**Accessible & Responsive WebApps**

**Lesson 1 - Why\_Responsiveness**

1.1 Chrome web tools for mobile size emulation

1.2. Udacity Front End Developer chrome extension

1.3. Connect your PC to your android

**Lesson 2 – Starting Small**

2.1 DeviceIndependentPixel

2.2 Device Pixel ratio

2.3. Website can look different on identical screens due to

2.4 ViewPort Calculation

2.5 Set the ViewPort in meta tag

2.6 Define max-width all the time!

2.7 Tap Screen

2.8 You start from small to large!

2.9 Project Summary

**Lesson 3 Building Up**

3.1 Media Queries

3.2 Pick Up BreakPoints

3.3 Flex Boxes

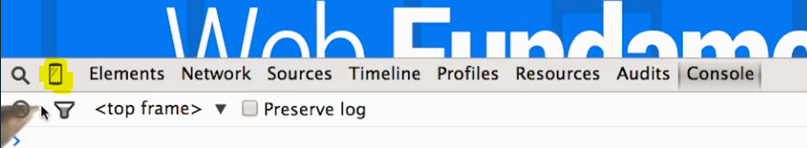
**Lesson 4 Common Responsive Patterns**

4.1 Patterns

4.2 Project Work:

**Lesson 1 - Why\_Responsiveness**

1.1 **Chrome web tools for mobile size emulation**



1.2. **Udacity Front End Developer chrome extension**

<https://chrome.google.com/webstore/detail/udacity-front-end-feedbac/melpgahbngpgnbhhccnopmlmpbmdaeoi>

1.3. **Connect your PC to your android** via USB to see your local page in your mobile

3.1 Enter developer mode in your android - > settings/aboutDevice/Build Number

Click on it seven times

3.2 Turn on your USB Debugging

3.3 PC – Chrome Canary, Mobile – Chrome Beta

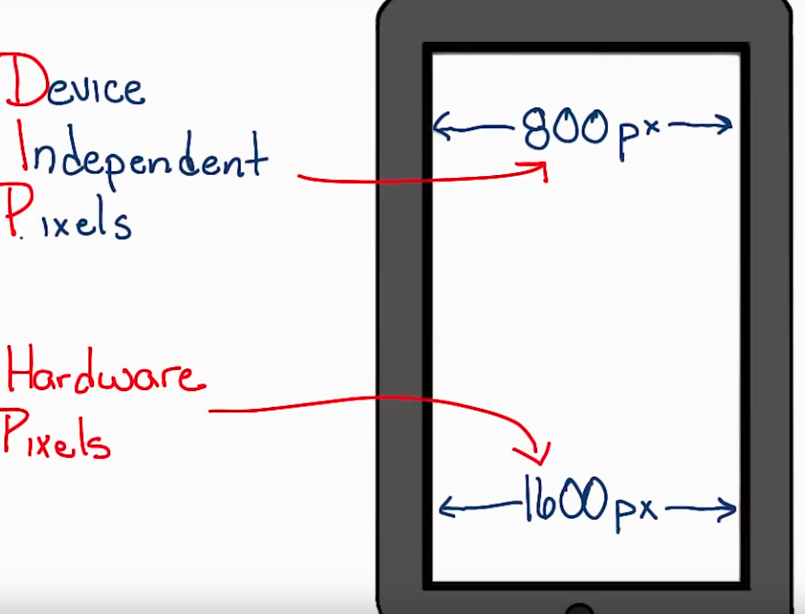
3.4 Connect them

<https://www.youtube.com/watch?time_continue=31&v=O091HfA9y_o>

**Lesson 2 – Starting Small**

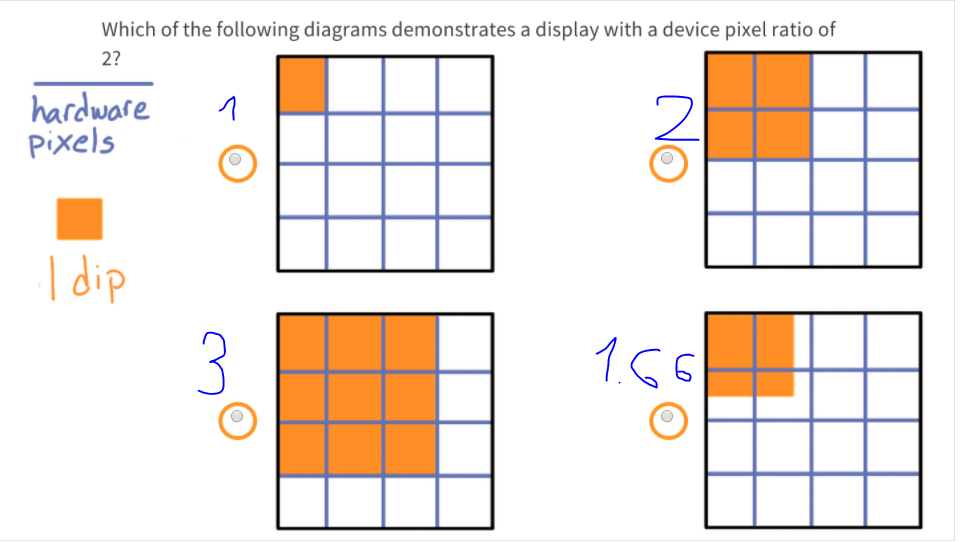
**2.1 DeviceIndependentPixel**: (some kind of extra measurement)

<https://stackoverflow.com/questions/4707320/basics-of-device-independent-pixels>



2.2 **Device Pixel ratio**

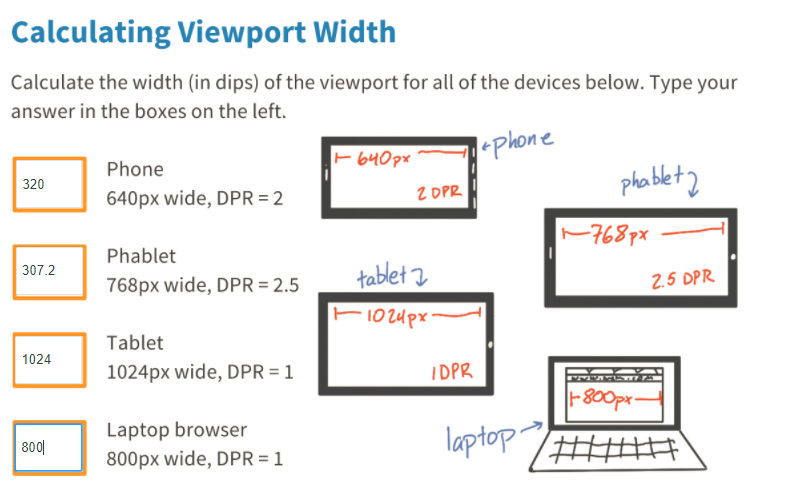
Will show you the following ratio hardwarePixel/DIP.



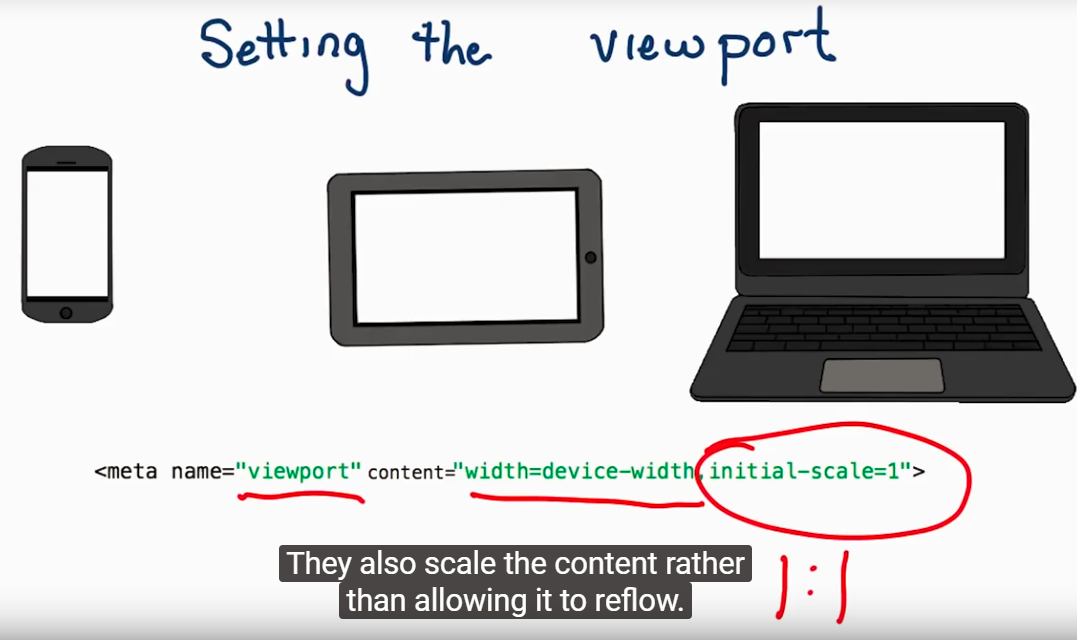
2.3. **Website can look different on identical screens due to:**

* Device Pixel Ratio is different
* Viewport was not set

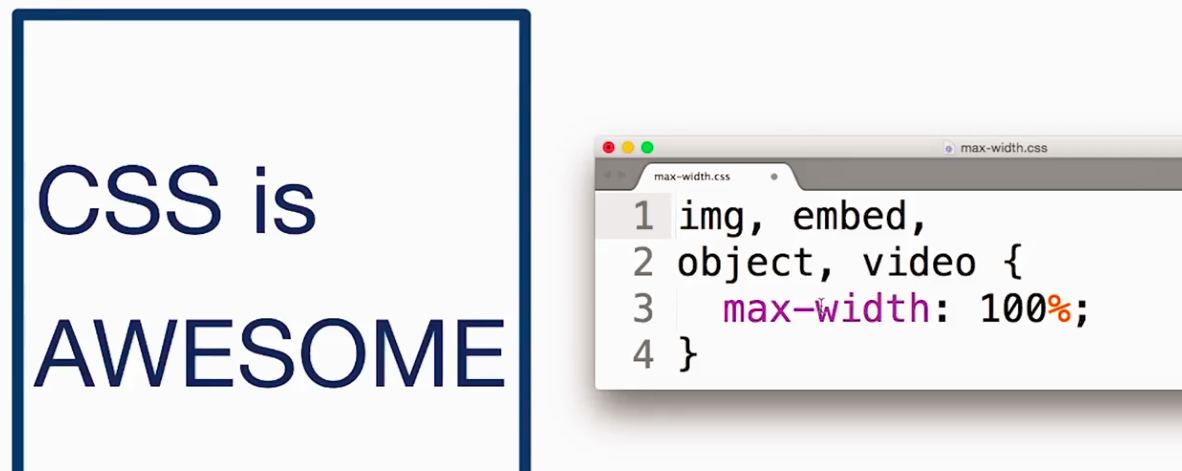
2.4 **ViewPort Calculation**



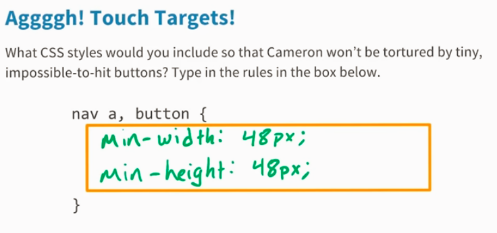
2.5 **Set the ViewPort in meta tag**



2.6 **Define max-width all the time!**

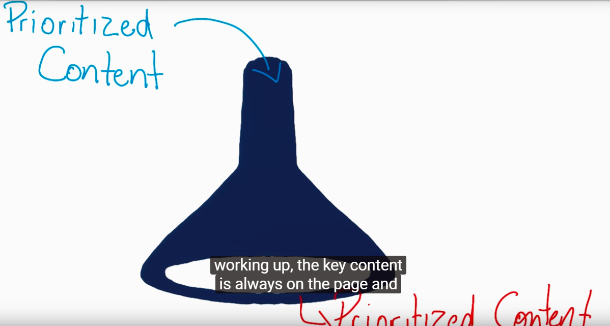


If you don’t the picture/text won’t fit the screen



2.7 **Tap Screen** (so user can tap the buttons in the screen)

It should be at least **48px X 48px**



2.8 **You start from small to large!**

* Start the coding from small
* Key Content is displayed
* Better User experience for mobile

2.9 **Project Summary**

In order to make it fully responsive you had to do the following:

* Add viewport meta tag
* Make all outer container element to min-width:100%;
* For any action button (**<a>)** add an inner padding of **1.5 em,** so it will be easy to tap

**Lesson 3 Building Up**

3.1 **Media Queries**

@media screen and (min-width: 401px) and (max-width: 599px) {

background-color:green;

}

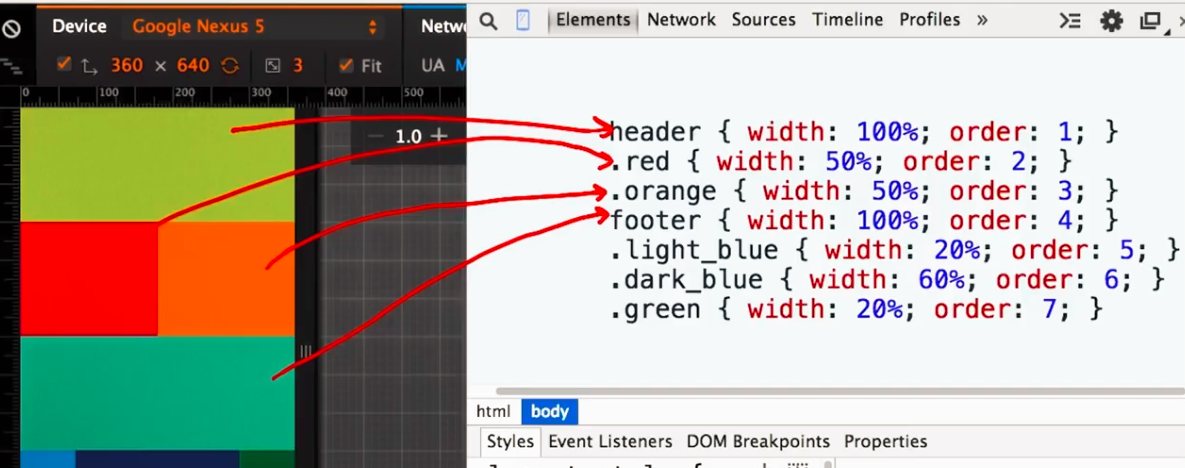
3.2 **Pick Up BreakPoints**

You determine the breakpoints based on the content!

3.3 **Flex Boxes**

**There is a great resource with lot of example here:**

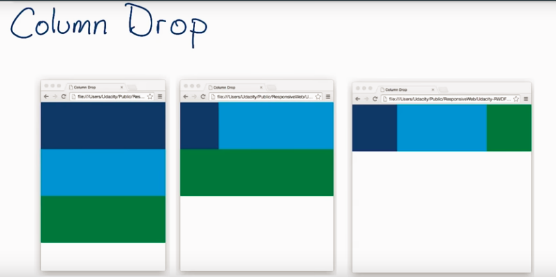
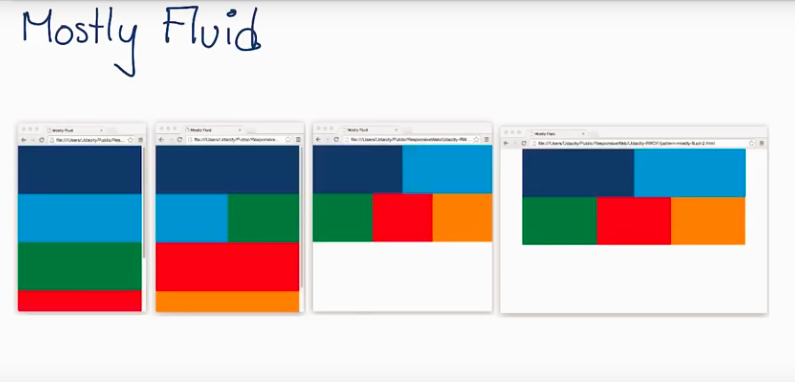
<https://www.w3schools.com/css/css3_flexbox.asp>



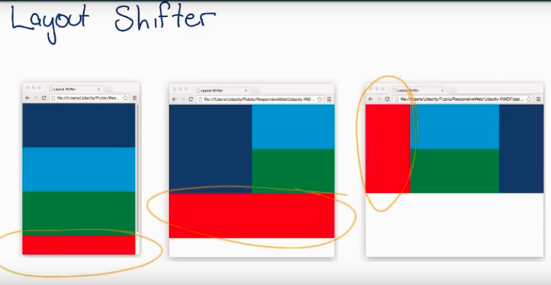
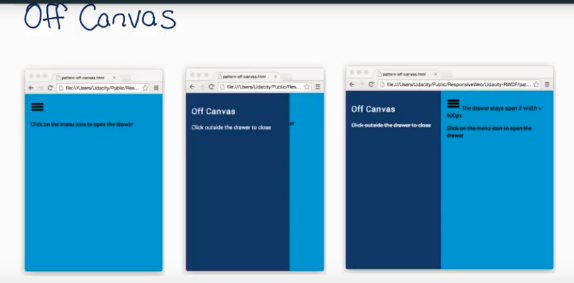
**Lesson 4 Common Responsive Patterns**

4.1 **Patterns**

**Column Drop** **Mostly Fluid**



**Layout Shifter (flexbox ordering)** **Off Canvas**



**Off Canvas:**

The elements are hidden until the screen is not big enough. (Hamburger menu)

You use translate to hide and unhide the elements.

nav {

width: 300px;

position: absolute;

*/\* This trasform moves the drawer off canvas. \*/*

-webkit-transform: translate(-300px, 0);

transform: translate(-300px, 0);

*/\* Optionally, we animate the drawer. \*/*

transition: transform 0.3s ease;

}

nav.open {

-webkit-transform: translate(0, 0);

transform: translate(0, 0);

}

**4.2 Project Work:**

(**MobileWebSpecialist\1\_Why\_Responsive\Lesson\_4\_Common\_Responsive\_Layout\rwdf-project-start**)

* Flexbox
* OffCanvas,ColumnDrop,LayoutShifter
* JQuery to toggle class

1. Add hamburger menu + position it with css (absolute position)
2. Stlye Navbar to be **off-canvas**  using **translate**

.nav {

width: 250px;

top:60px;

position: absolute;

z-index: 100;

background: white;

/\* This trasform moves the drawer off canvas. \*/

-webkit-transform: translate(-300px, 0);

transform: translate(-300px, 0);

/\* Optionally, we animate the drawer. \*/

transition: transform 0.3s ease;

}

nav.open {

-webkit-transform: translate(0, 0);

transform: translate(0, 0);

}

1. Add jquery to toggleClass of **open**
2. Media query, display nav normally
3. Add **flexbox**  to **content** class and determine the order + width of the sections

**Lesson 7 Units, Formats, Enviroments**

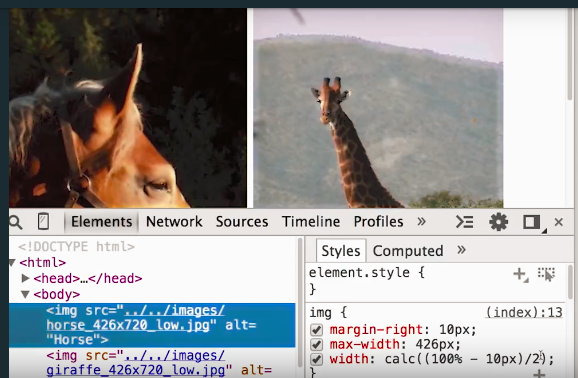
<https://www.w3schools.com/css/css3_images.asp>

**7.1 Images Relative Sizing**

//always good to have max size

min-width:100%;

if you want two images next to each other, you can use **calc()**



**7.2 Images Css tricks**

**full height image**

height: 100vh;

full width image

width: 100vw;

**Cover fully the webpage**

width: 100vmax;

height: 100vmax;

**Fit to the page height or width**

width: 100min;

height: 100min;

**7.3 Raster & Vector**

**Raster** Represented as a grid of dots of colors

**Vector:** Written shapes, lines, everything is determined by code (responsive)