**Mobile Responsive and Positions**

**28 Jan**

1. **What are media queries in CSS and their use case?**

Ans- The Media query in CSS is used to create a responsive web design. It means that the view of a web page differs from system to system based on screen or media types. The breakpoint specifies for what device-width size, the content is just starting to break or deform.

Media queries in CSS allow you to apply different styles to a web page based on various factors such as screen size, device type, orientation, and resolution. They enable you to create responsive web designs that adapt to the user's device and provide the optimal viewing experience.

Media queries are written as a set of rules that define the styles to be applied when certain conditions are met. For example, you could use a media query to apply a different layout or font size when the screen size is smaller than a certain value.

Here's an example of a media query in CSS:

**@media only screen and (max-width: 600px) {**

**/\* Styles to be applied for screens smaller than 600px \*/**

**}**

In this example, the styles within the curly braces will only be applied when the screen size is smaller than 600px.

1. **Difference between min-width and max-width in media queries in CSS?**

**Ans-** In CSS media queries, min-width and max-width are used to define the range of screen sizes that a particular set of styles that should be applied to.

min-width is a media query property that specifies the minimum screen width at which the styles should be applied. For example, if you set min-width: 600px, the styles within that media query will only apply to screens that are 600 pixels wide or wider.

Here's an example of a media query using min-width:

**@media only screen and (min-width: 600px) {**

/\* Styles to be applied for screens 600px wide or wider \*/

**}**

max-width, on the other hand, specifies the maximum screen width at which the styles should be applied. For example, if you set max-width: 768px, the styles within that media query will only apply to screens that are 768 pixels wide or narrower.

Here's an example of a media query using max-width:

**@media only screen and (max-width: 768px) {**

**/\* Styles to be applied for screens 768px wide or narrower \*/**

**}**

1. **Explain float property and clear property in CSS with examples?**

Ans- **The float Property**

The float property is used for positioning and formatting content e.g. let an image float left to the text in a container.

The float property can have one of the following values:

* left - The element floats to the left of its container
* right - The element floats to the right of its container
* none - The element does not float (will be displayed just where it occurs in the text). This is default
* inherit - The element inherits the float value of its parent

In its simplest use, the float property can be used to wrap text around images.

*Example*

the image will float to the right in the paragraph

**img {**

**float: right;**

**}**

**The clear Property**

When we use the float property, and we want the next element below (not on right or left), we will have to use the clear property.

The clear property specifies what should happen with the element that is next to a floating element.

The clear property can have one of the following values:

* none - The element is not pushed below left or right floated elements. This is default
* left - The element is pushed below left floated elements
* right - The element is pushed below right floated elements
* both - The element is pushed below both left and right floated elements
* inherit - The element inherits the clear value from its parent

When clearing floats, you should match the clear to the float: If an element is floated to the left, then you should clear to the left. Your floated element will continue to float, but the cleared element will appear below it on the web page.

*Example*

This example clears the float to the left. Here, it means that the <div2> element is pushed below the left floated <div1> element:

**div1 {  
  float: left;  
}**

**div2 {  
  clear: left;  
}**