1.What is position property in CSS and its type ?

Ans. Types of `position` in CSS

1. `static` **(Default)**
   - The default position for all elements.
   - Elements appear in the normal document flow.
   - `top`, `right`, `bottom`, and `left` properties do not work.
2. `relative`
   - The element is positioned relative to its normal position.
   - You can use `top`, `right`, `bottom`, and `left` to move it.
   - It still occupies space in the normal flow.
3. `absolute`
   - The element is removed from the normal document flow.
   - It is positioned relative to its nearest **positioned** ancestor (`relative`, `absolute`, or `fixed`).
   - If no positioned ancestor exists, it positions itself relative to the `<html>` element.
4. `fixed`
   - Similar to `absolute`, but the element is positioned relative to the **viewport** (browser window).
   - It does not move when scrolling the page.
5. `sticky`
   - The element toggles between `relative` and `fixed`, depending on the scroll position.
   - It remains fixed **only when it reaches a specified scroll position** and behaves like `relative` otherwise.

2. How many types of positioning are there in CSS ?

Ans. There are **five** types of positioning in CSS:

1. `static` **(Default Positioning)**
2. `relative` **(Relative Positioning)**
3. `absolute` **(Absolute Positioning)**
4. `fixed` **(Fixed Positioning)**
5. `sticky` **(Sticky Positioning)**

3. What is Z-index and why to use it?

Ans. The `z-index` property in CSS controls the **stacking order** of overlapping elements. It determines which element appears **in front of or behind** another when elements overlap.

How `z-index` Works?

- `z-index` only works on elements that have a **position other than `static`** (i.e., `relative`, `absolute`, `fixed`, or `sticky`).
- Higher `z-index` values place elements **closer to the viewer** (on top), while lower values push elements **further back**.

Why Use `z-index`?

- To **control the visibility** of overlapping elements.
- To ensure **important elements (modals, popups, dropdowns, tooltips, etc.)** appear on top of other content.
- To fix layering issues when multiple elements overlap.