CSS Media Queries

CSS Media Queries are a powerful tool in responsive web design. They allow you to apply different styles to different devices or screen sizes, ensuring that your website or web application looks good and is usable on various screen sizes (like desktops, tablets, and smartphones).

Basic Syntax

A media query consists of a media type (like `screen`, `print`, etc.) and one or more expressions that check for conditions like screen width, resolution, or orientation. Depending on whether the condition is true or false, certain styles will be applied.

Here is a basic syntax example of a media query:

```
@media (condition) {
    /* CSS rules */
}
```

Example: Targeting Screen Widths

To target specific screen widths, you can use conditions like `max-width` or `min-width`.

```
/* Styles for devices with a screen width of 600px or less */
@media (max-width: 600px) {
    body {
        background-color: lightblue;
    }
}

/* Styles for devices with a screen width of 768px or greater */
@media (min-width: 768px) {
    body {
        background-color: lightgreen;
    }
}
```

In the example:

- Devices with a screen width of 600px or less will have a light blue background.
- Devices with a screen width of 768px or more will have a light green background.

Media Query Structure

Media queries can contain:

1. **Media Type**: Specifies the type of media (like `screen`, `print`, etc.).
2. **Logical Operators**: Combines multiple conditions using `and`, `not`, or `only`.
3. **Media Features**: Describes specific conditions like width, height, resolution, and more.

Media Types

Some common media types are:

- `screen`: Used for computer screens, tablets, smartphones, etc.
- `print`: Used when printing documents.
- `all`: Applies to all devices.

Example: Media Query with a Media Type

```
@media screen and (max-width: 800px) {
    /* Styles for screens with max width of 800px */
    body {
        font-size: 14px;
    }
}
```

This rule applies only to screens and ensures that if the screen width is 800px or smaller, the font size will be reduced.

Logical Operators

- **and**: Combines multiple conditions. All must be true for the styles to be applied.
- **not**: Excludes certain media types or features.
- **only**: Hides the styles from older browsers that don't support media queries.

Example with `and`:

```
@media screen and (min-width: 600px) and (max-width: 1024px) {
    /* Styles for screens between 600px and 1024px */
    .container {
        width: 80%;
    }
}
```

This query applies styles only when the screen width is between 600px and 1024px.

Example with `not`:

```
@media not print {
    /* Styles for anything except print media */
    body {
        color: black;
    }
}
```

Common Media Features

1. **Width and Height**: Useful for adjusting layouts for different screen sizes.
   - `max-width` and `min-width`: Sets maximum and minimum width for applying styles.
   - `max-height` and `min-height`: Sets maximum and minimum height for applying styles.

```
@media (max-width: 768px) {
    /* Styles for devices 768px wide or less */
    body {
        font-size: 12px;
    }
}
```

2. **Aspect Ratio**: Useful for adjusting styles based on the aspect ratio of the screen.
   - `aspect-ratio`: Checks the width-to-height ratio of the screen.

```
@media (min-aspect-ratio: 16/9) {
    /* Styles for screens with a 16:9 aspect ratio or greater */
    .video {
        height: auto;
    }
}
```

3. **Orientation**: Detects whether the device is in `portrait` or `landscape` mode.
   - `orientation: landscape`: The screen is wider than it is tall.
   - `orientation: portrait`: The screen is taller than it is wide.

```
@media (orientation: landscape) {
    /* Styles for landscape mode */
    .menu {
        display: none;
    }
}
```

4. **Resolution**: Targets high-DPI (dots per inch) or retina displays.

    - `min-resolution`: Specifies a minimum resolution (like 300dpi or 2dppx).

```
@media (min-resolution: 2dppx) {
    /* Styles for high-resolution displays */
    .image {
        background-image: url('high-res-image.png');
    }
}
```

Example: Creating a Responsive Layout

```
/* Base styles (for larger screens) */
.container {
    width: 1000px;
    margin: 0 auto;
}

/* For tablets (768px and below) */
@media (max-width: 768px) {
    .container {
        width: 90%;
    }
}

/* For smartphones (480px and below) */
@media (max-width: 480px) {
    .container {
        width: 100%;
        padding: 10px;
    }
}
```

This approach ensures the layout adjusts based on the device's screen width:

- On large screens, the container has a width of 1000px.
- On tablets, the width is reduced to 90%.
- On smartphones, the container takes the full width of the screen with some padding.

Conclusion

Media queries are essential for creating responsive websites. They allow you to apply styles based on different conditions like screen size, orientation, resolution, and more, ensuring your website adapts to various devices and screen sizes without breaking the design.