1. How are inline and block elements different from each other?

**Block Level Elements**: A block-level element is an HTML element that begins a new line on a Web page and extends the full width of the available horizontal space of its parent element, they have the following characterstics:

* If no width is set, will expand naturally to fill its parent container
* Can have margins and/or padding
* If no height is set, will expand naturally to fit its child elements (assuming they are not floated or positioned)
* By default, will be placed below previous elements in the markup (assuming no floats or positioning on surrounding elements)
* Ignores the vertical-align property

Examples: <p>, <div>, <form>, <header>, <nav>, <ul>, <li>, and <h1>.

**Inline Elements**: Unlike the block level elements, inline elements begin within a line rather starting a new line, they have the following characteristics :

* Flows along with text content, thus
* Will not clear previous content to drop to the next line like block elements
* Is subject to white-space settings in CSS
* Will ignore top and bottom margin settings, but will apply left and right margins, and any padding
* Will ignore the width and height properties
* If floated left or right, will automatically become a block-level element, subject to all block characteristics
* Is subject to the vertical-align property

2. Explain the difference between visibility:hidden and display:none

**display:none** removes the element from the normal flow of the page, allowing other elements to fill in. It commonly used with JavaScript to hide and show elements without deleting and recreating them.

e.g:-

<div style=”display:none”>

The contents inside will not display on screen and even will not take space.

</div>

**visibility:hidden** leaves the element in the normal flow of the page such that is still occupies space.

e.g :

h1.hidden {

visibility: hidden

}

<h1 class="visible">This is a visible heading</h1>

<h1 class="hidden">This is an invisible heading</h1>

<p> the invisible heading still takes up space.</p>

3. Explain the clear and float properties.

Clear property: Clear is a CSS property that specifies on which sides of an element floating elements are not allowed to float. It is a sister property of float.

Float property: Float is a CSS property specifies that an element should be placed along the left or right side of its container, where text and inline elements will wrap around it. This way, the element is taken from the normal flow of the web page.

4. Explain difference between absolute, relative,fixed and static.

Static Position:-

Static is the default type of positioning. When elements don’t have a position specifically set, they default to static. There’s not much you can do with a statically positioned element. These elements will stack in a standard one-after-another order. So in your code, whatever comes first will be displayed first, then the next element will be below it, and so on.

Relative position:-

The relative positioning is interesting because if you just give an element position:relative it will initially seem to do nothing. In order to see a relatively positioned element move you also need to tell it where to go using one of the following top: XXX ; bottom: XXX; left: XXX; right: XXX;. When you begin to move around a relatively positioned element, two things happen. First, you will see the element move off from the side specified, so if you wrote top:50px; the element will move 50px off from the top, or basically down.

Absolute Position :-

An absolutely positioned element is actually removed from the DOM and positioned based on its nearest relatively positioned parent element. What does this mean?… First off, unlike a relatively positioned element which doesn’t effect other static elements, when you give an element position:absolute its like it no longer exists. This means that other static elements will move up to fill in the space where the absolute element would have been. The position of the absolute element is determined by its parent elements. If all of the parent elements are either static, or there are none, then the element is positioned based on the <body>.

Fixed Position:-

Fixed elements are completely independent of everything else on the web page. Regardless of any parents, a fixed position element will always be positioned based on the browser window. The interesting thing about fixed position elements is that when the page is scrolled, the element stays “fixed” and is always visible.

5. Write the HTML code to create a table in which there are 4 columns( ID , Employee Name, Designation, Department) and at least 6 rows. Also do some styling to it.

Table.html

<Html>

<head>

<link rel="stylesheet" href="table\_style.css"

</head>

<Body>

<div class ="new1">

EMPLOYEE

</div>

<table>

<td>

<tr> <th>ID</th><th>Employee Name</th><th>Designation</th><th>Project</th></tr>

<tr> <td>007</td> <td>shubham</td> <td>coder</td> <td>CS</td> </tr>

<tr> <td>008</td> <td>vrinda</td> <td>tester</td> <td>HDFC</td> </tr>

<tr> <td>009</td><td>shaan</td><td>technical lead</td><td>Set Max</td> </tr>

<tr> <td>010</td><td>aashish</td><td>team lead</td><td>NCPCR</td> </tr>

<tr> <td>011</td><td>randhir</td><td>Manager</td><td>LogMeIn</td> </tr>

<tr> <td>100</td><td>shreya</td><td>Dept. Head</td><td>AMD</td> </tr>

</td>

</table>

</body>

</html>

table\_style.css

table {

border-collapse: collapse;

width: 100%;

}

th, td {

text-align: center;

padding: 8px;

}

tr:nth-child(even){background-color: #f2f2f2}

th {

background-color: #4CAF50;

color: white;

}

tr:hover{background-color: #b1e549}

tr:active{background-color: #ff0000}

6. Why do we use meta tags?

Metadata is data (information) about data.

The <meta> tag provides metadata about the HTML document. Metadata will not be displayed on the page, but will be machine parsable.

Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata.

The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

HTML5 introduced a method to let web designers take control over the viewport (the user's visible area of a web page), through the <meta> tag (See "Setting The Viewport" example below).

7. Explain box model.

All HTML elements can be considered as boxes. In CSS, the term "box model" is used when talking about design and layout.The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:



* Content - The content of the box, where text and images appear
* Padding - Clears an area around the content. The padding is transparent
* Border - A border that goes around the padding and content
* Margin - Clears an area outside the border. The margin is transparent

8. What are the different types of CSS Selectors?

In CSS, selectors are patterns used to select the element(s) you want to style.

1. Element selector.

The element selector selects elements based on the element name.

p {

text-align: center;

color: blue;

font-family:Arial;

}

Now the above properties would be applied to all the paragraph tags.

2. Id Selector

The id selector uses the id attribute of an HTML element to select a specific element. The id of an element should be unique within a page.

#myID {

text-align: center;

color: blue;

font-family:Arial;

}

Now the properties held by #myid selector may be applied to the tag which would be assigned this ID.

.myClass{

color:blue;

}

<p class=” myClass”>this text would appear blue</p>

3. Class Selector

The class selector selects elements with a specific class attribute.

9. Define Doctype.

Doctype declaration is an instruction to the webpage that declares the precise version of html being used.

for e.g.

<!DOCTYPE html> declaration defines this document to be HTML5.

10. Explain 5 HTML5 semantic tags.

* <article> - Defines an article in the document. it specifies independent, self-contained content.

e.g.

<article>

<h1>Article heading</h1>

<p>Article content goes here</p>

</article>

* <aside> - Defines content aside from the page content like sidebar.

e.g.

<aside>

<h4>Aside info</h4>

<p>Aside info goes here.</p>

</aside>

* <footer> - Defines a footer for the document or a section

e.g.

<footer>

<p>Contact information: <a href="abcd@gmail.com">

abcd@gmail.com</a>.</p>

</footer>

* <header> - element represents a container for introductory content or a set of navigational links.

e.g.

<header>

<h1>Most important heading here</h1>

<h3>Less important heading here</h3>

<p>Content info here.</p>

</header>

* <nav> - Defines navigation links in the document

e.g.

<nav>

<a href="#">HomeL</a>

<a href="#">About us</a>

<a href="#">Coursest</a>

<a href="#">Login</a>

</nav>

11. Create HTML for web-page.jpg (check resources, highest weightage for answers)

12. Create HTML for form.png (check resources, highest weightage for answers)