly across entire width (but with space on edges)

align-items:

- Controls alignment of all items on the cross axis.
  For instance flex-direction: row (default)
  flex-start → align vertically top
  flex-end → align vertically end/bottom
  center → align vertically centered
  baseline → align elements vertically so that baselines align
  stretch → force element to be height of container.

39) How do you control the order of flex items using CSS flexbox?

→ order is a flex-item property.
The order property specifies the order of the flex items

```
<div class = "flex-container">
    <div style = "order: 3"> 1 </div>
    <div style = "order: 1"> 4 </div>
    <div style = "order: 2"> 2 </div>
</div>
```

40) What are flexbox breakpoints, and how can they be used for responsive design?

→ Flex box breakpoints are the media queries used for responsive design ie, based on screen size it should be displayed.

or

It is points at which a website's content and design will adapt in a certain way to provide the best possible user experience

for Desktop:

```
@ media screen and (min-width: 1024 px) {
    ....
}
```

for Tablets

```
@ media screen and (min-width: 768 px) and (max-width: 1023 px) {
    ....
}
```

for Smartphones

```
@ media screen and (max-width : 767 px) {
    ....
}
```

4) What are HTML attributes?

→ HTML attributes provide additional information about HTML elements.

① href attribute

 ...

② Src Attribute :-

<img src="... .jpg">

③ Width & height attributes :-

<img src=" " width="500" height="600">

④ alt attribute :-

<img src=" " alt="..... ...">

⑤ Style attribute :-

 ...

⑥ lang attribute :- Should always use inside an html element

<!DOCTYPE html>
<html lang="en"> — Define language of HTML page
<body>
...
</body>
</html>

⑦ title attribute :- give additional information about an element

 ....

42) Explain diff btwn global attribute and element-specific attribute in HTML?

→ Global Attributes :-

Global attributes are attributes that can be used with all html elements

title - Specifies extra information about an element.

style - Specifies an inline CSS style for an element

id - Specifies an unique id for an element

lang - Specifies the language of elements content

Event attributes :-

HTML has ability to let events trigger actions in a browser, like starting a JavaScript when a user clicks on element

onclick - Fires on a mouse click on the element

onafterprint - Script to be run after the document is printed

Ex:-

```
<body>
  <button onclick="myFunction()">Click Me</button>
  <p id="demo"></p>
  <p>.....</p>
  <script>
    function myFunction() {
      document.getElementById("demo").innerHTML = "Hello World";
    }
  </script>
</body>
```

How do you add attributes to an HTML element?

To add an attribute to HTML tag, you first put a space after the tag name. Then you would add the attribute name that you with to use followed by equal sign. Finally the value should be placed in quotation marks.

Ex:-

```
<p class = "opening" >
<p class = 'new' title = " ABC" >
```

Elements with required attributes

Elements like link and img must require attribute such as href and src attribute

```
<img src = " ... jpg' alt = " ... " >
<a href = " ... " >
```

Attributes as CSS hooks

You can use the attribute hooks in CSS to dictate the how structured page will display in web browser.

```
<div class = 'featured' >
/*CSS
.featured { background-color: black;
            font-size: 1.5 rem;
          }
```

What is the purpose of id Attribute in HTML and how it unique?

The global id Attribute is used to specify a unique identifier for an HTML element. It cannot be shared across multiple elements within a same HTML document. The id attribute can be used as a bookmark to navigate to a specific point in a document

45) What is diff between class attribute & id attribute?

→ A class name can be used by multiple HTML elements.

A/ Id name must be used only by one HTML element within page

Class Attribute

It is used for specify one or more class names for an HTML element. The class Attribute can be used on any HTML element. The class name in CSS stylesheet using . Symbol

Syntax:

```
<p' class = " geeks " >

// CSS part.
 .geeks {
      background-color : green;
    } ...
```

Id Attribute:

An Id name must be used only by one HTML el

In CSS, the id Attribute is written using the # Symbol followed by id

Syntax:

```
<p' id = " id-name' >

// CSS part
   # id-name { ....
          }
```

Explain the role of href attribute in HTML, particularly in the context of links and anchors?

→ The href attribute specifies the URL of the page the link goes to. If the href attribute is not present, the <a> tag will not be hyperlink.

You can use href="#top" or href="#" to link to the top of current page.

Syntax:

...

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

We can add alternative text to an image by using alt attribute in an <img> tag.

Note: the text should describe the image & use alt="" if the image is only for decoration

<img src="cars.jpg" alt="Ferrari Cars">

What is the purpose of target attribute in HTML links, and what are its possible values?

The target attributes specifies a name or a keyword that indicates where to display the response that is received after submitting the form.

Syntax:

<form target="_blank | _self | _top | framename | _parent">

-blank = Ruponse is displayed in new window or tab

-self = " " " " Same frame (Default tab)

-parent = " " " " parent frame

-top = " " " " full body of window

framename = " " " " a named iframe

```
<!DOCTYPE html>
<html>
<body>
  <h1> . . . . </h1>
  <form target="_blank">
    <label for="fname"> first Name: </label>
    <input type="text" id="fname" name="fname"> <br> </form>
  </form>
</body>
</html>
```

4q) How do you used src attribute to embed an external resource, such as image or video in HTML?

→ ① `<img src="..." alt="..." >`

② `<video src="..." controls>`

`. . . . . .`

`</video>`

5q) What is the purpose of disabled attribute and how it is used in HTML form element?

→ The Boolean disabled attribute, when present, makes the element not mutable, focusable, or even submitted with the form

Ex.
```
<form>
  <label for="name"> Name: </label>
  <input name="name" type="text">
  <label for="emp"> Employed: </label>
  <select name="emp" disabled>
    <option> No </option>
    <option> Yes </option>
  </select>
  <label for="empDate"> Employment Date: </label>
  <input name="empDate" type="date" disabled>
</form>
```

Is there any relationship between java & Javascript

There is no relationship between Java & Javascript. They are both programming languages but have different syntax & structure and use cases. Java is used for building server side web applications and Javascript is used for developing interactive web pages and client side web applications

Is Javascript a Compiled or Intrepreted language

Yes it is a Intrepreted language

Is Javascript a case-sensitive language?

Yes, Javascript is a case-sensitive language

What is node.js?

- Node.js is an open source server Environment
- It is free and runs on various platforms.
- It is platform to run Javascript files.

What is diff between let and var?

A variable declared using the var keyword is defined throughout the program. Can be redeclared and has global scope

A variable declared by let cannot be redeclared & must be declared before its use. It have Block level scope

Ex:- console.log(x)
var x = 5;
console.log(x)

o/p → undefined
5.

console.log(x)
let x = 5;
console.log(x);

o/p → "Error"

56) Differences between Undeclared & Undefined variables?

→

Undeclared	Undefined
• These are variables that do not exist in the memory heap.	• They are ones that do not exist memory but nothing is being assigned to them.
• The variables are considered to be undeclared ∵ programmer doesn't write them with var, let or const.	• The variables are considered to be undefined ∵ it is assigned by javascript to them.
• If try to access in execution phase it will throw error.	• It doesn't throw any error.

57) What is hoisting?

→ - Process of moving all declaration to top of script.
- Creating declaration of variables internally.

Note: Definition will not be moved to top only declaration will be moved

Ex:- x=10;
console.log(x);
o/p → Error

x=10;
console.log(x)
var x;
o/p = 10.

58) What is ~~Scope~~ in Javascript?

→ Scope determines the accessibility (visibility) of variables
JS have 3 types of scope:
- Block scope — let, const.
- Function scope — ~~let~~, var
- Global scope — var.

59) What are reserved words? Can I use reserved words as identifiers

→ It is word that cannot be used as ~~identifiers~~ such as name of a variable, function or label. No.
Ex:- char, break, default, else, false, for etc

Q) Why do you need strict mode? How do you declare strict mode

Strict mode makes it easier to write "secure" JavaScript.

Strict mode changes previously accepted "bad syntax" into real errors.

① Using a variable without declaring is not allowed

```
<script>
  "use strict";
  x = 3.14;
  x = {p1:10, p2:10}
</script>
```

O/P - Error

② Deleting a variable not allowed

```
"use strict"
let x = 3.14;
delete x;
```

O/P - Error

③ Deleting a function is not allowed

```
"use strict"
function x(p1, p2) { }
delete x;
```

O/P - Error

④ Duplicating a parameter name is not allowed

```
"use strict"
function x(p1, p1) { };
```

O/P - Error

What are global variables?

The variables created outside a function are known as global variables.

Q) What are problems with global variables?

Because once a global variable it declared, it can easily be modified by any successive function or block defined.

62) What is NaN property?

→ The Nan global property is a value representing Not-A-Number.

```
function x(a) {
    if (isNaN(a)) {
        return NaN;
    }
    return a;
}
console.log(x('1'));
// OP - '1'
console.log(x('NotaNumber'));
O/p - NaN.
```

63) What is the purpose of delete operator in JavaScript?

→ It is used to delete an object property. It is used used to delete an object property that already exists, it returns true and removes the property from object.

64) What is diff btwn null & undefined?

→ Undefined → Undefined means a variable has been declared but has yet not been assigned a value

Ex. var a;
console.log(a)

Null → It is an assignment value. It can be assigned to variable as a representation of no value.

var a = null;
console.log(a);

57 What are bitwise operators available in JavaScript?

Bitwise operations are performed on a 32-bit binary number.

Bitwise AND — & — Returns true if both operands are true
Bitwise OR — | — " true with one " " "
Bitwise XOR — ^ — " " if both operands are different
" Not — ~ — Flips the value of an operand
" Left Shift — << — Shifts the bit toward left
" Right Shift — >> — " " " " right

Zero fill Right Shift (>>>) — Shifts bit toward right but adds 0 from left

58 Can I redeclare let and const variables in JavaScript?

let and const can not be redeclared. let and const must be declared before use.

59 Does const variables make the value the immutable in JavaScript?

Yes, const variables make the values the immutable in JavaScript.

60 What is ES6? List down some of the feature of ES6?

ECMA script is also known as ES6. It is standard for scripting languages. It is a programming language adopted by European computer Manufacturers Association as a standard for performing computations in web application.

Features of ES6 are

1. let and const keywords
2. Arrow functions
3. Multi-line strings
4. Default parameters
5. Template literals
6. Destructuring Assignment
7. Enhanced object literals
8. Promises
9. Classes
10. Modules.

64) What are the possible ways to create objects in JavaScript?

→ ① Creating object with a function:

```
// Simple function
function x (name, maker, engine){
    this.name = name;
    this.maker = maker;
    this.engine = engine;
}
// New keyword to create an object
let car = new vehicle ('GT', 'BMW', '1498 cc');
console.log (car.name);
```

② Creating object with literals

```
let car = {
    name: 'GT',
    maker: 'BM',
    engine: '1498cc'
};
console.log (car['maker']);
console.log (car);
```

```
let car = {
    name: 'GT',
    maker: 'BMW',
    engine: '1498cc'
    start: function() {
        console.log('...start
    }
};
car.start();
car.stop = function() {
    console.log('...
}
car.stop();
```

③ Creating Object with Object.create() method

```
const x = {
    isStudying: false;
    printIntroduction: function() {
        console.log ('My name is ${this.name}')
    }
};
const me = Object.create(coder);
me.name = 'Mukul';
me.isStudying = true;
me.printIntroduction();
```

6 Using ES6 classes

```
class Vehicle {
    constructor (name, maker, engine){
        this.name = name;
        this.maker = maker;
        this.engine = engine
    }
}

let car1 = new Vehicle('Hi', 'Bmw', 'Inline 4');
console.log(car1.name);
```

What is the difference between splice and slice?

Slice()

- This method is used to get a new array by selecting a sub array of given array.
- It doesn't alter original array.
- The return value is new array with the values in selected sub array of given array.

Ex: `let y = arr.slice(1, 3)`
1 → starting of an array
3 → ending of an arr

Splice()

- This method is used to add/remove an item from the given array.
- It affects the original array
- The return value is an array containing deleted elements

`let x = arr.splice(2, 1, 3, 4, 5)`
2 → index of array from which to start.
1 → how many elements to be deleted
3, 4, 5 → elements to be added

7) What is the difference between
a. == and === operators
b. = and == operators
c. %= and =

→

(a) == → Operator tests for abstract equality. Thus it a==b:
does values.

=== → Operator tests for strict equality i.e, it will not do the type conversion hence if the two values are not of the same type, when compared it will return false Ex: a===b o/p → True

(b) = and == Operators

= → It is used for assigning values to a variable

== → It is used for comparing two variables, but it ignores the data type of variable.

(c) %= and = operat

%= → It is used to get division remainder is then, it is compared with other values

Ex:- x %= y x = x % y;

= → It is used for assigning an values for element.

Ex:- x = y.

What is Higher Order Function?

A function which takes another function as parameter.
A function which return other function.

```
function sqr1(num){
    return num * num;
}

function sqr2(num){
    return num ** 2;
}

function cube(x, num){
    return x(num) * num;
}

let out = cube(sqr1, 10)
console.log(out);
```

What is the currying function?

Currying is a technique in functional programming that performs the transformation of a function with multiple arguments into several functions containg a single argument in a sequence

Ex.-
```
function dummy(a1, a2, a3, ...){
}
```

Ex.-
```
<body>
<script>
function volume(length){
    return function(breadth){
        return function(height){
            return length * breadth * height;
        }
    }
}
</script>
console.log("volume of container is" + volume(4)(5)(6));
</body>
```

74) What are arrow functions?

→ Arrow functions provide you with an alternative way to write a shorter syntax compared to function expression.

Ex. let myFunction = (a,b) => a * b;

console.log(myFunction (3,5));

Ex. hello = () => {
return "Hello World";
}

hello = () => "Hello World";

Ex. hello = (val) => "Hello" + val;

75) What is a spread operator?

→ Spread operator (...) allows us to quickly copy all or part of an existing or object into another array or object.

const a = [1, 2, 3]
const b = [4, 5, 6]
const result = [...a, ...b]
o/p → [1, 2, 3, 4, 5, 6]

76) What is a rest Parameter?

→ The rest parameter syntax allows us to represent an indefinite number of arguments as an array. With the help of of a rest parameter, a function called be called with any number of arguments, no matter how it was defined.
(... triple day) is the rest parameter).

Ex. function fun(...input){
let sum = 0;
for (let i of input){
sum + = i;
}
return sum;
}

return sum;
}
console.log(fun(1,2)); // O/P - 3
console.log(fun(1,2,3)); // O/P - 6
Console.log(fun(1,2,3,4,5)); // O/P - 15.

} What happens if you do not use rest parameter as a last argument?

If we do not use rest parameter at last it throws an error.

Ex. function fn(a, ...rest, b) {

}

What are regular Expression pattern?

The regular expression is a pattern of characters. The pattern is used for searching and replacing characters in string.

Syntax:- /pattern/modifier(s);

[abc] or [a-z] - find any character between a-z

[A-Z] - find any character between A-Z

[0-9] - " " number between 0-9

[a-zA-Z0-9] - find any from a-z, A-Z and 0-9

[^abc] - character starting with abc

\d ⇒ only number

\D ⇒ NOT number

combination ⇒ \W

\W ⇒ Not character

7a) How do you search a string for a patterns in Javascript?

→ ① Using String.search() method :-

Syntax:

let a = str.search(expression);

Ex.-

```
var string = "ABc ... Ac";
var a = /G/;
var b = /c/;
var c = /2/;
console.log(string.search(a))
console.log(string.search(b))
            (string.search(c))
```

O/P = 0
17
-1

② Using string.match() method:-

```
let string = "Geeks 0 compute portal";
console.log(string.match(/rGe/g));
```

O/P - ['rGe']
0: "rGE"
length: 1
[[Prototype]]: Array(0);

8a) What is purpose of switch case?

→ The use of switch to select one of many blocks of code to be executed. This is the perfect solution for long, nested if/else statements.

Syntax:

```
switch (expression) {
    case n:
        code block;
        break;
    "
    "
    default:
        default code blow
}
```

What are the conventions to be followed for the usage of switch cases?

Each case is aligned with the switch. This avoids over indentation. A case label is not a statement, and should not be indented like one.

Each group of elements (except the default) should end with break, return or throw. Do not fall thru.

What are primitive Data types?

The predefined data types provided by Javascript language are known as primitive data types. They are also known as built-in data types.

1. Number
2. String
3. Undefined
4. Null
5. Boolean
6. Symbols

83) What are different ways to access the object properties?

→ ① Dot Property accessor.

The common way to access property of an object is the dot property accessor.

Syntax:

Expression.identifier

Ex: const hero = {
name: 'XYZ';
};
console.log (hero.name)

o/p → Batman

② Square Bracket Property accessor

Syntax:

expression[expression].

The first expression denoting object and second expression should evaluate to string

Ex:- const property = 'name';
const hero = {
name: 'BB'
};
console.log (hero['name']);
console.log (hero[property]);

③ Object Destructuring

Syntax:

const { identifier } = expression;

Identifier is the name of property to access and expression should evaluate to an object. After destructuring, the variable identifier contains the property value.

Ex:- const hero = {
name: 'Batman'
};

const {name} = hero;
console.log (name);

What are function parameters rule?

The Javascript function Parameters are the names that are defined in function definition and real values passed to the function in the function definition are known as arguments

Parameter rules:

1. There is no need to specify the data type for parameters in JS function definitions
2. It does not perform type-checking based on passed in JS functions
3. It does not check the number of received arguments.

Syntax:

```
function Name(parameter1, parameter2, ....) {
  //Statements
}
```

Different ways which create infinite loops in Javascript?

For creating an infinite loop in JS, use the most common looping technique including:

- for() loop
- while() loop

for loop Eg:

```
for (var i=0; i < Infinity; i++) {
    // Statements to be executed
}
```

while() loop

```
while (true)
{
    console.log("I am infinite loop")
}
```

86) What are template literals ?

→ Template literals make it easier to create multi-line strings and include expressions within them.

It provides an easy way to interpolate variables and expressions into strings.

It uses back-ticks (` `)

Ex: let text = `He is ... "Johny" ... .`

let a = `...
...
...`

87) What are default values in destructuring assignment ?

→ Destructuring assignment makes possible to unpack values from array etc.

Each destructed property can have a default value. The default value is used when the property is not present or has value undefined. It is not used if the property has value null.

Ex: const [a=1] = []; // a is 1

const {b=2} = {b: undefined} // b is 2

const {c=2} = {c: null} // c is null

) How do you swap variables in destructing assignment?

```
let a;
let b;
[a,b] = [1,2,3];
console.log(a);   // o/p-1
coosole.log(b);   // o/p-2
```

[a,b] = [1,2,3] is a destructuring assignment that destructures [1,2,3] array, 'a' variable is assigned with first item 1 and second item 2.

Knowing how to destructure an array, it's easy to use it for swapping variables.

```
let a = 1;
let b = 2;
[a,b] = [b,a];

console.log(a);
coosole.log(b);
```

o/p : 2
1

[a,b] = [b,a] is destructuring assignment that swaps the variables 'a' and 'b'. At first step on right side of destructuring, a temporary array [2,1] is created. then destructuring occurs. Then 'a' variable is assigned with 2 and 'b' is assigned with '1'.

Is that possible to use expressions in Switch cases?

Yes, it is possible to use expression in switch cases.

The switch statement evaluates an expression, matching expression value against a series of case clauses.

Syntax:

```
switch (expression) {
  case value:
    break;
  default:
    statement
}
```

89) What are the differences between for..of and for...in statements?

→ For..of → Access elements of an array

For...in → access index or position of an array

For of

Ex let frontEnd = ['html', 'CSS', 'JS', 'RJ']

```
for (let c of frontEnd)
{
  console.log(c)
}
```

For in

```
for (let c in frontEnd)
{
  console.log(frontEnd[c])
}
```

or

```
{
  console.log(frontEnd)
}
```

or

```
for (let i in frontEnd)
{
  console.log(`The value of index ${i} is ${frontEnd[i]}`);
}
```

90) What are the differences between argument objects and rest parameters in Javascript

→

Argument Objects	Rest Parameters
- It is not a real array	- It is array instances, meaning methods like sort(), maps() can be applied directly.
- It has additional callee property.	- It doesn't have callee property.
- The arguments object contains all of the parameters - including parameters in the ...restparam array - bundled into one array like object	- The rest parameter bundles all extra parameters into a single array, but doesn't contain any named arguments defined before...

a.) What are differences between spread operator and rest parameter?

→

Spread Operator

- It is represented by dots. followed by iterable
- It allows up to expand an iterable such as an array.
- Spread syntax is used to pass an array to functions that normally require a list of many arguments.

Syntax:

```
var a_name = [..... iterable]
```

```
var a = [10,20,30]
var b = [40,50,60]
var c = [...a, ...b]
console.log(c)
```

Rest parameter

- It is represented by three dots followed by iterable
- It compresses the iterable
- It is used to create functions that accept any number of arguments.

Syntax:

```
function a_name(...arguments){
    Statements;
}
```

Ex:

```
function average(...args){
    console.log(args);
    var avg = args.reduce(function(a,b){
        return a+b;
    },0)/args.length;
    return avg;
}
console.log("avg of numbers:" + average(1,2,3,4,5));
```

92) Explain all the array methods, what are outputs and whether the method modifies the original array.

→ ① Array length

The length property return length (size) of an array.

```
const = ["A", "B", "C", "D"];
let size = fruits.length;
console.log(size)
```

② fill

```
let arr1 = Array(5).fill("A")
let arr2 = Array(6).fill("B")
```

Output → [AAAAA]
'BBBBBB'

③ Concat

↳ It creates a new array by merging existing arrays

```
let new_array = arr1.concat(arr2)
```

④ indexOf Method

↳ It is used to search the element of any array by position

```
console.log (frontEnd.indexOf ('JS'));
```

⑤ includes Method

↳ It is used to search if, whether it is present or not.

```
console.log (frontEnd.includes ('JS'));
```

⑥ isArray Method ⇒ whether it is array or not.

```
console.log (Array.isArray (arr1))
console.log (Array.isArray (frontEnd));
```

(7) Methods to add elements to Existing array :-

```
frontEnd[frontEnd.length] = 'VueJS'      // at end
frontEnd[6] = 'JQ'                       // at particular index
frontEnd.push('VJS')                     // at end
frontEnd.unshift('BS')                   // at beginning
frontEnd.splice(2,0,1,2,3,4,5)           // 1st parameter indicates beginning of array
                                         // 2nd parameter - how many elements to be deleted
                                         // next adding elements to array
```

(8) Methods to delete elements to Existing array :-

```
frontEnd.pop()
frontEnd.shift()
frontEnd.splice(-1,3)
```

(9) Create a sub array from an array :-

```
let new_array1 = frontEnd.splice(1, @frontEnd.length)
```

(10) Reverse an array :-

```
frontEnd.reverse()
```

(11) Join

↳ This methods also joins all elements into a string

```
const fruits = ["A", "B", "C"]

fruits.join("*")
```

(12) Flattening an array

It is process of reducing dimensionality of array.
It creates a new array with sub array elements concatenated to specified depth.

```
const myArr = [[1, 2], [3, 4], [5, 6]]
const newArr = myArr.flat();
```

o/p - 1, 2, 3, 4, 5, 6