Responsive Web Design (RWD) and Media Queries (MQ)

Avinash Maskey

What is Responsive Design?

- When you are designing a website, it is really important that your content looks good on all screen sizes.
- **Responsive Design** is the practice of making sure your content looks good on all screen sizes.
- Everything in the website including layouts, fonts and images should automatically adapt to the user's device.
- In the early 2000's, developers focused on making sure their websites looked good on larger screen sizes like laptops and desktop computers.
- In today's world, you have to consider devices like mobile phones, tablets, and even watches.
- An important component of responsive design are **media queries.**

What is a Media Query?

- **CSS Media queries** are a way to target browser by certain characteristics, features, and user preferences, then apply styles or run other code based on those things.
- Perhaps the most common media queries in the world are those that target particular viewport ranges and apply custom styles, which birthed the whole idea of responsive design.
- There are lots of other things we can target ***beside viewport width***. That might be ***screen resolution***, ***device orientation***, ***operating system preference***, or even more among a whole bevy of things we can query and use to style content.

Anatomy of a Media Query

Basic Syntax of Media Query (1)

@media media-type (media-feature){

/*Styles go here*/

}

- **Let's break down what this syntax means.**
  - The **@media** is a type of **At-rule in CSS**. These rules will dictate what the CSS will look like based on certain conditions.
  - The media type refers to the ***category of media for the device.***
  - The different **media types** include all, print, screen and speech:
    - **all -** works for all devices
    - **print -** works for devices where the media is in print preview mode
    - **screen -** works for devices with screens
    - **speech -** works for devices like screen readers where the content is read out loud to the user

Basic Syntax of Media Query (2)

- **Media Features:**
  - Once we define the type of media we're trying to match, we can start defining what features we are trying to match it to.
  - **Features:**
    - ❖**width:** Defines the widths of the viewport. This can be a specific number (e.g. 400px) or a range (using min-width and max-width). **Values accepted are 'length'.**
    - ❖**height:** Defines the height of the viewport. This can be a specific number (e.g. 400px) or a range (using min-height and max-height). **Values accepted are 'length'.**
    - ❖**orientation:** The way the screen is oriented, such as tall (portrait) or wide (landscape) based on how the device is rotated. **Values accepted are 'portrait' and 'landscape'.**

Basic Syntax of Media Query (3)

- **Operators:**
  - **Media queries support logical operators** like many programming languages so that we can match media types based on certain conditions.
  - The ***@media rule is itself a logical operator*** that is basically stating that **"if"** the following types and features are matches, then do some stuff.
  - If you wanted to create more complex media queries, then you can use **logical operators like:**
    - ❖**and:** This operator is used to join multiple media features. If all of the media features are true then the styles inside the curly braces will be applied to the page.
    - ❖**not:** This operator reverses a true query into a false and a false query into a true.
    - ❖**, (comma):** This operator will separate multiple media features by commas and apply the styles inside the curly brace if one of the conditions is true.

Basic Syntax of Media Query (4)

and

But we can use the **and** operator if we want to target screens within a range of widths:

```css
/* Matches screen between 320px AND 768px */
@media screen (min-width: 320px) and (max-width: 768px) {
  .element {
    /* Styles! */
  }
}
```

Basic Syntax of Media Query (5)

or (or comma-separated)

We can also comma-separate features as a way of using an `or` operator to match different ones:

```css
/*
  Matches screens where either the user prefers dark mode or the screen is at least 1200px wide */
@media screen (prefers-color-scheme: dark), (min-width 1200px) {
  .element {
    /* Styles! */
  }
}
```

Basic Syntax of Media Query (6)

not

Perhaps we want to target devices by what they do **not** support or match. This declaration removes the body's background color when the device is a printer and can only show one color.

```css
@media print and ( not(color) ) {
  body {
    background-color: none;
  }
}
```

Media Queries Examples (1)

- In this example below, we want ***the background color to change to blue*** when ***the width of the device is 600px or less.***
- In the CSS, we want to add a (max-width: 600px) for the media query which tells the computer to target devices with a screen width of 600px and less.
- **Syntax:**

```
@media (max-width: 600px) {
  body {
    background-color: blue;
  }
}
```

Media Queries Examples (2)

- In this example below, we ***want to change the background color from blue to red*** if the device has a ***width between 600 and 768px.***
- We can use the and operator to accomplish this.
- **Syntax:**

```
@media (min-width: 600px) and (max-width: 768px) {
  body {
    background-color: red;
  }
}
```

Should you write separate media queries for every single device on the market?

- **The short answer to that question is no.**
- There are way too many devices out on the market to try to write a media query for each device.
- Technology is always changing which means new devices will always be coming out.
- It is more important ***that you target a range of devices using media queries.***
- Some **common breakpoints** used for media queries are:
  - **320px—480px:** Mobile devices
  - **481px—768px:** iPads, Tablets
  - **769px—1024px:** Small screens, laptops
  - **1025px—1200px:** Desktops, large screens
  - **1201px and more:** Extra large screens, TV

THANK YOU!