INTERVIEW QUESTIONS ON HTML AND CSS

1. Are the HTML tags and elements the same thing?

No. HTML elements are defined by a starting tag, may contain some content and a closing tag.For example, <h1>Heading 1</h1> is a HTML element but just <h1> is a starting tag and </h1> is a closing tag.

2. What are tags and attributes in HTML?

Tags are the primary component of the HTML that defines how the content will be structured/ formatted, whereas Attributes are used along with the HTML tags to define the characteristics of the element. For example, <p align=” center”>Interview questions</p>, in this the ‘align’ is the attribute using which we will align the paragraph to show in the center of the view.

3. What are void elements in HTML?

HTML elements which do not have closing tags or do not need to be closed are Void elements. For Example <br />, <img />, <hr />, etc.

4. What is the advantage of collapsing white space?

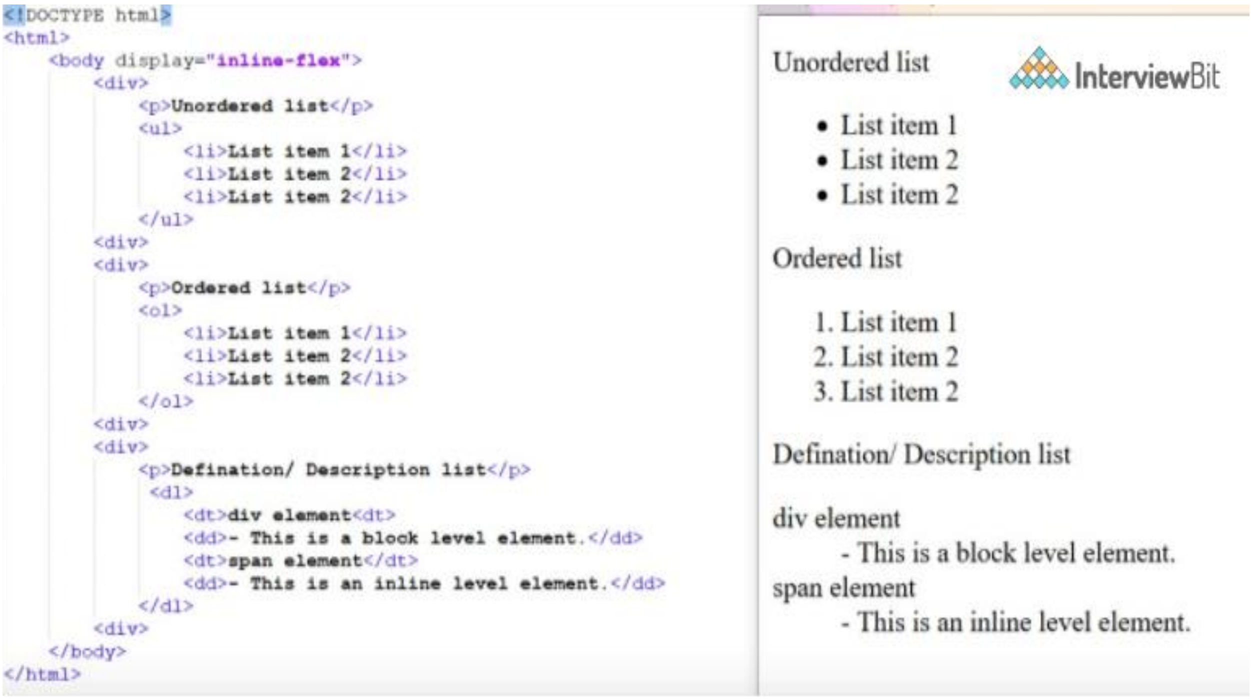
In HTML, a blank sequence of whitespace characters is treated as a single space character, Because the browser collapses multiple spaces into a single space character and this helps a developer to indent lines of text without worrying about multiple spaces and maintain readability and understandability of HTML codes.

5. What are HTML Entities?

In HTML some characters are reserved like ‘<’, ‘>’, ‘/’, etc. To use these characters in our webpage we need to use the character entities called HTML Entities. Below are a few mapping between the reserved character and its respective entity character to be used.

Character	Entity Name	Entity Number
<	&lt;	&#60;
>	&gt;	&#62;
&	&amp;	&#38;
(non-breaking space) Eg. 10 PM	&nbsp; Eg. <p>10&nbsp&nbspPM</p>	&#160;

6. What are different types of lists in HTML?



```
<!DOCTYPE html>
<html>
    <body display="inline-flex">
        <div>
            <p>Unordered list</p>
            <ul>
                <li>List item 1</li>
                <li>List item 2</li>
                <li>List item 2</li>
            </ul>
        <div>
        <div>
            <p>Ordered list</p>
            <ol>
                <li>List item 1</li>
                <li>List item 2</li>
                <li>List item 2</li>
            </ol>
        <div>
        <div>
            <p>Defination/ Description list</p>
             <dl>
                <dt>div element<dt>
                <dd>- This is a block level element.</dd>
                <dt>span element</dt>
                <dd>- This is an inline level element.</dd>
            </dl>
        <div>
    </body>
</html>
```

Unordered list

- List item 1
- List item 2
- List item 2

Ordered list

1. List item 1
2. List item 2
3. List item 2

Defination/ Description list

div element
    - This is a block level element.
span element
    - This is an inline level element.

7. What is the ‘class’ attribute in HTML?

The class attribute is used to specify the class name for an HTML element. Multiple elements in HTML can have the same class value. Also, it is mainly used to associate the styles written in the stylesheet with the HTML elements.

8. What is the difference between the ‘id’ attribute and the ‘class’ attribute of HTML elements?

Multiple elements in HTML can have the same class value, whereas a value of id attribute of one element cannot be associated with another HTML element.

9. Define multipart form data?

Multipart form data is one of the values of the enctype attribute. It is used to send the file data to the server-side for processing. The other valid values of the enctype attribute are text/plain and application/x-www-form-urlencoded.

10. Describe HTML layout structure.

Every web page has different components to display the intended content and a specific UI. But still, there are few things which are templated and are globally accepted way to structure the web page, such as:

- **<header>**: Stores the starting information about the web page.
- **<footer>**: Represents the last section of the page.

- **<nav>**: The navigation menu of the HTML page.
- **<article>**: It is a set of information.
- **<section>**: It is used inside the article block to define the basic structure of a page.
- **<aside>**: Sidebar content of the page.

11. How to optimize website assets loading?

To optimize website load time we need to optimize its asset loading and for that:

- **CDN hosting -** A CDN or content delivery network is geographically distributed servers to help reduce latency.
- **File compression -** This is a method that helps to reduce the size of an asset to reduce the data transfer
- **File concatenation -** This reduces the number of HTTP calls
- **Minify scripts -** This reduces the overall file size of js and CSS files
- **Parallel downloads -** Hosting assets in multiple subdomains can help to bypass the download limit of 6 assets per domain of all modern browsers. This can be configured but most general users never modify these settings.
- **Lazy Loading -** Instead of loading all the assets at once, the non-critical assets can be loaded on a need basis.

12. What are the various formatting tags in HTML?

HTML has various formatting tags:

- <**b**> **-** makes text bold
- <**i**> **-** makes text italic
- <**em**> - makes text italic but with added semantics importance
- <**big**> **-** increases the font size of the text by one unit
- <**small**> **-** decreases the font size of the text by one unit
- <**sub**> **-** makes the text a subscript
- <**sup**> **-** makes the text a superscript
- <**del**> **-** displays as strike out text
- <**strong**> **-** marks the text as important
- <**mark**> **-** highlights the text
- <**ins**> - displays as added text

13. What are the different kinds of Doctypes available?

The three kinds of Doctypes which are available:

- Strict Doctype
- Transitional Doctype
- Frameset Doctype

14. Please explain how to indicate the character set being used by a document in HTML?

The character set is defined in <meta> tag inside <head> element.

```
<!DOCTYPE html>
<html>
 <head>
  <meta charset="UTF-8">
  ...
  ...
 </head>
 ...
</html>
```

15. What is the difference between <strong>, <b> tags and <em>, <i> tags?

The effect on a normal webpage of the tags <strong>, <b> and <em>, <i> is the same. <b> and <i> tags stands for bold and italic. These two tags only apply font styling and bold tag <b>, just adds more ink to the text, these tags don't say anything about the text.

Whereas, <strong> and <em> tags represent that the span of text is of strong importance or more importance and emphatic stress respectively than the rest of the text. These tags have semantic meaning.

16. What is the significance of <head> and <body> tag in HTML?

<head> tag provides the information about the document. It should always be enclosed in the <html> tag. This tag contains the metadata about the webpage and the tags which are enclosed by head tag like <link>, <meta>, <style>, <script>, etc. are not displayed on the web page. Also, there can be only 1 <head> tag in the entire Html document and will always be before the <body> tag.

<body> tag defines the body of the HTML document. It should always be enclosed in the <html> tag. All the contents which needs to be displayed on the web page like images, text, audio, video, contents, using elements like <p>, <img>, <audio>, <heading>, <video>, <div>, etc. will always be enclosed by the <body> tag. Also, there can be only 1 body element in an HTML document and will always be after the <head> tag.

17. Can we display a web page inside a web page or Is nesting of webpages possible?

Yes, we can display a web page inside another HTML web page. HTML provides a tag <iframe> using which we can achieve this functionality.

```
<iframe src=”url of the web page to embed” />
```

18. How is Cell Padding different from Cell Spacing?

Cell Spacing is the space or gap between two consecutive cells. Whereas, Cell Padding is the space or gap between the text/ content of the cell and the edge/ border of the cell. Please refer to the above figure example to find the difference.

19. How can we club two or more rows or columns into a single row or column in an HTML table?

HTML provides two table attributes “rowspan” and “colspan” to make a cell span to multiple rows and columns respectively.

20. Is it possible to change an inline element into a block level element?

Yes, it is possible using the “display” property with its value as “block”, to change the inline element into a block-level element.

21. In how many ways can we position an HTML element? Or what are the permissible values of the position attribute?

There are mainly 7 values of position attribute that can be used to position an HTML element:

1. **static**: Default value. Here the element is positioned according to the normal flow of the document.
2. **absolute**: Here the element is positioned relative to its parent element. The final position is determined by the values of left, right, top, bottom.
3. **fixed**: This is similar to absolute except here the elements are positioned relative to the <html> element.
4. **relative**: Here the element is positioned according to the normal flow of the document and positioned relative to its original/ normal position.
5. **initial**: This resets the property to its default value.
6. **inherit**: Here the element inherits or takes the property of its parent.

22. In how many ways you can display HTML elements?

1. **inline**: Using this we can display any block-level element as an inline element. The height and width attribute values of the element will not affect.
2. **block**: using this, we can display any inline element as a block-level element.
3. **inline-block**: This property is similar to inline, except by using the display as inline-block, we can actually format the element using height and width values.
4. **flex**: It displays the container and element as a flexible structure. It follows flexbox property.
5. **inline-flex**: It displays the flex container as an inline element while its content follows the flexbox properties.

6. **grid**: It displays the HTML elements as a grid container.
7. **none**: Using this property we can hide the HTML element.

Below are some of the display types which are rarely used:

1. table
2. inline-table
3. table-cell
4. table-column
5. table-row
6. inline-grid
7. list-item
8. inherit
9. initial
10. table-caption

23. What is the difference between “display: none” and “visibility: hidden”, when used as attributes to the HTML element.

When we use the attribute “visibility: hidden” for an HTML element then that element will be hidden from the webpage but still takes up space. Whereas, if we use the “display: none” attribute for an HTML element then the element will be hidden, and also it won’t take up any space on the webpage.

24. How to specify the link in HTML and explain the target attribute?

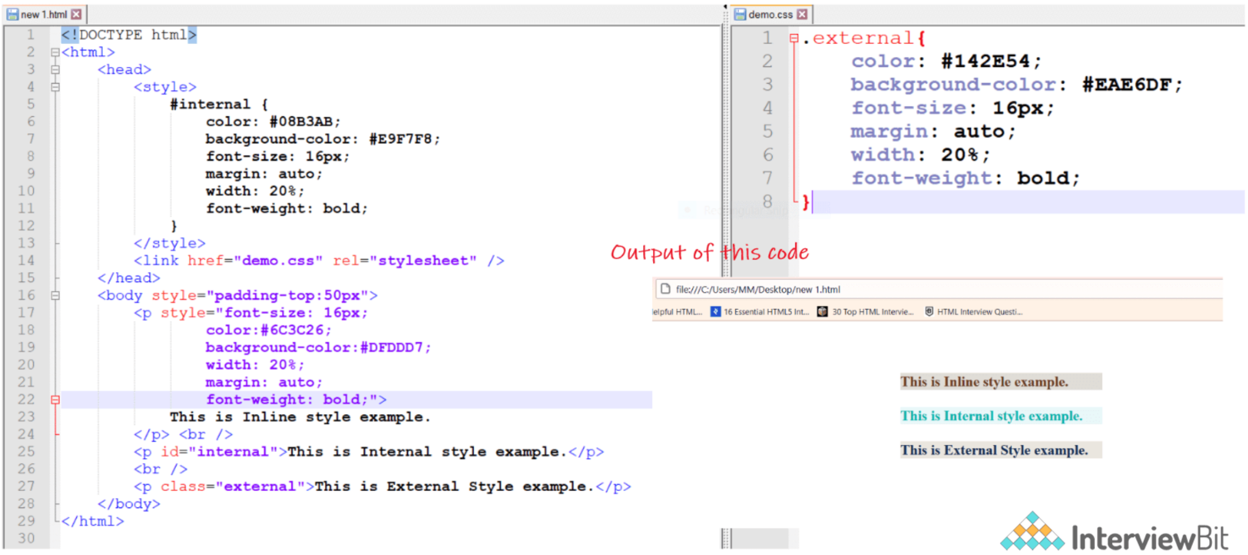
HTML provides a hyperlink - <a> tag to specify the links in a webpage. The ‘href’ attribute is used to specify the link and the ‘target’ attribute is used to specify, where do we want to open the linked document. The ‘target’ attribute can have the following values:

1. **_self**: This is a default value. It opens the document in the same window or tab as it was clicked.
2. **_blank**: It opens the document in a new window or tab.
3. **_parent**: It opens the document in a parent frame.
4. **_top**: It opens the document in a full-body window.

25. In how many ways can we specify the CSS styles for the HTML element?

There are three ways in which we can specify the styles for HTML elements:

- **Inline**: Here we use the ‘style’ attribute inside the HTML element.
- **Internal:** Here we use the <style> tag inside the <head> tag. To apply the style we bind the elements using ‘id’ or ‘class’ attributes.
- **External**: Here we use the <link> tag inside <head> tag to reference the CSS file into our HTML code. Again the binding between elements and styles is done using ‘id’ or ‘class’ attributes.



new 1.html

```html
<!DOCTYPE html>
<html>
    <head>
        <style>
            #internal {
                color: #08B3AB;
                background-color: #E9F7F8;
                font-size: 16px;
                margin: auto;
                width: 20%;
                font-weight: bold;
            }
        </style>
        <link href="demo.css" rel="stylesheet" />
    </head>
    <body style="padding-top:50px">
        <p style="font-size: 16px;
                color:#6C3C26;
                background-color:#DFDDD7;
                width: 20%;
                margin: auto;
                font-weight: bold;">
            This is Inline style example.
        </p> <br />
        <p id="internal">This is Internal style example.</p>
        <br />
        <p class="external">This is External Style example.</p>
    </body>
</html>
```

demo.css

```css
.external{
        color: #142E54;
        background-color: #EAE6DF;
        font-size: 16px;
        margin: auto;
        width: 20%;
        font-weight: bold;
}
```

Output of this code

```
file:///C:/Users/MM/Desktop/new 1.html

This is Inline style example.

This is Internal style example.

This is External Style example.
```

26. Difference between link tag <link> and anchor tag <a>?

The anchor tag <a> is used to create a hyperlink to another webpage or to a certain part of the webpage and these links are clickable, whereas, link tag <link> defines a link between a document and an external resource and these are not clickable.

27. How to include javascript code in HTML?

HTML provides a <script> tag using which we can run the javascript code and make our HTML page more dynamic.

```html
<!DOCTYPE html>
<html>
   <body>
    <h1>
         <span>This is a demo for </span>
         <u><span id="demo"></span></u>
   </h1>
   <script>
      document.getElementById("demo").innerHTML = "script Tag"
   </script>
   </body>
</html>
```

28. When to use scripts in the head and when to use scripts in the body?

If the scripts contain some event-triggered functions or jquery library then we should use them in the head section. If the script writes the content on the page or is not inside a function then it should be placed inside the body section at the bottom. In short, follow below three points:

1. Place library scripts or event scripts in the head section.
2. Place normal scripts that do not write anything on the page, in the head section until there is any performance issue.
3. Place scripts that render something on the web page at the bottom of the body section.

29. What are forms and how to create forms in HTML?

The HTML form is used to collect the user inputs. HTML provides a <form> tag to create forms. To take input from the user we use the <input> tag inside the form so that all collected user data can be sent to the server for processing. There are different input types like ‘button’, ‘checkbox’, ‘number’, ‘text’, ‘password’, ‘submit’ etc.

```
<form action="/submit_data.php">
  <label>Enter your name: </label>
  <input type="text" name="name" />
  <label>Enter Mobile number </label>
  <input type="number" name="mobile_no"/>
  <input type="submit" value="Submit">
</form>
```

30. How to handle events in HTML?

HTML allows event trigger actions in browsers using javascript or JQuery. There are a lot of events like ‘onclick’, ‘ondrag’, ‘onchange’, etc.

```
<!DOCTYPE html>
<html>
  <body style="padding-top:50px">
     <h3 id="event_demo">0</h3>
     <input type="button" onclick="myFunction()" value="Click Me" />
     <input type="reset" onclick="reset()" value="Reset" />
  </body>

  <script>
     function myFunction() {
        var value = document.getElementById("event_demo").innerHTML
        value = parseInt(value) + 1;
        document.getElementById("event_demo").innerHTML = value;
     }
     function reset() {
        document.getElementById("event_demo").innerHTML = 0;
     }
  </script>
</html>
```

31. What are some of the advantages of HTML5 over its previous versions?

Some advantages of HTML5 are:-

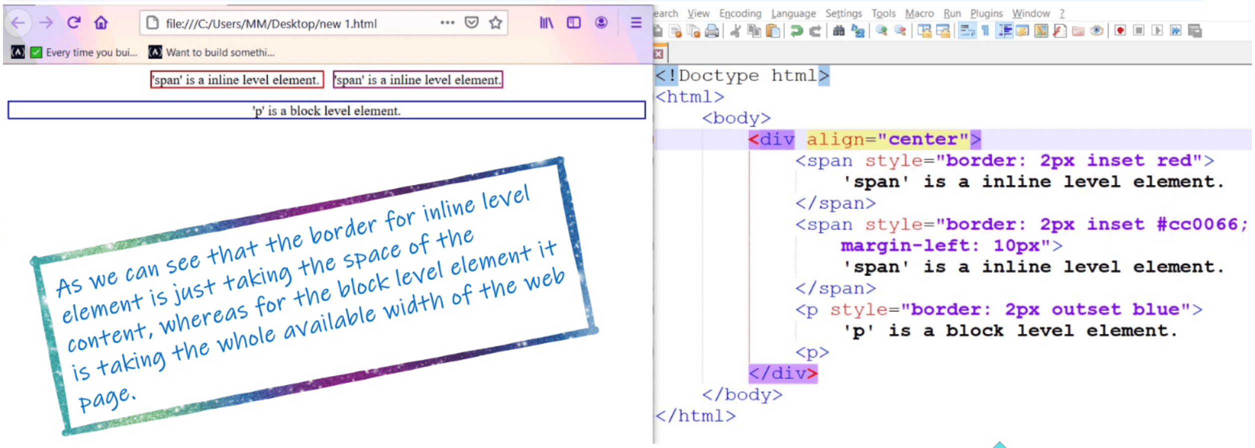
- It has Multimedia Support.
- It has the capabilities to store offline data using SQL databases and application cache.
- Javascript can be run in the background.
- HTML5 also allows users to draw various shapes like rectangles, circles, triangles, etc.
- Included new Semantic tags and form control tags.

32. How can we include audio or video in a webpage?

HTML5 provides two tags: <audio> and <video> tags using which we can add the audio or video directly in the webpage.

33. Inline and block elements in HTML5?

Inline	Block
Inline elements just take up the space that is absolutely necessary for the content and does not start from a new line. Example:- <span>, <a>, <strong>, <img>, <button>, <em>, <select>, <abbr>, <label>, <sub>, <cite>, <abbr>, <script>, <label>, <i>, <input>, <output>, <q>, etc.	Block elements start on a new line and consume the full width of the page available. Example:- <div>, <p>, <header>, <footer>, <h1>...<h6>, <form>, <table>, <canvas>, <video>, <blockquote>, <pre>, <ul>, <ol>, <figcaption>, <figure>, <hr>, <article>, <section>, etc.


file:///C:/Users/MM/Desktop/new 1.html
Every time you bui...
Want to build somethi...
earch View Encoding Language Settings Tools Macro Run Plugins Window ?

'span' is a inline level element. 'span' is a inline level element.

'p' is a block level element.

As we can see that the border for inline level element is just taking the space of the content, whereas for the block level element it is taking the whole available width of the web page.

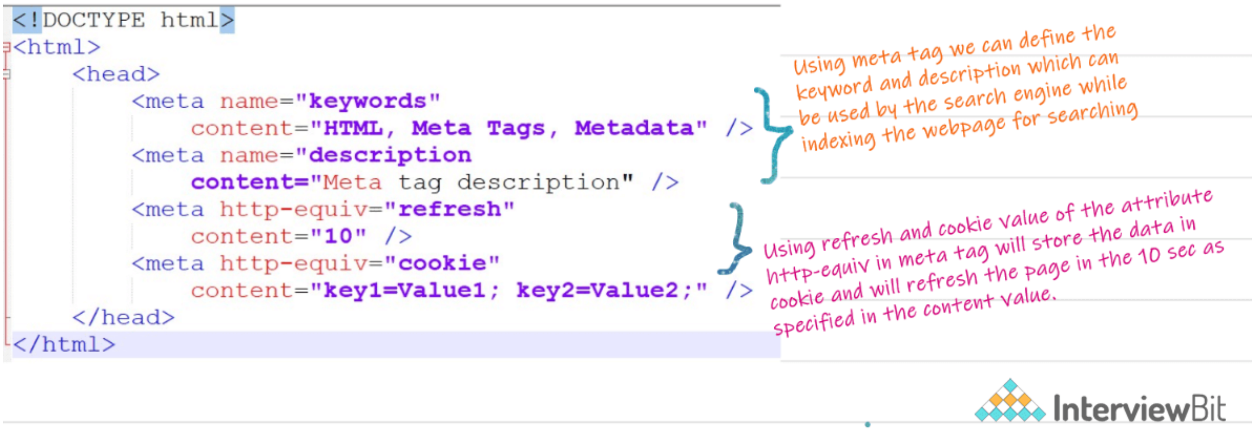
```
<!Doctype html>
<html>
    <body>
        <div align="center">
            <span style="border: 2px inset red">
                'span' is a inline level element.
            </span>
            <span style="border: 2px inset #cc0066;
                margin-left: 10px">
                'span' is a inline level element.
            </span>
            <p style="border: 2px outset blue">
                'p' is a block level element.
            <p>
        </div>
    </body>
</html>
```

34. What is the difference between <figure> tag and <img> tag?

The <figure> tag specifies the self-contained content, like diagrams, images, code snippets, etc. <figure> tag is used to semantically organize the contents of an image like image, image caption, etc., whereas the <img> tag is used to embed the picture in the HTML5 document.

35. How to specify the metadata in HTML5?

To specify we can use <meta> tag which is a void tag,i.e., it does not have a closing tag. Some of the attributes used with meta tags are name, content, http-equiv, etc. The below image tells how to specify the metadata.



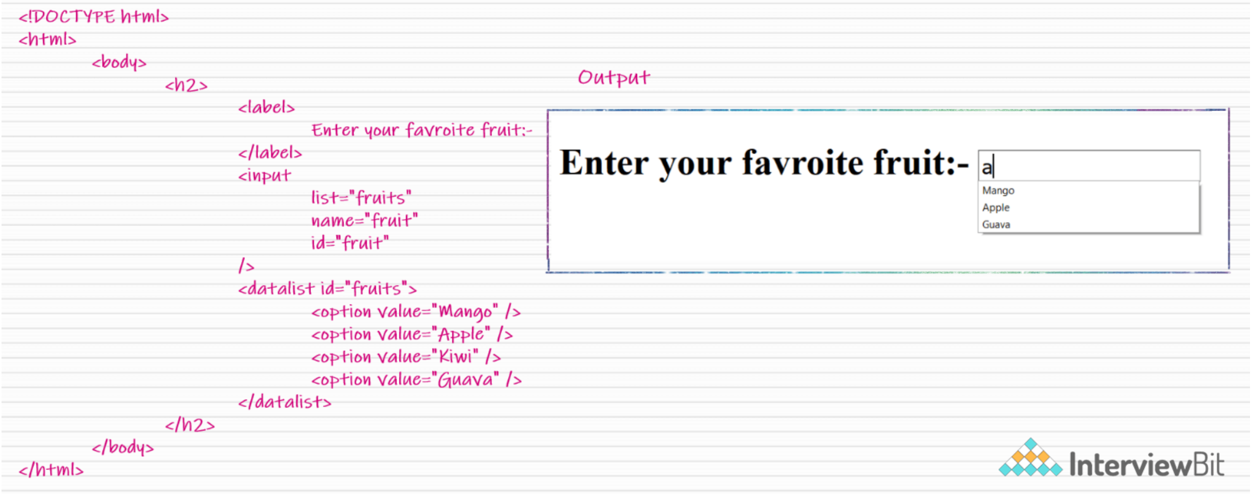
```
<!DOCTYPE html>
<html>
    <head>
        <meta name="keywords"
            content="HTML, Meta Tags, Metadata" />
        <meta name="description
            content="Meta tag description" />
        <meta http-equiv="refresh"
            content="10" />
        <meta http-equiv="cookie"
            content="key1=Value1; key2=Value2;" />
    </head>
</html>
```

Using meta tag we can define the keyword and description which can be used by the search engine while indexing the webpage for searching

Using refresh and cookie value of the attribute http-equiv in meta tag will store the data in cookie and will refresh the page in the 10 sec as specified in the content value.

36. Is the <datalist> tag and <select> tag same?

No. The <datalist> tag and <select> tag are different. In the case of <select> tag a user will have to choose from a list of options, whereas <datalist> when used along with the <input> tag provides a suggestion that the user selects one of the options given or can enter some entirely different value.



```
<!DOCTYPE html>
<html>
        <body>
                <h2>
                        <label>
                                Enter your favroite fruit:-
                        </label>
                        <input
                                list="fruits"
                                name="fruit"
                                id="fruit"
                        />
                        <datalist id="fruits">
                                <option value="Mango" />
                                <option value="Apple" />
                                <option value="Kiwi" />
                                <option value="Guava" />
                        </datalist>
                </h2>
        </body>
</html>
```

Output

Enter your favroite fruit:- a

37. Define Image Map?

Image Map lets a developer map/link different parts of images with the different web pages. It can be achieved by the <map> tag in HTML5, using which we can link images with clickable areas.

```
<img  src=”image_url” ,  usemap=”#workspace” />
<map  name=”workspace”>
    <area shape=”rect”  coords=”34, 44, 270, 350” ,  href=”xyz.html” />
    <area shape=”rect”  coords=”10, 120, 250, 360” ,  href=”xyz.html” />
</map>
```

38. What are Semantic Elements?

Semantic elements are those which describe the particular meaning to the browser and the developer. Elements like <form>, <table>, <article>, <figure>, etc., are semantic elements.

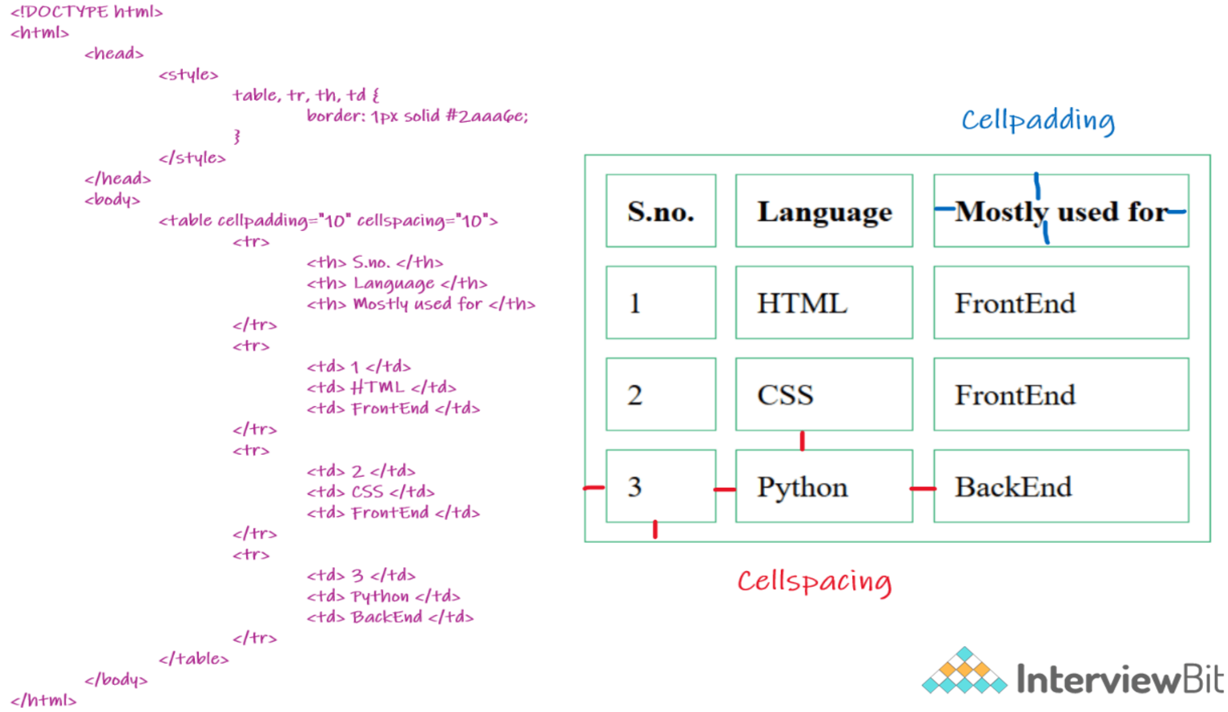
39. Convert the below data into Tabular format in HTML5?

S.no., Language, Mostly used for

1, HTML, FrontEnd

2, CSS, FrontEnd

3, Python, BackEnd



```
<!DOCTYPE html>
<html>
    <head>
        <style>
            table, tr, th, td {
                border: 1px solid #2aaa6e;
            }
        </style>
    </head>
    <body>
        <table cellpadding="10" cellspacing="10">
            <tr>
                <th> S.no. </th>
                <th> Language </th>
                <th> Mostly used for </th>
            </tr>
            <tr>
                <td> 1 </td>
                <td> HTML </td>
                <td> FrontEnd </td>
            </tr>
            <tr>
                <td> 2 </td>
                <td> CSS </td>
                <td> FrontEnd </td>
            </tr>
            <tr>
                <td> 3 </td>
                <td> Python </td>
                <td> BackEnd </td>
            </tr>
        </table>
    </body>
</html>
```

S.no.	Language	Mostly used for
1	HTML	FrontEnd
2	CSS	FrontEnd
3	Python	BackEnd

40. What is the difference between <meter> tag and <progress> tag?

<progress> tag should be used when we want to show the completion progress of a task, whereas if we just want a scalar measurement within a known range or fraction value. Also, we can specify multiple extra attributes for <meter> tags like ‘form’, ‘low’, ‘high’, ‘min’, etc.

41. Is drag and drop possible using HTML5 and how?

Yes, in HTML5 we can drag and drop an element. This can be achieved using the drag and drop-related events to be used with the element which we want to drag and drop.

42. Difference between SVG and Canvas HTML5 element?

SVG	Canvas
SVG is a vector based i.e., composed of shapes.	It is Raster based i.e., composed of pixels.
SVG works better with a larger surface.	Canvas works better with a smaller surface.
SVG can be modified using CSS and scripts.	Canvas can only be modified using scripts.
SVG is highly scalable. So we can print at high quality with high resolution.	It is less scalable.

43. What type of audio files can be played using HTML5?

HTML5 supports the following three types of audio file formats:

1. Mp3
2. WAV
3. Ogg

44. What are the significant goals of the HTML5 specification?

These were the target area of the HTML5 specs:

- Introduction of new element tags to better structure the web page such as <header> tag.
- Forming a standard in cross-browser behavior and support for different devices and platforms
- Backward compatible with the older version HTML web pages
- Introduction of basic interactive elements without the dependency of plugins such as <video> tag instead of the flash plugin.

45. Explain the concept of web storage in HTML5.

This web storage helps in storing some of the static data in the local storage of the browser so that we do not need to fetch it from the server every time we need it. There is a size limit based on different browsers. This helps in decreasing the load time and a smooth user experience. There are two types of web storage that are used to store data locally in HTML5:

- **Local Storage -** This helps in storing data that will be retained even though the user reopens the browser. It is stored for each webapp on different browsers.
- **Session Storage -** This is used for one session only. After the user closes the browser this gets deleted.

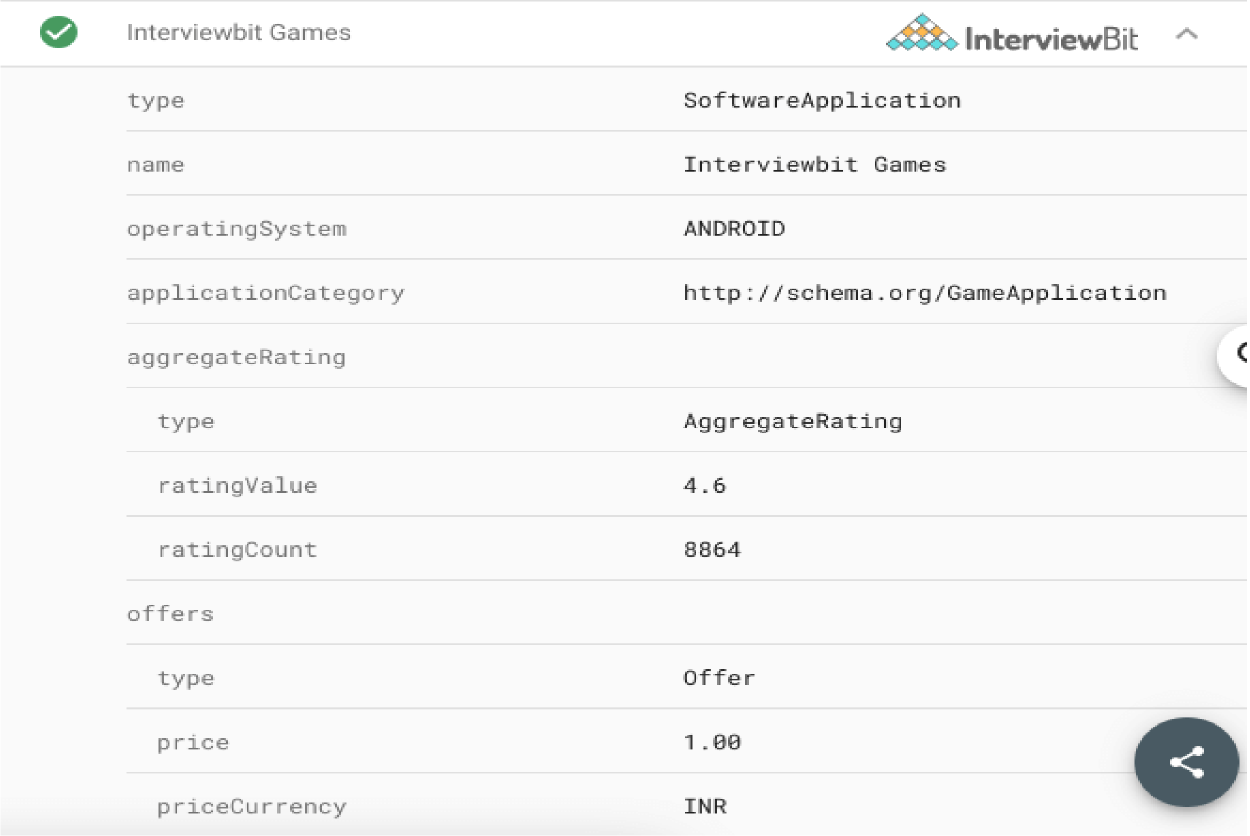
46. What is Microdata in HTML5?

It is used to help extract data for site crawlers and search engines. It is basically a group of name-value pairs. The groups are called items, and each name-value pair is a property. Most of the search engines like Google, Microsoft, Yandex, etc follow schema.org vocabulary to extract this microdata.

```
<div itemscope itemtype="http://schema.org/SoftwareApplication">
 <span itemprop="name">Interviewbit Games</span> -
 REQUIRES <span itemprop="operatingSystem">ANDROID</span><br>
 <link itemprop="applicationCategory"
href="http://schema.org/GameApplication"/>
 <div itemprop="aggregateRating" itemscope
itemtype="http://schema.org/AggregateRating">
RATING:
<span itemprop="ratingValue">4.6</span> (
<span itemprop="ratingCount">8864</span> ratings )
 </div>
 <div itemprop="offers" itemscope itemtype="http://schema.org/Offer">
Price: Rs.<span itemprop="price">1.00</span>
<meta itemprop="priceCurrency" content="INR" />
 </div>
</div>
```

- **itemid** – The unique, global identifier of an item.
- **itemprop** – Used to add properties to an item.
- **itemref** – Provides a list of element ids with additional properties.
- **itemscope** – It defines the scope of the itemtype associated with it.
- **itemtype** – Specifies the URL of the vocabulary that will be used to define itemprop.

The above example will be parsed by Google as



Interviewbit Games	
type	SoftwareApplication
name	Interviewbit Games
operatingSystem	ANDROID
applicationCategory	http://schema.org/GameApplication
aggregateRating	
type	AggregateRating
ratingValue	4.6
ratingCount	8864
offers	
type	Offer
price	1.00
priceCurrency	INR

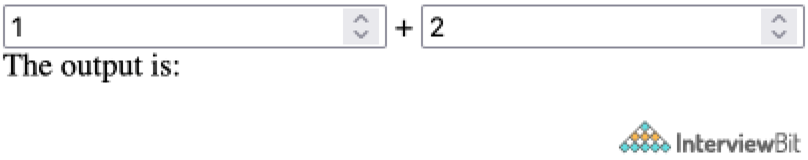
47. Which tag is used for representing the result of a calculation? Explain its attributes.

The <output> tag is used for representing the result of a calculation. It has the following attributes:

- **for -** It defines the relationship between the elements used in calculation and result.
- **form -** This is used to define the form the output element belongs to.
- **name -** The name of the output element.

```
<form oninput = "result.value=parseInt(n1.value)+parseInt(n2.value)">
    <input type = "number" name = "n1" value = "1" /> +
    <input type = "number" name = "n2" value = "2" /><br />
    The output is: <output name = "result"></output>
</form>
```

The above example looks like



48. What is new about the relationship between the <header> and <h1> tags in HTML5?

As HTML5 was all about better semantics and arrangements of the tags and elements, the <header> tag specifies the header section of the webpage. Unlike in previous version there was one <h1> element for the entire webpage, now this is the header for one section such as <article> or <section>. According to the HTML5 specification, each <header> element must at least have one <h1> tag.

49. Explain HTML5 Graphics.

HTML5 supports two kinds of graphics:

- **Canvas -** It is like drawing on a whitepaper or a blank webpage. We can add different graphic designs on web pages with available methods for drawing various geometrical shapes.

```
<!DOCTYPE HTML>
<html>
 <head>
 </head>
 <body>
  <canvas width="300" height="100" style="border:2px solid;"></canvas>
 </body>
</html>
```

- **SVG -** Scalable Vector Graphics are used mostly for diagrams or icons. It follows the XML format.

```
<!DOCTYPE html>
<html>
 <body>
  <svg width="400" height="110">
   <rect width="300" height="100" style="fill:#FFF;stroke-width:2;stroke:#000" />
  </svg>
 </body>
</html>
```

Both of the above examples produce this output and represent two different approaches provided by HTML5 to implement graphical aspects in the webpage.

50. Explain new input types provided by HTML5 for forms?

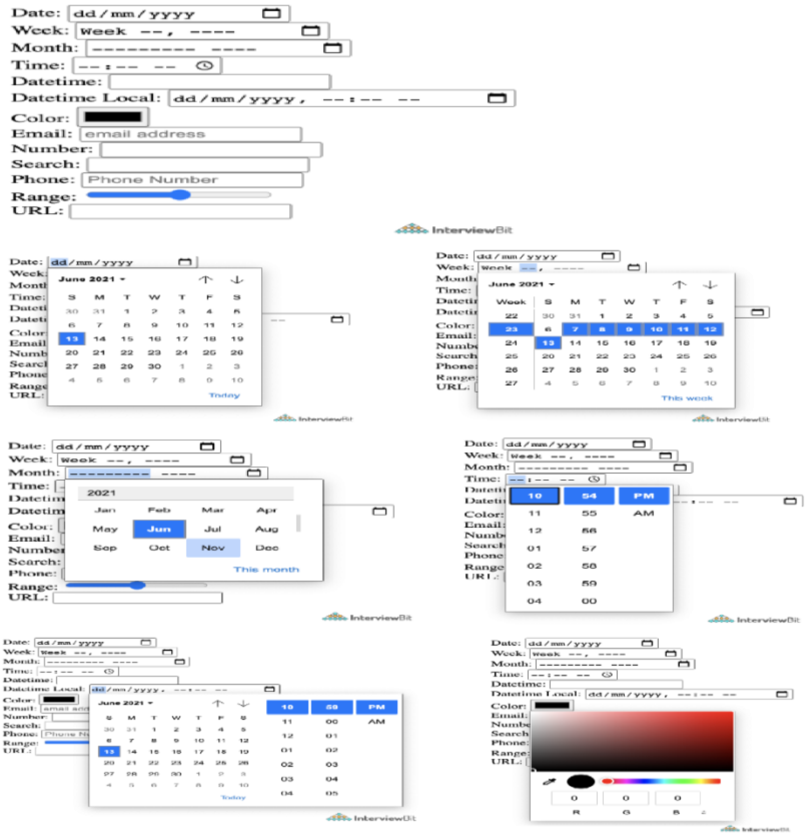
Following are the significant new data types offered by HTML5:

- Date - Only select date by using type = "date"
- Week - Pick a week by using type = "week"

- Month - Only select month by using type = "month"
- Time - Only select time by using type = "time".
- Datetime - Combination of date and time by using type = "datetime"
- Datetime-local - Combination of date and time by using type = "datetime-local." but ignoring the timezone
- Color - Accepts multiple colors using type = "color"
- Email - Accepts one or more email addresses using type = "email"
- Number - Accepts a numerical value with additional checks like min and max using type = "number"
- Search - Allows searching queries by inputting text using type = "search"
- Tel - Allows different phone numbers by using type = "tel"
- Placeholder - To display a short hint in the input fields before entering a value using type = "placeholder"
- Range - Accepts a numerical value within a specific range using type = "range"
- Url - Accepts a web address using type = "url”

```
<form>
    <div>
      <label>Date:</label>
      <input type="date" id="date" />
      <br>
      <label>Week:</label>
      <input type="week" id="week" />
      <br>
      <label>Month:</label>
      <input type="month" id="month" />
      <br>
      <label>Time:</label>
      <input type="time" id="time" />
      <br>
      <label>Datetime:</label>
      <input type="datetime" id="datetime" />
      <br>
      <label>Datetime Local:</label>
      <input type="datetime-local" id="datetime-local" />
      <br>
      <label>Color:</label>
      <input type="color" id="color"/>
      <br>
      <label>Email:</label>
      <input type="email" id="email" placeholder="email address" />
      <br>
      <label>Number:</label>
      <input type="number" id="number" />
      <br>
      <label>Search:</label>
```

```
         <input type="search" id="search" />
         <br>
         <label>Phone:</label>
         <input type="tel" id="phone" placeholder="Phone Number"
pattern="\d{10}$" />
         <br>
         <label>Range:</label>
         <input type="range" id="range" />
         <br>
         <label>URL:</label>
         <input type="url" id="url"/>
      </div>
   </form>
```



Date: dd / mm / yyyy
Week: Week --, ----
Month: --------- ----
Time: --:-- --
Datetime:
Datetime Local: dd / mm / yyyy, --:-- --
Color:
Email: email address
Number:
Search:
Phone: Phone Number
Range:
URL:

51. What are the New tags in Media Elements in HTML5?

- **<audio>** - Used for sounds, audio streams, or music, embed audio content without any additional plug-in.
- **<video>** - Used for video streams, embed video content etc.
- **<source>** - Used for multiple media resources in media elements, such as audio, video, etc.
- **<embed>** - Used for an external application or embedded content.
- **<track>** - Used for subtitles in the media elements such as video or audio.

52. Why do you think the addition of drag-and-drop functionality in HTML5 is important? How will you make an image draggable in HTML5?

The drag and drop functionality is a very intuitive way to select local files. This is similar to what most of the OS have copy functionality thus making it very easy for the user to comprehend. Before the native drag and drop API, this was achievable by writing complex Javascript programming or external frameworks like jQuery.

To enable this functionality there is a draggable attribute in the <img> tag and need to set ondrop and ondragover attribute to an eventhandler available in scripts.

53. Why do we need the MathML element in HTML5?

MathML stands for Mathematical Markup Language. It is used for displaying mathematical expressions on web pages. For this <math> tag is used.

54. What are the server-sent events in HTML5?

The events pushed from the webserver to the browsers are called server-sent events. DOM elements can be continuously updated using these events. This has a major advantage over straight-up polling. In polling, there is a lot of overhead since every time it is establishing an HTTP connection and tearing it down whereas, in server-sent events, there is one long-lived HTTP connection. To use a server-sent event, <eventsource> element is used. The src attribute of this element specifies the URL from which sends a data stream having the events.

```
<eventsource src = "/cgi-bin/myfile.cgi" />
```

55. What are Web Workers?

These are added to bring parallelism and async capability. It runs in the background to do the computationally expensive tasks without yielding to make the page responsive. It is achieved by starting a separate thread for such tasks. These are not meant to perform UI operations. There are three types of web workers:

- **Dedicated Workers -** These are workers that are utilized by a single script.
- **Shared Workers -**These are workers that are utilized by multiple scripts running in different windows, IFrames, etc.
- **Service Workers -** These act as proxy servers between web applications, the browser, and the network. Mostly used for push notifications and sync APIs.

56. What is the usage of a novalidate attribute for the form tag that is introduced in HTML5?

Its value is a boolean type that indicates whether or not the data being submitted by the form will be validated beforehand. By making this false, forms can be submitted without validation which helps users to resume later also.

```
<form action = "" method = "get" novalidate>
       Name:<br><input type="name" name="sname"><br>
      Doubt:<br><input type="number" name="doubt"><br>
       <input type="submit" value="Submit">
</form>
```

57. What are raster images and vector images?

Raster Images - The raster image is defined by the arrangement of pixels in a grid with exactly what color the pixel should be. Few raster file formats include PNG(.png), JPEG(.jpg), etc.
Vector Images - The vector image is defined using algorithms with shape and path definitions that can be used to render the image on-screen written in a similar markup fashion. The file extension is .svg

58. How to support SVG in old browsers?

To support old browsers instead of defining the resource of svg in src attribute of <img> tag, it should be defined in srcset attribute and in src the fallback png file should be defined.

```
<img src="circle.png" alt="circle" srcset="circle.svg">
```

59. What are different approaches to make an image responsive?

- **Art direction -** Using <picture> element the landscape image fully shown in desktop layout can be zoomed in with the main subject in focus for a portrait layout.

```
<picture>
 <source media="(min-width: 650px)" srcset="img_cup.jpg">
 <img src="img_marsh.jpg" style="width:auto;">
</picture>
```

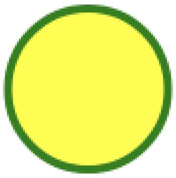
Bigger Screen (>650px)

For any other screen

- **Resolution switching -** Instead of zoom and crop the images can be scaled accordingly using vector graphics. Also, this can be further optimized to serve different pixel density screens as well.

For example SVG

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



60. What is a manifest file in HTML5?

The manifest file is used to list down resources that can be cached. Browsers use this information to make the web page load faster than the first time. There are 3 sections in the manifest file

- **CACHE Manifest -** Files needs to be cached
- **Network -** File never to be cached, always need a network connection.
- **Fallback -** Fallback files in case a page is inaccessible

```
CACHE MANIFEST
# 2012-06-16 v1.0.0
/style.css
/logo.gif
/main.js
NETWORK:
login.php
FALLBACK:
/html/ /offline.html
<!DOCTYPE HTML>
<html manifest="tutorial.appcache">
...
...
</html>
```

61. What is the Geolocation API in HTML5?

Geolocation API is used to share the physical location of the client with websites. This helps in serving locale-based content and a unique experience to the user, based on their location. This works with a new property of the global navigator object and most of the modern browsers support this.

```
var geolocation = navigator.geolocation;
```

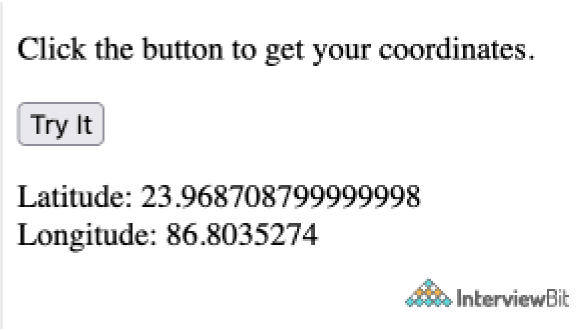
62. Write HTML5 code to demonstrate the use of Geolocation API.

```
<!DOCTYPE html>
<html>
  <body>
    <p>Click "try it" button to get your coordinates.</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 functionality is not supported by this browser.";
        }
      }

      function showPosition(position) {
        x.innerHTML = "Latitude: " + position.coords.latitude +
        "<br>Longitude: " + position.coords.longitude;
      }
    </script>
  </body>
</html>
```

The above example asks for user permission for accessing the location data via geolocation API and after clicking the button the coordinates of the physical location of the client get displayed.



Click the button to get your coordinates.

Try It

Latitude: 23.968708799999998
Longitude: 86.8035274

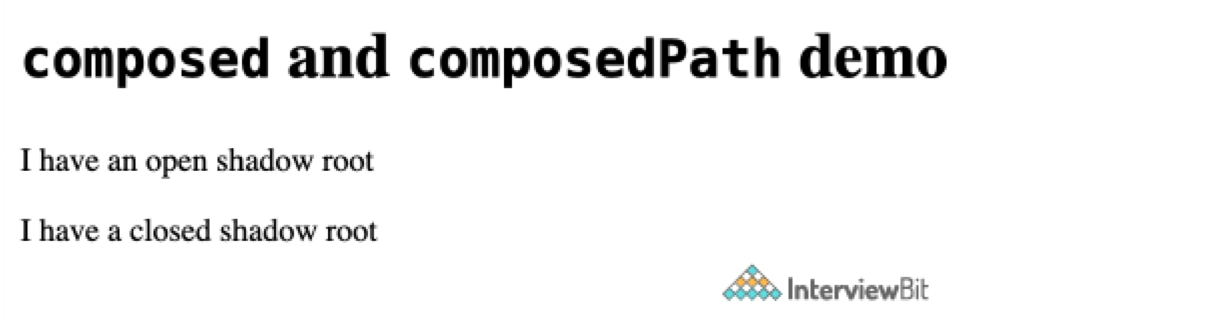
63. Explain Web Components and it’s usage.

These are used to create reusable custom elements which are very difficult in traditional HTML. It consists of three technologies:

- **Custom elements -** These are JavaScript APIs that help in defining custom elements and their behavior.
- **Shadow DOM -** These are JavaScript APIs that attach an encapsulated shadow DOM tree to an element to keep the element’s features private and unaffected by other parts.

```
<!DOCTYPE html>
<html>
 <head>
<meta charset="utf-8">
<title>composed and composedPath demo</title>
<script src="main.js" defer></script>
 </head>
 <body>
```

```
<h1><code>composed</code> and <code>composedPath</code> demo</h1>
<open-shadow text="I have an open shadow root"></open-shadow>
<closed-shadow text="I have a closed shadow root"></closed-shadow>
 </body>
</html>
customElements.define('open-shadow',
 class extends HTMLElement {
constructor() {
  super();
  const pElem = document.createElement('p');
  pElem.textContent = this.getAttribute('text');
  const shadowRoot = this.attachShadow({mode: 'open'});
  shadowRoot.appendChild(pElem);
}
 }
);
customElements.define('closed-shadow',
 class extends HTMLElement {
constructor() {
  super();
  const pElem = document.createElement('p');
  pElem.textContent = this.getAttribute('text');
  const shadowRoot = this.attachShadow({mode: 'closed'});
  shadowRoot.appendChild(pElem);
}
 }
);
document.querySelector('html').addEventListener('click', e => {
 console.log(e.composed);
 console.log(e.composedPath());
});
```



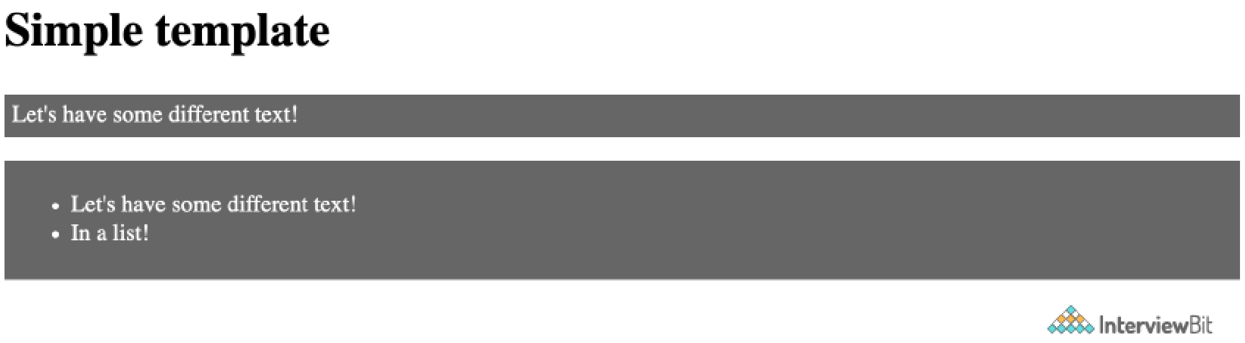
composed and composedPath demo

I have an open shadow root

I have a closed shadow root

Here 2 custom elements are defined <open-shadow> and <closed-shadow> which takes their text content and inserts them into a shadow DOM as content of a <p> element.

- **HTML templates -** The markup templates are written using <template> and <slot> elements which can be reused multiple times as the basis of a custom element's structure.



Simple template

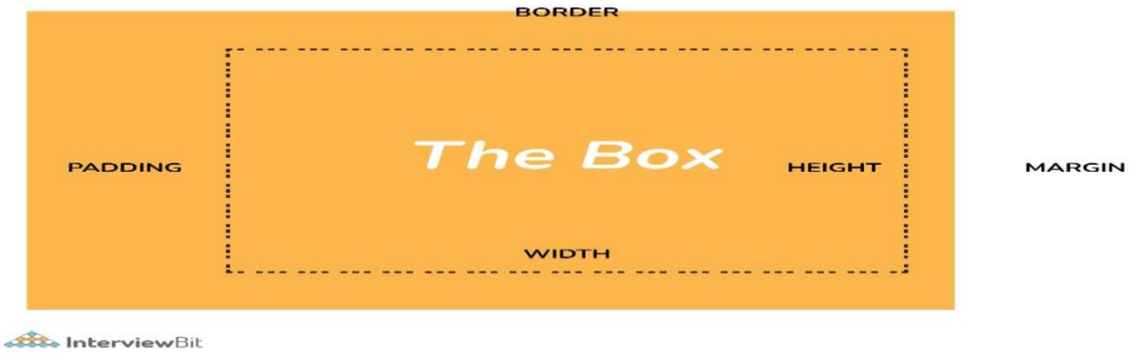
Let's have some different text!

- Let's have some different text!
- In a list!

Here we are reusing the <my-paragraph> template.

1. What is the Box model in CSS? Which CSS properties are a part of it?

A rectangle box is wrapped around every HTML element. The box model is used to determine the height and width of the rectangular box. The CSS Box consists of Width and height (or in the absence of that, default values and the content inside), padding, borders, margin.



- **Content:** Actual Content of the box where the text or image is placed.
- **Padding:** Area surrounding the content (Space between the border and content).
- **Border:** Area surrounding the padding.
- **Margin:** Area surrounding the border.

2. What are the advantages of using CSS?

The main advantages of CSS are given below:

- **Separation of content from presentation -** CSS provides a way to present the same content in multiple presentation formats in mobile or desktop or laptop.
- **Easy to maintain -** CSS, built effectively can be used to change the look and feel complete by making small changes. To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Bandwidth -** Used effectively, the style sheets will be stored in the browser cache and they can be used on multiple pages, without having to download again.

3. What are the limitations of CSS?

Disadvantages of CSS are given below:

- **Browser Compatibility:** Some style selectors are supported and some are not. We have to determine which style is supported or not using the @support selector).
- **Cross Browser issue:** Some selectors behave differently in a different browser).
- **There is no parent selector:** Currently, Using CSS, you can't select a parent tag.

4. How to include CSS in the webpage?

There are different ways to include a CSS in a webpage,

1 - External Style Sheet: An external file linked to your HTML document: Using link tag, we can link the style sheet to the HTML page.

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

2 - Embed CSS with a style tag: A set of CSS styles included within your HTML page.

```
<style type="text/css">

/*Add style rules here*/

</style>
```

Add your CSS rules between the opening and closing style tags and write your CSS exactly the same way as you do in stand-alone stylesheet files.

3 - Add inline styles to HTML elements(CSS rules applied directly within an HTML tag.): Style can be added directly to the HTML element using a style tag.

```
<h2 style="color:red;background:black">Inline Style</h2>
```

4 - Import a stylesheet file (An external file imported into another CSS file): Another way to add CSS is by using the @import rule. This is to add a new CSS file within CSS itself.

```
@import "path/to/style.css";
```

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

A CSS selector is the part of a CSS ruleset that actually selects the content you want to style. Different types of selectors are listed below.

Universal Selector: The universal selector works like a wildcard character, selecting all elements on a page. In the given example, the provided styles will get applied to all the elements on the page.

```
* {
  color: "green";
  font-size: 20px;
  line-height: 25px;
}
```

Element Type Selector: This selector matches one or more HTML elements of the same name. In the given example, the provided styles will get applied to all the ul elements on the page.

```
ul {
  line-style: none;
  border: solid 1px #ccc;
}
```

ID Selector: This selector matches any HTML element that has an ID attribute with the same value as that of the selector. In the given example, the provided styles will get applied to all the elements having ID as a container on the page.

```
#container {
  width: 960px;
  margin: 0 auto;
}

<div id="container"></div>
```

Class Selector: The class selector also matches all elements on the page that have their class attribute set to the same value as the class. In the given example, the provided styles will get applied to all the elements having ID as the box on the page.

```
.box {
  padding: 10px;
  margin: 10px;
  width: 240px;
}

<div class="box"></div>
```

Descendant Combinator: The descendant selector or, more accurately, the descendant combinator lets you combine two or more selectors so you can be more specific in your selection method.

```
#container .box {
        float: left;
        padding-bottom: 15px;
}

<div id="container">
        <div class="box"></div>

        <div class="box-2"></div>
</div>

<div class=”box”></div>
```

This declaration block will apply to all elements that have a class of box that is inside an element with an ID of the container. It’s worth noting that the .box element doesn’t have to be an immediate child: there could be another element wrapping .box, and the styles would still apply.

Child Combinator: A selector that uses the child combinator is similar to a selector that uses a descendant combinator, except it only targets immediate child elements.

```
#container> .box {
        float: left;
        padding-bottom: 15px;
}

<div id="container">
        <div class="box"></div>

        <div>
                <div class="box"></div>
```

```
	</div>
</div>
```

The selector will match all elements that have a class of box and that are immediate children of the #container element. That means, unlike the descendant combinator, there can’t be another element wrapping .box it has to be a direct child element.

General Sibling Combinator: A selector that uses a general sibling combinator to match elements based on sibling relationships. The selected elements are beside each other in the HTML.

```
h2 ~ p {
	margin-bottom: 20px;
}

<h2>Title</h2>
<p>Paragraph example.</p>
<p>Paragraph example.</p>
<p>Paragraph example.</p>
<div class=”box”>
	<p>Paragraph example.</p>
</div>
```

In this example, all paragraph elements (<p>) will be styled with the specified rules, but only if they are siblings of <h2> elements. There could be other elements in between the <h2> and <p>, and the styles would still apply.

Adjacent Sibling Combinator: A selector that uses the adjacent sibling combinator uses the plus symbol (+), and is almost the same as the general sibling selector. The difference is that the targeted element must be an immediate sibling, not just a general sibling.

```
p + p {
	text-indent: 1.Sem;
	margin-bottom: 0;
}

<h2>Title</h2>
<p>Paragraph example.</p>
<p>Paragraph example.</p>
<p>Paragraph example.</p>

<div class=”box”>
	<p>Paragraph example.</p>
	<p>Paragraph example.</p>
</div>
```

The above example will apply the specified styles only to paragraph elements that immediately follow other paragraph elements. This means the first paragraph element on a page would not receive these styles. Also, if another element appeared between two paragraphs, the second paragraph of the two wouldn’t have the styles applied.

Attribute Selector: The attribute selector targets elements based on the presence and/or value of HTML attributes, and is declared using square brackets.

```
input [type=”text”] {
            background-color: #444;
            width: 200px;
}

<input type="text">
```

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

A CSS Preprocessor is a tool used to extend the basic functionality of default vanilla CSS through its own scripting language. It helps us to use complex logical syntax like – variables, functions, mixins, code nesting, and inheritance to name a few, supercharging your vanilla CSS.

SASS: Sass is the acronym for “Syntactically Awesome Style Sheets”. SASS can be written in two different syntaxes using SASS or SCSS

SASS vs SCSS

- SASS is based on indentation and SCSS(Sassy CSS) is not.
- SASS uses .sass extension while SCSS uses .scss extension.
- SASS doesn’t use curly brackets or semicolons. SCSS uses it, just like the CSS.

SASS Syntax

```
$font-color: #fff
$bg-color: #00f

#box
            color: $font-color
            background: $bg-color
```

SCSS Syntax

```
$font-color: #fff;
$bg-color: #00f;

#box{
```

```
            color: $font-color;
            background: $bg-color;
}
```

LESS: LESS is an acronym for “Leaner Stylesheets”. LESS is easy to add to any javascript projects by using NPM or less.js file. It uses the extension .less.

LESS syntax is the same as the SCSS with some exceptions. LESS uses @ to define the variables.

```
@font-color: #fff;
@bg-color: #00f

#box{
            color: @font-color;
            background: @bg-color;
}
```

Stylus: Stylus offers a great deal of flexibility in writing syntax, supports native CSS as well as allows omission of brackets, colons, and semicolons. It doesn’t use @ or $ for defining variables.

```
/* STYLUS SYNTAX WRITTEN LIKE NATIVE CSS */
font-color= #fff;
bg-color = #00f;

#box {
            color: font-color;
            background: bg-color;
}

/* OR */

/* STYLUS SYNTAX WITHOUT CURLY BRACES */
font-color= #fff;
bg-color = #00f;

#box
            color: font-color;
            background: bg-color;
```

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

It’s a CSS unit used to measure the height and width in percentage with respect to the viewport. It is used mainly in responsive design techniques. The measure VH is equal to 1/100 of the height of the viewport. If the height of the browser is 1000px, 1vh is equal to 10px. Similarly, if the width is 1000px, then 1 vw is equal to 10px.

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

Reset CSS: CSS resets aim to remove all built-in browser styling. For example margins, paddings, font-sizes of all elements are reset to be the same.

Normalize CSS: Normalize CSS aims to make built-in browser styling consistent across browsers. It also corrects bugs for common browser dependencies.

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

Block Element: The block elements always start on a new line. They will also take space for an entire row or width. List of block elements are <div>, <p>.

Inline Elements: Inline elements don't start on a new line, they appear on the same line as the content and tags beside them. Some examples of inline elements are <a>, <span> , <strong>, and <img> tags.

Inline Block Elements: Inline-block elements are similar to inline elements, except they can have padding and margins and set height and width values.

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

It’s most important to test a website in different browsers when you’re first designing it, or when making major changes. However, it’s also important to repeat these tests periodically, since browsers go through a lot of updates and changes.

11. What are Pseudo elements and Pseudo classes?

Pseudo-elements allows us to create items that do not normally exist in the document tree, for example ::after.

- ::before
- ::after
- ::first-letter
- ::first-line
- ::selection

In the below example, the color will appear only on the first line of the paragraph.

```
p: :first-line {
```

```
            color: #ffOOOO;
            font-variant: small-caps;
}
```

Pseudo-classes select regular elements but under certain conditions like when the user is hovering over the link.

- :link
- :visited
- :hover
- :active
- :focus

Example of the pseudo-class, In the below example, the color applies to the anchor tag when it’s hovered.

```
/* mouse over link */
a:hover {
            color: #FFOOFF;
}
```

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

There are different ways to specify units in CSS like px, em, pt, percentage (%). px(Pixel) gives fine-grained control and maintains alignment because 1 px or multiple of 1 px is guaranteed to look sharp. px is not cascade. em maintains relative size. you can have responsive fonts. Em, will cascade 1em is equal to the current font-size of the element or the browser default. If u sent font-size to 16px then 1em = 16px. The common practice is to set default body font-size to 62.5% (equal to 10px).

pt(point) are traditionally used in print. 1pt = 1/72 inch and it is a fixed-size unit.

%(percentage) sets font-size relative to the font size of the body. Hence, you have to set the font-size of the body to a reasonable size.

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

No, it doesn’t affect the inline elements. Inline elements flow with the contents of the page.

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

We can use the font-family property for achieving this. The font-family property is used for specifying what font needs to be applied on the targetted DOM element. It can hold several font names as part of “fallback” mechanism in case the browser does not support the fonts. For example, we can use:

```
p {
 font-family: "Times New Roman", Times, serif;
}
```

In the above piece of code, we are applying font-family property to the paragraph element.

- It tells the browser to look for "Times New Roman" font and apply it.
- If the "Times New Roman" font is not installed or supported, then it asks the browser to use Times font.
- If both "Times New Roman" and Times are not supported, then it asks the browser to use any supported generic font belonging to serif.

If you do not want the font-face of the paragraph element to be Times New Roman/Times/serif font, and you want to use the Arial/Helvetica/sans-serif font, then we can just update the CSS property of paragraph element as:

```
p {
 font-family: Arial, Helvetica, sans-serif;
}
```

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

Adaptive Design	Responsive Design
Adaptive design focuses on developing websites based on multiple fixed layout sizes.	Responsive design focuses on showing content on the basis of available browser space.
When a website developed using adaptive design is opened on the desktop browser, first the available space is detected and then the layout with most appropriate sizes are picked and used for the display of contents. Resizing of browser window has no affect on the design.	When a website developed using responsive design is opened on a desktop browser and when we try to resize the browser window, the content of the website is dynamically and optimally rearranged to accomodate the window.
Usually, adaptive designs use six standard screen widths - 320 px, 480 px, 760 px, 960 px, 1200 px, 1600 px. These sizes are detected and appropriate layouts are loaded.	This design makes use of CSS media queries for changing styles depending on the target devices properties for adapting to different screens.
It takes a lot of time and effort to first examine the options and realities of the end users and then design best possible adaptive solutions them.	Generally, Responsive design takes much less work to build and design fluid websites that can accomodate content from screen depending on the screen size.
Gives a lot of control over the design to develop sites for specific screens.	No much control over the design is offered here.

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

The order of matching selectors goes from **right to left** of the selector expression. The elements in the DOM are filtered by browsers based on the key selectors and are then traversed up to the parent elements for determining the matches. The speed of determining the elements depends on the length of the chain of selectors. Consider an example:

```
p span{
   color: black;
}
```

```
p span{
        color:   black;
}
```

Here, the browser first finds all span elements in the DOM and then it traverses to each of its parent elements to check if they are the paragraph p elements.

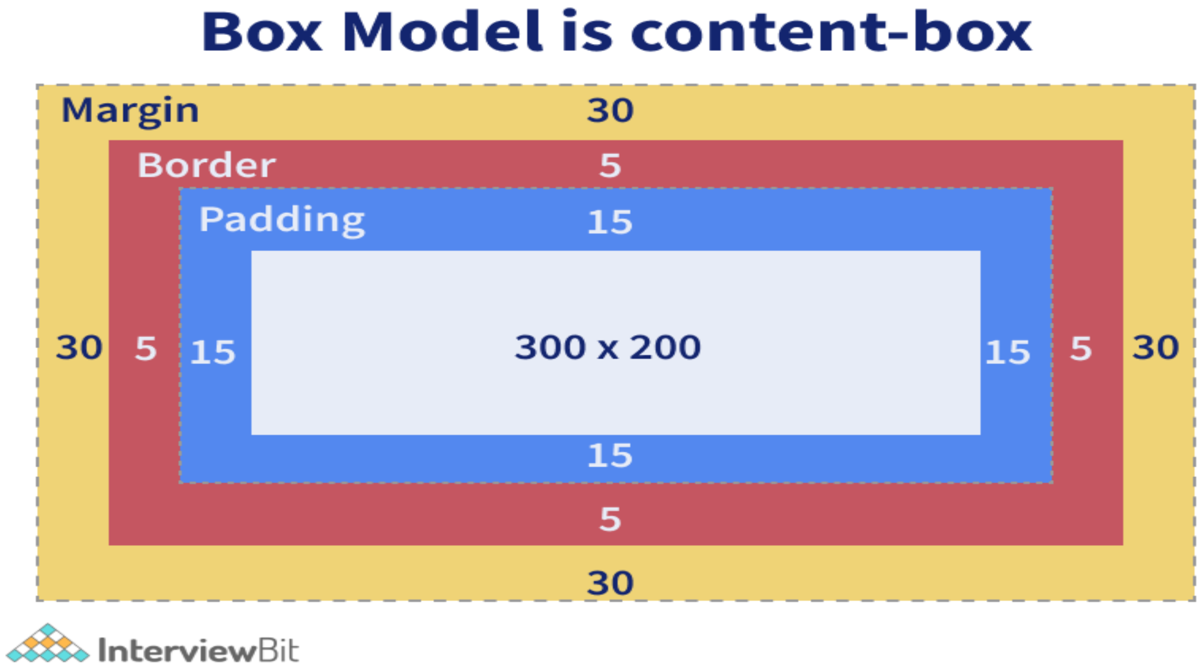
Once the browser finds **all** matching span tags having paragraph elements as parent and applies the color of black to the content, the matching process is stopped.

17. How is border-box different from content-box?

content-box is the default value box-sizing property. The height and the width properties consist only of the content by excluding the border and padding. Consider an example as shown:

```
div{
   width:300px;
   height:200px;
   padding:15px;
   border: 5px solid grey;
   margin:30px;
   -moz-box-sizing:content-box;
   -webkit-box-sizing:content-box;
   box-sizing:content-box;
}
```

Here, the box-sizing for the div element is given as content-box. That means, the height and width considered for the div content exclude the padding and border. We will get full height and width parameters specified for the content as shown in the below image.



The border-box property includes the content, padding and border in the height and width properties. Consider an example as shown:

```
div{
   width:300px;
   height:200px;
   padding:15px;
   border: 5px solid grey;
   margin:30px;
   -moz-box-sizing:border-box;
   -webkit-box-sizing:border-box;
   box-sizing:border-box;
}
```

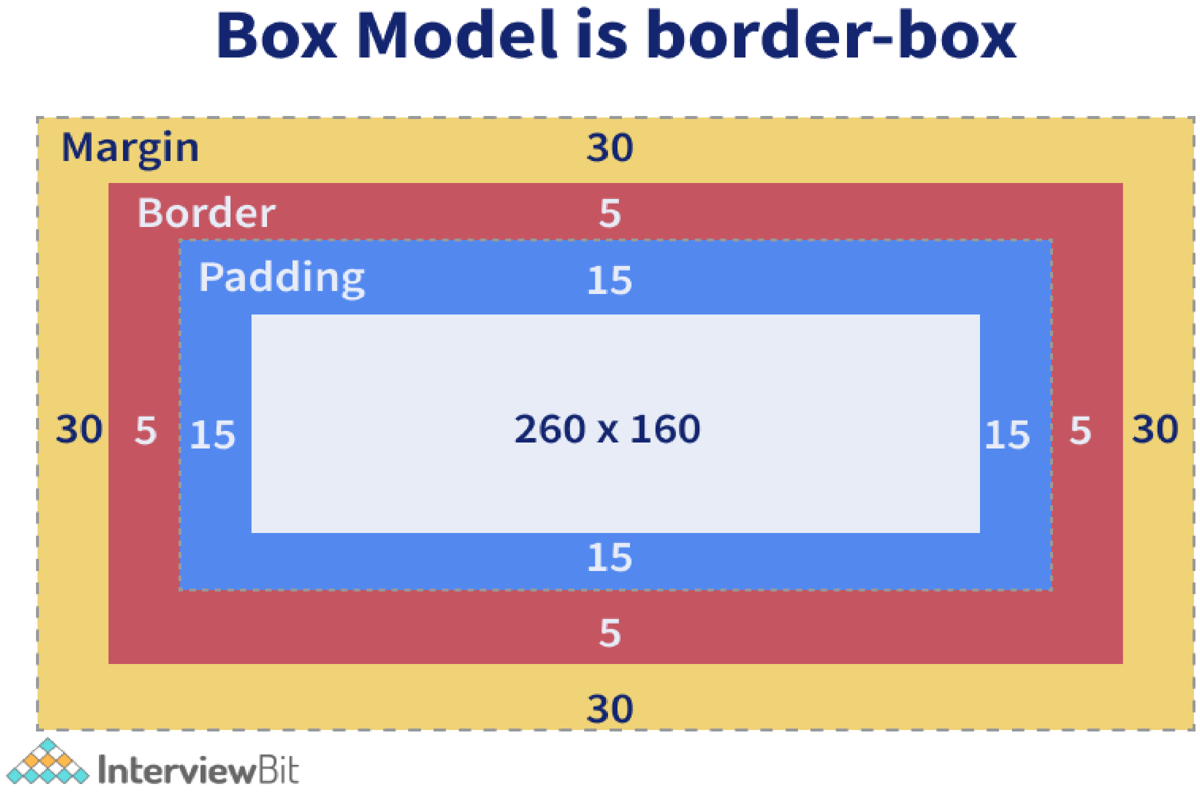
Here, the box-sizing for the div element is given as border-box. That means the height and width considered for the div content will also include the padding and border. This means that the actual height of the div content will be:

```
actual height = height -
            padding on top and bottom -
            border on top and bottom
          = 200 - (15*2) - (5*2)
          = 160 px
```

and the actual width of the div content would be:

```
actual width  = width -
               padding on right and left -
               border on right and left
             = 300 - (15*2) - (5*2)
             = 260 px
```

This is represented in the image below:



18. How is opacity specified in CSS3?

Opacity refers to the degree to which the content is transparent or opaque. We can use the property named opacity which takes the values ranging from 0 to 1. 0 specifies that the element is completely transparent where 1 means that the element is completely opaque. We can use the opacity property as follows:

```
 div {
   opacity: 0.6;
}
```

In the above example, an opacity of 60% is applied to the div section. The opacity property is not supported by the internet explorer browser. To make it work there, we need to use filter property as polyfill as shown in the example below.

```
div {
   opacity: 0.6;
   filter: alpha(opacity=60);
}
```

10% opacity
30% opacity
60% opacity
opacity 1
InterviewBit

19. Why should we use float property in CSS?

The float property is used for positioning the HTML elements horizontally either towards the left or right of the container. For instance,

```
float-demo {
    float: right;
}
```

Here, the element to which the class is applied ensures that the element is positioned on the right of the container. If you specify the value of float as left, then the element will be placed on the left side of the container.

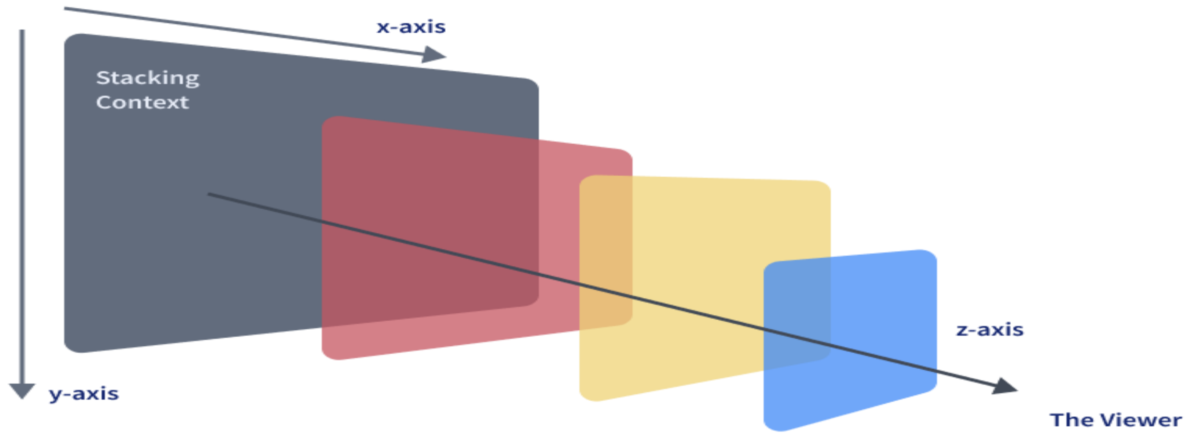
20. What is a z-index, how does it function?

z-index is used for specifying the vertical stacking of the overlapping elements that occur at the time of its positioning. It specifies the vertical stack order of the elements positioned that helps to define how the display of elements should happen in cases of overlapping.

The default value of this property is 0 and can be either positive or negative. Apart from 0, the values of the z-index can be:

- Auto: The stack order will be set equal to the parent.
- Number: The number can be positive or negative. It defines the stack order.
- Initial: The default value of 0 is set to the property.
- Inherit: The properties are inherited from the parent.

The elements having a lesser value of z-index is stacked lower than the ones with a higher z-index.



The z-index property determines the order of the element on the z-axis of the stacking contect.

InterviewBit

From the above figure, we can see that as the value of the z-index increases along the z-axis, the order of stacking would be towards the top of other elements along the vertical axis.

22. What are the properties of flexbox?

Flexbox stands for flexible box and it was introduced around 2017 in CSS with the purpose of providing an efficient way to handle layouts, align elements within them and distribute spaces amongst the items in dynamic/responsive conditions. It provides an enhanced ability to alter the dimensions of the items and make use of the available space in the container efficiently. In order to achieve this, CSS3 provides some properties.

The properties of flexbox are as follows:

- **flex-direction**: This property helps in defining the direction the container should stack the items targetted for flex. The values of this property can be
    - row: Stacks items horizontally from left to right in the flex container.
    - column: Stacks items vertically from top to bottom in the flex container.

    - row-reverse: Stacks items horizontally from right to left in the flex container.
    - column-reverse: Stacks items vertically from bottom to top in the flex container.
- **flex-wrap**: This property specifies of the flex items should be wrapped or not. Possible values are:
    - wrap: The flex items would be wrapped if needed.
    - nowrap: This is the default value that says the items won't be wrapped.
    - wrap-reverse: This specifies that the items will be wrapped if needed but in reverse order.
- **flex-flow**: This property is used for setting both flex-direction and flex-wrap properties in one statement.
- **justify-content**: Used for aligning the flex items. Possible values are:
    - center: It means that all the flex items are present at the center of the container.
    - flex-start: This value states that the items are aligned at the start of the container. This is the default value.
    - flex-end: This value ensures the items are aligned at the end of the container.
    - space-around: This value displays the items having space between, before, around the items.
    - space-between: This value displays items with spaces between the lines.
- **align-items**: This is used for aligning flex items.
- **align-content**: This is used for aligning the flex lines.

23. What is cascading in CSS?

"Cascading" refers to the process of going through the style declarations and defining weight or importance to the styling rules that help the browser to select what rules have to be applied in times of conflict. The conflict here refers to multiple rules that are applicable to a particular HTML element. In such cases, we need to let the browser know what style needs to be applied to the element. This is done by cascading down the list of style declarations elements.

For example, if we have the below style:

```
p{
   color:white;
}
```

and we also have the following declaration below it or in another stylesheet that has been linked to the page:

```
p{
   color: black;
}
```

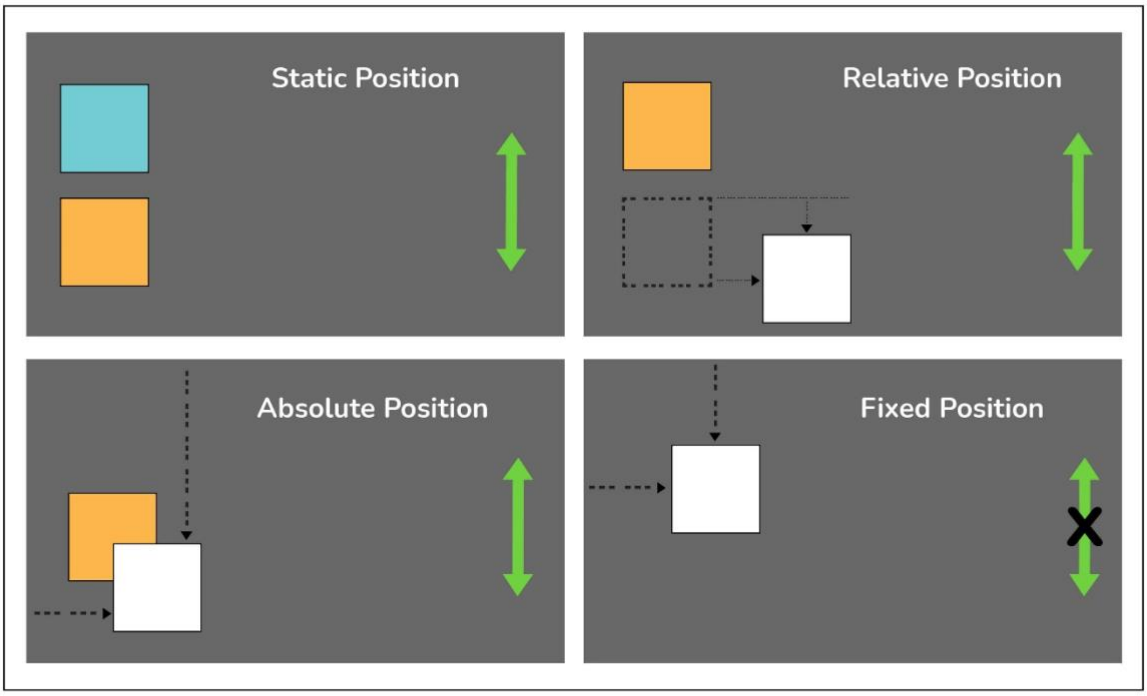
We have a conflict in color property here for the paragraph elements. Here, the browser just cascades down to identify what is the most recent and most specific style and applies that. Since we have the color:black; as the most specific declaration, the color black is applied to the paragraph elements. Now if you want to ensure color white is applied to the paragraph, we can define weight to that style by adding !important as shown below:

```
p{
   color:white !important;
}
```

!important ensures that the property has the maximum weight in presence of other conflicting properties.

24. Explain CSS position property?

- **Absolute:** To place an element exactly where you want to place it. absolute position is actually set relative to the element's parent. if no parent is available then the relative place to the page itself (it will default all the way back up to the element).
- **Relative:** "Relative to itself". Setting position: relative; on an element and no other positioning attributes, it will no effect on its positioning. It allows the use of z-index on the element and it limits the scope of absolutely positioned child elements. Any child element will be absolutely positioned within that block.
- **Fixed:** The element is positioned relative to the viewport or the browser window itself. viewport doesn't change if you scroll and hence the fixed element will stay right in the same position.
- **Static:** Static default for every single page element. The only reason you would ever set an element to position: static is to forcefully remove some positioning that got applied to an element outside of your control.
- **Sticky:** Sticky positioning is a hybrid of relative and fixed positioning. The element is treated as relative positioned until it crosses a specified threshold, at which point it is treated as fixed positioned.



InterviewBit

25. When does DOM reflow occur?

Reflow is the name of the web browser process for re-calculating the positions and geometries of elements in the document, for the purpose of re-rendering part or all of the document.

Reflow occurs when:

- Insert, remove or update an element in the DOM.
- Modify content on the page, e.g. the text in an input box.
- Move a DOM element.
- Animate a DOM element.
- Take measurements of an element such as offsetHeight or getComputedStyle.
- Change a CSS style.

26. Different Box Sizing Property?

The box-sizing CSS property sets how the total width and height of an element are calculated.

- **Content-box:** The default width and height values apply to the element's content only. The padding and border are added to the outside of the box.
- **Padding-box:** Width and height values apply to the element's content and its padding. The border is added to the outside of the box. Currently, only Firefox supports the padding-box value.

- **Border-box:** Width and height values apply to the content, padding, and border.

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

- **Centering with Table:**

HTML:

```
<div class=”cn”><div class=”inner”>your content</div></div>
```

CSS:

```
.cn {
        display: table-cell;
        width: 500px;
        height: 500px;
        vertical-align: middle;
        text-align: center;
}

.inner {
        display: inline-block;
        width: 200px; height: 200px;
}
```

- **Centering with Transform**

HTML:

```
<div class="cn"><div class="inner">your content</div></div>
```

CSS:

```
.cn {
        position: relative;
        width: 500px;
        height: 500px;
}

.inner {
        position: absolute;
        top: 50%; left: 50%;
        transform: translate(-50%,-50%);
        width: 200px;
        height: 200px;
}
```

- **Centering with Flexbox**

HTML:

```
<div class="cn"><div class="inner">your content</div></div>
```

CSS:

```
.cn {
        display: flex;
        justify-content: center;
        align-items: center;
}
```

- **Centering with Grid**

HTML:

```
<div class=”wrap_grid”>
        <div id=”container”>vertical aligned text<br />some more text here
        </div>
</div>
```

CSS:

```
.wrap-grid {
        display: grid;
        place-content: center;
}
```

28. Can you name the four types of @media properties?

The four types of @media properties are:

1. All → It’s the default property. Used for all media-type devices.
2. Screen → Used for computer screen, mobile screen.
3. Print → Used for printers.
4. Speech → Used for screen readers.

29. What is the grid system?

CSS Grid Layout is the most powerful layout system available in CSS. It is said to be a 2-dimensional system, meaning it can handle both columns and rows, unlike flexbox which is largely a 1-dimensional system.

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

- Using display property(**display: none**). It’s not available for screen readers. The element will not exist in the DOM if display: none is used.
- Using visibility property(**visibility: hidden**), will take up the space of the element. It will be available to screen reader users. The element will actually be present in the DOM, but not shown on the screen.
- Using position property (**position: absolute**). Make it available outside the screen.

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

The :root selector allows you to target the highest-level “parent” element in the DOM, or document tree. It is defined in the CSS Selectors Level 3 specification.

32. What does Accessibility (a11y) mean?

Accessibility refers to how software or hardware combinations are designed to make a system accessible to persons with disabilities, such as visual impairment, hearing loss, or limited dexterity.

For example, a website developed with accessibility in mind might have text-to-speech capabilities. In the USA public websites have to have accessible compliance. It’s defined in 508 compliance. It gives the guidelines and best practices for all website users that should be met with key areas of accessibility.

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

The keyword initial can be used to reset it to its default value.

34. Difference between CSS grid vs flexbox?

- CSS Grid Layout is a two-dimensional system, meaning it can handle both columns and rows. Grid layout is intended for larger-scale layouts which aren’t linear in design.
- Flexbox is largely a one-dimensional system (either in a column or a row). Flexbox layout is most appropriate to the components of an application.

35. How does Calc work?

The CSS3 calc() function allows us to perform mathematical operations on property values. Instead of declaring, for example, static pixel values for an element's width, we can use calc() to specify that the width is the result of the addition of two or more numeric values.

```
.foo {
        Width: calc(100px + 50px)
}
```

36. What do CSS Custom properties variables mean?

Custom properties (sometimes referred to as CSS variables or cascading variables) are defined by users that contain specific values to be reused throughout a document. The value is set using -- notion. And the values are accessed using the var() function.

```
:root {
        --main-bg-color: brown
}

.one {
        color: white;
        background-color· var (--main-bg-color);
        margin: l0px,
        width: 50px,
        height: 5Opx;
        display: inline-block;
}
```

37. What is the difference between CSS variables and preprocessor(SASS, LESS, Stylus) variables?

- CSS variables can be used without the need for a preprocessor. Currently, all the major browsers support the CSS variables.
- CSS variable cascade. But the preprocessor variables don’t cascade.
- CSS variable can be accessed and manipulated in javascript.

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

- It makes every element in the document include the padding and border in the element’s inner dimension for the height and width computation.
- In box-sizing: border-box, The height of an element is now calculated by the content's height + vertical padding + vertical border width.
- The width of an element is now calculated by the content's width + horizontal padding + horizontal border width.

39. What does !important mean in CSS?

The style is having the important will have the highest precedence and it overrides the cascaded property.

```
 p {
        color: red !important;
 }
 #thing {
```

```
            color: green;
 }
 <p id="thing">Will be RED.</p>
```

40. What is specificity? How to calculate specificity?

A process of determining which CSS rule will be applied to an element. It actually determines which rules will take precedence. Inline style usually wins then ID then the class value (or pseudo-class or attribute selector), the universal selector (*) has no specificity. ID selectors have a higher specificity than attribute selectors.

41. What is progressive rendering? How do you implement progressive rendering in the website?. What are the advantages of it?

Progressive rendering is the name given to techniques used to improve the performance of a webpage (in particular, improve perceived load time) to render content for display as quickly as possible.

We can implement the progressive rendering of the page by loading the lazy loading of the images. We can use Intersection Observer API to lazy load the image. The API makes it simple to detect when an element enters the viewport and take an action when it does. Once the image enters the viewport, we will start loading the images.

A sample snippet is given below.

```
<img class="lazy"
src="placeholder-image.jpg"
data-src="image-to-lazy-load-1x.jpg"
data-srcset="image-to-lazy-load-2x.jpg 2x, image-to-lazy-load-1x.jpg 1x"
alt="I'm an image!">

document.addEventListener("DOMContentLoaded", function() {
  var lazyImages = [].slice.call(document.querySelectorAll("img.lazy"));

  if ("IntersectionObserver" in window) {
    let lazyImageObserver = new IntersectionObserver(function(entries, observer) {
      entries.forEach(function(entry) {
        if (entry.isIntersecting) {
          let lazyImage = entry.target;
          lazyImage.src = lazyImage.dataset.src;
          lazyImage.srcset = lazyImage.dataset.srcset;
          lazyImage.classList.remove("lazy");
          lazyImageObserver.unobserve(lazyImage);
        }
      });
    });
    lazyImages.forEach(function(lazyImage) {
      lazyImageObserver.observe(lazyImage);
    });
  } else {
    // Possibly fall back to event handlers here
  }
});
```

42. What are the advantages of using translate() instead of absolute position?

Translate() does not cause the browser to trigger repaint and layout and instead only acts on the compositor. The absolute position triggers the repaint or DOM reflow. So, translate() gives better performance.

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

```
<head>
        <link h ref=" stylel. css" rel=" stylesheet">
        <link href="style2.css" rel="stylesheet">
</head>
```

No, the browsers will download the CSS in the order of its appearance on the HTML page.

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

The @support in CSS can be very useful to scan if the current browser has support for a certain feature.

```
@supports (display: grid) {
        div {
                display: grid;
        }
}
```

45. How does the absolute positioning work?

Absolute positioning is a very powerful positioning mechanism that allows users to place any element wherever they want in an exact location. The CSS properties right, left, top, bottom and define the exact locations where you need to place the element. In absolute positioning, the following points need to be considered:

- The element to which the absolute positioning is applied is removed from the normal workflow of the HTML document.
  - The HTML layout does not create any space for that element in its page layout.

- The element is positioned relative to the closest positioned ancestor. If no such ancestor is present, then the element is placed relative to the initial container block.
- The final position of the element is determined based on values provided to the top, right, left, bottom.

46. How does this property work overflow: hidden?

The overflow property in CSS is used for specifying whether the content has to be clipped or the scrollbars have to be added to the content area when the content size exceeds the specified container size where the content is enclosed. If the value of overflow is hidden, the content gets clipped post the size of the container thereby making the content invisible. For example,

```
div {
   width: 150px;
   height: 50px;
   overflow: hidden;
}
```

If the content of the div is very large and exceeds the height of 50px, the content gets clipped post 50px and the rest of the content is not made visible.

47. How will you align content inside the p tag at the exact center inside the div?

We can add the text-align: center property inside the parent div for aligning the contents horizontally. But it will not align the contents vertically. We can align the content vertically by making the parent element have relative positioning and the child element have absolute positioning. The child element should have the values of top, bottom, right, left as 0 to center it in the middle vertically. Then we need to set the margin as auto. It is assumed that both the child and mother elements will have height and width values.

Consider we have a div element of height and width taking 20% of the screen size, and we have a paragraph element taking the height of 1.2em and width of 20%. If we want to align the paragraph element at the center (vertically and horizontally), we write the following styles:

```
div {
   position : relative;  // Make position relative
   height : 20%;
   width : 20%;
   text-align : center; //Align to center horizontally
}
p {
   position : absolute; // Make position absolute
   top:0;              // Give values of top, bottom,left, right to 0
   bottom:0;
   left:0;
   right:0;
   margin : auto;       // Set margin as auto
   height : 1.2 em;
   width : 20%;
}
```

48. How is margin different from padding in CSS?

Margin property using which we can create space around the elements. We can also create space for borders defined at the exteriors. We have the following properties for defining the margin:

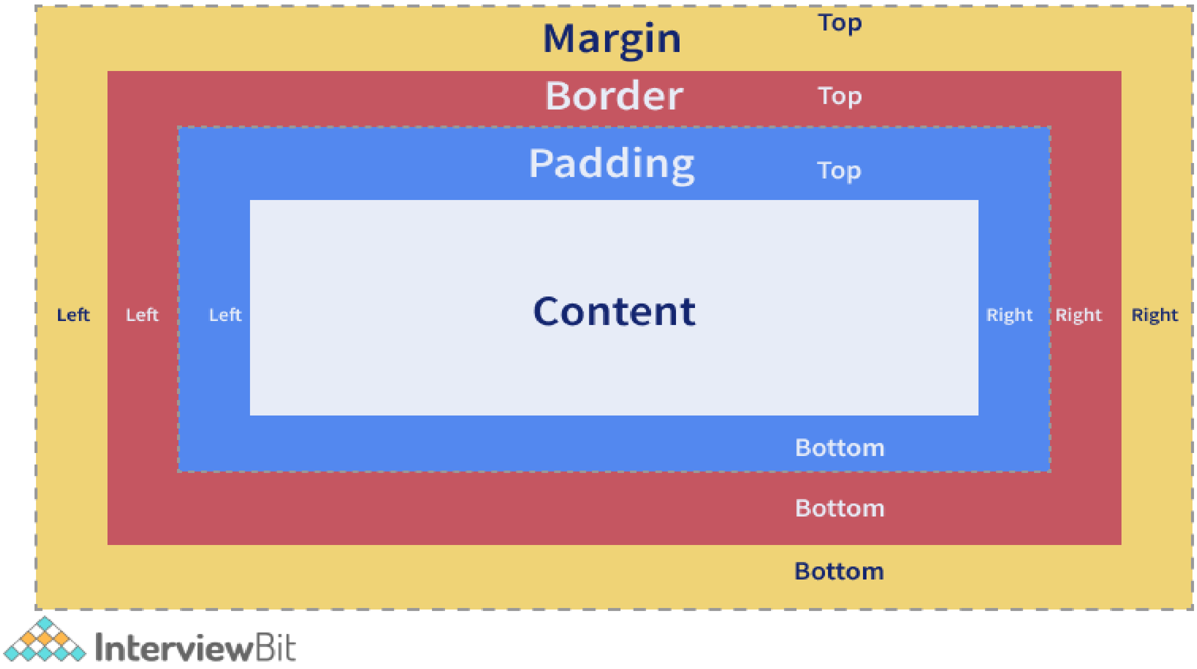
- margin-top
- margin-right
- margin-bottom
- margin-left
  margin property by itself has the values as:
- auto – The browser auto-calculates the margin while we use this.
- length – The value of this property can be in px, pt, cm, em etc. The values can be positive or negative.
- % – We can also give percentage value as margin to the element.
- inherit – Using this property, the margin properties can be inherited from the parent elements.

The padding property is used for generating the space around the element's content and inside any known border. The padding also has sub-properties like:

- padding-top
- padding-right
- padding-bottom
- padding-left

It is to be noted that the padding does not allow negative values.

From the below image, we can see that the Margin is the outermost entity of the CSS Box Model that lies outside of the borders whereas the padding lies within the borders.



49. What do you have to do to automatically number the heading values of sections and categories?

We can use the concept of CSS counters. This lets us adjust the appearance of the content based on the location in a document. While using this, we need to first initialize the value of the counter-reset property which is 0 by default. The same property is also used for changing the value to any number that we need. Post initialization, the counter's value can be incremented or decremented by using the counter-increment property. The name of the counter cannot be CSS keywords like "none", "initial", "inherit" etc. If the CSS keywords are used, then the declaration would be ignored.

Consider an example as shown below:

```
body {
  counter-reset: header;   /* define counter named 'header' whose initial value is 0 by default */
}

h2::before {
  counter-increment: header;   /* The value of header counter by 1.*/
  content: "Header " counter(header) ": ";  /* To display word Header and the value of the counter with colon before it.*/
}
```

Here, we are trying to achieve auto count increment and display feature for the h2 tag. Wherever we use h2 tag, the content will be prefixed by "Header 1 : " , "Header 2 : ", "Header 3 : " etc.

50. How is the nth-child() different from nth of type selectors?

Both are pseudo-classes (Pseudo-classes are those keywords that specifies the special state of the selected element). The nth-child() pseudo-class is used for matching elements based on the number that represents the position of an element based on the siblings. The number is used to match an element on the basis of the element’s position amongst its siblings.

For example, in the below piece of code, if we give nth-child(4) for the example class, then the 4th child of the example class is selected irrespective of the element type. Here, the fourth child of the example class is the div element. The element is selected and a background of black is added to it.

```
.example:nth-child(4) {
   background: black;
}
<div class="example">
          <p>This is a paragraph.</p>
          <p>This is a paragraph.</p>
          <p>This is a paragraph.</p>
          <div>This is a div.</div> <!-- 4th Element to select and apply style-->
          <div>This is a div.</div>
          <p>This is a paragraph.</p>
          <p>This is a paragraph.</p>
          <div>This is a div.</div>
</div>
```

The nth-of-type() pseudo-class is similar to the nth-child but it helps in matching the selector based on a number that represents the position of the element within the elements that are the siblings of its same type. The number can also be given as a function or give keywords like odd or even.

For example, in the below piece of code, if we give p:nth-of-type(even) for the example class, then all the even paragraph tags are selected within the example class and the style of background black is applied to them. The selected elements are marked in comments in the below code:

```
.example p:nth-of-type(even) {
   background: black;
}
<div class="example">
          <p>This is a paragraph.</p>
          <p>This is a paragraph.</p> <!-- Select this and apply style-->
```

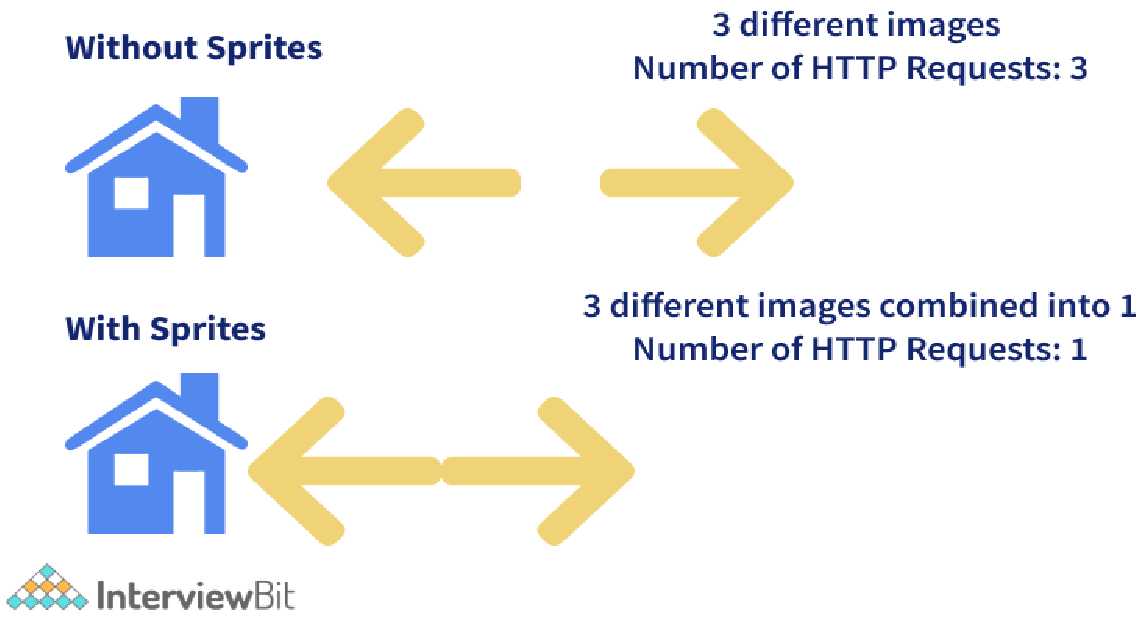
```
        <p>This is a paragraph.</p>
        <div>This is a div.</div>
        <div>This is a div.</div>
        <p>This is a paragraph.</p> <!-- Select this and apply style-->
        <p>This is a paragraph.</p>
        <div>This is a div.</div>
        <p>This is a paragraph.</p> <!-- Select this and apply style-->
        <div>This is a div.</div>
</div>
```

51. What is the importance of CSS Sprites?

CSS sprites are used for combining multiple images in a single larger image. They are commonly used for representing icons that are used in the user interfaces. The main advantages of using sprites are:

- It reduces the number of HTTP requests to get data of multiple images as they are acquired only by sending a single request.
- It helps in downloading assets in advance that help display icons or images upon hover or other pseudo-states.
- When there are multiple images, the browser makes separate calls to get the image for each of them. Using sprites, the images are combined in one and we can just call for that image using one call.

Consider an example where our application requires 3 images as shown below (Without Sprites Section). If we are trying to load the images independently, we require 3 different HTTP Requests to get the data. But if we have CSS Sprites where all 3 images are combines into 1 separated by some spaces, then we require only 1 HTTP Request.


Without Sprites
3 different images
Number of HTTP Requests: 3
With Sprites
3 different images combined into 1
Number of HTTP Requests: 1
InterviewBit

We can access each image from the sprite by accessing the positioning properties as shown in the below code:

```
<!DOCTYPE html>
<html>
<head>
<style>
#home-icon {
  left: 0px;
  width: 46px;
  background: url('spriteFile.gif') 0 0;
}

#prev-icon {
  left: 63px;
  width: 43px;
  background: url('spriteFile.gif') -47px 0;
}

#next-icon {
  left: 129px;
  width: 43px;
  background: url('spriteFile.gif') -91px 0;
}
</style>
</head>
<body>

<img id="home-icon" src="spriteFile.gif" width="1" height="1">    <!-- To display home icon here -->
<img id="next-icon" src="spriteFile.gif" width="1" height="1">    <!-- To display next icon icon here -->
<img id="prev-icon" src="spriteFile.gif" width="1" height="1">    <!-- To display previous icon icon here -->

</body>
</html>
```

In the above code, we are trying to access each element - house, previous and next icon - from the sprite file by using the left, width properties. The image is displayed in the img section by means of the background property. Do note that the source of the image (src attribute of the img tag) is just one file which is the spriteFile.gif and depending on the rules specified in the id selectors, the images are loaded accordingly.

52. What do you understand by tweening in CSS?

Tweening is the process of filling the gaps between the key sequences, i.e between the keyframes that are already created. Keyframes are those frames that represent start and end point of animation action. Tweening involves generating intermediate keyframes between two images that give the impression that the first one has evolved smoothly to the second image. For this purpose, we use properties like transforms - matrix, translate, scale, rotate etc.

In the below example, we are generating intermediate frames of paragraph elements to slide through from the start to the right edge of the browser.

```
p {
  animation-duration: 2s;
  animation-name: slidethrough;
}

@keyframes slidethrough {
  from {
    margin-left: 100%;
    width: 300%;
  }

  to {
    margin-left: 0%;
    width: 100%;
  }
}
```

Here, the paragraph element specifies that the animation process should take 2 seconds for execution from start to the finish. This is done by using the animation-duration property. The animation-name of the @keyframes is defined by using the property animation-name. The intermediate keyframes are defined by using @keyframes rule. In the example, we have just 2 keyframes. The first keyframe starts at 0% and runs till the left margin of 100% which is the rightmost edge of the containing element. The second keyframe starts at 100% where the left margin is set as 0% and the width to be set as 100% which results in finishing the animation flush against the left edge of the container area.

53. Why do we need to use clear property along with floats in CSS?

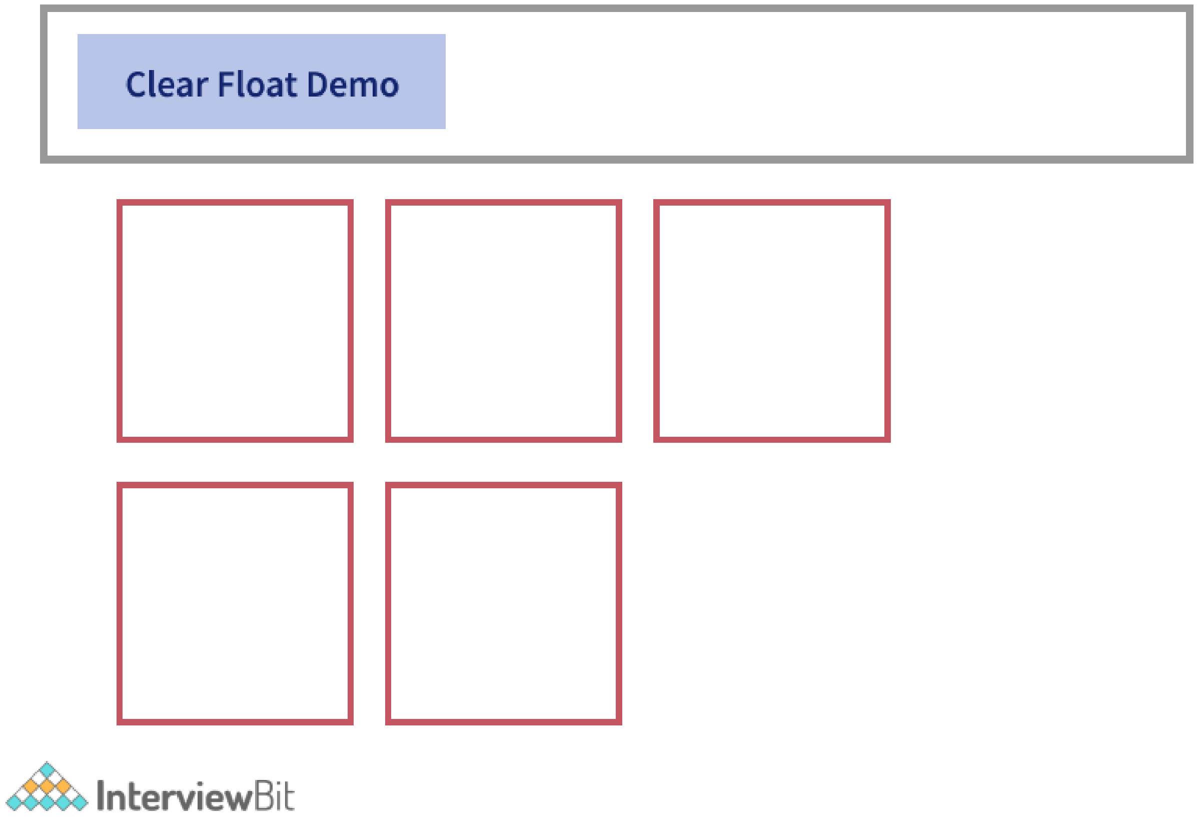
The clear property along with floats is used for specifying which side of floating elements is not supposed to float. An element having clear property ensures that the element does not move up adjacent to the float. But the element will be moved down past the float.

Let us understand this with the help of an example. We know that the floated objects do not add to the height of the objects where they reside. Consider we have a div element with class “floated_div” within another div element with id “main_div”.

```
<html>
    <head>
    <style>
        #main_div {
            width: 400px;
            margin: 10px auto;
            border: 4px solid #cccccc;
            padding: 5px;
        }

        .floated_div {
            float: left;
            width: 50px;
            height: 50px;
            border: 2px solid #990000;
            margin: 10px;
        }
    </style>
    </head>
    <body>
        <div id="main_div">
            <p>Clear Float Demo</p>
            <div class="floated_div"></div>
            <div class="floated_div"></div>
            <div class="floated_div"></div>
            <div class="floated_div"></div>
            <div class="floated_div"></div>
        </div>
    </body>
</html>
```

The result of this code would be as shown below. We see that the squares that are expected to be within dev are not within the main parent div. How do we fix this?



We can do it just by adding <div style="clear:both"></div> line at the end of the last floated element so that the floated elements are fit in properly within the main div container.

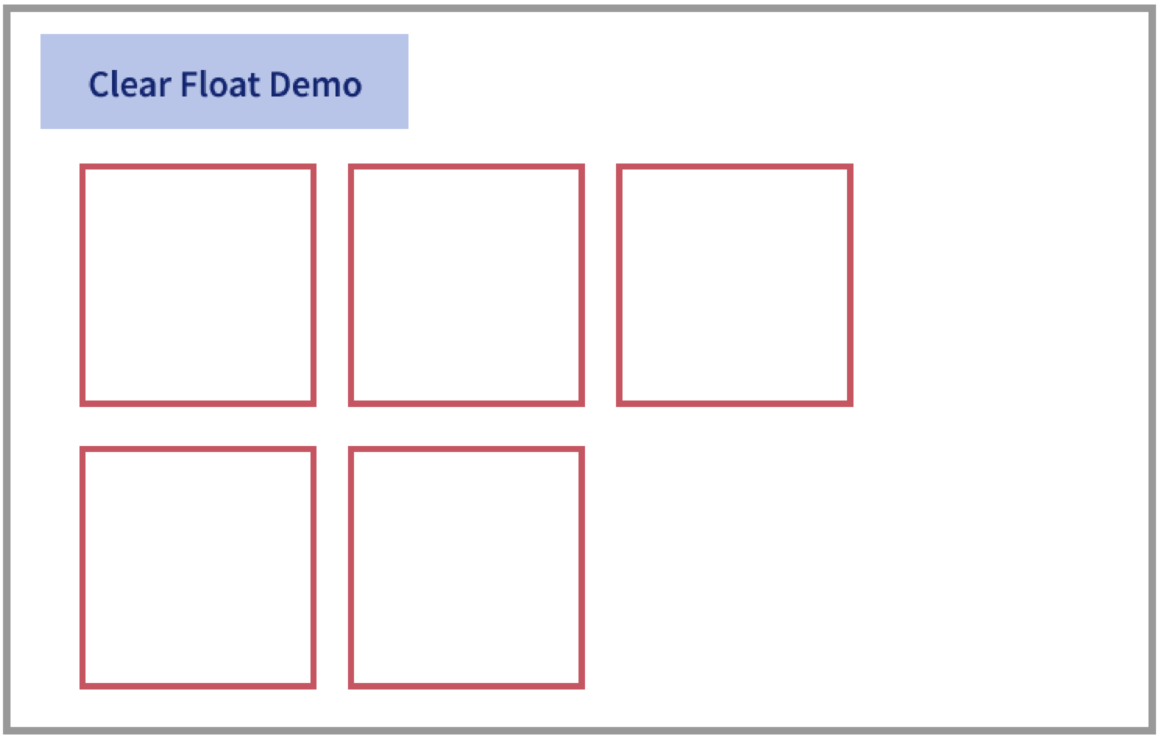
```
<html>
   <head>
   <style>
      #main_div {
          width: 400px;
          margin: 10px auto;
          border: 4px solid #cccccc;
          padding: 5px;
      }

      .floated_div {
          float: left;
          width: 50px;
          height: 50px;
          border: 2px solid #990000;
          margin: 10px;
      }
   </style>
   </head>
   <body>
```

```
        <div id="main_div">
            <p>Clear Float Demo</p>

            <div class="floated_div"></div>
            <div class="floated_div"></div>
            <div class="floated_div"></div>
            <div class="floated_div"></div>
            <div class="floated_div"></div>
            <div style="clear:both"></div>    <!-- Adding this fixed the issue -->
        </div>
    </body>
</html>
```

The result of this will be:



54. How will you fix browser-specific styling issues?

Different ways to fix browser-specific issues.

- We can write browser-specific styles separately in different sheets and load that only when the specific browser is used. This makes use of the server-side rendering technique.
- We can use auto-prefix for automatically adding vendor prefixes in the code.
- We can also use normalize.css or reset CSS techniques.

There are some ways for avoiding browser compatibility issues too. They are as follows:

- **Validate HTML and CSS:** We know that the code will be read, interpreted and handled differently by different browsers. We need to validate our HTML and CSS files for the missing closing tags, or missing semicolons in the syntaxes because there are chances that the old browsers will throw errors while rendering the code. We can avoid those errors by:
    - Maintaining well-aligned code that helps in easy readability.
    - Inserting comments at necessary places.
    - Make use of validation tools like Jigsaw CSS validator, W3C HTML Validators to identify syntax issues in the code.
- **Maintain Cross-Browser Compatibility in the Layouts:** Cross-Browser compatibility is a must while developing web applications. We expect our application to be responsive across all devices, browsers and platforms. Some of the effects of layout incompatibilities are unresponsiveness of the layouts in mobile devices, the difference in layout rendering between modern and old browsers, etc. These incompatibilities can be avoided by using:
    - CSS Multi-Column layouts - For maintaining proper layouts w.r.t columns and containers.
    - HTML viewport metatag – For ensuring content is properly spanned across mobile devices.
    - CSS Flexbox and Grids - To layout child elements depending on the content and available space.
    - CSS resets stylesheets - For reducing browser inconsistencies in default line heights, font sizes, margins etc.
- **Check JavaScript Library issues:** Ensure the libraries are used judiciously and the ones used are supported by the browsers.
- **Check DOCTYPE tag keyword:** The DOCTYPE keyword is meant for defining rules of what needs to be used in the code. Older browser versions check for DOCTYPE tag at the beginning and if not found, the application rendering won't be proper.
- **Test on real devices:** Although applications can be tested on virtual environments, it would be more beneficial if the testing is carried out on real devices and platforms. We can use tools like Testsigma for this purpose that enables us to test in real devices parallelly.

1) What is HTML?

HTML stands for Hyper Text Markup Language. It is a language of World Wide Web. It is a standard text formatting language which is used to create and display pages on the Web. It makes the text more interactive and dynamic. It can turn text into images, tables, links.

2) What are Tags?

HTML tags are composed of three things: an opening tag, content and ending tag. Some tags are unclosed tags.

HTML documents contain two things:

- content, and
- tags

When a web browser reads an HTML document, the browser reads it from top to bottom and left to right. HTML tags are used to create HTML documents and render their properties. Each HTML tags have different properties.

Syntax

1. **<tag>** content **</tag>**

Content is placed between tags to display data on the web page.

3) Do all HTML tags have an end tag?

No. There are some HTML tags that don't need a closing tag. For example: <image> tag, <br> tag. More details.

4) What is formatting in HTML?

The HTML formatting is a process of format the text for a better look and feel. It uses different tags to make text bold, italicized, underlined. More details.

5) How many types of heading does an HTML contain?

The HTML contains six types of headings which are defined with the <h1> to <h6> tags. Each type of heading tag displays different text size from another. So, <h1> is the largest heading tag and <h6> is the smallest one. For example:

1. **<h1>**Heading no. 1**</h1>**
2. **<h2>**Heading no. 2**</h2>**
3. **<h3>**Heading no. 3**</h3>**
4. **<h4>**Heading no. 4**</h4>**
5. **<h5>**Heading no. 5**</h5>**
6. **<h6>**Heading no. 6**</h6>**

6) How to create a hyperlink in HTML?

The HTML provides an anchor tag to create a hyperlink that links one page to another page. These tags can appear in any of the following ways:

- Unvisited link - It is displayed, underlined and blue.
- Visited link - It is displayed, underlined and purple.
- Active link - It is displayed, underlined and red.

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

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

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

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

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

- Ordered list - The ordered list displays elements in numbered format. It is represented by <ol> tag.
- Unordered list - The unordered list displays elements in bulleted format. It is represented by <ul> tag.
- Definition list - The definition list displays elements in definition form like in dictionary. The <dl>, <dt> and <dd> tags are used to define description list.

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

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

10) What is semantic HTML?

Semantic HTML is a coding style. It is the use of HTML markup to reinforce the semantics or meaning of the content. For example: In semantic HTML <b> </b> tag is not used for bold statement as well as <i> </i> tag is used for italic. Instead of these we use <strong></strong> and <em></em> tags.

11) What is an image map?

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

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

You can insert a copyright symbol by using &copy; or &#169; in an HTML file.

13) How to create a nested webpage in HTML?

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

```
<!DOCTYPE html>
<html>
<body>
<h2>HTML Iframes example</h2>
<p>Use the height and width attributes to specify the size of the iframe:</p>
```

```
<iframe src="https://www.javatpoint.com/" height="300" width="400"></iframe>
</body>
</html>
```

14) How do you keep list elements straight in an HTML file?

You can keep the list elements straight by using indents.

15) Does a hyperlink only apply to text?

No, you can use hyperlinks on text and images both. The HTML anchor tag defines a hyperlink that links one page to another page. The "href" attribute is the most important attribute of the HTML anchor tag.

Syntax

```
<a href = "..........."> Link Text </a>
```

16) What is a style sheet?

A style sheet is used to build a consistent, transportable, and well-designed style template. You can add these templates on several different web pages. It describes the look and formatting of a document written in markup language.

17) Can you create a multi-colored text on a web page?

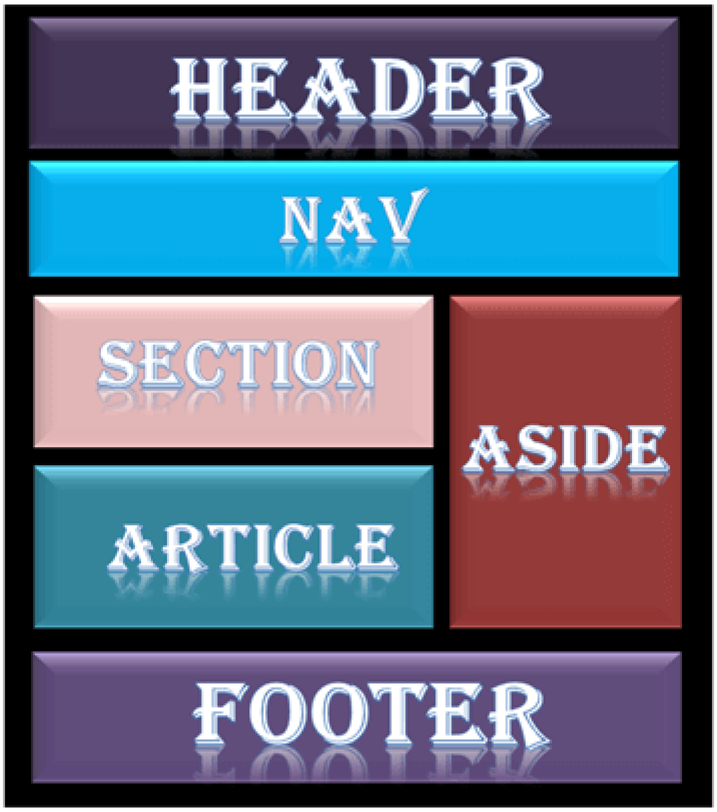
Yes. To create a multicolor text on a web page you can use <font color ="color"> </font> for the specific texts you want to color.

18) Is it possible to change the color of the bullet?

The color of the bullet is always the color of the first text of the list. So, if you want to change the color of the bullet, you must change the color of the text.

19) Explain the layout of HTML?

HTML layout specifies a way in which the web page is arranged.



Every website has a specific layout to display content in a specific manner.

Following are different HTML5 elements which are used to define the different parts of a webpage.

- <header>: It is used to define a header for a document or a section.
- <nav>: It is used to define a container for navigation links
- <section>: It is used to define a section in a document
- <article>: It is used to define an independent, self-contained article
- <aside>: It is used to define content aside from the content (like a sidebar)
- <footer>: It is used to define a footer for a document or a section

20) What is a marquee?

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

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

Three tags are used to separate the texts.

- <br> tag - Usually <br> tag is used to separate the line of text. It breaks the current line and conveys the flow to the next line
- <p> tag - The <p> tag contains the text in the form of a new paragraph.
- <blockquote> tag - It is used to define a large quoted section. If you have a large quotation, then put the entire text within <blockquote>.............</blockquote> tag.

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

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

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

Here, replace the "image.gif" with the name of your image file which you want to display on your web page.

23) What are empty elements?

HTML elements with no content are called empty elements. For example: <br>, <hr> etc.

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

The span tag is used for following things:

- For adding color on text
- For adding background on text
- Highlight any color text

Example:

```
<p>
<span style="color:#ffffff;">
In this page we use span.
</span>
</p>
```

25) What is the use of an iframe tag?

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

Syntax:

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

Example:

```
<iframe src="demo_iframe.html" width="200px" height="200px"></iframe>
```

Target to a link:

```
<iframe src="http://www.javatpoint.com" name="iframe_a"></iframe>
```

26) What are the entities in HTML?

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

27) Why is a URL encoded in HTML?

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

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

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

29) What is the canvas element in HTML5?

The <canvas> element is a container that is used to draw graphics on the web page using scripting language like JavaScript. It allows for dynamic and scriptable rendering of 2D shapes and bitmap images. There are several methods in canvas to draw paths, boxes, circles, text and add images. For Example:

```
<canvas id="myCanvas1" width="300" height="100" style="border:2px solid;">
Your browser does not support the HTML5 canvas tag.
</canvas>
```

30) What is SVG?

HTML SVG is used to describe the two-dimensional vector and vector/raster graphics. SVG images and their behaviors are defined in XML text files. So as XML files, you can create and edit an SVG image with the text editor. It is mostly used for vector type diagrams like pie charts, 2-Dimensional graphs in an X, Y coordinate system.

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

31) What are the different new form element types in HTML 5?

Following is a list of 10 frequently used new elements in HTML 5:

- Color
- Date
- Datetime-local
- Email
- Time
- Url
- Range
- Telephone
- Number
- Search

32) Is there any need to change the web browsers to support HTML5?

No. Almost all browsers (updated versions) support HTML 5. For example Chrome, Firefox, Opera, Safari, IE.

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

HTML 5 supports three types of video format:

- mp4
- WebM
- Ogg

34) Is audio tag supported in HTML 5?

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

1. mp3
2. WAV
3. Ogg

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

```
<audio controls>
  <source src="koyal.mp3" type="audio/mpeg">
Your browser does not support the html audio tag.
</audio>
```

Instead of koyal.mp3, you can pass any mp3 file name.

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

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

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

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

```
<p>The Taj Mahal is widely recognized as "the jewel of Muslim art in India and one of the universally admired masterpieces of the world's heritage."</p>
<figure>
  <img src="htmlpages/images/tajmahal.jpg" alt="Taj Mahal"/>
</figure>
```

37) What is the use of figcaption tag in HTML 5?

The <figcaption> element is used to provide a caption to an image. It is an optional tag and can appear before or after the content within the <figure> tag. The <figcaption> element is used with <figure> element and it can be placed as the first or last child of the <figure> element.

```
<figure>
  <img src="htmlpages/images/tajmahal.jpg" alt="Taj Mahal"/>
<figcaption>Fig.1.1 - A front view of the great Taj Mahal in Agra.</figcaption>
</figure>
```

38) What is button tag?

The button tag is used in HTML 5. It is used to create a clickable button within the HTML form on the web page. It is generally used to create a "submit" or "reset" button. Let's see the code to display the button.

```
<button name="button" type="button">Click Here</button>
```

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

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

40) What is datalist tag?

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

```
<label>
 Enter your favorite cricket player: Press any character<br />
 <input type="text" id="favCktPlayer" list="CktPlayers">
 <datalist id="CktPlayers">
 <option value="Sachin Tendulkar">
 <option value="Brian Lara">
 <option value="Jacques Kallis">
 <option value="Ricky Ponting">
 <option value="Rahul Dravid">
 <option value="Shane Warne">
 <option value="Rohit Sharma">
```

```
<option value="Donald Bradman">
<option value="Saurav Ganguly ">
<option value="AB diVilliers">
<option value="Mahendra Singh Dhoni">
<option value="Adam Gilchrist">
</datalist>
</label>
```

41) How are tags migrated from HTML4 to HTML5?

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

Header and Footer Example

HTML 4 Header and Footer:

```
<div id="header">
  <h1>Monday Times</h1>
</div>
.
.
.
<div id="footer">
  <p>&copy; JavaTpoint. All rights reserved.</p>
</div>
```

HTML 5 Header and Footer:

```
<header>
  <h1>Monday Times</h1>
</header>
.
.
.
<footer>
  <p>© JavaTpoint. All rights reserved.</p>
</footer>
```

Menu Example

HTML 4 Menu:

```
<div id="menu">
  <ul>
    <li>News</li>
    <li>Sports</li>
    <li>Weather</li>
  </ul>
</div>
```

HTML 5 Menu:

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

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

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

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

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

Example:

1. Name: <**input** type="text" name="name" required>

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

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

Example:

1. <**input** type="email">

Q1. What is HTML?

HTML stands for Hyper Text Markup Language. It is a language of the World Wide Web. It is a standard text formatting language which is used to create and display pages on the Web. HTML makes the text more interactive and dynamic. It can turn text into images, tables, links. HTML pages are saved by adding .html or .html in web page name.

Q2. What is the difference between HTML elements and tags?

Elements	**Tags**
The element is an individual component of the HTML web page or document. It represents semantics or meaning. For example, the title element represents the title of the document.	It is the root of the HTML document which is used to specify that the document is HTML. For example, the Head tag is used to contain all the head element in the HTML file.

Q3. What are Attributes and how do you use them?

Each tag has additional attributes that change the way the tag behaves or is displayed. For example, a **<input>** tag has a type attribute, which you can use to specify whether it's a text field, checkbox, radio button or one of many more options. **Attributes** are specified directly after the name of the **tag**, inside the two angled brackets. They should only ever appear in opening tags or in self-closing tags. But, they can never be in **closing tags**.

Example:

```
<!-- Text field -->
<input type="text" />
```

```
<!-- Checkbox -->
<input type="checkbox" />
<!-- Radio button -->
<input type="radio" value="on" />
```

Q4. What is the difference between a block-level element and an inline element?

Block	Inline
A block-level element is drawn as a block that stretches to fill the full width available to it i.e, the width of its container and will always start on a new line. **Elements** that are block-level by default: **<div>, <img>, <section>, <form>, <nav>.**	Inline elements are drawn where they are defined and only take up space that is absolutely needed. The easiest way to understand how they work is to look at how text flows on a page. **Examples** of elements that are inline by default: **<span>, <b>, <strong>, <a>, <input>.**

Q5. When are comments used in HTML?

To understand the code easily, you can add **code comments** to your HTML document. These are not displayed in the browser, but they help you in leaving notes for yourself and other developers as to what a section of HTML is for. The start of the comment is denoted by **<!–** and the end is marked by **—** >. Anything in the middle will be completely ignored, even if it contains valid **HTML**.

For example:

```
<!-- This is a comment! -->
<!-- Comments can span multiple lines too -->
<!-- This part is ignored in the browser -->
```

Q6. What are the HTML tags used to display the data in the tabular form?

The list of HTML tags used to display data in the tabular form include:

Tag	Decsription
<table>	It defines a table
<tr>	This tag defines a row in a table
<th>	It defines a header cell in a table
<td>	This is used to define a cell in a table
<caption>	It defines the table caption
<colgroup>	It specifies a group of one or more columns in a table for formatting
<col>	This is used with <colgroup> element to specify column properties for each column
<tbody>	This tag is used to group the body content in a table.
<thead>	It is used to group the header content in a table
<tfooter>	It is used to group the footer content in a table

Q7. How to create a Hyperlink in HTML?

The HTML provides an anchor tag to create a hyperlink that links one page to another page. These tags can appear in any of the following ways:

- **Unvisited link** – It is displayed, underlined and blue.
- **Visited link** – It is displayed, underlined and purple.
- **Active link** – It is displayed, underlined and red.

The **syntax** of Hyperlink in HTML is:

```
<a href = "..........."> Link Text </a>
```

Q8. Name some common lists that are used when designing a page.

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

- **Ordered list** – The ordered list displays elements in a numbered format. It is represented by <ol> tag.
- **Unordered list** – The unordered list displays elements in a bulleted format. It is represented by <ul> tag.
- **Definition list** – The definition list displays elements in definition form like in a dictionary. The <dl>, <dt> and <dd> tags are used to define description list.

Q9. What is semantic HTML?

Semantic HTML is a coding style. It is the use of **HTML markup** to reinforce the semantics or meaning of the content. For example: In semantic HTML **<b> </b>** tag is not used for bold statement as well as **<i> </i>** tag is used for italic. Instead of these we use **<strong></strong>** and **<em></em>** tags.

Q10. How to create a nested webpage in HTML?

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

```
<!DOCTYPE html>
<html>
<body>
<h2>HTML example</h2>
Use the height and width attributes to specify the size of the iframe:
<iframe src="https://www.edureka.co/" height="300" width="400"></iframe>
```

```
</body>
</html>
```

Q11. What is an image map?

An image map is used for linking many different web pages using a single image. It is represented by <map> tag. You can define shapes in images that you want to include as part of an image mapping.

Q12. Does a hyperlink only apply to text?

No, hyperlinks can be used both on **texts** and **images**. The HTML anchor tag defines a hyperlink that links one page to another page. The “href” attribute is the most important attribute of the HTML anchor tag.

Syntax:

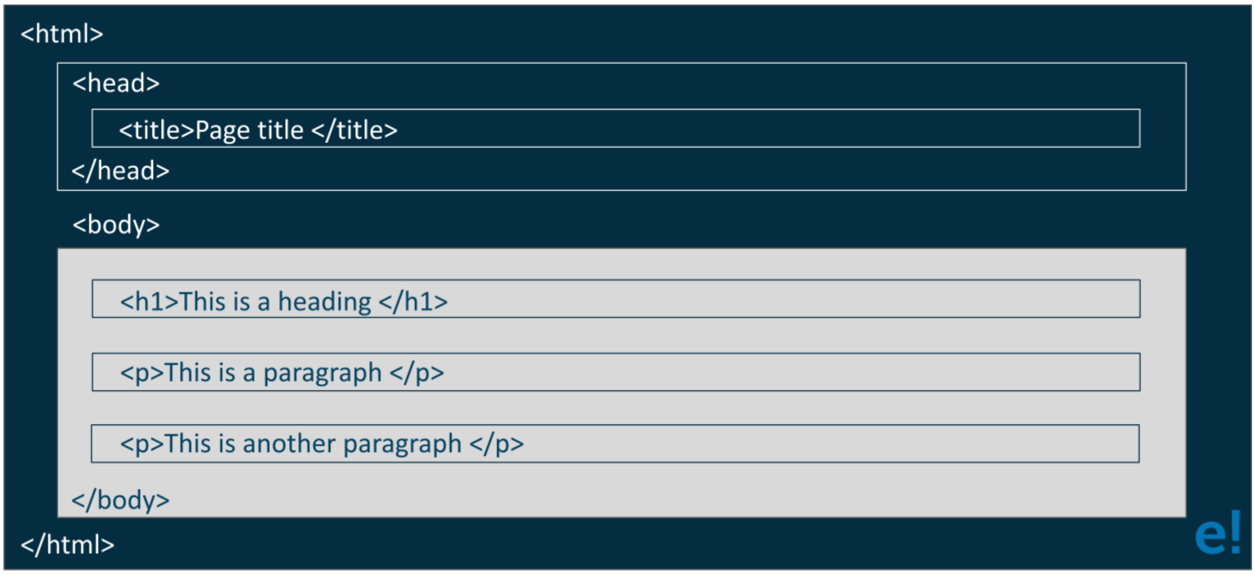
```
<a href = "...........">  Link Text </a>
```

Q13. What is a Style Sheet?

A style sheet is used to build a consistent, transportable, and well-designed **style template**. You can add these templates on several different web pages. It describes the look and formatting of a document written in the markup language.

Q14. Explain the layout of HTML.

HTML layout specifies a way in which the web page is arranged. Every website has a specific layout to display content in a specific manner. Following are different **HTML elements** which are used to define the different parts of a webpage:



```
<html>
    <head>
        <title>Page title </title>
    </head>
    <body>
        <h1>This is a heading </h1>
        <p>This is a paragraph </p>
        <p>This is another paragraph </p>
    </body>
</html>
```

- **<header>**: It is used to define a header for a document or a section.
- **<nav>**: This defines a container for navigation links

- **<section>**: It is used to define a section in a document
- **<article>**: This is used to define an independent, self-contained article
- **<aside>**: It is used to define content aside from the content
- **<footer>**: It is used to define a footer for a document or a section

Q15. What is a marquee?

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

Q16. What are the tags used to separate a section of texts?

There are three tags that can be used to separate the texts:

- **<br>** tag – Usually <br> tag is used to **separate** the **line of text**. It breaks the current line and conveys the flow to the next line
- **<p>** tag – This contains the **text** in the form of a new **paragraph**.
- **<blockquote>** tag – It is used to define a large quoted section. If you have a large quotation, then put the entire text within **<blockquote>……….</blockquote>** tag.

Q17. What is the difference between DIV and SPAN in HTML?

The difference between **span** and **div** is that a span element is **in-line** and usually used for a small chunk of HTML inside a line,such as inside a paragraph. Whereas, a div or division element is **block-line** which is equivalent to having a line-break before and after it and used to group larger chunks of code.

Example:

```
<div id="HTML">
This is <span class="Web Dev">interview</span>
</div>
```

Q18. What is the purpose of using alternative texts in images?

The purpose of using alternative texts is to define what the image is about. During an image mapping, it can be confusing and difficult to understand what hotspots correspond to a particular link. These alternative texts come in action here and put a description at each link which makes it easy for users to understand the hotspot links easily.

Q19. How to create a new HTML element?

You can create new elements for the document in the following way:

```
<script>
```

```
document.createElement("myElement")
</script>
```

It can be also be used in the HTML as:

```
<myElement>hello edureka!</myElement>
```

Q20. Is the <!DOCTYPE html> tag considered as a HTML tag?

No, the <!DOCTYPE html> declaration is not an HTML tag.

There are many type of HTML, such as, HTML 4.01 Strict, HTML 4.01 Transitional, HTML 4.01 Frameset, XHTML 1.0 Strict, XHTML 1.0 Transitional, XHTML 1.0 Frameset, XHTML 1.1 etc. So, <!DOCTYPE html> is used to instruct the web browser about the HTML page.

Q21. Why is a URL encoded in HTML?

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

Q22. What is the use of an iframe tag?

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

Syntax:

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

Example:

```
<iframe src="demo_iframe.html" width="200px" height="200px"></iframe>
```

Target to a link:

```
<iframe src=http://www.edureka.co" name="iframe_a"></iframe>
```

Q23. What are the entities in HTML?

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

Q24. Can you create a multi-colored text on a web page?

Yes, we can create a multi-colored text on a web page. To create a multicolor text, you can use <font color ="color"> </font> for the specific texts that you want to color.

Q25. How to make a picture of a background image of a web page?

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

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

Here, replace the “image.gif” with the name of your image file which you want to display on your web page.

Q26. What is the use of a span tag? Explain with example.

The span tag is used for following things:

- For adding color on text
- To add background on text
- Highlight any color text

Example:

```
<span style="color:#ffffff;">
In this page we use span.
</span>
```

Q27. What is the advantage of collapsing white space?

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

Q28. Is there any way to keep list elements straight in an HTML file?

By using indents, you can keep the list elements straight. If you indent each sub nested list in further than the parent list, you can easily determine the various lists and the elements that it contains.

Q29. Explain The Key Differences Between LocalStorage And SessionStorage Objects.

The key differences between localStorage and sessionStorage objects are as follows:

- The localStorage object stores the data without an expiry date. However, **sessionStorage** object stores the data for only one session.
- In the case of a localStorage object, data will not delete when the browser window closes. However, the data gets deleted if the browser window closes, in the case of sessionStorage objects.

- The data in sessionStorage is accessible only in the current window of the browser. But, the data in the localStorage can be shared between multiple windows of the browser.

Q30. When is it appropriate to use frames?

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

Q31. How to insert a picture into a background image of a web page?

To insert a picture into the background image, you need to place a tag code after the </head> tag in the following way:

```
<body background = “image.gif”>
```

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

Q32. What happens if you open the external CSS file in a browser?

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

Q33. What is the hierarchy that is being followed when it comes to style sheets?

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

Q34. How do you create text on a webpage that allows you to send an email when clicked?

To change the text into a clickable link to send an email, you need to use the *mailto* command within the *href* tag. You can write it in the following way:

```
<a href=”mailto:youremailaddress”>text to be clicked</a>
```

Q35. How are active links different from normal links?

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

Q36. What are the different tags to separate sections of text?

The **<br>** tag is one way to separate the lines of text. There are other tags like the <p> tag and <blockquote> tag that are also used to separate sections of text.

Q37. Are there instances where the text will appear outside of the browser?

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

Q38. Write an HTML table tag sequence that outputs the following:
50 pcs 100 500
10 pcs 5 50

The HTML Code for the above problem is:

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

Q39. What is the advantage of grouping several checkboxes together?

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

Q40. What happens if there is no text between the tags? Does this affect the display of the HTML file?

If there is no text present between the tags, there is nothing to format. Therefore, no formatting will appear. Some tags, such as the tags without a closing tag like the **<img>** tag, do not require any text between them.

Q41. What are the limits of the text field size?

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

Q42. What is the relationship between the border and rule attributes?

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

Q43. What is SVG?

HTML SVG is used to describe the two-dimensional vector and vector or raster graphics. SVG images and their behaviors are defined in XML text files. So as XML files, you can create and edit an **SVG image** with the text editor. It is mostly used for vector type diagrams like pie charts, 2-Dimensional graphs in an X, Y coordinate system.

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

Q44. What is button tag?

The button tag is used in HTML 5. It is used to create a clickable button within the HTML form on the web page. This tag creates a “submit” or “reset” button. The button tag code is as follows:

```
<button name="button" type="button">Click Here</button>
```

Q45. List the media types and formats supported by HTML.

HTML supports a wide range of media formats for sound, music, videos, movies, and animations. Some of the extensions supported by each media format are:

- **Images**– png, jpg, jpeg, gif, apng, svg, bmp, BMP ico, png ico
- **Audio**– MIDI, RealAudio, WMA, AAC, WAV, Ogg, MP3, MP4
- **Video**– MPEG, AVI, WMV, QuickTime, RealVideo, Flash, Ogg, WebM, MPEG-4 or MP4

Q46. What is Cell Spacing and Cell Padding?

Cell Spacing is referred to as the space or gap between the two cells of the same table. Whereas, Cell Padding is referred to as the gap or space between the content of the cell and cell wall or cell border.

Example:

```
<table border cellspacing=3>
<table border cellpadding=3>
<table border cellspacing=3 cellpadding=3>
```

Q47. What is difference between HTML and XHTML?

The differences between HTML and XHTML are:

- HTML is an application of Standard Generalized Markup Language. Whereas, XML is an application of Extensible Markup Language.
- The first one is a static Web Page whereas the later one is a dynamic Web Page.
- HTML allows programmer to perform changes in the tags and use attribute minimization whereas XHTML when user need a new markup tag then user can define it in this.
- HTML is about displaying information whereas XHTML is about describing the information.

Q48. How many types of CSS can be included in HTML?

There are **three** ways to include the CSS with HTML:

- **Inline CSS:** It is used for styling **small contexts**. To use inline styles add the style attribute in the relevant tag.
- **External Style Sheet**: This is used when the style is applied to **many pages**. Each page must link to the style sheet using the **<link>** tag. The <link> tag goes inside the head section.

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

- **Internal Style Sheet:** It is used when a single document has a unique style. Internal styles sheet needs to put in the head section of an HTML page, by using the **<style>** tag in the following way:

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

Q49. What are logical and physical tags in HTML?

Logical tags are used to tell the meaning of the enclosed text. The example of the logical tag is **<strong> </strong>** tag. When we enclose the text in the strong tag, it tells the browser that enclosed text is more important than other texts.

Physical tags are used to tell the browser how to display the text enclosed in the physical tag. Some of the examples of physical tags are **<b>, <big>, <i>.**

Q50. How can you apply JavaScript to a web page?

In order to make your webpage more interactive, you need JavaScript. It is a scripting language that allows you to interact with certain elements on the page, based on user input. As with CSS, there are three main ways of including JavaScript:

Inline

Certain HTML elements allow you to execute a piece of JavaScript when a certain event occurs. For example, a button allows you to run a script when you click on it. These events are accessed through attributes and differ based on the events that are available on each element. Here is an example that shows an alert with a message when the user clicks on it:

```
<button onclick= "alert('Click the Buton!');">Click me!</button>
```

Script block

You can define a script block anywhere on the page, which will get executed as soon as the browser reaches that part of the document. This can be inside the <head> or <body> section of your document.

```
<script>
    var x = 5;
    var y = 6;
    var result = x + y;
    alert(“X + Y is equal to " + result);
</script>
```

Link to a JavaScript file

It allows you to keep the content of the page separate to how users interact with that content. Also, it allows you to load the same script on multiple pages. As with the script block, you can load a JavaScript file from the **<head>** or **<body>.**

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

1. Name some CSS frameworks.

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

2. What do you understand by the universal sector?

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

Example:

```
<style>

* {

   color: blue;

   font-size: 10px;

}

</style>
```

3. Tell us about the use of the ruleset.

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

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

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

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

5. Differentiate between CSS3 and CSS2.

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

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

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

7. Explain a few advantages of CSS.

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

8. What is meant by RGB stream?

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

9. What was the purpose of developing CSS?

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

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

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

1. Define z-index.

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

2. What are the benefits of CSS Sprites?

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

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

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

Multiple elements can be targeted by separating with a comma:

h2, h3 {color: red;}

4. Name media types allowed by CSS.

The different media types allowed by CSS are:

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

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

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

```
h3 {

background-repeat: none;

 }
```

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

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

```
body {

  background-image: url(‘url_of_image’);

  background-repeat: no-repeat;

  background-attachment: fixed;

}
```

7. Name some font-related CSS attributes.

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

8. Define contextual selectors.

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

9. Explain responsive web design.

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

10. Tell us about the general CSS nomenclature.

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

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

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

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

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

3. What are mixins?

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

4. How can you optimize the webpage for prints?

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

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

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

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

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

ID Selector:

```
<style>

{

text-align: right;

color: blue;

}

</style>
```

CSS class Selector:

```
<style>

.right {
```

```
text-align: right;

color: blue;

}

</style>
```

7. Tell us about CSS float property.

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

8. What do you understand by pseudo-elements?

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

9. Differentiate between logical and physical tags.

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

10. How media types in CSS work?

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

```
@media print {

h2 {
```

```
background-color: blue;

 }

}
```

1. Tell us something about CSS3.

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

2. How is a CSS selector used?

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

3. What are CSS image scripts?

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

4. Explain CSS specificity.

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

5. Define gradients in CSS.

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

6. What are the properties of flexbox?

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

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

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

8. What are the position states in CSS?

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

9. Differentiate between absolute and relative in CSS.

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

10. What is common between class and ID?

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