[https://www.skillshare.com](https://www.skillshare.com/)

**Responsive Design**

The idea is to design by considering the end user and not a specific device. The user can visit your site any time of the day and with any device which might be varying display size.

So, responsive design means that you are designing one flexible experience that is delivered to every device that might visit that website. So, it is one website that you are delivering to all your users.

This helps you to reach more people and more devices than ever before.

Flexible in the web is something that we should see as a strength than something which limits us.

Responsive design ingredients, they allows us to control the flexibility in the way that is useful to us.

1. Fluid Layout
2. Flexible Images
3. Media Queries

**FLUID LAYOUT**

Every Responsive design begins with Fluid or Flexible grid layout. A layout that is built with percentage and proportions rather than inflexible pixels.

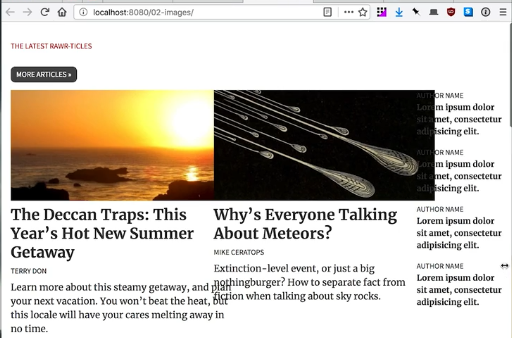
We can change the layouts at different breakpoints so the application will change its display with change in size of browser window.

This can be achieved by using CSS Grid layout and keeping column widths and column gaps in percentage or fractions.

**FLEXIBLE IMAGES**

Images: when they are displayed to the end user, Images actually have native dimensions they have a height and a width. As we drop them into our design without doing anything with them in CSS they are rendering at their native dimensions which does not helps when the browser display size changes (especially decreases).

*Problem Statement:* When browser size reduces == > Image size > size of its container element width and we want the image to not exceed the size of its container



We can control the presentation of our images with our CSS. Now if you want to have flexible images inside of your responsive design, there’s one very straight forward piece of CSS that you should know to make images work inside your flexible grids.

That is, max-width: 100%.

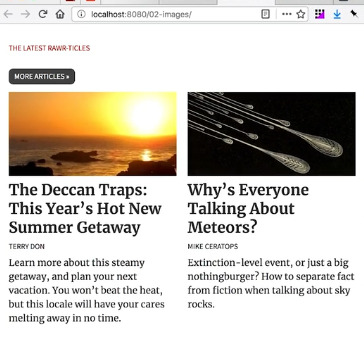
img {

max-width: 100%;

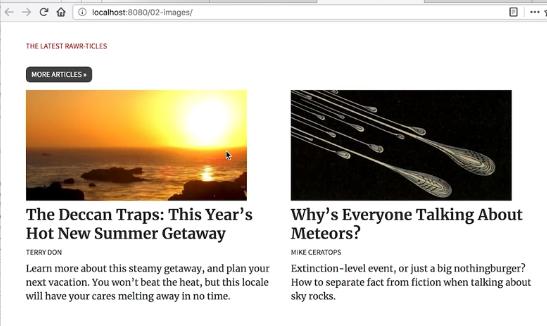
}

***This chunk of code says that width of image should never exceed the width of its container.***

In other words, an image can render at whatever size it wants but if the column size is smaller than the image, that image is going to resize proportionally and shrink up to fit inside of that column.



*Problem Statement:* When browser size Increases == > Image size < size of its container element width and we want the image to fit 100% the size of its container



Box highlighted in blue is the empty space left by the image.

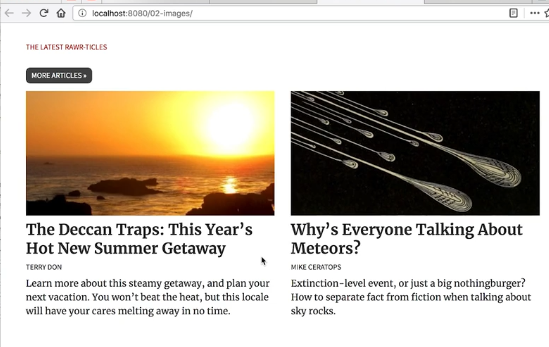
That is, width: 100%.

img {

width: 100%;

}

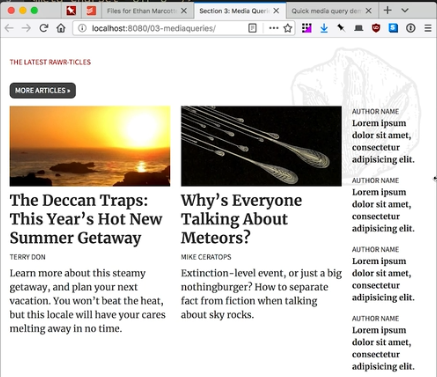
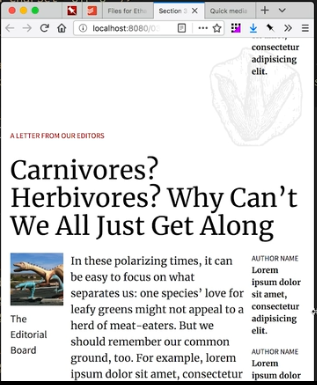
***This piece of code says that I always want image to match the width of its containing elements.***



***object-fit property is a property applied to inline images. I works like background-size: cover but for inline images.***

**MEDIA QUERIES**

Browser size is reduced to 50% and the look that we get is below,

This cannot be called a best user experience, as the text start wrapping dramatically and this layout is not really a pleasure to read anymore. The top headline looks heavy and prominent.

So we modify the layout to make somethings which doesn’t just fit but also feels at home on any size screen. This is where media queries come in.

Media query is like a question we are asking a browser if it meets certain characteristics, conditions or dimensions so we can apply style rules just for those circumstances.

body {

color: #000;

}

@media screen and (min-widht: 500px){

body {

font-size: 1.4rem;

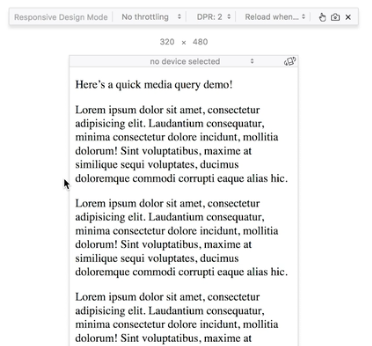
color: #F00;

}

}

What we are saying here is on the body element we want the text color to be black but when I have a screen width that is at least 500px wide, I want the color to be red and text to be bold.

Screen width < 500px:



Screen width > 500px:

