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

Some advantages of HTML5 are:-

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

**2. 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.

**3. 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.

**4. What are tags and attributes in HTML?**

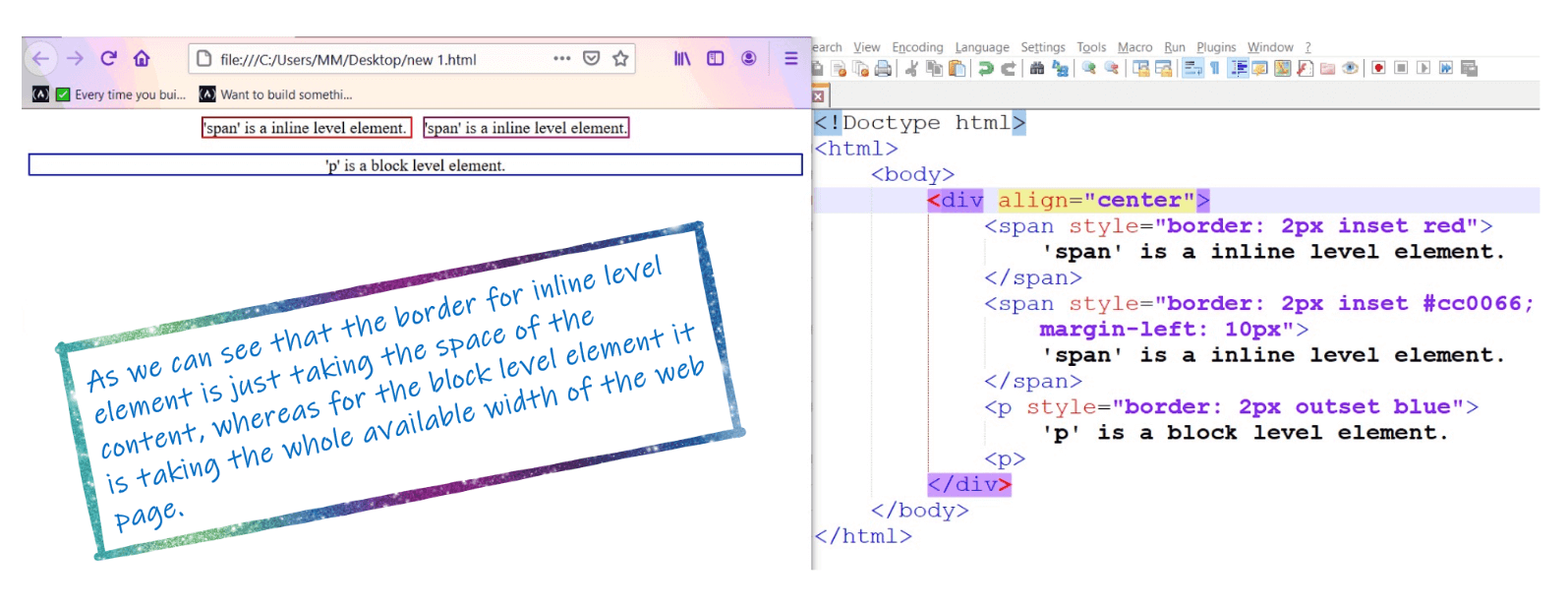
Tags are the primary component of the HTML which defines how the content will be structured/ formatted, whereas Attributes are used along with the HTML tags to define the characteristics of the element. For example, <p align=”center”>Interview questions</p>, in this the ‘align’ is the attribute using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

**5. 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.

**6. Inline and block elements in HTML?**

|  |  |
| --- | --- |
| **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. |



**7. 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.

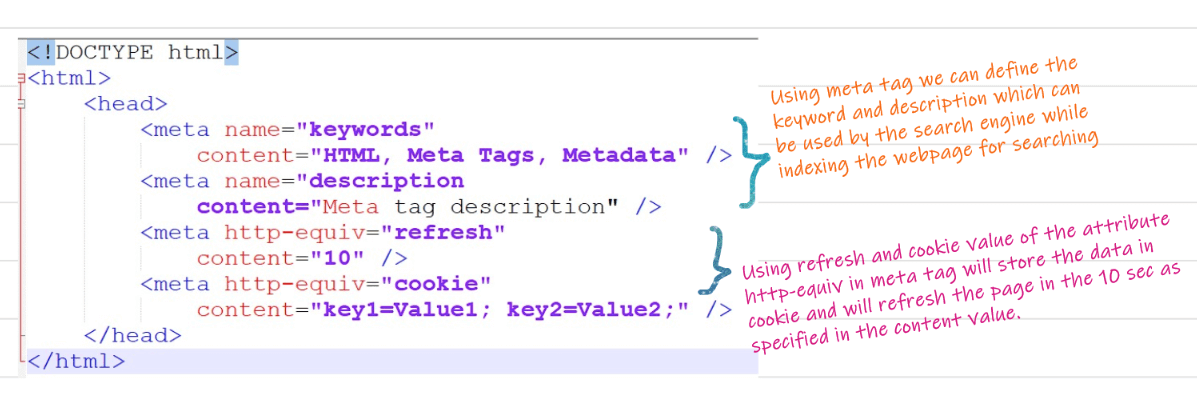
Whereas, <strong> and <em> tags represents that the span of text is of strong importance or more importance and emphatic stress respectively, than the rest of the text. These tags have semantic meaning.

**8. 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 HTML document.

**9. How to specify the metadata in HTML?**

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.



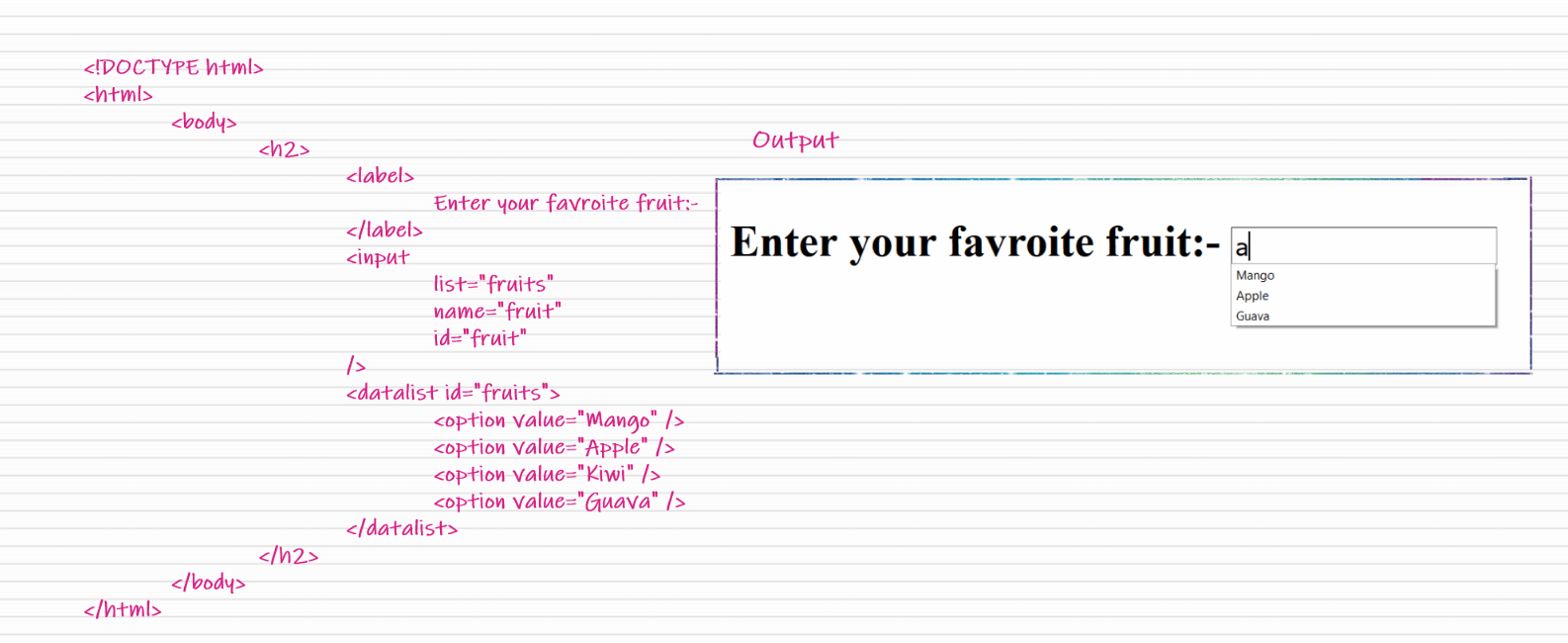
**10. What is the significance of <head> and <body> tag in HTML?**

**<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.

**11. Is the <datalist> tag and <select> tag same?**

No. The <datalist> tag and <select> tag are different. In 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.



**12. 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” />

**13. Define Image Map?**

Image Map, lets a developer to map/link different parts of images with the different webpages. It can be achieved by the <map> tag in HTML, 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>

**14. 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.

**15. What are Semantic Elements?**

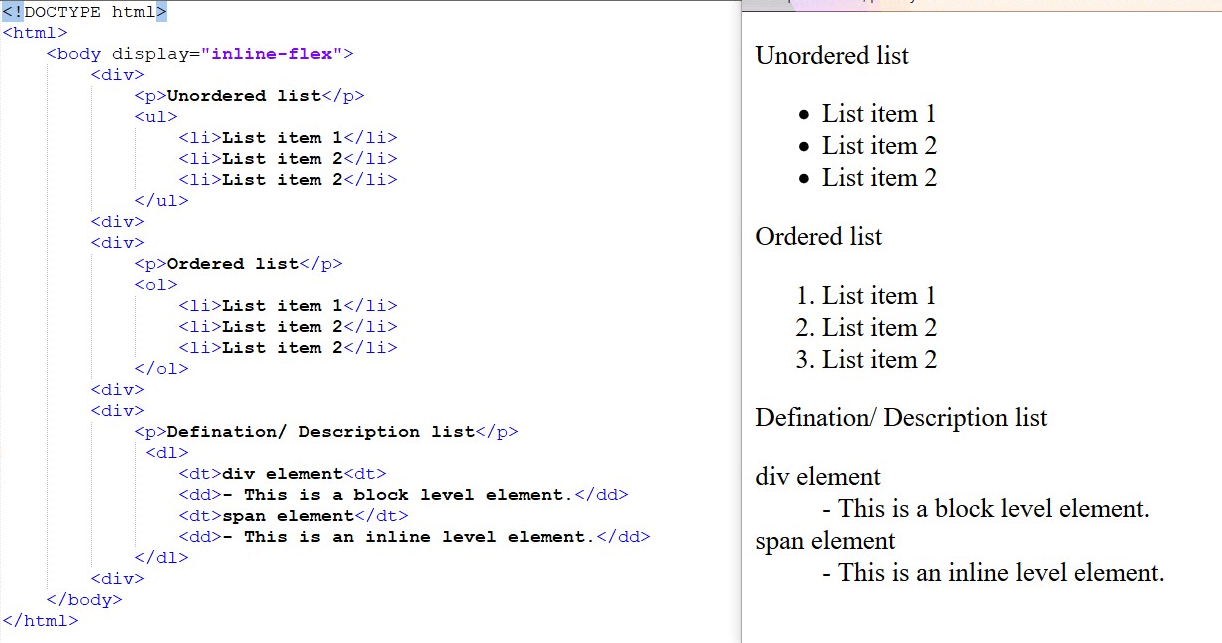
Semantic elements are those which describe the particular meaning to the browser and the developer. Elements like <form>, <table>, <article>, <figure>, etc., are semantic elements.

**16. 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; |

**17. What are different types of lists in HTML?**



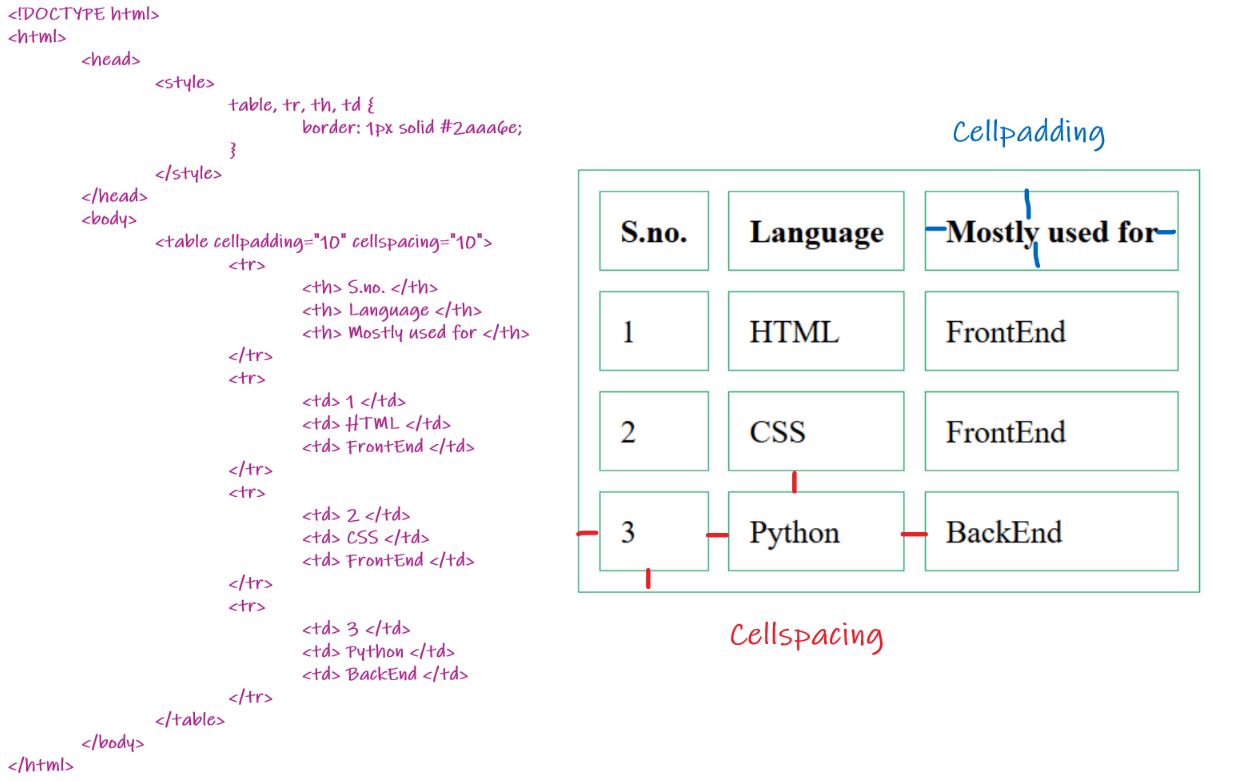
**18. Convert the below data into Tabular format in HTML?**

**S.no., Language, Mostly used for**

1, HTML, FrontEnd

2, CSS, FrontEnd

3, Python, BackEnd



**19. 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.

**20. 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.

**21. 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.

**22. 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.

**23. 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 it’s 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

**24. 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.

**25. 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.

**26. 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.

**27. 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.

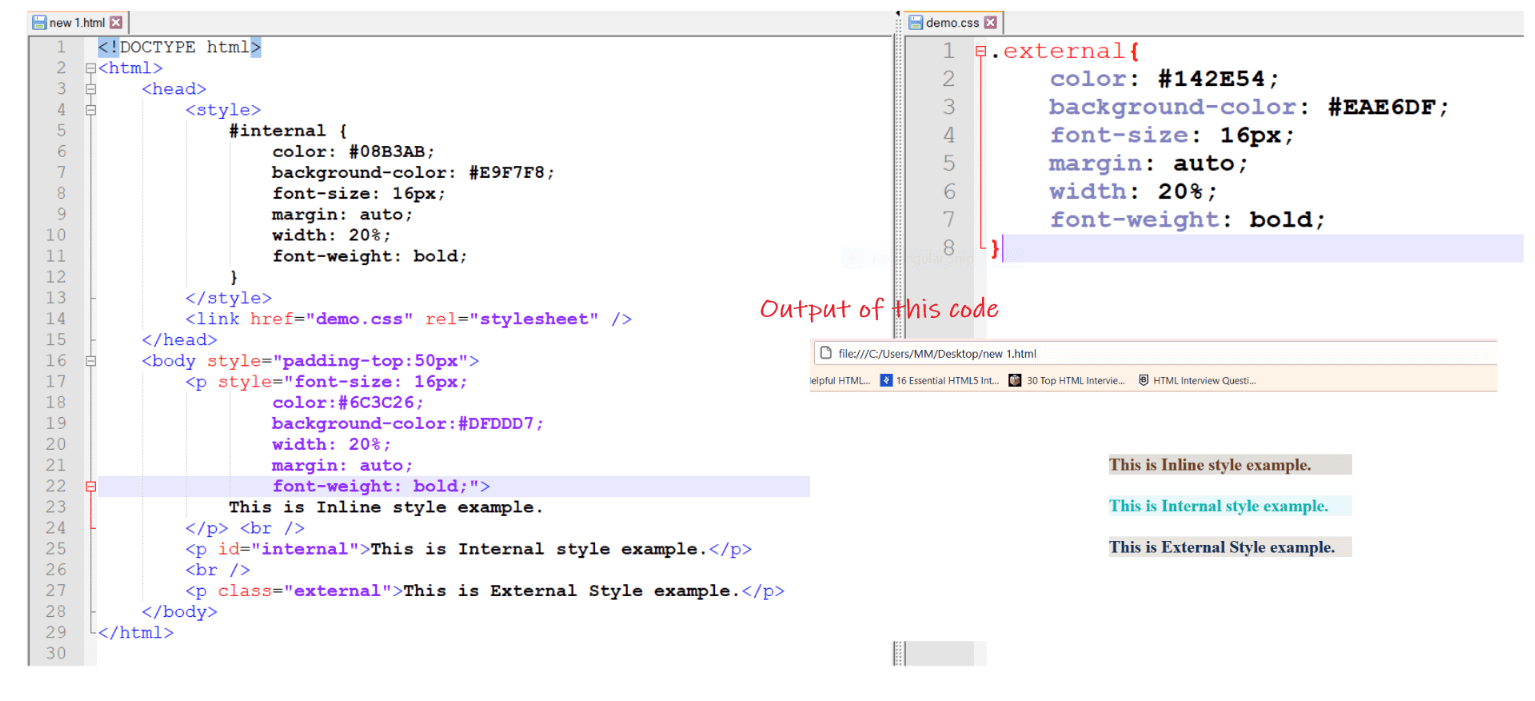
**28. What is the difference between the ‘id’ attribute and the ‘class’ attribute of HTML elements?**

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.

**29. 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 element:

1. **Inline:** Here we use the ‘style’ attribute inside the HTML element.
2. **Internal:**Here we use the <style> tag inside the <head> tag. To apply the style we bind the elements using ‘id’ or ‘class’ attributes.
3. **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.



**30. 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.

**31. 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.

<!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>**

**32. 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.

**33. 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>**

**34. 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>**

**35. 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.

**36. 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.

**37/ Difference between SVG and Canvas HTML element?**

|  |  |
| --- | --- |
| SVG | Canvas |
| SVG is 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 | Canas can only be modified using scripts. |
| SVG is highly scalable. So we can print at high quality with high resolution | It is less scalable. |

**38. What type of audio files can be played using HTML?**

HTML5 supports the following three types of audio file formats:

1. Mp3
2. WAV
3. Ogg

**Part 1 – HTML Interview Questions (Basic)**

This first part covers basic interview questions and answers.

**1. What do you mean by HTML?**

Answer:  
HTML is known as Hypertext Markup Language. This Language is used for World Wide Web. It’s a standard language that is used for creating web pages.

**2. What are the parts of the HTML page?**

**Answer:**  
Basically, there are two parts of the web pages: Content and Tags, which are responsible for the format of an HTML page.

Let us move to the next HTML Interview Questions.

**3. What do you mean by Tags?**

**Answer:**  
In HTML, page content is placed between the Tags, which are basically responsible for the formatting of the page. Tags are written between less than symbol (<) and greater than (>) symbol. For Example: <h1>text</h1>  
In the above example <h1> is the opening tag and </h1> is the closing tag.

 Popular Course in this category

**[HTML Training (12 Courses, 19+ Projects, 4 Quizzes)](https://www.educba.com/software-development/courses/html-course/?btnz=edu-blg-inline-banner1" \t "_blank)**[12 Online Courses | 19 Hands-on Projects | 89+ Hours | Verifiable Certificate of Completion | Lifetime Access | | 4 Quizzes with Solutions  
4.5 (6,276 ratings)](https://www.educba.com/software-development/courses/html-course/?btnz=edu-blg-inline-banner1" \t "_blank)

[Course Price](https://www.educba.com/software-development/courses/html-course/?btnz=edu-blg-inline-banner1" \t "_blank)**[View Course](https://www.educba.com/software-development/courses/html-course/?btnz=edu-blg-inline-banner1" \t "_blank)**

Related Courses

[Bootstrap Training (2 Courses, 6+ Projects)](https://www.educba.com/software-development/courses/bootstrap-training-course/?btnz=edu-blg-inline-banner1)[XML Training (5 Courses, 6+ Projects)](https://www.educba.com/software-development/courses/xml-training-course/?btnz=edu-blg-inline-banner1)[CSS Training (9 Courses, 9+ Projects)](https://www.educba.com/software-development/courses/css-course/?btnz=edu-blg-inline-banner1)

**4. Do all HTML tags are written in a pair?**

**Answer:**  
This is the most common HTML Interview Questions asked in an interview. No, there are some HTML tags are present which can be used as single. For Eg:  
<img>, <br>.

**5. What are the list types available in HTML?**

**Answer:**  
The common list type are available in HTML are given below:  
**– Ordered list  
– Unordered list  
– Definition list  
– Menu list  
– Directory list**

**6. Give the example for putting a comment on the HTML page?**

**Answer:**  
<!—Text for comment ->

**7. How to insert the copyright symbol in the HTML file?**

**Answer:**  
For copyright symbol, we can use type &copy; or ©

**8. How to apply Hyperlink in an HTML page?**

**Answer:**  
We can use <ahref> tag for HTML page. For eg: <ahref> Text </a>

**9.How to change the font colour on the HTML page?**

**Answer:**  
we can give this command for changing the font color  
<font color=”color”>…</font>

**10. What is the Marquee tag?**

**Answer:**  
For the scrolling text, we [use a marquee tag](https://www.educba.com/html-marquee-tag/). For Eg: <marquee>text</marquee>

Let us move to the next HTML Interview Questions.

**11. How to create an email enable text on the web page?**

**Answer:**  
For email enable text, you have to write the following command on the web page. <A HREF=”mailto:emailaddress”>text to be clicked</A>

**12. How to write the paragraph on the HTML page?**

**Answer:**  
For the paragraph tag <p> text you want to show the paragraph </p> will be used.

**13. How will you make the image as the background of the web pages?**

**Answer:**  
The command which for making the image as a background is:  
<body background = “image.gif”> where image.gif will be the path of the image.

**14. What are the frames?**

**Answer:**  
By using frames, we can make the navigation of the site easier.

**15. What is HTML5?**

**Answer:**  
HTML5 is the fifth version of [HTML language](https://www.educba.com/what-is-html/), and it is the currently [running version of HTML](https://www.educba.com/versions-of-html/).

Let us move to the next HTML Interview Questions.

**16. Which new tags are included in the HTML5?**

**Answer:**  
<video> and <audio> are the new tags which are introduced in the HTML5. They are basically used as a replacement for flash player and Silverlight to play multimedia items on the web pages.

**17. Which browser supports HTML5?**

**Answer:**  
Google Chrome, Apple Safari, Mozilla Firefox, and Opera all support most of the HTML5 features.

**18.What is the <figure> in the HTML5?**

**Answer:**  
For representing the self-contained flow content, we use a <figure> tag.

**19. What is the Canvas element in the HTML?**

**Answer:**  
For representing charts, 2D images, graphs on the web page, we use the Canvas element.

**20. What are the storage types of HTML5?**

**Answer:**  
Two storage type of HTML5 are:  
**Session Storage**– It will store the data related to the current.  
**Local Storage-** In this data will not be erased when the browser is closed

**Part 2 – HTML Interview Questions (Advanced)**

Let us now have a look at the advanced Interview Questions.

**21. What do you mean by Application Cache in HTML5?**

**Answer:**  
The Application cached means the web application is cached. So It can be accessed without an internet connection.

**22. Explain the input types of HTML5 for forms?**

**Answer:**  
Date, DateTime-local, DateTime, month, email.

**23. What are applets?**

**Answer:**  
This is the frequently asked HTML Interview Questions in an HTML Interview. Applets are a small program that can be embedded with web pages to perform some specific functionality like computation, animation.

**24. Applets program are written in which language?**

**Answer:**  
Java

**25. How can we get the geographic position of a user in HTML5?**

**Answer:**  
By using Geolocation API, we can retrieve the location of the user.

Let us move to the next HTML Interview Questions.

**26. What do you mean by HTML attribute?**

**Answer:**  
Additional information given with the elements is known as an attribute. For Eg  
<font size=”10” color=”red”>

**27. What is the extension for the HTML page?**

**Answer:**  
.htm or .html is the extension for HTML

**28. What are the different types of heading format supported by HTML?**

**Answer:**  
HTML heading is use to highlight the content of HTML document. the heading tags which are used in HTML are <h1> to <h6>.

**29.How we use JavaScript with HTML?**

**Answer:**  
By using script tag we can use JavaScript with HTML. For Eg:  
<script>  
document.getElementById(“demo”).innerHTML = “Hello JavaScript!”;  
</script>

**30.What is the Get and Post Method?**

**Answer:**  
GET s use to request the data from the server, and POST is used for submitting the data to a server.

Let us move to the next HTML Interview Questions.

**31. Which editor is used for creating the HTML pages?**

**Answer:**  
There are so many editors are available for HTML pages like Notepad, Notepad++, Sublime Text editor.

**32. What is Longdesc in HTML?**

**Answer:**  
Longdesc is an attribute that allows you to provide a link to another page that contains a description of the frame contents. For example longdesc=”framedescription.html”.

**33. How to create a table in HTML?**

**Answer:**  
By using <table> tag we create the table in HTML. For, e.g. If you want to create a table with 3 rows and 3 columns:  
<table style="width:100%">  
<tr>  
<th>Firstname</th>  
<th>Lastname</th>  
<th>Age</th>  
</tr>  
<tr>  
<td>Jill</td>  
<td>Smith</td>  
<td>50</td>  
</tr>  
<tr>  
<td>Eve</td>  
<td>Jackson</td>  
<td>94</td>  
</tr>  
<tr>  
<td>John</td>  
<td>Doe</td>  
<td>80</td>  
</tr>  
<table>

**34. What is CSS?**

**Answer:**  
CSS stands for Cascading Style Sheet. it’s a list of rules that can assign to various HTML elements. It is not case sensitive.

**35. What are the web sockets?**

**Answer:**  
Web Sockets is a next-generation bidirectional communication technology for web applications that operates over a single socket and is exposed via a JavaScript interface in HTML 5 compliant browsers.

Let us move to the next HTML Interview Questions.

**36. What is the SPAN tag is used for?**

**Answer:**  
Span is used for formatting elements in the SPAN block. It is used to select inline text.

**37. What are cell spacing and cell padding?**

**Answer:**  
Cell spacing the attribute that defines the width of the border of the HTML page, and Cell padding is the attribute that defines the space between the cell content.

**38. What are the Internationalization Attributes for XHTML Element?**

**Answer:**  
XML, DIR, LANG

**39. What do you mean by Box Model?**

**Answer:**  
Every Element on a Page is a Rectangular Box and may have Width, Height, Padding, Borders, and Margins. Every section of the box model relates to a[CSS property](https://www.educba.com/what-is-css/): width, height, padding, border, and margin.

**40. Why we use the Embed tag?**

**Answer:**  
Embed Tag is used for including a Video or an Audio within an HTML Document. The Embed Tag requires a Closing Tag. It requires the Source of the Video or the Audio file that needs to be displayed on the Page.  
Syntax: <EM**1. What are the different elements and corresponding tags used by HTML5 media content?**

Answer:  
<audio> and <video> elements are used by HTML5 media content. Please find below the tags used for the same:

 Popular Course in this category

**[HTML Training (12 Courses, 19+ Projects, 4 Quizzes)](https://www.educba.com/software-development/courses/html-course/?btnz=edu-blg-inline-banner1" \t "_blank)**[12 Online Courses | 19 Hands-on Projects | 89+ Hours | Verifiable Certificate of Completion | Lifetime Access | | 4 Quizzes with Solutions  
4.5 (6,282 ratings)](https://www.educba.com/software-development/courses/html-course/?btnz=edu-blg-inline-banner1" \t "_blank)

[Course Price  
₹6999](https://www.educba.com/software-development/courses/html-course/?btnz=edu-blg-inline-banner1" \t "_blank)~~[₹41999](https://www.educba.com/software-development/courses/html-course/?btnz=edu-blg-inline-banner1" \t "_blank)~~**[View Course](https://www.educba.com/software-development/courses/html-course/?btnz=edu-blg-inline-banner1" \t "_blank)**

Related Courses

[Bootstrap Training (2 Courses, 6+ Projects)](https://www.educba.com/software-development/courses/bootstrap-training-course/?btnz=edu-blg-inline-banner1)[XML Training (5 Courses, 6+ Projects)](https://www.educba.com/software-development/courses/xml-training-course/?btnz=edu-blg-inline-banner1)[CSS Training (9 Courses, 9+ Projects)](https://www.educba.com/software-development/courses/css-course/?btnz=edu-blg-inline-banner1)

1. **<audio>:** Used for the sound elements
2. **<video>:** Used for the movie or any video elements
3. **<embed>:** Used for the external application purpose
4. **<track>:** Used for the text tracks clarification of <audio> and <video> contents
5. **<source>:** Used for the different media resources of <audio> and <video> elements.

**2.What are the different new form element types provided by HTML5?**

Answer:  
There are mainly 10 important form elements that are newly introduced in HTML5:

1. **Date :** <input type= “date” name = “day”>
2. **Color:** <input type= “color” name= “favrtcolor”>
3. **Email:** <input type= “email” name= “email”>
4. **Datetime-local:** <input type= “datetime-local” name= “day\_time”>
5. **Time:** <input type= “time” name= “user\_time”>
6. **Range:** <input type= “range” min= “0” max= “30” step= “3” value= “5”>
7. **URL:** <input type= “url” name= “websitename”>
8. **Telephone:** <input type= “tel” name=”usertel”>
9. **Number:** <input type= “number” name= “quantity” min= “1” max= “10”>
10. **Search:** <input type= “search” name=”test”>

Let us move to the next HTML5 Interview Questions.

**3.Explain the new <canvas> element in HTML5?**

Answer:  
HTML5 provides the new <canvas> element to facilitate 2D drawing. It uses the tag <canvas>which helps in drawing graphics using Java scripts. The following code defines the canvas area for drawing purpose:  
<canvas height=””600″” id=””mycanvas”” solid=”” style=””border:1px” width=””900″”></canvas>

**4.Explain the difference between Canvas and SVG?**

Answer:  
This is the basic HTML5 Interview Questions asked in an interview. Please find below the differences between Canvas and SVG:

1. Canvas is resolution-dependent, whereas SVG is resolution-independent.
2. In SVG, an event handler can be associated with the drawing object, whereas Canvas doesn’t support event handlers associated with the drawing objects.
3. SVG is slower than Canvas as in the case of SVG; co-ordinates need to be remembered for the later manipulation purpose
4. Canvas is suitable for graphics-intensive gaming purpose, whereas SVG is not suitable for gaming.

**5. Explain the differences between sessionStorage and localStorage objects in HTML5?**

Answer:  
Storage stores data as per the different web sessions availability. Thus, if any window or tab is permanently closed, then any data or records stored through sessionStorage will be deleted. But in the case of local storage, the storing procedure is permanent, and hence all the records will remain stored on the user’s device until the user instructs the browser to delete it.

**Part 2 – HTML5 Interview Questions (Advanced)**

Let us now have a look at the advanced Interview Questions.

**6. Explain the usage of HTML5 semantic elements?**

Answer:  
The usage of the different HTML5 semantic elements are described below:

1. **<header>:** It is mainly used to store and define the starting information about a web page section.
2. **<article>:** It is used to define a set of information that can be logically independent and also can be described with respect to the concerned web page business logic.
3. **<section>:** It mainly consists of a set of instructions that defines the basic structure and content of the page
4. **<footer>:** This is used to hold a set of information that is getting displayed at the last portion of a webpage

**7. Explain HTML5 Web Storage?**

Answer:  
Using HTML5, a web page can store local data on the opened browser page. It is generally prescribed as a more secure and faster approach with respect to any web page performance evaluation process.

Let us move to the next HTML5 Interview Questions.

**8. How can you draw a straight line on a Canvas?**

Answer:  
The user can actually follow different methods to implement the process:

1. **Move To(x,y):** It will define the start point procedures while creating the line
2. **line To(x,y):** It will define the endpoint procedures while creating the line
3. **Stroke():** Now, using this method, the actual object can be drawn.

**9. How can an image be drawn on a Canvas?**

Answer:  
The following method can be used to implement the image:  
draw image(image,x,y): This will define and implement the image on Canvas.

**10. Explain a few advantages and disadvantages of HTML5?**

Answer:  
This is the advanced HTML5 Interview Questions that are asked in an interview. Please find below the list mentioning a few advantages and disadvantages of HTML5:

**Advantages:**

1. It provides greater consistency in terms of the HTML used to code a web page.
2. It provides both Audio and Video support
3. It helps developers to use fancier forms
4. It provides geolocation support
5. HTML5 is the most mobile ready tool for developing mobile sites and apps.
6. It usually requires less maintenance support.
7. It also provides more reliable storage options functionality.

**Disadvantages:**

1. It only provides the modern browser support
2. It has media licensing issues
3. It has a fragmentation problem. So, the new HTML5 feature may be properly implemented in FireFox, but again it may create problems while implementing the same in internet explorer.

**Q #1) What does HTML stand for?**

**Answer:** HTML stands for Hypertext Markup Language.

**Q #2) Describe HTML.**

**Answer:** Hypertext Markup Language or HTML is a markup language that is used to create website templates or WebPages to present the content on the World Wide Web.

HTML pages are saved by adding .html or .html in web page name.

**Q #3) Write the basic structure of the HTML template?**

**Answer:** **The basic structure of the HTML template is:**

<html>

      <head>

                <title></title>

      </head>

      <body>

      </body></html>

**Q #4) What is HTML5?**

**Answer:** HTML5 is the latest or updated version of the markup language that defines HTML.

**Q #5) Name some new features which were not present in HTML but are added to HTML5?**

**Answer:** **Some new features in HTML5 include:**

* **DOCTYPE declaration**: <!DOCTYPE html>
* **section**: Section tag defines a section in the document, such as a header, footer or in other sections of the document. It is used to define the structure of the document. **<section></section>**
* **header:**Header tag defines the head section of the document. A header section always sticks at the top of the document. **<header></header>**
* **footer:**Footer tag defines the footer section of the document. A footer section always sticks at the bottom of the document. **<footer></footer>**
* **article:**Article tag defines an independent piece of the content of a document. **<article> </article>**
* **main:**The main tag defines the main section in the document which contains the main content of the document. **<main></main>**
* **figcaption:**Figcaption tag defines the caption for the media elements such as an image or video. **<figcaption></figcaption>**

**Q #6) What is Anchor tag and how can you open an URL into a new tab when clicked?**

**Answer:** Anchor tag in HTML is used to link between two sections or two different web pages or website templates.

To open an URL into a new tab in the browser upon a click, we need to add target attribute equal to **\_blank.**

**<a href=”#” target=”\_blank”></a>**

**Q #7) Write an HTML code to form a table to show the below values in a tabular form with heading as Roll No., Student name, Subject Name, and values as**

1. **Ram, Physics**
2. **Shyam, Math**
3. **Murli, Chemistry**

**Answer:** **To represent the above values in an HTML table format, the code will be:**

<!DOCTYPE html>

<html>

<head>

<style>

table, th, td {

border: 1px solid black;

}

</style>

</head>

<body>

<table >

<tr>

<th> Roll No. </th>

<th> Student Name </th>

<th> Subject Name </th>

</tr>

<tr>

<td> 1 </td>

<td>Ram</td>

<td> Physics </td>

</tr>

<tr>

<td> 2 </td>

<td> Shyam </td>

<td> Math </td>

</tr>

<tr>

<td> 3 </td>

<td> Murli </td>

<td> Chemistry </td>

</tr>

</table>

</body>

</html>

**Output:**

**Q #8) Define Semantic elements in HTML.**

**Answer:** Semantic elements are HTML elements that represent its meaning to the browser and developer about its contents.

**For Example** – **p** tag represents a paragraph, **a** tag represents anchor tag, form tag, table tag, article tag and many more are semantic elements in HTML. Whereas, div tag, span tag, bold tag are not semantic elements.

**Q #9) Define attributes in HTML tag.**

**Answer:** The HTML tag contains a field inside their tag which is called attributes of that tag.

**For Example:**

<img src=”#”> here in this tag src is img tag attributes.

<input type=” text”> here in this tag type is input tag attributes.

**Q #10) Can we modify the attribute’s value of the HTML tag dynamically?**

**Answer:** Yes, we can modify the value of the attributes by using JavaScript.

Below is the input element whose attribute will be modified from **text**to **password,**JS code to modify the attribute value:

<input type=“text” id=“inputField”>

document.getElementById(“inputField”).attr(“type”, “password”);

**Q #11) How can we comment in HTML?**

**Answer:**Comments are used by developers to keep a track of the code functionality and also help the other developers in understanding the code functionalities easily.

The commented outlines will not be shown in the browser. To comment a line, the line should start by this **<!–**and end by this **–>.**Comments can be of one line or of multiple lines.

**For Example:**

<!-- This is one line comment ?

<!-- This is multiple         line of two or         more line -->

**Q #12) What are inline elements and block-level elements in HTML?**

**Answer:** Block elements are the blocks that take the full available width and always start from a new line. It will stretch itself to the full available width of the available container width. Block-level elements are <div>, <p>, <img>, <section> and many more.

Inline elements are the elements that will only take the width that is required to fit into the container.

**For Example,** take the flow of text on the page. When the line of the text takes the full width of the container it wraps itself into a new line and again goes in the same way.

Whereas, the inline element will take only that much space or width that it is needed for them. Inline elements are <span>, <label>, <a>, <b> and many more.

**Q #13) Can we change inline elements into block-level elements?**

**Answer:** Yes, we can change inline elements into block-level elements by adding display equal to block in its CSS tag. Writing it will change the inline elements into block elements and then inline elements will also take the full width of the container.

**display: block;**

**Q #14) What are the different browsers that support HTML5?**

**Answer:** All modern browsers support HTML5 elements except some old browsers. But fortunately, most of the browsers will take html5 elements as inline elements.

**Q #15) What are <br> tags in HTML?**

**Answer:** <br> tags are used to enter a new line into the HTML contents. These tags are generally used to separate two different lines of text between each other.

**Q #16) Explain the structure of the HTML webpage.**

**Answer:** **The common structure which all HTML pages follow are enlisted below:**

**(i) DOCTYPE** – It is a special tag in HTML which is always written at the top of the HTML document, i.e. at the start of the HTML template. DOCTYPE is used to convey to the browser about the HTML version.

<!DOCTYPE html>

**(ii) HTML –**After DOCTYPE tag, the HTML tag is written to start the template. All the code will be placed into this HTMLtag. It works as the container for the whole HTML page elements.

<html>

            <!-- Rest of the html code will come inside it -->

</html>

**(iii) HEAD –**<head> tag is the first element inside the <html> tag. It is used to provide information to the browser about the page and its contents.

Search Engine Optimization (SEO) techniques are written inside this tag. <title>, <meta> tags are written inside these tag. CSS and JS external links or internal CSS and JS are also written inside this tag.

<head>

           <meta charset="UTF-8">

       <meta name="viewport" content="width=device-width, initial-scale = 1.0">

       <title>HTML Interview Questions</title>

</head>

**(iv) BODY –**<body> tags are written after the closing tag of the <head> tag, i.e. after </head>. Whatever HTML code is written inside these tags will be shown by the browser as website content.

<body>

<h2>Top HTML Interview Questions</h2>

<p>HTML stands for Hypertext Markup Language</p>

</body>

**Together the whole body will be:**

<! DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale = 1.0">

<title>HTML Interview Questions</title>

<style type="text/css">

<!-- CSS Code will be written into these -->

h2{

color: #1855b5;

}

p{

color: #3bd256;

font-weight: 600;

}

</style>

<script type="text/javascript">

<!-- Javascript code will be written into these -->

</script>

</head>

<body>

<h2>Top HTML Interview Questions</h2>

<p>HTML stands for Hypertext Markup Language</p>

</body>

</html>

**Output:**

**Q #17) Why Meta tags are used in HTML?**

**Answer:**Meta tags in HTML are used by the developer to tell the browser about the page description, author of the template, character set, keywords and many more.

Meta tags are used for search engine optimization to tell the search engine about the page contents.

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale = 1.0">

<meta name="description" content="HTML interview questions">

<meta name="author" content="Author Name">

<meta name="copyright" content="All Rights Reserved">

**Q #18) Explain list elements in HTML.**

**Answer:** **Enlisted below are the list elements in HTML:**

* **Ordered List (<ol>) –** An Ordered List or ol tag is the list that lists the items in an ordered way, i.e. numbered or alphabetically.
* **Unordered List (<ul>) –** An Unordered List or ul tag is the list which will list the items in an unordered way, i.e. in bulleted format or in any other format.
* **Definition List (<dl>)** – A Definition List or dl tag arrange the items in the way in which they are arranged in a dictionary.

**Ordered List:**

<!DOCTYPE html>

<html>

<head>

<title>HTML Interview Questions Ordered List</title>

</head>

<style>

h1{

color: red;

}

li{

color: #0070ff;

}

</style>

<body>

<h1>Ordered List : </h1>

<ol>

<li>HTML</li>

<li>CSS</li>

<li>Bootstrap</li>

<li>JavaScript</li>

</ol>

</body>

</html>

**Output:**

**Unordered List:**

<!DOCTYPE html>

<html>

<head>

<title>HTML Interview Questions Unordered List</title>

<style>

h1{

color: red;

}

li{

color: #0070ff;

}

</style>

</head>

<body>

<h1>Unordered List : </h1>

<ul>

<li>HTML</li>

<li>CSS</li>

<li>Bootstrap</li>

<li>JavaScript</li>

</ul>

</body>

</html>

**Output:**

**Definition List:**

<!DOCTYPE html>

<html>

<head>

<title>HTML Interview Question Definition List</title>

<style>

dt{

font-weight: 600;

font-size: 25px;

color: red;

}

dd{

font-weight: 500;

font-size: 15px;

color: #0070ff;

}

</style>

</head>

<body>

<dl>

<dt><strong>HTML</strong></dt>

<dd>HTML stands for Hypertext Markup Language</dd>

<dt><bold>CSS</bold></dt>

<dd>CSS stands for Cascading Style Sheets</dd>

</dl>

</body>

</html>

**Output:**

**Q #19) Define iframe in HTML.**

**Answer:** Iframe tag is written as <iframe>.

An iframe is used to display different document content inside the different document content in a rectangular region in the browser. When different document content is embedded into a current HTML content, then it is known as an inline iframe.

**The src attribute contains the path to the document that occupies the inline iframe.**

<!DOCTYPE html>

           <html>

                       <head>

                               <title>HTML Interview Questions Iframes</title>

                      </head>

                      <body>

Iframe HTML Code

                                 <iframe src = "demo\_iframe.htm" width = "300" height = "300">

                                            Sorry your browser does not support inline frames.

                                 </iframe>

                     </body>

         </html>

**Output:**

**Q #20) Define forms in HTML.**

**Answer:** Forms in HTML are required when we want to collect the user information whenever a user fills any form or provides any details and when we want to save it into our database.

<!DOCTYPE html>

<html>

<head>

<title>HTML Interview Question Form tag</title>

<style>

form {

width: 200px;

border: 2px solid blue;

margin: 0 auto;

padding: 60px 100px;

}

p{

color: red;

font-size: 16px;

font-weight: 600;

}

input::placeholder{

color: blue;

}

button{

line-height: 20px;

text-align: center;

background: green;

border: 0;

color: #ffffff;

font-size: 14px;

padding: 15px 64px;

margin-top: 20px;

}

</style>

</head>

<body>

<form >

<p>Name:</p>

<input type = "text" name = "user\_name" placeholder = "Enter Name"/>

<br/>

<br/>

<p>Email: </p>

<input type = "email" name = "user\_email" placeholder = "Enter email"/>

<br/>

<br/>

<p>Password: </p>

<input type="password" name = "user\_pwd" placeholder = "Enter Password" />

<br/>

<button> Submit </button>

</form>

</body>

</html>

**Output:**

**Q #21) In how many ways can a heading be written in HTML?**

**Answer:** A heading can be defined as a block-level element that is used to give a heading to a particular section or topic.

**Example using 6 types of headings in HTML:**

<!DOCTYPE html>

<html>

<head>

<style>

h1{

color: red;

}

h2{

color: blue;

}

h3{

color : green;

}

h4{

color: purple;

}

h5{

color: yellow;

}

h6{

color: orange;

}

</style>

</head>

<body>

<h1>Heading 1</h1>

<h2>Heading 2</h2>

<h3>Heading 3</h3>

<h4>Heading 4</h4>

<h5>Heading 5</h5>

<h6>Heading 6</h6>

</body>

**Output:**

**Q #22) How can we create a hyperlink in HTML?**

**Answer:** An anchor tag or <a> tag in HTML is used to create hyperlinks. This creates a path between two different HTML web pages.

**Hyperlinks can be displayed in three ways:**

* **Unvisited Link** – These links are blue in color and underlined.
* **Visited Link** – These links are purple in color and underlined.
* **Active Link** – These links are red in color and underlined.

**Q #23) Why do we use the required attribute in HTML?**

**Answer:** The required attribute is used in HTML to make the field mandatory. It forces the user to fill that particular field to submit the form.

If the field is input then it will throw a default HTML error.

<input type="email" name = "user\_email" required />

**Q #24) How can we include Google maps on a website?**

**Answer: HTML code to include Google maps on our web page:**

<!DOCTYPE html>

<html>

<body>

<h1>HTML Interview Questions Google Map</h1>

<div id="map" style="width: 400px; height: 400px; background: yellow">

</div>

<script>

function myMap() {

var mapOptions = {

center: new google.maps.LatLng(51.5, -0.12),

zoom: 10

}

var map = new google.maps.Map(document.getElementById("map"), mapOptions);

}

</script>

<script src= "https://maps.googleapis.com/maps/api/js?key=AIzaSyBu-916DdpKAjTmJNIgngS6HL\_kDIKU0aU&callback=myMap"></script>

</body></html>

**Output:**

**Q #25) Differentiate between HTML and XHTML.**

**Answer:** **The differences between HTML and XHTML are:**

* HTML stands for Hypertext Markup Language, whereas XHTML stands for Extensible Markup Language.
* A static webpage is an HTML web page and dynamic web pages are XHTML.
* XHTML are more stricter than HTML.
* An XML application of HTML is defined as XHTML.
* All modern browsers support XHTML.

**Q #26) What are Web Workers?**

**Answer:** Web Workers is a code of JavaScript which runs in the background threads without disturbing the performance of the page. It is used for computing-heavy tasks like an access database or function.

**Q #27) What is the SVG element?**

**Answer:**SVG is a followed XML format; it stands for Scalable Vector Graphics which is used to create vector graphics with the support for interactivity and animation.

SVG is resolution independent as it does not lose its quality when they are resized or zoomed.

**Q #28) Explain about Canvas.**

**Answer:**Canvas is a pixel-based graphics and it is one of the new features of HTML5.

It provides a space in the document where we can draw graphics by using JavaScript and it is resolution dependent, hence the quality will be affected when it’s zoomed or resized.

**Example:**

<!DOCTYPE html>

<html>

<body>

<canvas id="myCanvas" width="300" height="200" style="border:1px solid #d3d3d3; color: #c9cc18;">

Your browser does not support the HTML5 canvas tag.</canvas>

<script>

var can = document.getElementById("myCanvas");

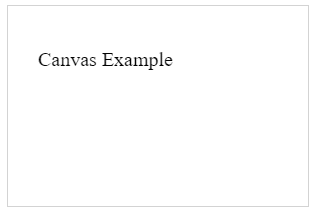
var canvas = can.getContext("2d");

canvas.font = "20px Hind-sanserif";

canvas.fillText("Canvas Example",30,60);

</script>

</body>

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2018/10/canvas.png)

**Q #29) Explain new form elements in HTML5.**

**Answer: The new form elements that were added into HTML5 are:**

* **Datalist –** It’s used as a list of options for input control.
* **Keygen –** This tag defines a key-pair generator (Private/Public) field.
* **Output –** It’s used to show the result of a calculation.

**Q #30) What is Quirks mode in HTML5?**

**Answer:**If we do not include the <!DOCTYPE> element in our HTML page or Document, it will go to Quirks Mode. In this mode, the HTML element depends on the browser. Hence the content will be displayed according to the browser.

1. What is HTML?

The odds are that you won’t be asked this directly, but it can’t hurt to remind yourself what [HTML](https://www.goskills.com/Course/Intro-HTML) stands for: Hyper Text Markup Language. This means that an HTML document, written in plain text, is used to describe the structure and content of web pages, with [links to other pages](https://www.goskills.com/Course/Intro-HTML/Lesson/1795/Creating-HTML-Links) and resources. In its most basic form, you can define blocks of content, which are displayed depending on the type of block you used.

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

Elements

Each part of a web page, such as a paragraph, an image, a link or anything else you can interact with, is an element. Each type of element has its own behavior - for example you can click on links, or type in text boxes.

Tags

An HTML document is a simple, plain text document, which you are able to open with any text editor on your computer. When you open one, you’ll see the document is made up of tags, which are keywords surrounded by angled brackets, each of which describes an HTML element. Here you can see HTML tags telling the browser how to render the text element inside:

<span>This text is surrounded by HTML tags!</span>

Most tags have opening and closing tags. The opening tag is written with the tag name in angled brackets, like <tagname> whereas the closing tag adds a forward slash: </tagname>. Anything between these opening and closing tags is considered to be contents of that tag.

Some tags, like the <img> tag are self-closing. This means that they cannot have any content. For example, an image can’t contain additional HTML elements within it. The only way to change their behavior or appearance is through attributes or CSS.

<img src="http://placekitten.com/200/300" />

3. What are attributes and how do you use them?

Each tag can also have additional attributes, which change the way the tag behaves or is displayed. For example, an <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 - they can never be in closing tags. They are followed by an equals sign = and the chosen value in double quotes ". Take care not to include any spaces before and after the equals sign! Multiple attributes can be defined on a single tag, separated by a space.

<!-- Text field -->

<input type="text" />

<!-- Checkbox -->

<input type="checkbox" />

<!-- Radio button -->

<input type="radio" value="on" />

There are some attributes that are available on every single tag - like the style attribute - however, most tags also have their own specific ones as those attributes simply wouldn’t make sense on tags of a different type (like an image source attribute on a paragraph tag).

4. When should you use comments?

Sometimes it can be useful to add code comments to your HTML document. These will not be displayed in the browser, but they can be useful to leave 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.

<!-- This is a comment! -->

<!-- Comments can

span multiple

lines too -->

<!-- All of this is ignored. Even valid HTML like this: <span>Ska—doosh!</span> -->

While comments are useful, try to keep them to a minimum. Only use them when something is not quite clear enough, or else your documents will become more ‘comment’ than code!

5. What’s the difference between a block-level element and an inline element?

Each element in HTML is displayed in one of a few ways. By default, most tags are either displayed as block-level or inline. This value can be overridden using [CSS](https://www.goskills.com/Course/Intro-CSS).

Block

As the name suggests, a block-level element is drawn as a block that stretches to fill the full width available to it (the width of its container) and will always start on a new line.

Examples of elements that are block-level by default: <div>, <img>, <section>, <form>, <nav>.

Inline

Unlike the block-level elements, 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. When a line of text gets to the end of the space available, it wraps onto the next line and happily keeps going. If you were to tack more text onto an existing line of text, it will stay on the same line, as if it was all part of the same text to begin with.

Examples of elements that are inline by default: <span>, <b>, <strong>, <a>, <input>.

6. Do you know what these tags are used for?

There are a lot of different HTML elements, which can be a little overwhelming. Luckily, they are also generally pretty intuitive! To help you prepare for a pop quiz in your next interview, we’ve organized some of the most common elements in sections below:

Text

| **Tag** | **Description** |
| --- | --- |
| **p** | Paragraph block. Has margin above and below by default. |
| **span** | Inline text, no distinguishing styling by default. Generally used to style parts of a block of text differently (e.g. underlined, different background or font) |
| **a** | Anchor or link. The href attribute defines where it takes you upon clicking it. This can be a reference point on the same page or a different page. |
| **button** | A clickable element styled differently depending on the browser and operating system (e.g. Windows, Mac, Linux) used, though these can be overridden. What happens upon click is up to you to decide! |
| **strong, b** | Renders a piece of text bold. |
| **i** | Renders a piece of text italic. |
| **h1, h2, …, h6** | These are headings of different levels. For example, you would generally have a single h1 tag, which can have multiple h2 tags inside of it. Each of those in turn can have multiple h3 tags inside them, and so on. |
| **br** | Denotes a line break. HTML ignores white space in your code when it becomes more than a single space. Therefore, to break text onto a different line, you can use this tag. Alternatively, you could put the different pieces of text in two separate block-level elements. |

Layout

| **Tag** | **Description** |
| --- | --- |
| **div** | This is your basic container element. It is a block-level element but has no additional styling by default. |
| **ul** | This stands for [unordered list](https://www.goskills.com/Course/Intro-HTML/Lesson/1800/Creating-Lists), also known as a bulleted list. Inside the ul element you can have any number of li elements. Using CSS you can define whether it displays as bullet points, empty circles or squares. |
| **ol** | Stands for ordered list. Each item inside this list will have an incremented number or symbol beside them (e.g. 1, 2, 3 or a, b, c). The symbols can be numbers, letters or roman numerals. |
| **li** | Stands for list item. These live inside ul or ol elements. Each li is a separate item in the list, denoted by bullet point, number or any other symbol chosen by you. |
| **table** | Sometimes you need to display related data in a table format. Hurray for tables! Just be sure not to use them for your page’s layout, or you might get a phone call from the 90s. |
| **tr** | Used to define table rows inside table elements. |
| **td** | Used to define table cells inside tr elements, which in turn are inside a table element. |
| **thead** | Optional (but recommended) part of your table. Use it to group a table row (tr) that contains the column titles of your table. |
| **tbody** | Like thead, this is optional. If you have a thead in your table, you should also include a tbody. It should contain all rows that are not in your thead. |
| **section** | Behaves like a div but it’s used to mark a specific section of a page. Each section can have its own h1 tag, whereas normally you should only have one h1 per page. Introduced in HTML5. |
| **nav** | Behaves like a ul but specifically for navigation items. Introduced in HTML5. |

Visual

| **Tag** | **Description** |
| --- | --- |
| **img** | Used to show images on your page. Use the src attribute to specify which file you’d like it to load. |
| **video** | Like img, this is used to display video on your page. If you don’t want to embed video from another source (like YouTube or Vimeo), then this is your best bet. Use source tags with the src attribute inside the video tag to specify which file to load, including backup options with different file types.    Just make sure your video is small in file size or you might get some angry emails. You can specify whether you want it to show video controls (like a play/pause button) and whether it autoplays. Older browsers do not support this. |
| **audio** | Similar to the above elements, but of course this only loads audio. As with the video element, this will display audio controls if you specify so. As before, use source tags with the src attribute inside the audio tag to specify which file to load, including backup options with different file types. |

Forms

| **Tag** | **Description** |
| --- | --- |
| **form** | As the name suggests, this creates a form. Every input element inside a form tag belongs to that form. The action and method attributes are used to specify what to do when submitting the form. |
| **input** | These elements are very versatile and can take on many forms using the type attribute, from text fields and radio buttons, to date fields and Submit buttons. |
| **textarea** | These are larger than simple text fields and allow the user to enter line breaks as well. The size can be adjusted. |
| **label** | This defines a label for an input element. When using the for attribute, this can be clicked on to select its associated input field. |
| **select** | Your classic dropdown. Each item inside the dropdown list is defined using the option tag, nested inside the select tag. You can of course have many option tags inside a select. |
| **option** | An individual item in a dropdown list. |

7. How is an HTML document structured?

While each page is different, there are a few rules around the overall structure of the HTML document. Let's look at the structure below:

DOCTYPE tag

DOCTYPE is a special tag and must be the first one in an HTML document. It tells the browser what version of HTML you want to use. You can read more about this and the latest version (HTML5) in the next section. For now, you’ll likely want to use the following:

<!DOCTYPE html>

HTML tag

Next, we have the <html> tag. Each document only has one <html> tag, and it serves as a ‘container’ for the rest of the page’s elements. You will place the rest of your html code for the page within the <html> tag.

<html>

<!-- The rest of your code -->

</html>

Head tag

The first element inside the <html> tag is the <head> tag. Content inside this tag is only meant for the browser and is not visible on the page directly. Among other things, it contains the title of your page (as shown in your browser tab), the character set used to display content and more metadata (meaning data about data). This is also where you define CSS and load some JavaScript (more about that later).

<head>

<title>The title of your page</title>

<meta charset="UTF—8">

<meta name="description" content="A description of your page">

</head>

Body tag

After the <head> tag, still inside the <html> tag, we have the <body> tag. It contains all the content that’s displayed in the browser.

<body>

<h1>Welcome to my website!</h1>

<p>This is where I'll put some content :)</p>

</body>

All together now

With all of the above rules combined, we get something like this:

<!DOCTYPE html>

<html>

<head>

<title>The title of your page</title>

<meta charset="UTF-8">

<meta name="description" content="A description of your page">

</head>

<body>

<h1>Welcome to my website!</h1>

<p>This is where I'll put some content :)</p>

</body>

</html>

8. What do you know about HTML5?

Throughout the history of the internet, there have been many versions of the HTML standard. The versions varied in features and how strict they were. In 2014, the official recommendation for HTML5 by the [World Wide Web Consortium](https://www.w3.org/) was released. It is a living standard, meaning new features can be added over time.

9. What new features were added to HTML5?

It introduced a number of semantic elements, which is to say elements that convey meaning. Some of the new semantic elements are <header>, <footer>, <section>, and <article> . They are semantic in that they are not just simple containers, but they tell the browser more about their contents.

There are additional form element types, like "number", "date", "calendar" and "range". Video and audio elements have also been added, as well as new graphic elements, such as <svg> and <canvas>.

10. Are there changes to DOCTYPE for HTML5?

The DOCTYPE for HTML5 is simple. Once you’ve added this to your page, the browser will interpret everything else as HTML5.

<!DOCTYPE html>

11. What browsers support HTML5?

All modern browsers support HTML5, however some older browsers do not. Luckily, most browsers will simply handle the new elements as inline elements. You can then use CSS to change certain elements to be displayed as block-level elements where needed.

12. How do you apply CSS styles to a web page?

When you’ve got a good grasp of HTML, your interviewer may ask you some questions on how HTML works with CSS and [JavaScript](https://www.goskills.com/Course/Intro-JavaScript), to tie everything together. If you are planning on working as a [front end developer](https://www.goskills.com/Development), employers prefer some proficiency in all three languages.

CSS (Cascading Style Sheets) allows you to change the look of elements on the page, transforming it from a simple text document to a fully fledged website. We won’t go into too much detail on CSS here, but you can read our article on [CSS interview questions](https://www.goskills.com/Development/Articles/CSS-interview-questions-answers) for more.

There are three main ways to apply CSS styles to a webpage:

Inline styles

You can add a style attribute to almost any tag. Inside this attribute you can write your CSS rules.

<div style="background-color: red;">A container with a red background.</div>

A style block

You are able to define one or more style blocks inside the head section of your HTML document. Inside these blocks you can write your CSS rules. You will have to specify which elements on the page you’d like to style. In the below example, we’re targeting the <body> tag and an element with a class attribute equal to .button.

<head>

<style>

body {

font-size: 16px;

}

.button {

padding: 10px;

}

</style>

</head>

Link to a CSS file

By far the most recommended option is to link to a CSS file. This way you are able to keep the content (HTML) separate from the way you present that content (CSS). It also means you can use the same styles on multiple pages. To link to a CSS file, you will have to add a <link> tag to the <head> section in your document with an href attribute that specifies the location of the CSS file.

<head>

<link rel="stylesheet" href="styles.css">

</head>

13. How do you apply JavaScript to a web page?

In order to add interactivity to your page, other than what’s already provided through HTML, you will 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. The following example shows an alert with a message when the user clicks on it.

<button onclick="alert('You clicked on me!');">Click me!</button>

A 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. Note that 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

Again, as with CSS, this is the preferred way of including JavaScript. It allows you to keep the content of the page separate to how users interact with that content, and 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>, but keep in mind it will be loaded in the order you’ve structured your document.

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

### Q1. What is HTML?

[HTML](https://www.edureka.co/blog/what-is-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:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <!-- 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**:

|  |  |
| --- | --- |
| 1  2  3 | <!-- 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:

|  |  |
| --- | --- |
| 1 | <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:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <!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:**

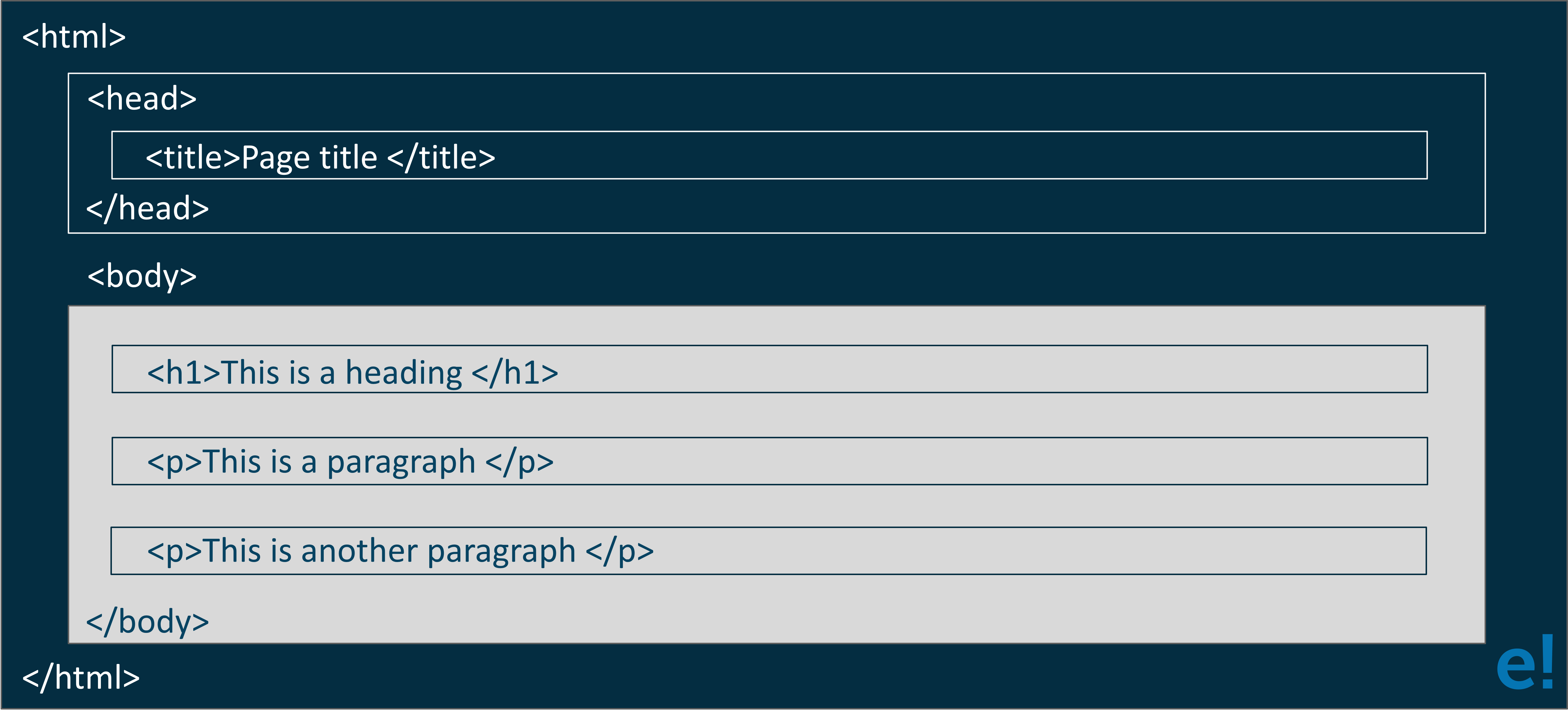
|  |  |
| --- | --- |
| 1 | <a href = "..........."> Link Text </a> |

**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:

****

* **<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:**

|  |  |
| --- | --- |
| 1  2  3 | <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.

[[](https://www.edureka.co/full-stack-developer-training)](https://www.edureka.co/full-stack-developer-training" \t "_blank)

**[Full Stack Web Developer Masters Program](https://www.edureka.co/full-stack-developer-training" \t "_blank)**

[Explore Curriculum](https://www.edureka.co/full-stack-developer-training" \t "_blank)

**Q19. How to create a new HTML element?**

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

|  |  |
| --- | --- |
| 1  2  3 | <script>  document.createElement﴾"myElement"﴿  </script> |

It can be also be used in the HTML as:

|  |  |
| --- | --- |
| 1 | <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:**

|  |  |
| --- | --- |
| 1 | <iframe src="URL"></iframe> |

**Example:**

|  |  |
| --- | --- |
| 1 | <iframe src="demo\_iframe.html" width="200px" height="200px"></iframe> |

**Target to a link:**

|  |  |
| --- | --- |
| 1 | <iframe src="[http://www.edureka.co](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.

|  |  |
| --- | --- |
| 1 | <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:**

|  |  |
| --- | --- |
| 1  2  3 | <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:

|  |  |
| --- | --- |
| 1 | <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.

**Programming & Frameworks Training**

**[FULL STACK WEB DEVELOPMENT INTERNSHIP PROGRAM](https://www.edureka.co/full-stack-web-development" \o "Full Stack Web Development Internship Program" \t "_blank)**

**[Full Stack Web Development Internship Program](https://www.edureka.co/full-stack-web-development" \o "Full Stack Web Development Internship Program" \t "_blank)**

*[Reviews](https://www.edureka.co/full-stack-web-development" \o "Full Stack Web Development Internship Program" \t "_blank)*

**[5](https://www.edureka.co/full-stack-web-development" \o "Full Stack Web Development Internship Program" \t "_blank)**[(2005)](https://www.edureka.co/full-stack-web-development" \o "Full Stack Web Development Internship Program" \t "_blank)

**[JAVA CERTIFICATION TRAINING COURSE](https://www.edureka.co/java-j2ee-training-course" \o "Java Certification Training Course" \t "_blank)**

**[Java Certification Training Course](https://www.edureka.co/java-j2ee-training-course" \o "Java Certification Training Course" \t "_blank)**

*[Reviews](https://www.edureka.co/java-j2ee-training-course" \o "Java Certification Training Course" \t "_blank)*

**[4](https://www.edureka.co/java-j2ee-training-course" \o "Java Certification Training Course" \t "_blank)**[(47003)](https://www.edureka.co/java-j2ee-training-course" \o "Java Certification Training Course" \t "_blank)

**[PYTHON SCRIPTING CERTIFICATION TRAINING](https://www.edureka.co/python-scripting" \o "Python Scripting Certification Training" \t "_blank)**

**[Python Scripting Certification Training](https://www.edureka.co/python-scripting" \o "Python Scripting Certification Training" \t "_blank)**

*[Reviews](https://www.edureka.co/python-scripting" \o "Python Scripting Certification Training" \t "_blank)*

**[5](https://www.edureka.co/python-scripting" \o "Python Scripting Certification Training" \t "_blank)**[(10053)](https://www.edureka.co/python-scripting" \o "Python Scripting Certification Training" \t "_blank)

**[PYTHON DJANGO TRAINING AND CERTIFICATION](https://www.edureka.co/python-django" \o "Python Django Training and Certification" \t "_blank)**

**[Python Django Training and Certification](https://www.edureka.co/python-django" \o "Python Django Training and Certification" \t "_blank)**

*[Reviews](https://www.edureka.co/python-django" \o "Python Django Training and Certification" \t "_blank)*

**[5](https://www.edureka.co/python-django" \o "Python Django Training and Certification" \t "_blank)**[(5635)](https://www.edureka.co/python-django" \o "Python Django Training and Certification" \t "_blank)

**[NODE.JS CERTIFICATION TRAINING](https://www.edureka.co/nodejs-certification-training" \o "Node.js Certification Training" \t "_blank)**

**[Node.js Certification Training](https://www.edureka.co/nodejs-certification-training" \o "Node.js Certification Training" \t "_blank)**

*[Reviews](https://www.edureka.co/nodejs-certification-training" \o "Node.js Certification Training" \t "_blank)*

**[5](https://www.edureka.co/nodejs-certification-training" \o "Node.js Certification Training" \t "_blank)**[(6526)](https://www.edureka.co/nodejs-certification-training" \o "Node.js Certification Training" \t "_blank)

**[SPRING FRAMEWORK CERTIFICATION TRAINING](https://www.edureka.co/spring-certification-course" \o "Spring Framework Certification Training " \t "_blank)**

**[Spring Framework Certification Training](https://www.edureka.co/spring-certification-course" \o "Spring Framework Certification Training " \t "_blank)**

*[Reviews](https://www.edureka.co/spring-certification-course" \o "Spring Framework Certification Training " \t "_blank)*

**[5](https://www.edureka.co/spring-certification-course" \o "Spring Framework Certification Training " \t "_blank)**[(8645)](https://www.edureka.co/spring-certification-course" \o "Spring Framework Certification Training " \t "_blank)

**[PHP & MYSQL WITH MVC FRAMEWORKS CERTIFICATION TRAINING](https://www.edureka.co/php-mysql-self-paced" \o "PHP & MySQL with MVC Frameworks Certification Training" \t "_blank)**

**[PHP & MySQL with MVC Frameworks Certification Training](https://www.edureka.co/php-mysql-self-paced" \o "PHP & MySQL with MVC Frameworks Certification Training" \t "_blank)**

*[Reviews](https://www.edureka.co/php-mysql-self-paced" \o "PHP & MySQL with MVC Frameworks Certification Training" \t "_blank)*

**[5](https://www.edureka.co/php-mysql-self-paced" \o "PHP & MySQL with MVC Frameworks Certification Training" \t "_blank)**[(3583)](https://www.edureka.co/php-mysql-self-paced" \o "PHP & MySQL with MVC Frameworks Certification Training" \t "_blank)

**[MASTERING PERL SCRIPTING CERTIFICATION TRAINING](https://www.edureka.co/mastering-perl-scripting-self-paced" \o "Mastering Perl Scripting Certification Training" \t "_blank)**

**[Mastering Perl Scripting Certification Training](https://www.edureka.co/mastering-perl-scripting-self-paced" \o "Mastering Perl Scripting Certification Training" \t "_blank)**

*[Reviews](https://www.edureka.co/mastering-perl-scripting-self-paced" \o "Mastering Perl Scripting Certification Training" \t "_blank)*

**[5](https://www.edureka.co/mastering-perl-scripting-self-paced" \o "Mastering Perl Scripting Certification Training" \t "_blank)**[(4358)](https://www.edureka.co/mastering-perl-scripting-self-paced" \o "Mastering Perl Scripting Certification Training" \t "_blank)

**[ADVANCED JAVA CERTIFICATION TRAINING](https://www.edureka.co/advanced-java-sp" \o "Advanced Java Certification Training" \t "_blank)**

**[Advanced Java Certification Training](https://www.edureka.co/advanced-java-sp" \o "Advanced Java Certification Training" \t "_blank)**

*[Reviews](https://www.edureka.co/advanced-java-sp" \o "Advanced Java Certification Training" \t "_blank)*

**[5](https://www.edureka.co/advanced-java-sp" \o "Advanced Java Certification Training" \t "_blank)**[(3843)](https://www.edureka.co/advanced-java-sp" \o "Advanced Java Certification Training" \t "_blank)

Next

**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:

|  |  |
| --- | --- |
| 1 | <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:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <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.

|  |  |
| --- | --- |
| 1  2  3 | <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:

|  |  |
| --- | --- |
| 1 | <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:**

|  |  |
| --- | --- |
| 1  2  3 | <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.

|  |  |
| --- | --- |
| 1  2  3 | <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:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <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:

[[](https://www.edureka.co/full-stack-developer-training)](https://www.edureka.co/full-stack-developer-training" \t "_blank)

**[Full Stack Web Developer Masters Program](https://www.edureka.co/full-stack-developer-training" \t "_blank)**

[Weekday / Weekend BatchesSee Batch Details](https://www.edureka.co/full-stack-developer-training" \t "_blank)

|  |  |
| --- | --- |
| 1 | <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.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <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>.**

|  |  |
| --- | --- |
| 1 | <script src="my-code.js"></script> |

**Question 1: What is HTML?**

Now you might be thinking - why on Earth would I be asked such a thing? And indeed, it does seem strange that you would be asked this question in a… Web developer’s job interview, doesn’t it?

Well, it’s not strange at all. By asking this type of question, your potential employers simply want to test your skills in defining things in your own, personal words. They want to see how you think and how you approach such tasks.

To put it in a very simple manner, [**HTML**](https://www.bitdegree.org/learn/html-basics) is *the language of the internet*. Abbreviation for “*Hypertext Markup Language*”, it is the language used to create websites, web apps, and so on.

Did you know?

**Have you ever wondered which online learning platforms are the best for your career?**

[**See & compare TOP3 online learning platforms side by side**](https://www.bitdegree.org/online-learning-platforms/coursera-vs-udacity)

**Question 2: What is a tag?**

These types of definition-requiring HTML interview questions are quite common in the beginning, but don't try to memorize: instead, explain in your own words.

**Tags** are *symbols between which the content is placed*. They are used to shape and format the content so that it would make sense when released to the live version of the webpage.

**Question 3: What’s the difference between ‘physical tags’ and ‘logical tags’?**

Comparison-type HTML interview questions are also very common in web development job interviews, so take notes.

The content that is placed in between **physical** tags will be formatted and displayed according to those tags (to their parameters) by adding style. **Logical** tags set the meaning and importance of the text that they encompass.

Example:

<p style="color: blue";><it>These are physical tags, describing the appearance.</it></p>  
<em>These are logical tags, describing that is an emphasized paragraph.</em>

**Question 4: Should you close all of your tags?**

Although at first, it might come off as obvious, this is one of the more sneaky HTML interview questions.

No, not all of the tags should be closed: not all HTML tags require a closing tag.

Example:

<img src="https://cdn.bitdegree.org/learn/pom-laptop.png?raw=true">

**Question 5: What’s an ‘attribute’?**

An **attribute** is simply *an additional function that a tag has*. This function sets a certain way in which the tag functions in the live version of a webpage.

These attributes are assigned to the tags in the same area in which the name of the tag is displayed.

Example: adding width to an image:

<img src="https://cdn.bitdegree.org/learn/pom-laptop.png?raw=true" width="100">

**Question 6: Name a few semantic tags that were introduced with HTML5.**

As you’ve probably know, HTML5 is the newest and current version of HTML, so it's no wonder you will have to talk about HTML5 in your HTML interview questions.

There were many different semantic tags introduced with the new version of HTML, but just to name a few examples: **header**, **footer**, **main**, **article**.

If you want to know about all of the new elements, read our [**HTML5 semantic tags guide**](https://www.bitdegree.org/learn/html5-semantic-tags).

**Question 7: Why are some characters often displayed as boxes?**

This is quite an often occurrence - this happens because some certain characters might not be supported by the web browser that your operating system is running.

**Question 8: How many types of headings can HTML document support?**

HTML supports **six** different types of headings. These headings differ in size, with Heading 1 being the largest (usually reserved for the title) and thus descending in size with each heading after that. You can read more about the headings [**here**](https://www.bitdegree.org/learn/h-element).

[**Example**](https://www.bitdegree.org/learn/best-code-editor/h-element-example-1):  
<h1>Heading text - Level 1. Most Important Heading</h1>  
<h2>Heading text - Level 2</h2>  
<h3>Heading text - Level 3</h3>  
<h4>Heading text - Level 4</h4>  
<h5>Heading text - Level 5</h5>  
<h6>Heading text - Level 6. Least Important Heading</h6>

**Question 9: What’s the difference between ‘block’ and ‘inline’ elements?**

**Block** elements are programmed so that they would take up as much space as possible. As opposed to that, **inline** elements are designed to take up a very minimal amount of space.

You can read more about these elements and their differences [**here**](https://www.bitdegree.org/learn/inline-elements).

**Question 10: What’s XHTML?**

This might either be one of the fifth version HTML interview questions, or it could be completely unrelated to the topic. This is because it deals with the different versions of HTML.

[**XHTML**](https://www.bitdegree.org/learn/what-is-xhtml) is *used to extend the most popular versions of HTML*.

**Question 11: How would you group form elements?**

If you want to collect and transfer certain form HTML elements into a single group, the **fieldset** tag is considered to be the best one to use.

You can read more about this tag [**here**](https://www.bitdegree.org/learn/html-fieldset).

**Question 12: What is an ‘image map’ used for?**

An **image map** is *a tool that lets you place many different links on one, specific image*. This is great because it lets you create awesome banners and website cover images while providing the ability to link anything you’d like.

[**Example**](https://www.bitdegree.org/learn/best-code-editor/image-map-html-example-1):  
<map name="creaturemap">  
<areashape="rect"coords="34, 44, 270, 350"alt="Doggo"href="https://www.bitdegree.org">  
<areashape="rect"coords="290, 172, 333, 250"alt="Gaming"href="http://www.bitdegree.org">  
<areashape="circle"coords="337, 300, 44"alt="Level up"href="http://www.bitdegree.org">  
</map>

**Question 13: Should you collapse white spaces?**

This might come off as one of the trick HTML interview questions, simply because of the wording. But don’t get fooled - collapsing white spaces is considered to be quite beneficial.

In HTML, white spaces are considered to be characters, too. That means that they take up space. If you collapse them, you might be able to improve the readability of your project.

**Question 14: What are ‘anchor’ tags?**

**Anchor tags** are used to create **hyperlinks**. These links are created on an already existing piece of content (text). In total, there are **three** types of anchor tags - **active**, **visited** and **unvisited**.

You can see an example of such links [**here**](https://www.bitdegree.org/learn/best-code-editor/html-link-example-11).

**Question 15: Do all browsers support HTML5?**

Well, yes and no, depending on what **\*all\*** means. Most of the current browsers do support HTML5 without a problem. However, this might not be the case with older versions of these browsers.

**Advanced HTML Questions**

Once you’re finished with the basics, your interviewers will probably want to transition to the more advanced questions. That’s why now we’ll cover some notable HTML interview questions and answers for experienced developers. These will determine how well you know HTML, and just how much preparation you’ve put into the web developer’s job interview.

**Question 1: What are ‘semantic elements’?**

While simple elements (tags) aim to define what a webpage should look like, **semantic elements** further this with actually bringing meaning into the webpage. Examples of **semantic elements**: <form> , <table> , and <article>. As you can see they clearly show what kind of content is going to be contained within.

**Question 2: How is data stored in HTML5?**

There are two ways to store data in HTML5 - with [**local storage**](https://www.bitdegree.org/learn/html5-local-storage) and with **session** storage.

During the HTML interview questions knowing all of its' versions features is vital. Data that is stored in the **local** **storage** is safe and will not be eliminated after the developer decides to exit the browser. In **session** **storage**, once you exit the browser, the data is automatically deleted.

**Question 3: What happens if two sets of tags overlap each other?**

If you accidentally overlap two or more sets of tags, you’ll probably notice it fast in the live version of the website.

If different tags overlap, only the first tags are activated in the live version of the website. The way that you can check this is by going to the actual webpage and checking your content for possible errors and inconsistencies.

**Question 4: How can you input JavaScript into your website?**

Currently, there are **two** main ways of how you could insert JavaScript into your web page - **inline**, by **inserting a script block** and by **linking to a JavaScript file**.

You can learn more about it [**here**](https://www.bitdegree.org/learn/html-javascript).

**Question 6: What is an ‘application cache’?**

An **application cache** is *a function that allows you to run your project (website) in an offline mode*. This is great for testing since it can load your resources in a much faster manner.

**Question 7: What is a ‘marquee’?**

A **marquee** is *a function that allows you to add a scrollable text into your webpage*. The way that you would do this is by inserting that text into the "**marquee**" tags.

**Question 8: What are APIs?**

This is one of the more often asked fifth version HTML interview questions. An **API** is an *Application Programming Interface*. These interfaces are used to create web applications. APIs use pre-existing tools and components, so that allows the developers to integrate them into their websites.

**Question 9: What’s the difference between normal links and active links?**

**Normal** links are those which appear in blue (if there were no additional styling added). These links may become **active** once a mouse cursor is placed unto them.

**Question 10: What’s WebSQL?**

**WebSQL** is *a database that stores certain specific information about the people that come to visit/register on your webpage*. The database stores their search preferences, certain actions and so on. An important point here is that WebSQL **does** **not** store any passwords, credit card information, etc. Remember to mention this if you get such interview questions on HTML.

**Question 11: What is an ‘entity’?**

If you get asked this or similar HTML interview questions, keep in mind that your employer is probably talking about special characters that HTML does not support. **Entities** are like placeholders - they fill a certain area in the file where another type of character should be. However, since the web browser does not support that character, you have to rely on an entity.

**Question 12: What is ‘cite’?**

The “**cite**” tag is used to - *you’ve guessed it* - mark a certain area of a text that has been taken from somewhere else. It’s an inline tag that simply shows cited text.

Example:  
<cite>Sophie's Choice</cite> by William Styron

**Question 13: Does HTML5 support videos?**

Yes, it supports videos in three separate formats - **MP4**, **OGG** and **WebM**.

**Question 14: What is the default size for a text field?**

This might also come off as one of the trick HTML interview questions, for not many people think about it and might be surprised by this question.

The answer is pretty simple, though - the maximum amount of character that can be present in an unaltered text field is **13**.

**Q-1. What are the new features introduced in HTML5?**

**Answer.**

HTML5 introduces a number of new elements and attributes that help in building an attractive webSite, that we see nowadays.

It supports following new features.

* **New Semantic Elements –**These are like <header>, <footer>, and <section>.
* **Forms 2.0 –**It contains improvements to HTML web forms. It has introduced new attributes for the <input> tag.
* **Persistent Local Storage –**With HTML5, it is possible to achieve this, without resorting to third-party plugins.
* **WebSocket –**It facilitates setting up a bidirectional communication for web applications.
* **Server-Sent Events(SSE) –**These events got introduced in HTML5. The direction of the flow of the execution of these events is from the server to the Web Browser.
* **Canvas –**It supports a two-dimensional drawing surface that is programmable using JavaScript.
* **Audio & Video –**It allows embedding audio or video on the web pages without resorting to third-party plugins.
* **Geolocation –**It facilitates the visitors to share their physical location with the web application.
* **Microdata –**It allows building our personal vocabulary beyond HTML5 and extends our web pages with those custom semantics.
* **Drag and drop –** It supports to Drag and drop the items from one location to another location on the same Web page.

**Q-2. What would happen if the HTML Document does not contain <!DOCTYPE>?**

**Answer.**

It instructs the Web Browser about the version of HTML used for creating the Web page.  
If the developer misses declaring the DOCTYPE information in the code, then new features and tags provided by HTML5, like <article>, <footer>, and <header> will not be supported. Additionally, the Browser may automatically go into Quirks or Strict Mode.

**Q-3. What is a tag in HTML?**

**Answer.**

A tag instructs the Browser about how to format the HTML properly. When you write an HTML page, you enter tags for many reasons to change the appearance of text, to show a graphic, or to make a link to another page. HTML uses symbols like “<” and “>” to enclose the tags. And symbol “\” for closing the tag.

**Q-4. What is the difference between HTML elements and tags?**

**Answer.**

HTML elements communicate to the Browser how to represent the text. They become HTML tags when enclosed within angular brackets <>.

**Q-5. What are the various elements provided by HTML5 for media content?**

**Answer.**

HTML provides the support of following elements for representing the media content.

* **<audio> –**It defines the sound content.
* **<video> –**It represents the video content that needs to be attached to a Web page.
* **<source> –**This tag defines the source of video and audio.
* **<embed > –** It provides a container for an external application.
* **<track> –**It defines text tracks for video and audio.

**Q-6. What are the new Form elements made available in HTML5?**

**Answer.**

When we want to collect some data from the person visiting our site, we use HTML Forms. An example is, the user has to enter his name, email id when he registers for the first time.

A form takes input from the site visitor and then posts it to a back-end application such as CGI, ASP Script or PHP script. The back-end application will then perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available in earlier version of HTML like, text fields, text area fields, drop-down menus, radio buttons, checkboxes, etc.

HTML5 provides the support of some new Form elements that are as follows.

* **<datalist> –**It represents a list of pre-defined options for input controls.
* **<keygen> –**It defines a key-pair generator field (for forms).
* **<output> –**It represents the result of the calculation.

**Q-7. What is a semantic element in HTML5? Also, explain the difference between semantic and non-semantic elements?**

**Answer.**

Semantic elements are one of the new features that are part of HTML5. They intend to help the developers in quickly creating the page structure.

A semantic element has its meaning expressed to both the browser and the developer. Also, all the modern browsers support this feature. However, it is possible for a developer to train old browsers to handle unknown elements.

Here is the list of some of the commonly used HTML5 Semantic Elements.

* **<article>**
* **<figcaption>**
* **<figure>**
* **<header>**
* **<footer>**
* **<nav>**
* **<section>**
* **<article>**
* **<aside>**
* **<summary>**

**The difference – Semantic Vs. Non-semantic.**

**Semantic –** These elements clearly describe their content like <img>, <form>, <table> etc.

**Non-semantic –** These elements are without any definition. They don’t describe anything about their structure such as <span> and <div>.

**Q-8. What are the various tags provided for better structuring in HTML5?**

**Answer.**

The various tags provided for better structuring in HTML 5 are:

* **<article> –** This tag defines an article.
* **<aside> –**It defines content other than the page content.
* **<bdi> –**This tag isolates a part of the text for formatting in a different direction, from another text present there.
* **<command> –** It defines a command button to be invoked by the user.
* **<details> –**It outlines the additional details that a user can hide or view as per choice.
* **<dialog> –**It defines a dialog box.
* **<figure> –**This tag specifies content like illustrations, diagrams, photos, code listings, etc.
* **<figcaption> –**It provides a caption for a <figure> element.
* **<footer> –**This tag defines a footer for a document or a section.
* **<header> –**This tag describes a header for a document or a section.
* **<hgroup> –**When there are multiple levels in a heading, it groups a set of <h1> to <h6> elements.

**Q-9. How can we get the geographical position of a user in HTML5?**

**Answer.**

Yes, HTML5 can retrieve the location of a user with the use of Geolocation API. It provides getCurrentPosition() method to get the user’s current position.

**Q-10. Describe Form Input Types in HTML5?**

**Answer.**

HTML5 is the new standard for HTML that provides 13 new input types for forms. Using these new input types, we can create more interactive and easy-to-use web forms. It also provides better data validation, input control, color picker controls and many others.

The new input types for forms provided by HTML5 are as follows.

* **color –**It’s applicable for HTML elements that represent color.
* **date –**It allows the user to select a date.
* **DateTime –**It enables the user to select a date and time (with time zone).
* **DateTime-local –**It allows the user to select a date and time (without time zone).
* **email –**It is applicable for input fields that contain an e-mail address.
* **month –**It permits the user to select a month and year.
* **number –**It is applicable for input fields that accept a numeric value. It allows setting restriction on the data type of the number, this field takes.
* **range –**It is applicable for input fields that accept a value from a range of numbers. It allows setting restriction on the data type of the number, this field takes.
* **search –**It gets used for search fields.
* **tel –** It defines a field for entering a telephone number.
* **time –**It allows the user to select a time.
* **URL –**It is applicable for the input fields that contain a URL address.
* **week –**It allows the user to select a week and a year.

**Q-10. How can we retrieve the geographical position of a user using HTML5?**

**Answer.**

HTML5 provides the support of Geolocation API to retrieve the location of a user.  
We can find out the current position of the user using getCurrentPosition() method of this API.

**Q-11. What is HTML5 Graphics?**

**Answer.**

In HTML5, there are two types of graphics.

**Scalable Vector Graphics (SVG).**

* The HTML5 <svg> element is a container for SVG graphics. It provides several methods for drawing boxes, paths, text, circles, and graphic images.
* SVG is beneficial as nowadays, people are using high-resolution devices (iPads and Monitors), so it becomes impressive as designs, logos, and charts scale according to the requirement, maintaining the picture quality.
* SVG is XML based, which means that every element is available within the SVG DOM. It treats every shape as an object. If the user changes the attributes of any SVG object, the browser will automatically re-render the shape.

**Canvas.**

* It is a rectangular area on the HTML page for drawing graphics on the fly, using JavaScript.
* The default size of the canvas is 300 PX × 150 PX (width × height).
* The HTML tag <canvas> is a container for the Canvas graphics. Canvas gets executed on the pixel by pixel basis.
* In Canvas, browser forgets the graphic, after drawing it. If the user tries to modify it, the entire scene needs to be redrawn, including all the objects present in the graphic.

**Q-12. Explain the key differences between SVG and Canvas?**

**Answer.**

* Canvas is resolution dependent while SVG is not.
* Canvas does not provide any support for event handlers while SVG does provide the support for event handlers.
* Canvas is suitable for graphic-intensive games while SVG is not suitable for gaming.
* Canvas is suitable for small rendering areas while SVG is suitable for large rendering areas like Google maps.
* Canvas provides a less interactive animated user interface. Whereas, the interface provided by SVG is very highly interactive.

**Q-13. How can we create a new HTML element?**

**Answer.**

We can even create new elements for the document as follows.

<script>

document.createElement﴾"myElement"﴿

</script>

It can be used in the HTML as.

<myElement>hello</myElement>

**Q-14. What is the use of Drag and Drop in HTML5?**

**Answer.**

Drag and Drop (DnD) is a powerful User Interface concept, which enables the user to copy, reorder and delete any number of items, just with the help of mouse click. To do this, the user has to click and hold the mouse button over an element, drag to the required location, and finally release the button to drop the element there.

With HTML4, developers have to either do complex Javascript programming or use other Javascript frameworks like jQuery to achieve this functionality.

HTML 5 introduced a Drag and Drop (DnD) API that provides the support of all the fundamental DnD operations to the browser. Thus making it easy for the developers to code it.

Latest versions of all the main browsers, like Chrome, Firefox, and Safari, support this DnD API of HTML5.

**Drag and Drop Events.**

Many events get triggered during various stages of the drag and drop operation. These events are listed below.

* **dragstart –**it gets triggered when the user starts dragging a draggable object.
* **dragenter –**it gets triggered when the user has dragged the draggable element over the target element.
* **drag –**it gets fired when the object is getting dragged.
* **dragend –**it gets fired when the user releases the mouse button after dragging an object.
* **dragleave –**This event gets triggered when the mouse leaves an element while a drag is occurring.
* **dragover –**This event gets fired when the mouse moves over an element while a drag is occurring.

**Q-15. What is HTML5 Web Storage?**

**Answer.**

HTML5 Web Storage, also known as DOM Storage is a way to preserve state on either the client or server which makes it much easier to work against the stateless nature of HTTP. It allows the web pages to store the data locally in the user’s browser.

Following are the advantages of HTML5 Web Storage.

* It can store 5 to 10 MB data. That is far more than what cookies allow.
* HTML5 never transfers Web storage data with any HTTP request. Thus creating less overhead than cookies and in turn, increase the performance of the application.
* Apps can work both online and offline.
* This API is easy to learn and use.

**Q-16. What are the different types of Web Storage provided by HTML5?**

**Answer.**

There are two types of Web Storage.

**1. Session Storage**

As its name indicates, it stores data of current session only. The data stored in Session Storage clears on closing the browser.

Following methods are available to access Session Storage.

* Use setItem() function to store data in Session Storage

**sessionStorage.setItem (‘key’,’value’);**

**For Example.**

sessionStorage.setItem (‘username’,’Meenakshi’)

* Use getItem() function to retrieve data from Session Storage

**sessionStorage.getItem(‘key’);**

**For Example.**

var username= sessionStorage.getItem(‘username’);

* We can only store String in Session Storage. To save the objects in Session, first, convert the object into JSON string and then store this string in Session Storage as in the following.

**sessionStorage.setItem (‘object’, JSON.stringify(object));**

* If JSON string gets stored in Session Storage, then first convert it into an object as follows.

var object=JSON.parse(sessionStorage.getItem(‘object’));

* Use removeItem() function to delete a particular key from Session Storage.

**sessionStorage.removeItem(‘key’);**

**2. Local Storage**

Local Storage is the second type of HTML Web Storage. It stores data as KEY/VALUE pair.

Following functions are available to access Local Storage.

* Use setItem() function to store data in Local Storage

**localStorage.setItem (‘key’,’value’);**

* Use getItem() function to retrieve data from Local Storage

**localStorage.getItem(‘key’);**

**Q-17. What is the need of introducing Local Storage in HTML5?**

**Answer.**

Before HTML5, LocalStores was done with Cookies. They are not very good for large amounts of data, because, with every request, it passes this data to the server, making it very slow and ineffective. However, HTML5 does not pass this data on every server request. It uses the data ONLY when required. In HTML5, it is possible to store large amounts of data without affecting the website’s performance. Data for different website gets stored in separate areas. However, a particular website can only access its own data.

**Q-18. Explain the key differences between localStorage and sessionStorage objects?**

**Answer.**

Following are the key differences between localStorage and sessionStorage objects.

* 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.

**Q-19. What is the concept of Application Cache in HTML5? What are its advantages?**

**Answer.**

HTML5 introduced the concept of Application Cache. It means that a web application is cached, and is accessible without an internet connection.

Following are the key advantages of Application Cache.

* **Offline browsing –** Users can use the application even when they are offline.
* **Speed  –**Cached resources load faster as compared to content that gets  
  downloaded, directly from the server.
* **Reduced server load –**The browser will only download updated/modified resources from  
  the server.

**Q-20. What is a Manifest file?**

**Answer.**

A Manifest file is a simple text file, that tells the browser what to cache and what not to cache.

A Manifest file contains three Sections as

* **CACHE MANIFEST –**HTML5 performs the caching of files listed under this section after  
  they get downloaded for the first time.
* **NETWORK –**Files listed here, always need a connection to the server. The browser can never cache them.
* **FALLBACK –**Files listed here specify the fallback pages, if any page in it is not accessible.

**Q-21. What is the difference between HTMl5 Application cache and regular HTML browser cache?**

**Answer.**

Following are the key differences between the two.

* In AppCache, we can define all the assets the browser should cache in a manifest file (even the entire site). For fetching this content, it is not necessary for the user to have accessed it previously. In other words, Application Cache can prefetch pages that have not been visited at all and are thereby unavailable in the regular browser cache. However, the browser cache will only store the pages (and associated assets) the user has visited actually.
* The AppCache allows web apps (and websites) to be made available offline, that too, with the same speed benefits as the regular browser cache could provide only when the user is online.

**Q-22. What is a Web Worker? How does it work?**

**Answer.**

JavaScript will hang the browser if it has to handle UI events, query large amounts of API data for processing, and manipulate the DOM simultaneously.  
Web Workers handle this situation by doing all the high computation tasks without interrupting the user interface. They do this by running on separate threads. Thus we can say that.

* A web worker is a script, which runs in the background. It exists in external files.
* The user can perform actions like clicking, selecting things and so on. Meanwhile, the Web worker runs in the background.
* It is appropriate, to use Web worker for CPU intensive tasks.

Since Web workers are in external files, they do not have access to the following JavaScript objects.

* The window object
* The document object
* The parent object

**How does a Web worker work?**

A Web worker gets initialized with the URL of a JavaScript file that contains its code. This code sets event listeners and starts communication with the script that invoked the worker from the main page. The Syntax is as follows.

var worker = new Worker("sample\_prog.js"﴿;

If the javascript file specified in the above code exists, the browser spawns a new worker thread.

Once the Web worker gets spawned, it starts the communication with the parent page, using the postMessage() method. The Web worker, in turn, returns a message, that gets accessed using the onmessage() event on the main page.

Let’s take an example, where the script spawns a Web worker to execute a loop having thousands of iterations. After that, the Web worker returns the calculated value to the HTML page.

<script>

var worker = new Worker('calculateLoop.js');

worker.onmessage = function (event) {

alert("Completed " + event.data + "iterations" );

};

</script>

Below is the code of “calculateLoop.js” file. It makes use of postMessage() API, to pass the communication back to the HTML page.

for (var i = 0; i <= 1000000000; i += 1){

var j = i;

}

postMessage(j);

**Q-23. What are the new attributes provided in HTML5 for <form>?**

**Answer.**

The new attributes provided in HTML5 for <form> are.

* **autocomplete**
  + It specifies if a form or an input field should have “autocomplete” feature set as on or off.
  + If autocomplete is set to on, it enables the browser to fill the values, based on the values that the user starts to enter.
  + autocomplete works for input types like text, search, URL, tel, email, password, date pickers, range, and color.
* **novalidate**
  + It is a boolean attribute.
  + Its presence signifies that the form-data should not get validated at the time of submission.

**Q-23. What is output element in HTML5?**

**Answer.**

Output element gets used when you design a form, that displays the result of a computation. Along with the standard global attributes, <output> also accepts for, form, and name attributes.  
Let’s see a simple example of the <output> element that adds two numbers and displays the resulting value.

<!DOCTYPE html>

<html>

<head>

<title>HTML Output Tag</title>

</head>

<body>

<form oninput="sumresult.value=parseInt(val1.value)+parseInt(val2.value)+parseInt(val3.value)">

<input type="range" name="va1" value="10" /> +

<input type="number" name="val2" value="20" /> +

<input type="number" name="val3" value="40" /><br />

The output is: <output name="sumresult"></output>

</body>

</html>

The form attribute associates the <output> with a form. It displays the output as “70” on the web page.

**Q-24. What are the new attributes provided in HTML5 for <input> element?**

**Answer.**

Following are the new attributes provided in HTML5 for <input>.

* **autofocus**
  + It is a Boolean attribute.
  + The presence of this attribute means that an <input> element should automatically come into focus when the page gets loaded.
* **form**
  + This attribute specifies about all the forms, to which a particular <input> element belongs.
* **formaction**
  + This attribute defines the URL of a file, that will process the input control after the form gets submitted.
  + This attribute is used along with type=”submit” and type=”image”.
  + Also, it overrides the action attribute of the <form> element.
* **formenctype**
  + This attribute defines, the method to encode the form data before submitting it to the server.
  + It gets used with type=”submit” and type=”image”.
  + Also, it overrides the enctype attribute of the <form> element.
* **formmethod**
  + It defines the HTTP method used for sending form related data to the action URL.
  + It gets used with type=”submit” and type=”image”.
  + It overrides the method attribute of the <form> element.
* **formnovalidate**
  + It is a boolean attribute.
  + It gets used with type= “submit”.
  + It indicates that the validation of the <input> element, should not be done at the time of submission.
  + It overrides the novalidate attribute of the <form> element.
* **formtarget**
  + It specifies a name or a keyword of the area where response received after submitting the form will be displayed.
  + It gets used with type=”submit” and type=”image”.
* **height and width**
  + It specifies the height and width of an <input> element.
  + It gets used only with <input type=”image”>.
* **list**
  + It refers to a <datalist> element, which contains a list of pre-defined options for an <input> element.
* **min and max**
  + It specifies the minimum and maximum value for an <input> element.
  + It works with the following input types, number, range, date, datetime, datetime-local, month, time, and week.
* **multiple**
  + It is a boolean attribute.
  + It specifies that the user is allowed to enter more than one value in the <input> element.
  + It works with the following input types: email and file.
* **pattern**
  + It specifies a regular expression with which the value of the <input> element gets compared.
  + It works with the following input types: text, search, URL, tel, email, and password.
* **placeholder**
  + It displays a short hint that indicates the expected value of an input field.
  + It works with the following input types: text, search, URL, tel, email, and password.
* **required**
  + It is a boolean attribute.
  + It indicates that it is mandatory to fill the particular field, before submitting the form.
* **step**
  + It specifies the legal number intervals for an <input> element.
  + It works with the following input types: number, range, date, datetime, datetime-local, month, time, and week.

**Q-25. What is the major difference between, Transitional and Strict doctype?**

**Answer.**

* **Strict –**This DTD contains all HTML components and properties. However, it does NOT INCLUDE presentational or expostulated components (like text style). It does not permit the use of Framesets.
* **Transitional –**This DTD contains all HTML components and properties, INCLUDING presentational and belittled components (like textual style). It does not allow the use of Framesets.

**Q-26. What is Audio Tag in HTML 5? What are its attributes>**

**Answer.**

This new element allows you to embed audio files in an HTML or XHTML document, without the need for any plug-ins. The Audio tag is a new tag introduced in HTML5. You can use it to play audio sound like .mp3, wav, and .ogg.  
Using the <source> tag, we can specify media along with media type and many other attributes. An audio element allows multiple source elements and browser will use the first recognized format.

<!DOCTYPE HTML>

<html>

<body>

<audio controls="controls" >

<source src="URL1" type="audio/mp3" />

<source src="URL2" type="audio/wma" />

<source src="URL3" type="audio/x-wav" />

Your browser does not support the audio element.

</audio>

</body>

</html>

The HTML5 audio tag supports following attributes to direct the look and feel and various functionalities of the control.

* **autoplay**
  + It is a boolean attribute. If, the value is set the audio track starts playing automatically. The System will not wait for data loading to complete.
* **autobuffer**
  + It is a boolean attribute. If set, the audio will automatically begin buffering, even if the automatic play is not enabled.
* **controls**
  + If this attribute is present, it will allow the user to control audio playback, including volume, seeking, and pause/resume playback.
* **loop**
  + Setting this boolean attribute would automatically restart the audio from the beginning, once it reaches to the end.
* **preload**
  + This attribute specifies that the audio will be loaded at page load, and will be ready to run. If autoplay is present, this attribute will not work.
* **src**
  + It represents the URL of the audio to embed. Its presence is optional.

**Q-27. What is the use of <fieldset> tag in HTML5?**

**Answer.**

The <fieldset> tag groups related form elements. It is like a box. In other words, it draws a box around related elements.

It must start with a <legend>tag because the <legend> tag defines the title of the fieldset.

Following is the Syntax of the <fieldset> tag in HTML5.

<fieldset>Controls</fieldset>

All the popularly known browsers provide the support for the <fieldset> tag.

HTML5 supports following attributes with the fieldset tag.

* **disabled**
  + Its value is disabled. It specifies, whether the fieldset will be displayed or not.
* **name**
  + Its value is in the form of text. It defines the name of the fieldset.
* **form**
  + Its value is the name of the form. It specifies the form related to the fieldset.

Let’s see an example, where we create a fieldset in a form. Here we use the <legend> tag to define the heading for the fieldset.

<html>

<body>

<form>

<fieldset>

<legend>Personal Information</legend>

First Name: <input type="text" />

<br/><br/> Last Name: <input type="text" />

<br/><br/> person\_Address: <input type="text" />

<br/><br/> person\_Qualification: <input type="text" />

</fieldset>

</form>

</body>

</html>

**Q-28. What are the HTML tags which get deprecated in HTML5?**

**Answer.**

Following are the tags that are deprecated in HTML5.

* **<basefont>**
* **<big>**
* **<center>**
* **<font>**
* **<s>**
* **<strike>**
* **<tt>**
* **<u>**
* **<frame>**
* **<frameset>**
* **<noframe>**
* **<acronym>**
* **<applet>**
* **<isindex>**
* **<dir>**

Some attributes from HTML4 are no longer allowed in HTML5 since their functionality is better handled by CSS. Below are some of the known attributes that got removed and the corresponding impacted element.

* **Attribute Removed Element**
* **rev, charset a, link**
* **longdesc, name img**
* **version html**
* **abbr th**
* **scope td**
* **align all block level elements**
* **background body**
* **hspace, vspace img**
* **bgcolor table, tr, td, th**
* **border, cell padding, cell spacing table**
* **height, width td, th**
* **valign table**

**Q-29. What is a meter tag? What is the difference between progress tag and a meter tag?**

**Answer.**

The <meter> tag defines a scalar measurement within a known range or a fractional value. We can also call it a gauge.  
Some of the items that can be represented using <meter> tag are Disk usage, the relevance of a query result, and so on.

Note: The <meter> tag should not be used to indicate progress (as in a progress bar). For progress bars, use the <progress> tag.

Following example demonstrates the use of the <meter> tag.

<li><meter min="0" max="100" value="25">25%</meter></li>

The <meter> tag provides the support of the following attributes.

* **min**
  + It is a number. It specifies the minimum value of the range.
* **max**
  + It is a number. It specifies the maximum value of the range.
* **low**
  + It is a number. It defines a range that represents <low> value.
* **high**
  + It is a number. It defines a range that represents “high” value.
* **value**
  + It is a number. It is a mandatory element. It defines the current value of the gauge.
* **optimum**
  + It is a mandatory element with a numeric value. It specifies the optimum, or the best value, for the element. If this value is higher than the “high” value, this indicates that the higher the value, the better it is. If it’s lesser than the <low> mark, it means that the lower values are better. If it is, in between, then it indicates that neither high nor low values are good.
* **form**
  + It specifies one or more forms that define the <meter> element. It has value form\_id.

**Q-30. Why do we need HTML5 Server-Sent Events?**

**Answer.**

HTML5 Server-Sent Events (SSE) is a new way for the web pages to communicate with the web server. It enables a webpage to get updates from a server automatically. It was possible earlier also, but for this, the web page needs to ask if any updates were available. The client makes a request and waits for the server to respond with data. Once the web server provides its response, the communication is over.

However, there are some situations, where web pages require a long-term connection with the web server. A typical example is stock quotes on finance websites where price update happens automatically. Other examples are news feeds, sports results that run continuously on media websites, Facebook/Twitter updates and so on.

We can achieve the above, using HTML5 using SSE. It enables a web page to hold an open connection to the web server so that it can send a response automatically at any time. Thus there’s no need to reconnect and run the same server script from scratch over and over again.

**Receive Server-Sent Event Notifications.**

The EventSource interface contains the Server-Sent event API. We need to create an EventSource object to receive the Server-Sent event notifications. Following is the code for the same.

var source = new EventSource("sse\_demo.php");

source.onmessage = function(event) {

document.getElementById("result").innerHTML += event.data + "<br>";

};

Above code performs following steps.

* First, create a new EventSource object, and specify the URL of the page sending the updates (in this example “sse\_demo.php”).
* Every time an update arrives, onmessage event gets triggered.
* When an onmessage event occurs, it places the received data into the element that has <id = result>.

**Server-Side Code Example.**

For the above example to work, we need a server capable of sending data updates. The server-side event stream syntax is simple. Set the “Content-Type” header to “text/event-stream”. Now you can start sending event streams. Following is the code (demo\_sse.php).

<?php

header('Content-Type: text/event-stream');

header('Cache-Control: no-cache');

$time = date('r');

echo "data: The server time is: {$time}\n\n";

flush();

?>

Following is the explanation of the above code.

* Set the “Content-Type” header to “text/event-stream”.
* Specify that the page should not cache.
* Output the data to send (Always start with “data: “).
* Flush the output data back to the web page.

**What does a DOCTYPE do?**

**DOCTYPE** is an abbreviation for **DOCument TYPE**. A DOCTYPE is always associated to a **DTD** - for **Document Type Definition**.

A DTD defines how documents of a certain type should be structured (i.e. a button can contain a span but not a div), whereas a DOCTYPE declares what DTD a document *supposedly* respects (i.e. this document respects the HTML DTD).

For webpages, the DOCTYPE declaration is required. It is used to tell user agents what version of the HTML specifications your document respects. Once a user agent has recognized a correct DOCTYPE, it will trigger the **no-quirks mode** matching this DOCTYPE for reading the document. If a user agent doesn't recognize a correct DOCTYPE, it will trigger the **quirks mode**.

The DOCTYPE declaration for the HTML5 standards is <!DOCTYPE html>.

**References**

* <https://html.spec.whatwg.org/multipage/syntax.html#the-doctype>
* <https://html.spec.whatwg.org/multipage/xhtml.html>
* <https://quirks.spec.whatwg.org/>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/html-questions.md#table-of-contents)

**How do you serve a page with content in multiple languages?**

I will assume that it is asking about the most common case, which is how to serve a page with content available in multiple languages, but the content within the page should be displayed only in one consistent language.

When an HTTP request is made to a server, the requesting user agent usually sends information about language preferences, such as in the Accept-Language header. The server can then use this information to return a version of the document in the appropriate language if such an alternative is available. The returned HTML document should also declare the lang attribute in the <html> tag, such as <html lang="en">...</html>.

Of course this is useless for letting a search engine know that the same content is available in different languages, and so you must also make use of the hreflang attribute in the <head>. Eg. <link rel="alternate" hreflang="de" href="http://de.example.com/page.html" />

In the back end, the HTML markup will contain i18n placeholders and content for the specific language stored in YML or JSON formats. The server then dynamically generates the HTML page with content in that particular language, usually with the help of a back end framework.

**References**

* <https://www.w3.org/International/getting-started/language>
* <https://support.google.com/webmasters/answer/189077>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/html-questions.md#table-of-contents)

**What kind of things must you be wary of when designing or developing for multilingual sites?**

* Use lang attribute in your HTML.
* Directing users to their native language - Allow a user to change his country/language easily without hassle.
* Text in raster-based images (e.g. png, gif, jpg, etc.), is not a scalable approach - Placing text in an image is still a popular way to get good-looking, non-system fonts to display on any computer. However, to translate image text, each string of text will need to have a separate image created for each language. Anything more than a handful of replacements like this can quickly get out of control.
* Restrictive words/sentence length - Some content can be longer when written in another language. Be wary of layout or overflow issues in the design. It's best to avoid designing where the amount of text would make or break a design. Character counts come into play with things like headlines, labels, and buttons. They are less of an issue with free-flowing text such as body text or comments.
* Be mindful of how colors are perceived - Colors are perceived differently across languages and cultures. The design should use color appropriately.
* Formatting dates and currencies - Calendar dates are sometimes presented in different ways. Eg. "May 31, 2012" in the U.S. vs. "31 May 2012" in parts of Europe.
* Do not concatenate translated strings - Do not do anything like "The date today is " + date. It will break in languages with different word order. Use a template string with parameters substitution for each language instead. For example, look at the following two sentences in English and Chinese respectively: I will travel on {% date %} and {% date %} 我会出发. Note that the position of the variable is different due to grammar rules of the language.
* Language reading direction - In English, we read from left-to-right, top-to-bottom, in traditional Japanese, text is read up-to-down, right-to-left.

**References**

* <https://www.quora.com/What-kind-of-things-one-should-be-wary-of-when-designing-or-developing-for-multilingual-sites>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/html-questions.md#table-of-contents)

**What are data- attributes good for?**

Before JavaScript frameworks became popular, front end developers used data- attributes to store extra data within the DOM itself, without other hacks such as non-standard attributes, extra properties on the DOM. It is intended to store custom data private to the page or application, for which there are no more appropriate attributes or elements.

These days, using data- attributes is generally not encouraged. One reason is that users can modify the data attribute easily by using inspect element in the browser. The data model is better stored within JavaScript itself and stay updated with the DOM via data binding possibly through a library or a framework.

However, one perfectly valid use of data attributes, is to add a hook for *end to end* testing frameworks such as Selenium and Capybara without having to create a meaningless classes or ID attributes. The element needs a way to be found by a particular Selenium spec and something like data-selector='the-thing' is a valid way to do so without convoluting the semantic markup otherwise.

**References**

* <http://html5doctor.com/html5-custom-data-attributes/>
* [https://www.w3.org/TR/html5/dom.html#embedding-custom-non-visible-data-with-the-data-\*-attributes](https://www.w3.org/TR/html5/dom.html#embedding-custom-non-visible-data-with-the-data-*-attributes)

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/html-questions.md#table-of-contents)

**Consider HTML5 as an open web platform. What are the building blocks of HTML5?**

* Semantics - Allowing you to describe more precisely what your content is.
* Connectivity - Allowing you to communicate with the server in new and innovative ways.
* Offline and storage - Allowing webpages to store data on the client-side locally and operate offline more efficiently.
* Multimedia - Making video and audio first-class citizens in the Open Web.
* 2D/3D graphics and effects - Allowing a much more diverse range of presentation options.
* Performance and integration - Providing greater speed optimization and better usage of computer hardware.
* Device access - Allowing for the usage of various input and output devices.
* Styling - Letting authors write more sophisticated themes.

**References**

* <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/html-questions.md#table-of-contents)

**Describe the difference between a cookie, sessionStorage and localStorage.**

All the above-mentioned technologies are key-value storage mechanisms on the client side. They are only able to store values as strings.

|  | **cookie** | **localStorage** | **sessionStorage** |
| --- | --- | --- | --- |
| Initiator | Client or server. Server can use Set-Cookie header | Client | Client |
| Expiry | Manually set | Forever | On tab close |
| Persistent across browser sessions | Depends on whether expiration is set | Yes | No |
| Sent to server with every HTTP request | Cookies are automatically being sent via Cookie header | No | No |
| Capacity (per domain) | 4kb | 5MB | 5MB |
| Accessibility | Any window | Any window | Same tab |

*Note: If the user decides to clear browsing data via whatever mechanism provided by the browser, this will clear out any cookie, localStorage, or sessionStorage stored. It's important to keep this in mind when designing for local persistance, especially when comparing to alternatives such as server side storing in a database or similar (which of course will persist despite user actions).*

**References**

* <https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies>
* <http://tutorial.techaltum.com/local-and-session-storage.html>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/html-questions.md#table-of-contents)

**Describe the difference between <script>, <script async> and <script defer>.**

* <script> - HTML parsing is blocked, the script is fetched and executed immediately, HTML parsing resumes after the script is executed.
* <script async> - The script will be fetched in parallel to HTML parsing and executed as soon as it is available (potentially before HTML parsing completes). Use async when the script is independent of any other scripts on the page, for example, analytics.
* <script defer> - The script will be fetched in parallel to HTML parsing and executed when the page has finished parsing. If there are multiple of them, each deferred script is executed in the order they were encoun­tered in the document. If a script relies on a fully-parsed DOM, the defer attribute will be useful in ensuring that the HTML is fully parsed before executing. There's not much difference in putting a normal <script> at the end of <body>. A deferred script must not contain document.write.

Note: The async and defer attrib­utes are ignored for scripts that have no src attribute.

**References**

* <http://www.growingwiththeweb.com/2014/02/async-vs-defer-attributes.html>
* <https://stackoverflow.com/questions/10808109/script-tag-async-defer>
* <https://bitsofco.de/async-vs-defer/>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/html-questions.md#table-of-contents)

**Why is it generally a good idea to position CSS <link>s between <head></head> and JS <script>s just before </body>? Do you know any exceptions?**

**Placing <link>s in the <head>**

Putting <link>s in the <head> is part of proper specification in building an optimized website. When a page first loads, HTML and CSS are being parsed simultaneously; HTML creates the DOM (Document Object Model) and CSS creates the CSSOM (CSS Object Model). Both are needed to create the visuals in a website, allowing for a quick "first meaningful paint" timing. This progressive rendering is a category optimization sites are measured in their performance scores. Putting stylesheets near the bottom of the document is what prohibits progressive rendering in many browsers. Some browsers block rendering to avoid having to repaint elements of the page if their styles change. The user is then stuck viewing a blank white page. Other times there can be flashes of unstyled content (FOUC), which show a webpage with no styling applied.

**Placing <script>s just before </body>**

<script> tags block HTML parsing while they are being downloaded and executed which can slow down your page. Placing the scripts at the bottom will allow the HTML to be parsed and displayed to the user first.

An exception for positioning of <script>s at the bottom is when your script contains document.write(), but these days it's not a good practice to use document.write(). Also, placing <script>s at the bottom means that the browser cannot start downloading the scripts until the entire document is parsed. This ensures your code that needs to manipulate DOM elements will not throw an error and halt the entire script. If you need to put <script> in the <head>, use the defer attribute, which will achieve the same effect of downloading and running the script only after the HTML is parsed.

Keep in mind that putting scripts just before the closing </body> tag will create the illusion that the page loads faster on an empty cache (since the scripts won't block downloading the rest of the document). However, if you have some code you want to run during page load, it will only start executing after the entire page has loaded. If you put those scripts in the <head> tag, they would start executing before - so on a primed cache the page would actually appear to load faster.

**References**

* <https://developer.yahoo.com/performance/rules.html#css_top>
* <https://www.techrepublic.com/blog/web-designer/how-to-prevent-flash-of-unstyled-content-on-your-websites/>
* <https://developers.google.com/web/fundamentals/performance/critical-rendering-path/>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/html-questions.md#table-of-contents)

**What is progressive rendering?**

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.

It used to be much more prevalent in the days before broadband internet but it is still used in modern development as mobile data connections are becoming increasingly popular (and unreliable)!

Examples of such techniques:

* Lazy loading of images - Images on the page are not loaded all at once. JavaScript will be used to load an image when the user scrolls into the part of the page that displays the image.
* Prioritizing visible content (or above-the-fold rendering) - Include only the minimum CSS/content/scripts necessary for the amount of page that would be rendered in the users browser first to display as quickly as possible, you can then use deferred scripts or listen for the DOMContentLoaded/load event to load in other resources and content.
* Async HTML fragments - Flushing parts of the HTML to the browser as the page is constructed on the back end. More details on the technique can be found [here](http://www.ebaytechblog.com/2014/12/08/async-fragments-rediscovering-progressive-html-rendering-with-marko/).

**References**

* <https://stackoverflow.com/questions/33651166/what-is-progressive-rendering>
* <http://www.ebaytechblog.com/2014/12/08/async-fragments-rediscovering-progressive-html-rendering-with-marko/>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/html-questions.md#table-of-contents)

**Why you would use a srcset attribute in an image tag? Explain the process the browser uses when evaluating the content of this attribute.**

You would use the srcset attribute when you want to serve different images to users depending on their device display width - serve higher quality images to devices with retina display enhances the user experience while serving lower resolution images to low-end devices increase performance and decrease data wastage (because serving a larger image will not have any visible difference). For example: <img srcset="small.jpg 500w, medium.jpg 1000w, large.jpg 2000w" src="..." alt=""> tells the browser to display the small, medium or large .jpg graphic depending on the client's resolution. The first value is the image name and the second is the width of the image in pixels. For a device width of 320px, the following calculations are made:

* 500 / 320 = 1.5625
* 1000 / 320 = 3.125
* 2000 / 320 = 6.25

If the client's resolution is 1x, 1.5625 is the closest, and 500w corresponding to small.jpg will be selected by the browser.

If the resolution is retina (2x), the browser will use the closest resolution above the minimum. Meaning it will not choose the 500w (1.5625) because it is greater than 1 and the image might look bad. The browser would then choose the image with a resulting ratio closer to 2 which is 1000w (3.125).

srcsets solve the problem whereby you want to serve smaller image files to narrow screen devices, as they don't need huge images like desktop displays do — and also optionally that you want to serve different resolution images to high density/low-density screens.

**References**

* <https://developer.mozilla.org/en-US/docs/Learn/HTML/Multimedia_and_embedding/Responsive_images>
* <https://css-tricks.com/responsive-images-youre-just-changing-resolutions-use-srcset/>

[[↑] Back to top](https://github.com/yangshun/front-end-interview-handbook/blob/master/contents/en/html-questions.md#table-of-contents)

**Have you used different HTML templating languages before?**

Yes, Pug (formerly Jade), ERB, Slim, Handlebars, Jinja, Liquid, and EJS just to name a few. In my opinion, they are more or less the same and provide similar functionality of escaping content and helpful filters for manipulating the data to be displayed. Most templating engines will also allow you to inject your own filters in the event you need custom processing before display.