**Mobile Responsive and Positions**

**29Jan**

1. **What is the Position property in CSS and its type?**

Ans- **The position Property**

The position property specifies the type of positioning method used for an element.

There are five different position values:

* static
* relative
* fixed
* absolute
* sticky

Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the position value.

1. **How many types of positioning are there in CSS?**

Ans- There are five types of positioning in CSS:

* Static
* Relative
* Fixed
* Absolute
* Sticky

**position: static;**

HTML elements are positioned static by default.

Static positioned elements are not affected by the top, bottom, left, and right properties.

An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:

Example:

**div {**

**position: static;**

**}**

**position: 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.

Example:

div {

position: relative;

top: 20px;

left: 50px;

}

In this example, the **div** element will be positioned relative to its normal position, and will be moved 20 pixels down from its original position and 50 pixels to the right.

**position: 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.

Example:

div {

position: fixed;

top: 0;

left: 0;

}

In this example, the **div** element will be positioned at the top-left corner of the viewport, and will remain in the same position even if the page is scrolled.

**position: absolute;**

An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

If no positioned ancestor is found, it is positioned relative to the initial containing block.

The top, bottom, left, and right properties can be used to adjust the position of the element.

Example:

div {

position: absolute;

top: 100px;

left: 200px;

}

In this example, the div element will be positioned 100 pixels down from the top of its nearest positioned ancestor, and 200 pixels to the right.

**position: sticky;**

An element with position: sticky; is positioned based on the user's scroll position.

A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).

Example:

div {

position: sticky;

top: 50px;

}

In this example, the div element will behave as relative positioning until it reaches a threshold (when it reaches 50 pixels from the top of the viewport), at which point it becomes fixed positioning, and will remain in that position even if the page is scrolled further.

1. **What is Z-Index and why to use it?**

Ans- z-index is a CSS property that controls the vertical stacking order of elements on a web page. It determines which elements will appear in front of or behind other elements, based on the element's z-index value.

Elements with a higher z-index value will appear in front of elements with a lower z-index value. If two elements have the same z-index value, the one that appears later in the HTML source will be displayed on top.

z-index is useful for creating a visual hierarchy on a web page, allowing you to control which elements appear on top of others. For example, you may want to ensure that a navigation bar or a header element always appears on top of other elements on the page, even as the user scrolls down.

Here's an example of how to use z-index in CSS:

Example

**Set the z-index for an image:**

**img {  
  position: absolute;  
  left: 0px;  
  top: 0px;  
  z-index: -1;  
}**

Because the image has a z-index of -1, it will be placed behind the heading