

Making your site accessible

Step 1 – View your **interface** in the **browser** of your choice

Example: I would suggest opening up your interface in the **Chrome** browser

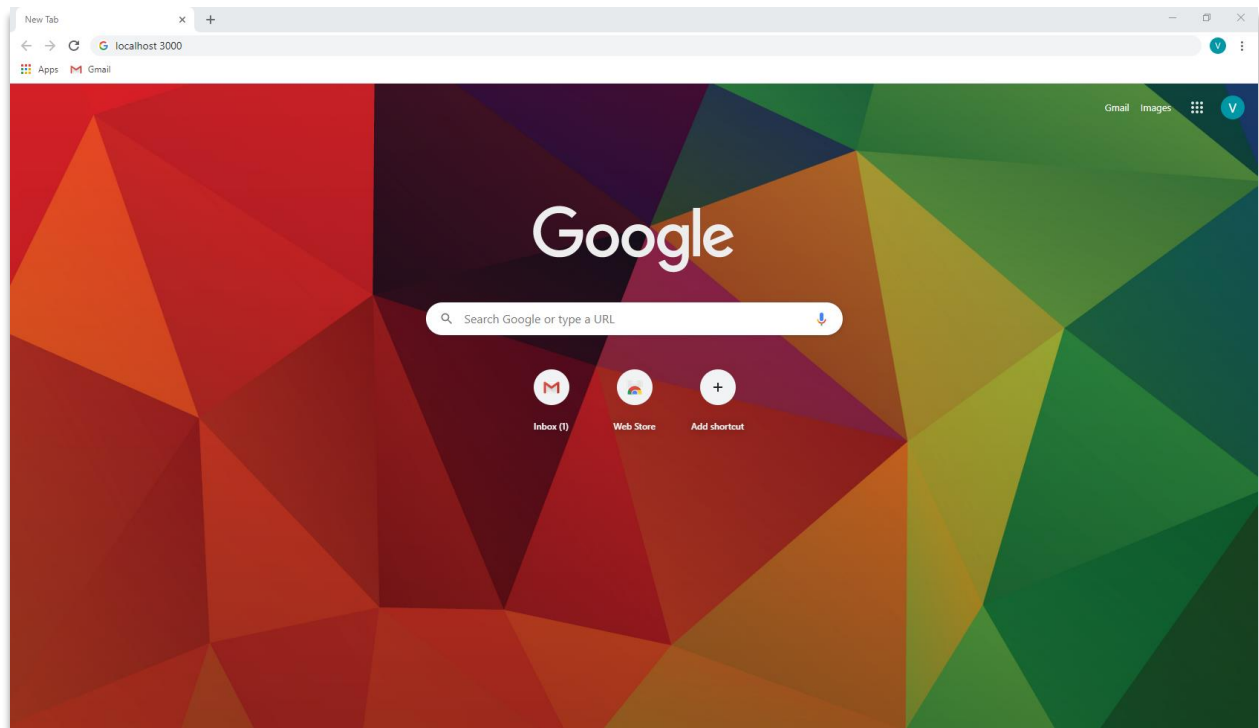


Figure 1 - Open a browser of your choice

Step 2 – Adjust the **viewport** to see how the **elements** move on the page

There are two commonly used methods for you to adjust your viewport represented by A (**manually adjusting the window**) or B (**using an inspector tool**)

Step A.2.1 - Select the **Restore Down** button in the top right corner of the page

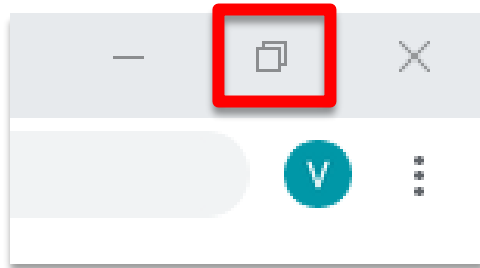


Figure 2 - Select the Restore Down button

Step A.2.2 – Now that you have access to sides of the window, you can use your mouse to resize the window via click and drag

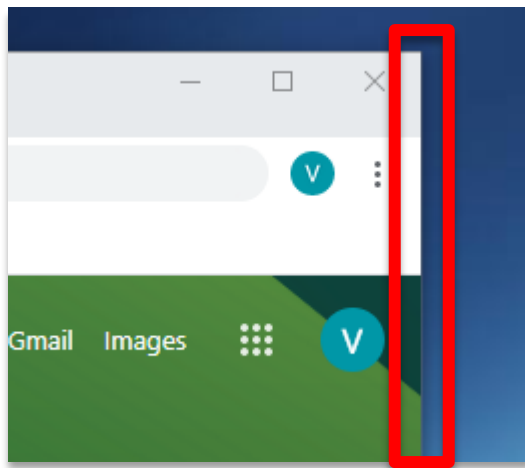


Figure 3 - Select the side of the window and use your mouse to resize the window

Step B.2.1 – Press the **F12** key or hold down “**ctrl + shift + i**” to activate the **inspector tool**

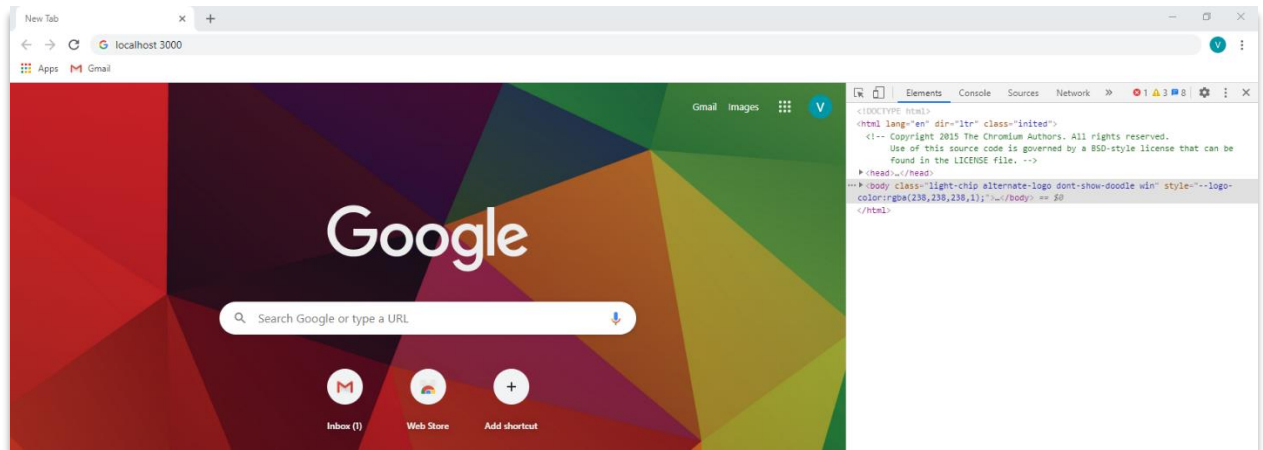


Figure 4 - Opening the inspector tool

Step B.2.2 – Use the inspector tool’s **Device Tool Bar Selection Tool** in the top left of the inspector tool

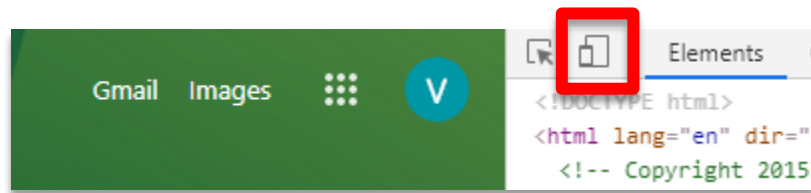


Figure 5 - Selecting the Device Tool Bar Selection Tool

Step B.2.3 – You chose a mode to change the **viewport** or just change it manually in the **Responsive** mode



Figure 6 - Changing the viewport size

Step 3.1 – Now, you want to **adjust** the **viewport** to multiple different sizes to make sure that the content is still **accessible** to the users

An example of this can be found in the CSS framework [Materialize's parallax example page](#). Notice how in Figure 7, when the page is at full size, there are three columns following the parallax image.

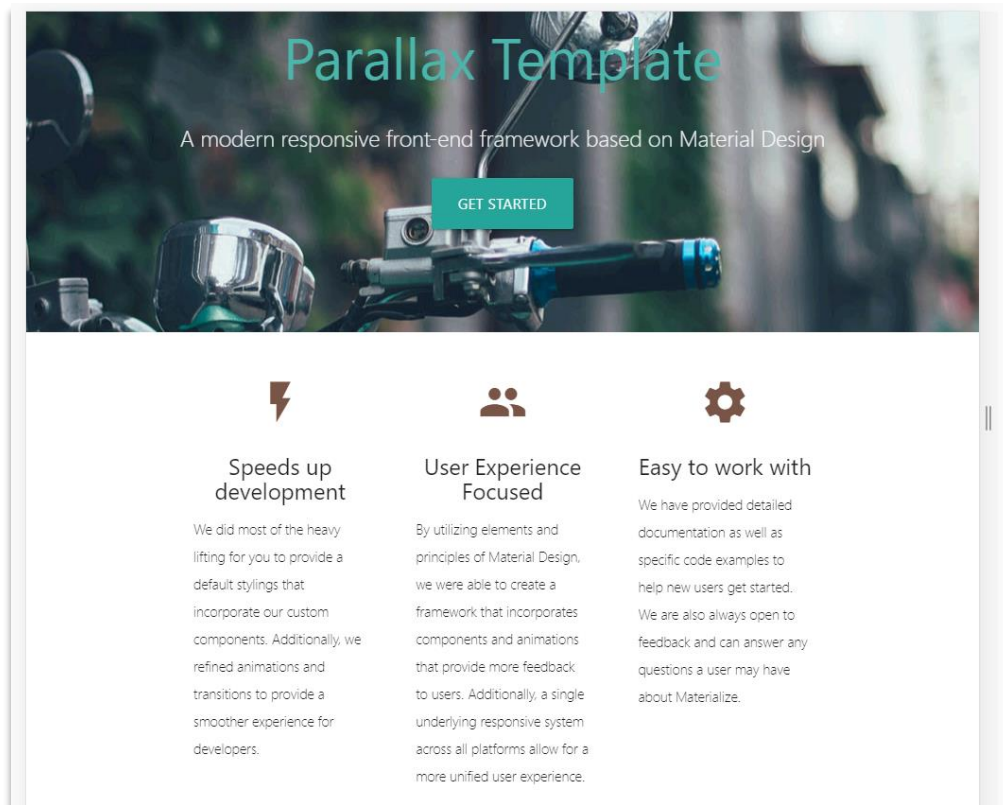


Figure 7 - Example page full size

In Figure 8 however, there was not enough space for the **columns** in a smaller viewport, so the content had to be converted into **rows** using media queries

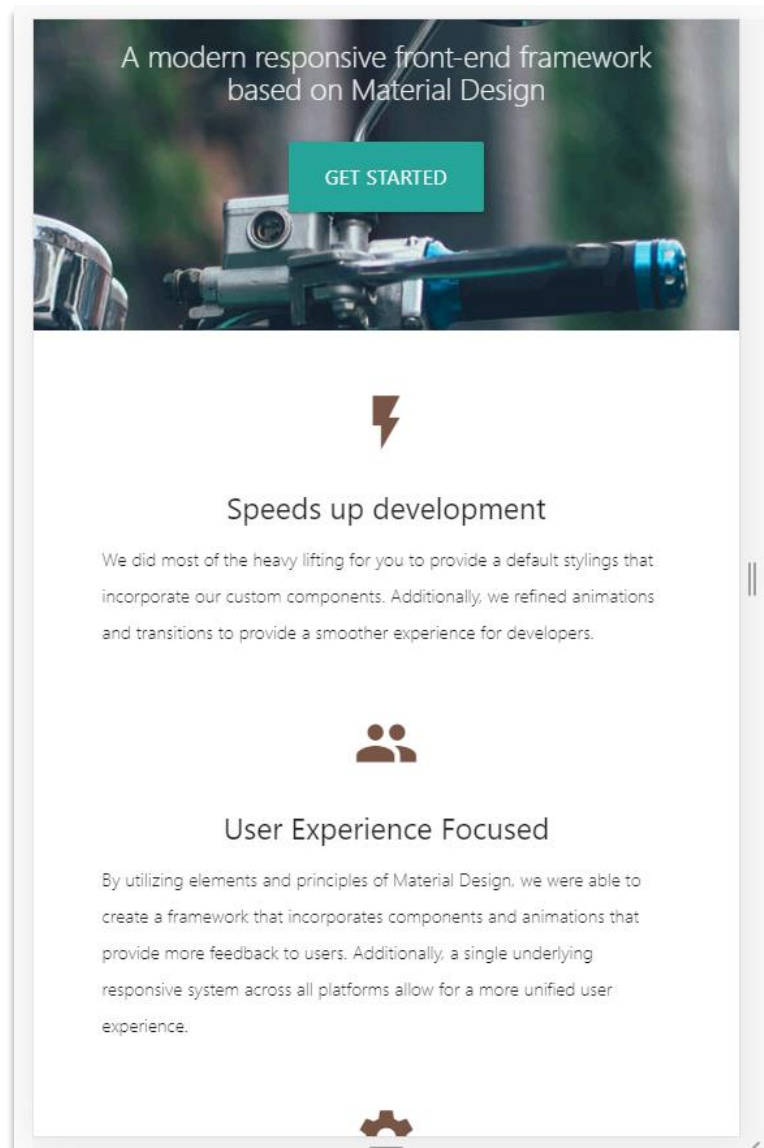


Figure 8 - Example page zoomed in

Step 3.2 – This is not the only location you have to be worried about content squishing into itself, you also want to be aware of the navigation

Take, for example, the top of the example site (Figure 9)



Figure 9 - Example site navigation not zoomed in

Here the **Navbar Link** is displayed as a website would typically display it, though usually there would be more than one link.



Figure 10 - Example site navigation zoomed in

In Figure 10, the viewport had been shrunk to accommodate for a mobile version where the content was not hidden nor was it removed, but simply aggregated to not have all the navbar links fighting for space at the top of the page.

Step 4 – If you notice that your content is not quite lining up or squishing too tightly together, it is time to break out the **media queries**

Referring to Figure 7, here is the media query making sure that the columns stay columns on viewports bigger than 601px.

```
@media only screen and (min-width: 601px)
  .row .col.m4 {
    width: 33.3333333333%;
    margin-left: auto;
    left: auto;
    right: auto;
  }
```

materialize.min.css:6

Figure 11 - Example page's column media query

Why would the media query be taking effect in the desktop view rather than the mobile view? Well, that is a little out of the scope of our **Reflow** guide, however, because you are special, I will tell you. We **design** our **media queries** like this because mobile phones do not currently have the same processing power of most computers. It is easier and faster for a computer to interpret the **CSS** than mobile devices. This essentially saves on the time it takes to load a page which can be a life or death matter for a website in this day and age.

It is also true that this will result in some structural changes, but this is done in order to make your interface more readable as well as comprehensible.

```
@media only screen and (max-width: 992px)
.hide-on-med-and-down materialize.min.css:6
{
  display: none !important;
}
```

Figure 12 - Example page's navigation media query

The **Navigation** was changed when the max-width was less than 992px because, in order to get the hamburger effect, there was actually a hidden unordered list tag ready to show itself under a similar media query. This, as with most things in our technical domain, is not what is required in every case, but was what happened in this case.

Step 5 – Repeat process until you have ensured that people can access all the content of your site with little to no structural changes