

Media Queries and Responsive Practices

CSS3



Mobile Websites versus Responsive Websites

If you visit a website like Facebook or Twitter from most mobile devices, watch the address bar carefully. Notice that the URL redirects from “facebook.com” or “twitter.com” to “m.facebook.com” or “mobile.twitter.com”.

What this tells us, is that there are two different websites in these cases. One intended specifically for the desktop experience, and one specifically for a more mobile or tablet experience. Much of the web does not use this practice, as it can mean a lot of work to support two websites, and it can cause confusion for users if there are differences in features. The potential pros lie mostly in optimizations that can be made to improve performance and load-times for mobile devices.

The alternative to mobile websites, are [responsive](#) websites. Most websites you visit likely use this approach. If you visit a website like ualberta.ca and resize your browser, you’ll notice that items in the page move or change size to accommodate the new browser viewport size. Visiting from your phone will not result in a redirect to another site—instead the CSS is written carefully to support many screen sizes.

As more and more users frequent their phone as their primary browsing device, [mobile-first](#) responsive has become a best practice.

Thinking Mobile–First Responsive

To write responsive code, usually a developer tries to think in device width or height “breakpoints.”

Start with mobile, and consider the layout you’re trying to achieve. From there, work your way up. Make note of which device widths may look odd, be hard to read, or break the layout. Perhaps once the design is brought to the width of a tablet, you can make use of more horizontal space and lay things out a bit differently.

Work your way up again. Perhaps at around the size of the average laptop monitor you notice things start looking a little strange again. At this point, you may want to adjust things again.

How does this thinking translate to code?

Glad you asked! In CSS, we can write [media queries](#). Media queries are blocks of CSS rules that will only apply if the query rings true.

For example, we can ask the browser, “is this browser viewport at least 600px wide?” If the answer is yes, the block of rules will then kick into effect! As soon as the answer is no, those rules will stop applying.

The question we posed above would look something like so...

```
@media screen and ( min-width: 600px ) { /* Regular selectors and rules here. */ }
```

You may be wondering what sort of widths are common in modern devices. Thankfully, CSS Tricks has done the groundwork on this for us. See their article, “[Media Queries for Standard Devices](#).”

Testing Breakpoints

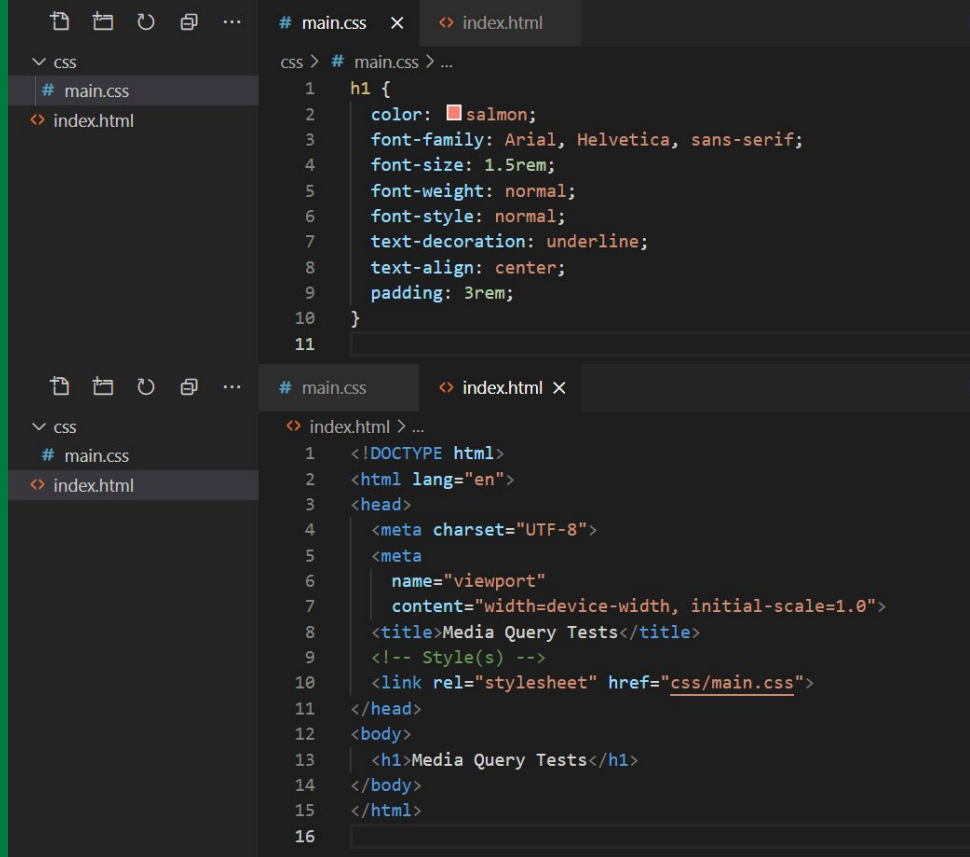
In major browsers' developer tools, there are usually options for emulating various devices and screen-sizes.

See the following guides to understand your browser of choice:

- [Google Chrome](#)
- [Microsoft Edge](#)
- Apple Safari ([Lifewire Article](#))
- [Mozilla Firefox](#)

Mobile-First Styles

Let's start a small new website to run a few tests!



The screenshot shows a code editor with two files open: `main.css` and `index.html`. The `main.css` file contains a single `h1` selector with the following styles:

```
1 h1 {  
2   color: salmon;  
3   font-family: Arial, Helvetica, sans-serif;  
4   font-size: 1.5rem;  
5   font-weight: normal;  
6   font-style: normal;  
7   text-decoration: underline;  
8   text-align: center;  
9   padding: 3rem;  
10 }  
11
```

The `index.html` file contains the following HTML structure:

```
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head>  
4   <meta charset="UTF-8">  
5   <meta  
6     name="viewport"  
7     content="width=device-width, initial-scale=1.0">  
8   <title>Media Query Tests</title>  
9   <!-- Style(s) -->  
10  <link rel="stylesheet" href="css/main.css">  
11 </head>  
12 <body>  
13   <h1>Media Query Tests</h1>  
14 </body>  
15 </html>  
16
```

Media Query Tests

Let Mobile Styles Act As Your Base

Notice the way we're starting out in this example, writing styles that are not inside of any media query blocks.

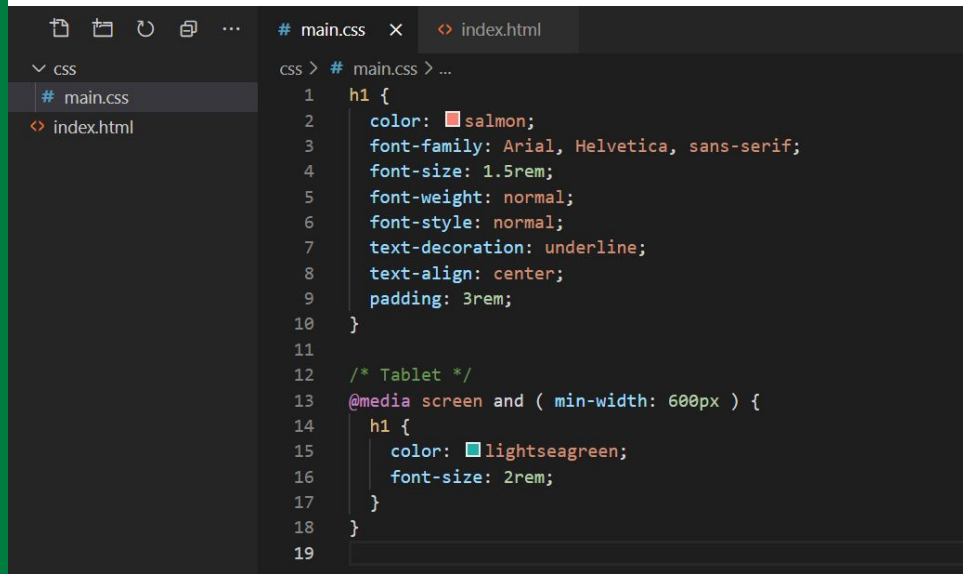
Using these base global styles is a common and effective approach to writing mobile-first styles. If you write your queries after this, you can use the cascade to comfortably override only those styles if or when necessary.

Note that styles that were not overridden, like the font and "text-decoration: underline;", were carried over into the subsequent browser viewport size.

Adding a Media Query

Let's add a breakpoint targeting a viewport of 600px or wider.

Use your browser's responsive / emulation tool to compare the H1 appearance on smaller and larger displays.

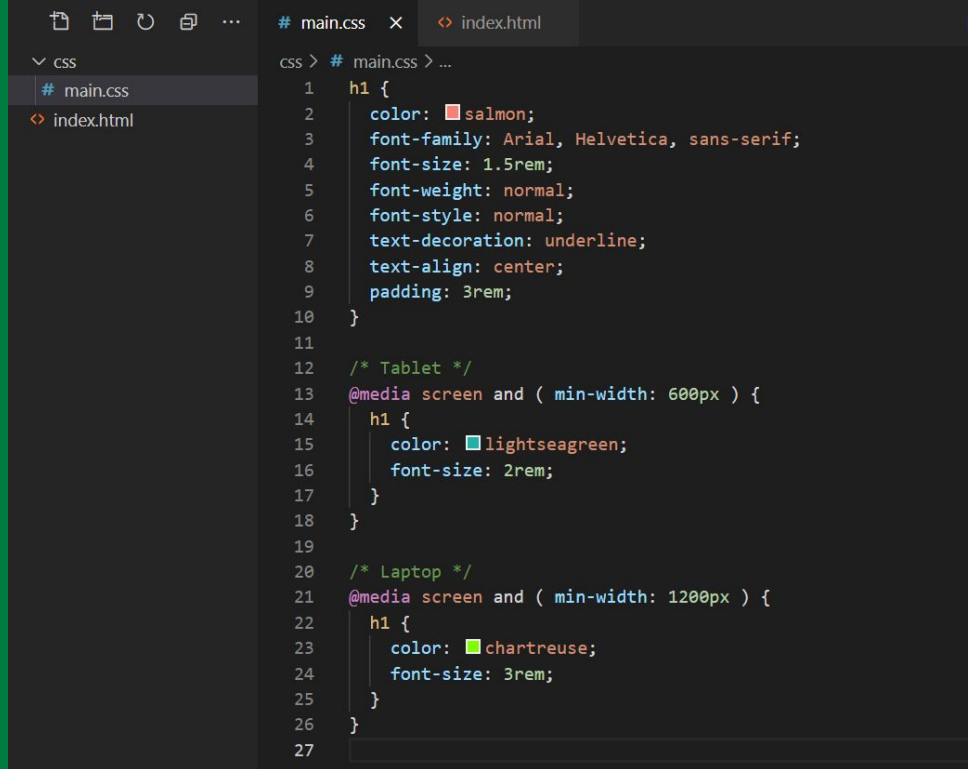
A screenshot of a code editor with a dark theme. The left sidebar shows a file explorer with 'css' expanded, containing 'main.css' and 'index.html'. The main editor area shows the 'main.css' file with the following CSS code:

```
1 h1 {  
2   color: salmon;  
3   font-family: Arial, Helvetica, sans-serif;  
4   font-size: 1.5rem;  
5   font-weight: normal;  
6   font-style: normal;  
7   text-decoration: underline;  
8   text-align: center;  
9   padding: 3rem;  
10 }  
11  
12 /* Tablet */  
13 @media screen and ( min-width: 600px ) {  
14   h1 {  
15     color: lightseagreen;  
16     font-size: 2rem;  
17   }  
18 }  
19
```

Media Query Tests

Let's Try Another

Again, use your browser's responsive tools to find sizes that trigger each of the three h1 styles!

A screenshot of a code editor with a dark theme. The left sidebar shows a file explorer with 'css' expanded, containing 'main.css' and 'index.html'. The main editor area shows the content of 'main.css'. The code defines three styles for the 'h1' element using media queries. The first style is the default, with salmon color, 1.5rem font size, and underline. The second style, labeled 'Tablet', has lightseagreen color and 2rem font size. The third style, labeled 'Laptop', has chartreuse color and 3rem font size. Line numbers 1 through 27 are visible on the left of the code.

```
# main.css
index.html

css > # main.css > ...
1  h1 {
2      color: salmon;
3      font-family: Arial, Helvetica, sans-serif;
4      font-size: 1.5rem;
5      font-weight: normal;
6      font-style: normal;
7      text-decoration: underline;
8      text-align: center;
9      padding: 3rem;
10 }
11
12 /* Tablet */
13 @media screen and ( min-width: 600px ) {
14     h1 {
15         color: lightseagreen;
16         font-size: 2rem;
17     }
18 }
19
20 /* Laptop */
21 @media screen and ( min-width: 1200px ) {
22     h1 {
23         color: chartreuse;
24         font-size: 3rem;
25     }
26 }
27
```

Media Query Tests

Recommended Reading

Look into more image-related content here:

- [Meyer, E. A; Weyl, E. \(October 2017\). CSS: The Definitive Guide, 4th Edition. O'Reilly Media, Inc.](#)
 - [Chapter 1. CSS and Documents](#)
 - [Media Queries](#)
 - [Chapter 20. Media-Dependent Styles](#)