🎨 CSS Media Queries — Notes

1. What Are Media Queries?

Media queries allow your website to **adapt its styling** depending on the **device's screen size or features** (like width, height, orientation, etc.).
They're the backbone of **responsive web design** — the art of making websites behave properly on phones, tablets, laptops, and desktops.

Syntax:

```
@media (condition) {

  /* CSS rules here */

}
```

---

2. The Basic Idea

Media queries define **breakpoints** — specific screen widths where the layout or styling should change.

Example:

```
@media (max-width: 600px) {

  h1 {

    font-size: 1.5rem;

  }

}
```

📱 When the screen width is **600px or smaller**, the <h1> text becomes smaller — perfect for mobile.

---

3. Two Main Keywords: max-width and min-width

Keyword	Meaning	Targets
max-width	"Up to this width, apply these styles."	Smaller screens
min-width	"From this width and larger, apply these styles."	Bigger screens

Example:

```
@media (max-width: 600px) { ... }  /* Phones and small tablets */

@media (min-width: 600px) { ... }  /* Larger tablets and up */
```

Visualize it like this:

```
<---- smaller screens ---- 600px ---- larger screens ---->
```

↑ uses max-width rules ↑ uses min-width rules

---

4. Combining Breakpoints

You can combine both min-width and max-width to target **specific ranges**.

```
@media (min-width: 600px) and (max-width: 900px) {
 body {
  background-color: lightblue;
 }
}
```

🎯 Targets screens **between 600px and 900px** only.

---

5. Example: Responsive Box

```
div {
 width: 200px;
 height: 200px;
 background-color: blue;
}


@media (max-width: 600px) {
 div {
  width: 100px;
  height: 100px;
 }
}
```

When viewed on a small screen (≤600px), the box shrinks — efficiently adapting to available space.

---

6. Media Types

Media queries can target specific **output types**:

Type	Description
screen	Default — applies to digital displays

Type	Description
print	Applies when the page is printed
all	Applies to all media types

Example:

```
@media print {
  body {
    color: black;
    background: white;
  }
}
```

🖨️ Makes sure your website doesn’t waste ink or look awful when printed.

---

7. Common Device Breakpoints (Approximate)

Device Type	Screen Width Range	Example CSS
Mobile	319px – 480px	@media (max-width: 480px)
Tablet (iPad)	481px – 1200px	@media (min-width: 481px) and (max-width: 1200px)
Laptop	1201px – 1600px	@media (min-width: 1201px) and (max-width: 1600px)
Desktop	1601px and above	@media (min-width: 1601px)

---

8. Practical Example: Background Colors

```
body {
  background-color: aquamarine;
}

/* Mobile */
@media (max-width: 480px) {
  body {
    background-color: salmon;
  }
}
```

```
/* Tablet */
@media (min-width: 481px) and (max-width: 1200px) {
  body {
    background-color: powderblue;
  }
}

/* Laptop */
@media (min-width: 1201px) and (max-width: 1600px) {
  body {
    background-color: limegreen;
  }
}

/* Desktop */
@media (min-width: 1601px) {
  body {
    background-color: seagreen;
  }
}
```

✅ When you resize your browser, the background color changes — a visual cue that your media queries are working.

---

9. Testing Responsiveness

Use **Chrome Developer Tools**:

- Right-click → *Inspect* → Click the *device toolbar icon* (top-left).
- Choose different devices (iPhone, iPad, etc.).
- Adjust screen width to test your breakpoints.

Super handy for debugging responsive issues.

---

10. Summary Table

Concept	**Description**	**Example**
@media	Defines a conditional CSS rule	@media (max-width: 600px)
max-width	Targets screens **below or equal to** given width	@media (max-width: 480px)
min-width	Targets screens **greater than or equal to** given width	@media (min-width: 1200px)
Combined Query	Targets a **specific range**	@media (min-width: 600px) and (max-width: 900px)
screen	Applies styles to digital displays	@media screen and (max-width: 600px)
print	Applies when printing	@media print { ... }

11. Key Takeaways

- Media queries = responsive magic.
- Use max-width for mobile-first design (most common).
- Combine min + max for specific ranges.
- screen and print target device types.
- Use Chrome DevTools to test responsiveness.
- Keep it simple — you don’t need complex conditions for most projects.