

Here are comprehensive notes on media queries in Markdown format:

## # Media Queries in CSS

### ## Introduction

Media queries are a CSS technique that allows you to apply specific styles to your web pages based on the characteristics of the user's device, such as screen size, orientation, and resolution. They are essential for creating responsive web designs that adapt to different devices.

### ## Basic Syntax

```
```css
@media only screen and (max-width: 768px) {
  /* CSS styles go here */
}
```

- **@media**: Starts the media query.
- **only screen**: Specifies the media type (e.g., screen, print).
- **(max-width: 768px)**: The condition for applying the styles. Here, styles are applied if the screen width is 768 pixels or less.

## Common Use Cases

### 1. Responsive Layouts

Adjusting layouts for different screen sizes (e.g., mobile, tablet, desktop).

```
/* Mobile Styles */
@media only screen and (max-width: 600px) {
  body {
    background-color: lightblue;
  }

  .container {
    flex-direction: column;
  }
}

/* Tablet Styles */
@media only screen and (min-width: 601px) and (max-width: 768px) {
  body {
    background-color: lightgreen;
  }

  .container {
    flex-direction: row;
  }
}
```

```

    }
}

/* Desktop Styles */
@media only screen and (min-width: 769px) {
    body {
        background-color: lightyellow;
    }

    .container {
        flex-direction: row;
    }
}

```

## 2. Hiding/Showing Elements

Hiding navigation menus on small screens and showing a hamburger icon instead.

```

/* Hide the menu on small screens */
@media only screen and (max-width: 600px) {
    .menu {
        display: none;
    }

    .hamburger-icon {
        display: block;
    }
}

/* Show the menu on larger screens */
@media only screen and (min-width: 601px) {
    .menu {
        display: block;
    }

    .hamburger-icon {
        display: none;
    }
}

```

## 3. Changing Font Sizes

Making text larger on smaller screens for better readability.

```

@media only screen and (max-width: 768px) {
    body {
        font-size: 18px;
    }
}

```

```
}

@media only screen and (min-width: 769px) {
  body {
    font-size: 16px;
  }
}
```

## Using Breakpoints

A breakpoint is a specific point where your design changes based on screen size. Here are some commonly used breakpoints:

- **Small devices (phones):** 600px and below
- **Medium devices (tablets):** 601px to 768px
- **Large devices (desktops):** 769px and above

You can define your own breakpoints based on your design requirements.

## Example: Responsive Web Page

Below is an example of a simple responsive web page using media queries.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Responsive Web Design</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }

    .container {
      display: flex;
      flex-direction: row;
      justify-content: space-between;
      padding: 20px;
    }

    .box {
      width: 30%;
      padding: 20px;
      background-color: lightcoral;
      text-align: center;
      color: white;
      border-radius: 8px;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="box">
      <h1>Hello World</h1>
    </div>
  </div>
</body>
</html>
```

```

    }

    /* Mobile Styles */
    @media only screen and (max-width: 600px) {
        .container {
            flex-direction: column;
        }

        .box {
            width: 100%;
            margin-bottom: 10px;
        }
    }

    /* Tablet Styles */
    @media only screen and (min-width: 601px) and (max-width: 768px) {
        .box {
            width: 45%;
        }
    }
}
</style>
</head>
<body>
    <div class="container">
        <div class="box">Box 1</div>
        <div class="box">Box 2</div>
        <div class="box">Box 3</div>
    </div>
</body>
</html>

```

Explanation:

- **Default Layout:** On screens wider than 768px, the boxes are laid out side by side.
- **Tablet Layout:** On screens between 601px and 768px, the boxes are still side by side but take up more space.
- **Mobile Layout:** On screens 600px wide and below, the boxes stack vertically, taking up 100% of the width.

## Testing Media Queries

You can use your browser's developer tools (like Chrome DevTools) to simulate different device sizes and see how your media queries affect the layout.

## Conclusion

Media queries are a powerful tool in responsive web design, allowing you to create layouts that work across a wide range of devices. By mastering media queries, you can ensure your websites are accessible and user-friendly on any screen size.

You can copy and paste this Markdown into any Markdown editor, like Visual Studio Code, or directly into GitHub, and it will render properly with the appropriate headers, code blocks, and formatting.