Question 1--> **Define and give some examples of inline and block level elements in HTML.**

Answer-->

Block-level elements: Block-level elements initiate a new line and utilize the entire available width within their parent container. They establish a "block" that displaces other elements horizontally. Block-level elements can accommodate other block-level and inline elements."

Examples of block-level elements:

: A generic container for other elements and used for layout purposes.

<p>: Represents a paragraph of text.

<h1> to <h6>: Headings with different levels of importance.

 and <ol>: Unordered and ordered lists, respectively.

: List items within lists.

Inline elements: Inline elements do not start on a new line and only take up as much width as necessary to fit their content. They flow within the text and do not disrupt the surrounding content. Inline elements cannot contain block-level elements but can contain other inline elements.

Examples of Inline elements:

: A generic inline container used for styling and scripting purposes.

<a>: Represents a hyperlink.

 and : Used for emphasizing text.

<img>: Displays an image.

<input>: Used for input fields in forms.

<button>: Represents a clickable button.

Question 2 --> **What do you mean by semantic tag in HTML? Give some examples of semantic and non-semantic tags**

Answer -->

Semantic tags in HTML are elements that carry meaning and convey the structure of the content they enclose. They provide context and define the purpose of the content, making it easier for both machines (like search engines) and developers to understand the document's layout. Semantic tags improve accessibility, SEO, and maintainability of the web page by organizing the content in a meaningful way.

Examples of Semantic Tags:

<header>: Represents the header section of a document or a section within an article.
<nav>: Defines a navigation menu for the page.
<main>: Represents the main content of the document.
<article>: Represents a self-contained, independent piece of content, like a blog post or news article.
<section>: Represents a generic section of content within the document.

Examples of Non-Semantic Tags:

: A generic container that does not carry any inherent meaning on its own.
: A generic inline container used for styling or scripting purposes.
<b>: Represents bold text, but it does not convey the reason for the text's importance or significance.
<i>: Represents italicized text, but it does not specify why the text is in italics.
<font>: Used to apply font styles, sizes, and colors but lacks semantic meaning.

Question 3--> **Discuss about HTML ordered and unordered list**

Answer -->

Ordered list:

An ordered list is used to represent a list of items in a particular sequence, where each item is preceded by a number or letter (default is numbers). The order of the list items is significant and can be indicated by numerical or alphabetical ordering.

Ordered Lists (<ol>):
Used to create a numbered list.
Each list item is preceded by a number (by default, starting from 1).
The order of items in the list matters.

Unordered Lists :
An unordered list is used to represent a list of items with no particular sequence or order. Each item in the list is preceded by a bullet point by default, although this can be customized using CSS.

Unordered Lists (<ul>):

Used to create a bullet-pointed list.
Each list item is preceded by a bullet point (by default).
The order of items in the list doesn't matter; they are just points in a list.

Question 4--> **How many ways are there for inserting stylesheet in HTML? Give some examples of all the ways. Discuss about CSS Box Model.**

Answer -->

Inline CSS:

The inline styles will only affect the HTML element to which the style attribute with CSS-property values is applied. The first paragraph in the example below will be styled in red with a 20px font size. The properties apply just to the first line of the code, not the full code.

Example :

```
<body>

    <p style=”color:red; font-size: 20px;”>This is our first HTML code.</p>

    <p>This is our second HTML code</p>

</body>
```

Internal CSS:

Internal CSS is one of the most popular CSS forms for updating, customizing, and modifying a single web page's unique styles. You can use the internal CSS by integrating the <style> element in the <head> section of a HTML web page. Internal CSS can be used to style a single web page, but not multiple web pages, and we can style numerous web pages with the same code.

Example :

```
<!DOCTYPE html>

<html>

<head>

 <title>Internal Stylesheet Example</title>

 <style>

  body {

   font-family: Arial, sans-serif;

   background-color: #f0f0f0;

  }
```

```
    h1 {
      color: #007bff;
    }
  </style>
</head>
<body>
  <!-- Your HTML content goes here -->
</body>
</html>
```

External CSS:

In this method, you link an external CSS file to your HTML document using the <link> element within the <head> section of the HTML file.

Example:

```
<!DOCTYPE html>
<html>
<head>
  <title>Inline Styles Example</title>
</head>
<body>
  <h1 style="color: #007bff;">This is a heading with inline style.</h1>
  <p style="font-size: 16px; color: #333;">This is a paragraph with inline style.</p>
</body>
```

</html>

Question 5--> **Discuss about CSS Box Model.**

Consider the following example and find out the total width the div element will have.

```
div {
        width: 300px;
        border: 15px solid green;
        padding: 50px;
        margin: 20px;
      }
```

Answer -->

1.Content width: The content width is explicitly set to 300px.

2.Border width: The total border width is 30px (15px on the left + 15px on the right) because the border is 15px wide on each side.

3.Padding width: The total padding width is 100px (50px on the left + 50px on the right) because the padding is 50px wide on each side.

4.Margin width: The total margin width is 40px (20px on the left + 20px on the right) because the margin is 20px wide on each side.

To calculate the total width of the element:
Total width = Content width + Total Padding width + Total Border width + Total Margin width

Total width = 300px + 100px + 30px + 40px = 470px

So, the element will have a total width of 470px.

Question 6--> **What are Pseudo-classes? Why do we use Pseudo-classes?**

Answer -->

A pseudo-class is a selector that selects elements that are in a specific state, e.g. they are the first element of their type, or they are being hovered over by the mouse pointer. They tend to act as if you had applied a class to some part of your document, often helping you cut down on excess classes in your markup, and giving you more flexible, maintainable code.

1.Styling based on state: Pseudo-classes allow you to style elements based on their current state or user interaction. For example, you can change the style of a link when it is hovered over or clicked, style a form element when it is in focus, or style an element when it is being selected.

2.Styling based on position: Pseudo-classes also enable you to target elements based on their position

within the document tree. For instance, you can style the first child or last child of a parent element differently, select even or odd elements, or target specific elements based on their relationship with other elements.

3.Creating interactive effects: Pseudo-classes are crucial for creating interactive and dynamic effects on web pages. They help in creating animations, transitions, and other effects based on user interactions or element states.

Question 7--> **Discuss the following CSS rule/style :**
margin: 15px 70px;

Answer -->

The CSS declaration `margin: 15px 70px;` applies spacing around an element, establishing a gap between it and nearby elements.

In this particular rule, the `margin` property is utilized with two values: 15px and 70px. These values designate the top/bottom margin and left/right margin, respectively. The order of the values is crucial and adheres to the shorthand notation for specifying margins.

The significance of the values can be further elucidated as follows:

- 15px: This value denotes the top and bottom margins, setting a margin of 15 pixels above and below the element.

- 70px: This value indicates the left and right margins, establishing a margin of 70 pixels on the left and right sides of the element.

Hence, the CSS declaration `margin: 15px 70px;` will create a margin of 15 pixels above and below the element, as well as a margin of 70 pixels on the left and right sides of the element.

This approach is commonly employed to provide spacing between an element and its adjacent content or other elements on the page. The specific values used can be adjusted as necessary to achieve the desired layout and visual appearance.

Question 8--> **Discuss about CSS descendant selectors.**

Answer -->

The CSS descendant selector is used to match the descendant elements of a particular element. The word Descendant indicates nested anywhere in the DOM tree. It can be a direct child or deeper than five levels, but it will still be referred to as a descendant.

The Descendant combinator is represented using a single space. It combines two selectors in which the first selector represents an ancestor (parent, parent's parent, etc.), and the second selector represents

descendants. The elements matched by the second selector are selected if they have an ancestor element that matches the first selector. Descendant selectors use the descendant combinators.