***The Small Things***

**Defining the Viewport**

**Things you should learn from this lesson:**

* There is a difference between css and device independence.
* Understand there are considerations when choosing sizes for elements.
* Understand that when the viewport is not set properly your website may not display correctly in a device screen.
* Understand that the viewport defines the area of the screen the browser can render content to.
* Understand that a pixel is not always what we know a pixel to be – more to come on this.

– Created by [Utimacy](https://classroom.udacity.com/courses/ud893)

**Pixels, Pixels & More Pixels!**

**Things you should learn from this lesson:**

* Understand that browsers report the viewport area in [Device Independent Pixels (DIP)](https://material.io/guidelines/layout/units-measurements.html#units-measurements-density-independent-pixels-dp).
* Understand that there are Hardware Pixels and [Device Independent Pixels (DIP)](https://material.io/guidelines/layout/units-measurements.html#units-measurements-density-independent-pixels-dp).
* Understand that browsers report the viewport area in [Device Independent Pixels (DIP)](https://material.io/guidelines/layout/units-measurements.html#units-measurements-density-independent-pixels-dp).
* When the viewport is not set the browser will try to scale the page.
* Understand that Font Boosting increases the font size of text in wide columns, so you won’t have to zoom in on them as much and hence every column fits onscreen at a legible text size.

– Created by [Utimacy](https://classroom.udacity.com/courses/ud893)

**Setting the Viewport**

**Things you should learn from this lesson:**

* understand that the meta viewport tag sets the device width and scale.
* – Created by [Utimacy](https://classroom.udacity.com/courses/ud893)

**Large Fixed Width Elements**

**Things you should learn from this lesson:**

* understand that relative widths are better suited for responsive design.
* understand that large CSS widths or absolute positions can result in elements that are too big or that don’t fit on the screen

– Created by [Utimacy](https://classroom.udacity.com/courses/ud893)

**Max-Width On Elements**

**Things you should learn from this lesson:**

* understand that relative units should be used when specifying widths to prevent them from overflowing.
* understand that, by default, CSS will allow a containers content to overflow
  + this can be managed/mitigated by setting a max-width: 100%
  + it is a good idea to actually add a catch all to do this for img, embed, object, and video elements

– Created by [Utimacy](https://classroom.udacity.com/courses/ud893)

**Tap Target Sizes**

**Things you should learn from this lesson:**

* understand that it is a good rule of thumb, lol, to make tap targets (buttons) at least 48 pixels wide by 48 pixels tall

– Created by [Utimacy](https://classroom.udacity.com/courses/ud893)

**Start Small**

**Things you should learn from this lesson:**

* understand that it is a good rule of thumb to begin your design small to ensure your core styles and layout support the goals
  + reduces code
  + increases performance
* The [mobile-first](https://www.uxpin.com/studio/blog/a-hands-on-guide-to-mobile-first-design/) approach is exactly as it sounds: designing for the smallest screen and working your way up. It is one of the best strategies to create either a [responsive or adaptive design](https://www.uxpin.com/studio/ebooks/ux-design-trends-responsive-adaptive-web-design/).

– Created by [Utimacy](https://classroom.udacity.com/courses/ud893)