Basics to start with..

Default html font-size 16px

**CSS UNITS**

* Absolute Units which are easy to understand.

We mostly use pixels(px). Rarely we also use pt, cm, mm, in, etc.

* Percentage are mainly used for widths, and easy to understand. Relative to their parent.
  + Ex:

\*\*\*\*{

Width: 80%

Max-width: 620px

}

* Relative Units
  + Relative to font-size of its parent component
    - em and rem (and many other less common ones)
    - em:
      * 1em (relative to parent size)
      * 1em means size equal to parent component font size
      * 1.5em means 1.5 times to parent component font size
    - Problems with em:
      * Cascading effect
    - rem (Root em):
      * 1rem (relative to root of document i.e. html element)
  + Relative to viewport
    - vw, vh, vmin, vmax

**General Rule of Thumb**

* Font-Size = rem
* Padding and margin = em
* Widths = em or percentage

It is just a general rule of thumb, not a hard and fast rule!

@media query basics

Min-width, max-width, orientation, screen

**CSS FLEXBOX**

Flex-Container properties

* display
  + creates a flex container. flex, inline-flex
* flex-direction
  + sets the direction of main axis. row, row-reverse, column, column-reverse
* flex-wrap
  + control the wrapping of flex items within the container. nowrap, wrap, wrap-reverse
* flex-flow
  + flex-flow: takes two values simultaneously. 1st one is of flex-direction and other is flex-wrap. So, we can use flex-flow directly instead of using individually flex-direction and flex-wrap.
* justify-content
  + Align items and distribute any extra spacing in the parent container. flex-start, flex-end, center, space-between, space-around, space-evenly
* align-items
  + Align items along the cross-axis. flex-start, flex-end, center, baseline, stretch
* align-content
  + Aligns lines of content along the cross axis and distribute any extra spacing in the parent container. flex-start, flex-end, center, stretch, space-between, space-around.

Flex-item properties

* Order
  + Controls the order of items in the flex container. Integer value.
* flex-grow
  + dictates what amount of available space inside the flex container the item should take up. Relative to the other items in the container.

Default value is 0 – items do not grow.

Flex-grow value of 1 –flex items grow evenly.

* flex-shrink
  + dictates the shrink factor of the flex items when the default size of flex items is larger than the flex container. Relative to the other items in the container.

Default value is 1.

* flex-basis
  + To set initial size of a flex item. Pixels, percentages or relative units. Default value is auto.
* Flex
  + Shorthand for flex grow, shrink and basis.

.item{ .item{

flex-grow: 2; flex: 2 5 200px;

flex-shrink: 5; }

flex-basis: 200px;

}

* align-self
  + Align the items individually.

Values like auto, flex-start, flex-end, center and stretch.

Overrides the align-items value of the flex container.