CSS Import

Use @import url(…); in css say to import a font or something from online without the need to use link in html

User Agent Stylesheet

It is the default stylesheet given by browser like chrome, when you write only html with no css, that default style of font, size of different heading etc is what that default stylesheet is giving you.

CSS Selectors:

Class

.abc {

}

Id

#abc {

}

Element

P {

}

Universal

\* {

}

Grouping

.a, #b {

}

Nested – it will find inside, and do nothing if incorrect hierarchy

.abc>#abcde>salmslamsldmalsdm … {

}

Pseudo selectors

p:hover {

}

CSS Priority/Override:

Internal and External CSS hv same priority, their import/definition order define who comes first in cascading.

Inline is more priority then the internal and external, as it is defined most close (last) to the actual content.

Also, in a child, all its ancestors styles is inherited in inverse manner, the most closest ancestor, i.e. the parent is most priority, then its parent and so on till the first styled ancestor (maybe root or body)

Inside an element, priority is: inline > id > class > element

For multiple classes, the order in which the classes is defined in the css file will decide the final resulting override. Above defined is overrided by below devined class. Eg: class=”abc abc-r”, then if abc-r is defined after abc then it is overrides abc

CSS Colors:

rgb(255, 0, 123)

#ff00ff

hsl(10, 100%, 20%) saturation gives color to dark, lightness gives any to light(max white), and hue is the actual base color

hwb(10, 34%, 40%): same as hsl, here lightness is whiteness(w) and saturation opposite is blackness(b)

why hsl/hwb need? Because they are very intuitive for people who don’t know how primary colors rgb mix to give all other colors. Very hard to predict color in rgb model.

CSS Widths:

* 50%: relative to parent
* 100vw: 100% of view width, 50vw: 50% of view width – it is absolute wrt parent, so in a parent whose width is relative if u add a child with vw units, that child size may go beyond parent size on scale/normally. But vw is relative wrt the view width. (unlike pixel)
* Just like vw we have vh for height units
* If you use PX for font, then browser’s increase font size option will not increase your absolute font size. You must always use **em** to define font size
* There are other absolute units like PX like
  + in: inch
  + pc: picas 1/6 inch
  + pt: point 1/72 inch
  + cm
  + mm
  + Q: quarter millimeter (¼mm)
* Not only is em a relative unit for browser, it is parent-relative too. Every element is by default 1em. If u make a parent specifically 2em and a child in it is also 2em, then that child is actually 2 times bigger than it would have been in a 1em default parent. This is because the em in the child takes parent’s base font size as the base which was already 2 in our case.
* By the same logic, if parent is 2em, instead of by default 1em, then any child, by default will be twice as big, bcoz say h3 was 1em, and parent was 1em then h3 remains 1em. But now parent is 2em and child h3 is relative to parent, so that 1em of child is becomes twice as parent unit is now twice.
* To solve this unwanted scaling of em based unit font, rem is used, which is root em. So only relative to the root the font size will change. Root size can be set in css of html {} tag
* Use of pixel: like border width

CSS Box Model

* box-sizing: border-box (instead of width and height as content size, it will be size of box till border including border size and padding size, and it will automatically change content size to make sum = border-box width mentioned below as width: say 500px)

CSS Border

Always going in clockwise:

1 value: all sides same value

2 values: top-bottom + left-right

3 values: top, right, bottom

4 values: different side different values

* border <size> <type> <color>
* border-color
* border-style
* border-width

CSS Text

Color

Background-color

Text-align: justify/right/end/start/left

The text-align-last property specifies how to align the last line of a text.

Text-decoration:

* text-decoration-line: line-through
* text-decoration-color:
* text-decoration-style:

Text-shadow

Font-family: fallback when one not found to next font

Web safe fonts: the ones that are supported by majority of the browsers

Font-size

Font-weight

Font-style

Text-transform

Word-spacing

Line-height

Letter-spacing

CSS-Display

Inline (pure):

* a
* span
* Inline ele also not have top and down margins

Block ele:

* p, h1-h6, div, list, form
* They basically block the flow and must continue on next line vertically

Inline-Block ele:

* Img
* Here we can not only have inline continuation, but also specify width and height. IMPORTANT: For Pure Inline We Cannot Define Width or Height