Examples of properties that need units:

**Font-size, Padding, border, margin, height, width, top and bottom, left and right.**

Examples of properties that don’t need units much:

**background,**

**Category of units:**

* **Absolute length:** These mostly ignore user settings. Example – px.
* **Viewport length:** It adjust elements according to viewport (visible area of website in the browser). Examples – vh, vw, vim, vmax
* **Font-Relative length:** Adjust to default font size. Example – rem, em
* **Percent length:** It is the percentage of the containing element. 3 rules to remember-
  1. If we have element with fixed position property, the containing block is the viewport.
  2. If we have element with absolute property, the containing block refers to the ancestor element’s (content + padding). The closest ancestor which has the position property defined, will be the containing element.
  3. If element property is static or relative, it will be ancestor but with only the content.

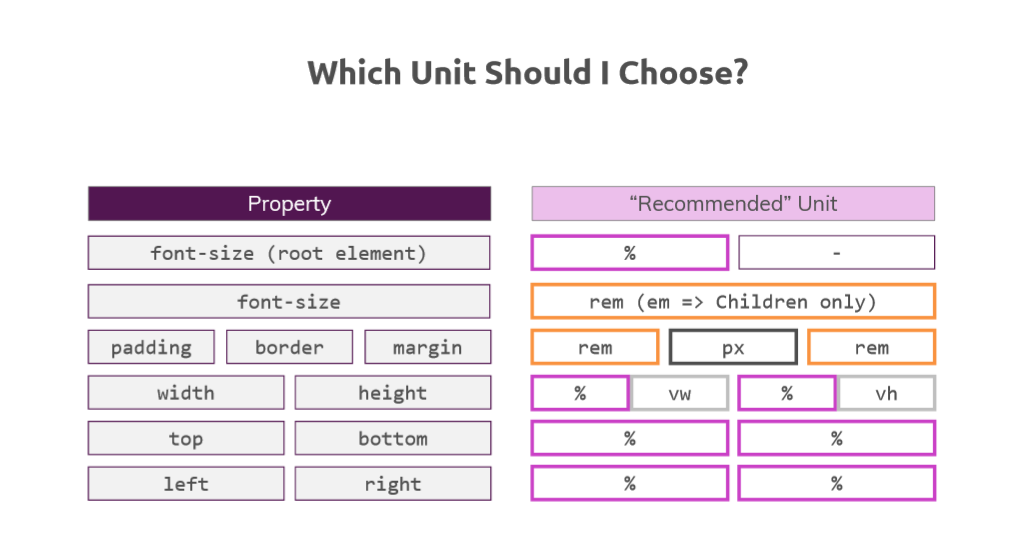
**Height 100% issue** – set the property to absolute.

**Max-width and min-width:** helpful for defining the breakpoint.

**Rem and Em:** Rem is relative the body font-size, and em is relative to the parent font-size. In case of em, if we have multiple em defined in the ancestor hierarchy, it will keep multiplying as it goes towards children.

**Vw and vh:** it always refers our sizes of actual viewport.

**Vmin and vmax:** 80vmin means it will take smaller of the elements height and width and set it to 80% of it. 80vmax will work the other way around, it will take the more of the two.



**Font-size of root element**: we can use default or % if we want to change the default

**Font-size**: better is rem, can use em too but recommended it to use on children only.

**Padding/margin**: rem is recommended.

**Border**: pixel

**Width/height:** we can use % (just have to aware of containing block), also can use vh/vw

**Top/bot/left/right:** % is recommended.

These are just rule of thumb, not absolute