

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

Ans: **Inline elements:** Basically, an inline element does not cause a line break (start on a new line) and does not take up the full width of a page, only the space bounded by its opening and closing tag. It is usually used within other HTML elements. Examples include: anchor tag and emphasis tag.

**Block elements:** A block-level element always starts on a new line and takes up the full width of a page, from left to right. A block-level element can take up one line or multiple lines and has a line break before and after the element. Examples include: div tag and paragraph tag.

**Visually:**

## BLOCK-LEVEL ELEMENTS:



## INLINE ELEMENTS:



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

Ans: **The display: none** property is used to hide elements without deleting them. It does not take up any space. On the other hand, **The visibility: hidden** property also hides an element, but affects the layout i.e. takes up space.

3. Explain the clear and float properties.

Ans.**float** will make an element align to the left or right (the parameter) inside its parent. **float: none** does nothing, unless the element was already floating. The float element lose it's automatically filled width, and reduce it to as small as it can get.

Clear will make sure there are no floating elements on the side you tell. If there is one, it will move down until there is none in the given direction. **clear: both** will check this for both directions.

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

Ans: **Static:**An element with position: static; is not positioned in a special away but it is positioned according to the flow of the page:They are not affected by properties like top, bottom, left, and right.HTML elements are positioned static by default.

**Relative:** An element with position: relative; is positioned relative to its normal position.Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

**Fixed:** An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.A fixed element does not leave a gap in the page where it would normally have been located.

**Absolute:** An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.

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.

Ans.

file:///home/ishrat/Desktop/table.html			
ID	Employee Name	Designation	Department
1	Vikram	trainee	AEM
2	Naman	trainee	Android
3	Gagan	trainee	JVM
4	Chirag	trainee	AEM
5	suman	trainee	FEEN
6	Aaryan	trainee	Android

6. Why do we use meta tags?

Ans: 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.

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="description" content="Free Web tutorials">
```

```
<meta name="keywords" content="HTML,CSS,XML,JavaScript">
```

```
<meta name="author" content="John Doe">
```

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

```
</head>
```

7. Explain box model.

Ans: CSS box model is a container which contains multiple properties including borders, margin, padding and the content itself. It is used to create the design and layout of web pages. It can be used as a toolkit for customizing the layout of different elements. The web browser renders every element as a rectangular box according to the CSS box model.

Box-Model has multiple properties in CSS. Some of them are given below:

- **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.



**BOX MODEL**

8. What are the different types of CSS Selectors?

Ans: CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class).
- Combinator selectors (select elements based on a specific relationship between them).
- Pseudo-class selectors (select elements based on a certain state).
- Pseudo-elements selectors (select and style a part of an element).
- Attribute selectors (select elements based on an attribute or attribute value).

### **CSS Selectors:**

**The CSS element Selector:** The element selector selects HTML elements based on the element name.

**The CSS id selector:** The id selector uses the id attribute of an HTML element to select a specific element. The id of an element is unique within a page, so the id selector is used to select one unique element. To select an element with a specific id, write a hash (#) character,

followed by the id of the element.

**The CSS class selector:**The class selector selects HTML elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the class name.

**The CSS Universal Selector:**The universal selector (\*) selects all HTML elements on the page.

## 9. Define Doctype.

Ans: The <!DOCTYPE> declaration must be the very first thing in your HTML document, before the <html> tag. The <!DOCTYPE> declaration is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in.

In HTML 4.01, the <!DOCTYPE> declaration refers to a DTD, because HTML 4.01 was based on SGML. The DTD specifies the rules for the markup language, so that the browsers render the content correctly. HTML5 is not based on SGML, and therefore does not require a reference to a DTD.

## 10.Explain 5 HTML5 semantic tags.

Ans: A semantic element clearly describes its meaning to both the browser and the developer.

Five semantic tags are as follows:

1. **<aside>**: The <aside> element is intended for content that is not part of the flow of the text in which it appears, however still related in some way. This of <aside> as a sidebar to your main content.

2. **<header> and <hgroup>**: The <header> element is generally found at the top of a document, a section, or an article and usually contains the main heading and some navigation and search tools. The <hgroup> element should be used where you want a main heading with one or more subheadings.

3. **<footer>**: If there is a <header> there must be a <footer>. A <footer> is generally found at the bottom of a document, a section, or an article. Just like the <header> the content is generally meta-information, such as author details, legal information, and/or links to related information. It is also valid to include <section> elements within a footer.

4. **<small>**: The <small> element often appears within a <footer> or <aside> element which would usually contain copyright information or legal disclaimers, and other such fine print. However, this is not intended to make the text smaller. It is just describing its content, not prescribing presentation.

5. **<time>**: The <time> element allows an unambiguous ISO 8601 date to be attached to a human-readable version of that date.

