**Media Queries:**

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.

Media queries allow you to apply CSS styles depending on a device's general type (such as print vs. screen) or other characteristics such as screen resolution or browser viewport width. Media queries are used for the following:

1. To conditionally apply styles with the CSS @media and @import at-rules.
2. To target specific media for the <style>, <link>, <source>, and other HTML elements with the media= attribute.
3. To test and monitor media states using the Window.matchMedia() and EventTarget.addEventListener() methods.

Basic Syntax:

@media media-type and (condition: breakpoint){

// Css rules

}

Example:

@media screen and (min-width: 600px) {

.element {

/\* Apply some styles \*/

}

}

Syntax - 2

@media media-type (media-feature){

/\*Styles go here\*/

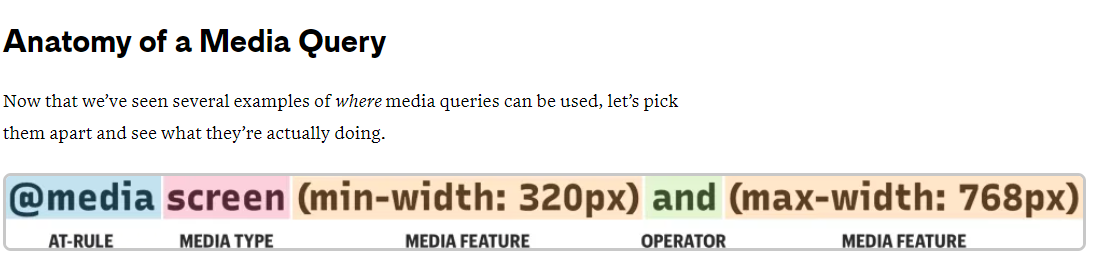
}

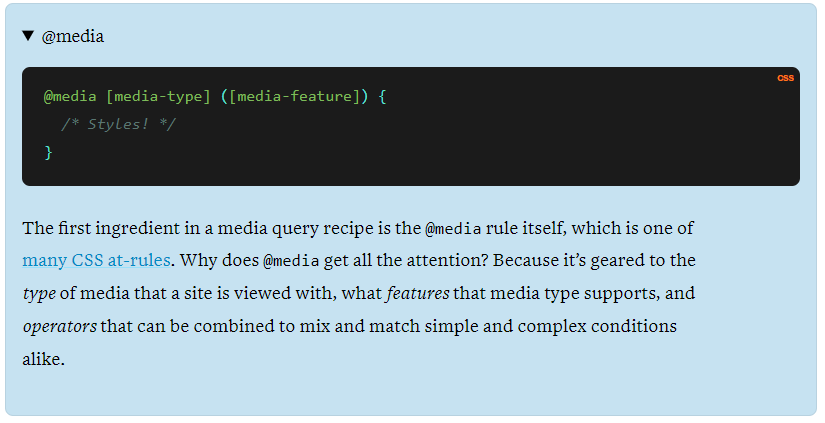
Let's break down what this syntax means.

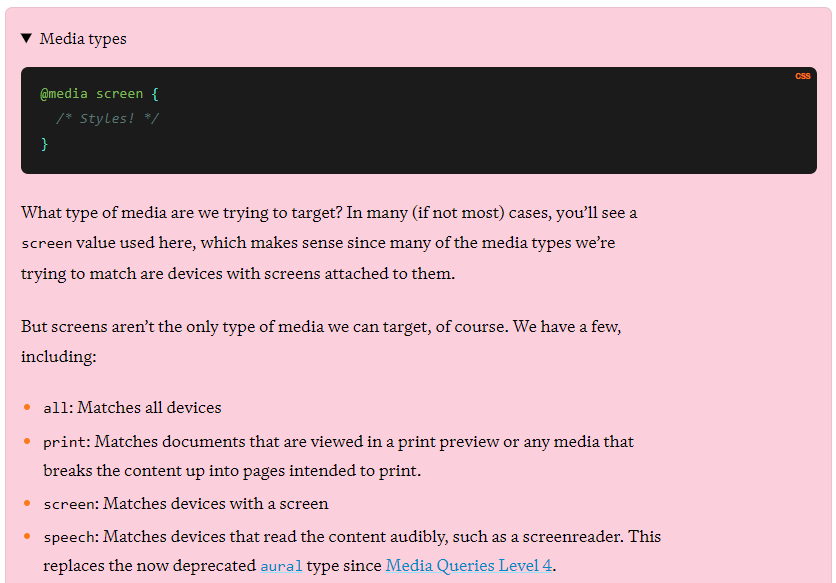
The @media is a type of At-rule in CSS. These rules will dictate what the CSS will look like based on certain conditions.

The media type refers to the category of media for the device. The different media types include all, print, screen and speech.

1. all - works for all devices
2. print - works for devices where the media is in print preview mode
3. screen - works for devices with screens
4. speech - works for devices like screen readers where the content is read out loud to the user







**Media features**

describe a specific characteristic of the user agent, output device, or environment:

1. any-hover
2. any-pointer
3. aspect-ratio
4. color
5. color-gamut
6. color-index
7. device-aspect-ratio Deprecated
8. device-height Deprecated
9. device-width Deprecated
10. display-mode
11. dynamic-range
12. forced-colors
13. grid
14. min & max height
15. hover
16. inverted-colors
17. monochrome
18. orientation
19. overflow-block
20. overflow-inline
21. pointer
22. prefers-color-scheme
23. prefers-contrast
24. prefers-reduced-motion
25. prefers-reduced-transparency Experimental
26. resolution
27. scripting
28. update
29. video-dynamic-range
30. min & max width.

**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.

**and**

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

*/\* Matches screen between 320px AND 768px \*/*

@media screen (min-width: 320px) and (max-width: 768px) {

.element {

*/\* Styles! \*/*

}

}

**or (or comma-separated)**

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

*/\**

*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! \*/*

}

}

**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.

@media print and ( not(color) ) {

body {

background-color: none;

}

}

@media (min-width: 600px) and (max-width: 768px) {

body {

background-color: #de3163;

}

}

**Common Media Query Breakpoints:**

We can add the breakpoint to see the screen width along with the width and height of the viewport for the different devices. A breakpoint is a point or key that determines when to change the layout by reshaping & adding new rules inside the media queries. There are some common breakpoints, not a standard resolution, that can be used for the different widths & heights of devices:

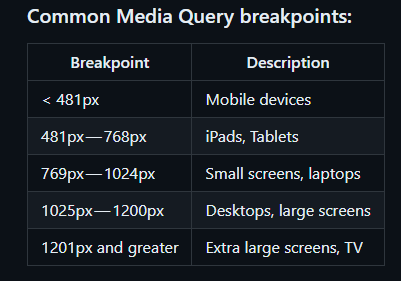
320px — 480px: Mobile devices

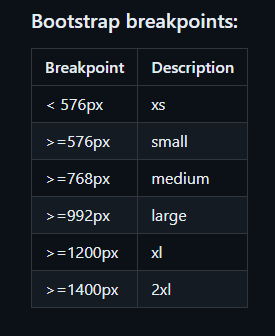
481px — 768px: iPads, Tablets

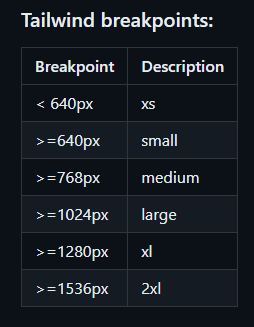
769px — 1024px: Small screens, laptops

1025px — 1200px: Desktops, large screens

1201px and more —  Extra large screens, TV







## **How Min- and Max-Width Queries Work**

How media queries function can be a bit confusing. Let’s take a look at the queries which are commonly used in email.

### **Max-width**

Here is an example of a max-width query.

@media only screen and (max-width: 600px) {...}

What this query really means, is “If [device width] is less than or equal to 600px, then do {…}”

So if the email is opened on an iPhone 5S with a screen width of 320px, the media query will trigger and all of the styles contained in { … } will take effect.

### **Min-width**

Here is an example of a min-width query.

@media only screen and (min-width: 600px) {...}

What this query really means, is “If [device width] is greater than or equal to 600px, then do {…}”

So if the email is opened on an iPhone 5S with a screen width of 320px, the media query *will not*trigger and the styles contained in { … } *will not*take effect.

### **Combining media query expressions**

Max-width and min-width can be used together to target a specific range of screen sizes.

@media only screen and (max-width: 600px) and (min-width: 400px) {...}

The query above will trigger only for screens that are 600-400px wide. This can be used to target specific devices with known widths.

[CSS Tricks has a list of standard device widths and the media queries to use](https://css-tricks.com/snippets/css/media-queries-for-standard-devices/).

## **Breakpoints**

Most media queries are set to trigger at certain screen widths or breakpoints. Exactly what these should be set to is a matter of some debate amongst email developers.

iPhones and iPads provide us with a few easy breakpoints to start from. Coding styles for these specific clients will ensure emails look great on these screens.

Android devices, on the other hand, vary widely in screen size because there are so many different manufacturers and devices. I recommend creating two to four breakpoints, based on popular Apple devices, which will cover most devices.

* iPhone 5S is an example of a **Breakpoint 1 with 320px**
* iPhone 6+ is an example of a **Breakpoint 2 with 414px**
* iPad Mini is an example of a **Breakpoint 3 with 703px**
* iPad Air is an example of a **Breakpoint 4 with 768px**

Breakpoints 3 and 4 are optional, as most emails will look fine showing the desktop version on an iPad or large tablet. You could even get away with using just breakpoint 2, if you code your container tables to expand to 100% width (and not a set width, which may or may not match the device well).