Assignment

Topic: Mobile responsive and Positions

Q1. What are media queries in CSS and their use case?

Ans: In CSS, a media query is a technique that allows you to apply specific styles based on certain conditions related to the user's device, like screen size, orientation, resolution, or device type, essentially enabling responsive design by tailoring the layout and appearance of a website depending on the viewing environment; the primary use case is to create websites that adapt seamlessly across different devices and screen sizes, ensuring optimal viewing experiences on everything from large desktops to small mobile phones.

Key points about media queries:

- **Functionality:**

Media queries use the @media rule in CSS to define conditions and apply styles only when those conditions are met.

- **Common media features:**
    - **Viewport size:** max-width, min-width, max-height, min-height
    - **Orientation:** landscape, portrait
    - **Device type:** screen, print, speech

- **Example usage:**
    - **Responsive layout:** Display a navigation menu horizontally on large screens and vertically on small screens.
    - **Font size adjustments:** Increase font size for smaller screens to improve readability
    - **Hiding elements:** Hide certain elements on smaller screens if they are not necessary

Q2. Difference between min-width and max-width in media queries in CSS.

Ans: In CSS, min-width and max-width are used in media queries to apply styles based on the width of the viewport or device. They serve different purposes:

1. min-width

- Definition: Applies styles when the viewport width is greater than or equal to the specified value.
- Use Case: Used for mobile-first design, where you start with styles for smaller screens and add styles for larger screens as the viewport width increases.
- Example:

```
@media (min-width: 768px) {
  /* Styles for screens wider than or equal to 768px */
  body {
    background-color: lightblue;
  }
```

}

This rule applies when the viewport width is 768px or wider.

---

2. max-width

- Definition: Applies styles when the viewport width is less than or equal to the specified value.
- Use Case: Used for desktop-first design, where you start with styles for larger screens and adjust them for smaller screens as the viewport width decreases.
- Example:

```
@media (max-width: 767px) {
  /* Styles for screens narrower than or equal to 767px */
  body {
    background-color: lightcoral;
  }
}
```

This rule applies when the viewport width is 767px or narrower.

Key Differences:

Property	Applies When	Common Use Case
min-width	Viewport width ≥ specified value	Mobile-first design
max-width	Viewport width ≤ specified value	Desktop-first design

Q3. Explain float property and clear property in CSS with examples.

Ans: 1. float Property

The float property is used to position an element to the left or right of its container, allowing text and inline elements to wrap around it. It is commonly used for creating layouts where elements are aligned side by side.

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 (default).
- inherit: The element inherits the float value from its parent.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Float Example</title>
  <style>
    .float-left {
      float: left;
      width: 100px;
      height: 100px;
      background-color: lightblue;
```

```
            margin-right: 10px;
        }

        .float-right {
            float: right;
            width: 100px;
            height: 100px;
            background-color: lightcoral;
            margin-left: 10px;
        }

        .clear {
            clear: both;
        }
    </style>
</head>
<body>
    <div class="float-left"></div>
    <p>This text wraps around the left-floated div. The div is floated to the left, and the text flows around it on the right side.</p>

    <div class="float-right"></div>
    <p>This text wraps around the right-floated div. The div is floated to the right, and the text flows around it on the left side.</p>

    <div class="clear"></div>
```

```
    <p>This paragraph is placed after clearing the floats, so it will not wrap around the floated elements.</p>

</body>

</html>
```

2. clear Property

The clear property is used to control the behavior of elements that follow floated elements. It specifies whether an element should be moved below (cleared) any preceding floated elements.

Values:

- none: The element is not moved down to clear past floats (default).
- left: The element is moved down to clear past left floats.
- right: The element is moved down to clear past right floats.
- both: The element is moved down to clear past both left and right floats.
- inherit: The element inherits the clear value from its parent.

Example:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Clear Example</title>

    <style>

        .float-left {
```

```
    float: left;
    width: 100px;
    height: 100px;
    background-color: lightblue;
    margin-right: 10px;
}

.float-right {
    float: right;
    width: 100px;
    height: 100px;
    background-color: lightcoral;
    margin-left: 10px;
}

.clear-left {
    clear: left;
}

.clear-right {
    clear: right;
}

.clear-both {
    clear: both;
}
```

```
    </style>
</head>
<body>
    <div class="float-left"></div>
    <p>This text wraps around the left-floated div.</p>
    <p class="clear-left">This paragraph clears the left float, so it will not wrap around the left-floated div.</p>

    <div class="float-right"></div>
    <p>This text wraps around the right-floated div.</p>
    <p class="clear-right">This paragraph clears the right float, so it will not wrap around the right-floated div.</p>

    <div class="float-left"></div>
    <div class="float-right"></div>
    <p>This text wraps around both floated divs.</p>
    <p class="clear-both">This paragraph clears both floats, so it will not wrap around any floated elements.</p>
</body>
</html>
```