

~~CSS~~

(Loading style sheets)

body {

background-color: red;

CSS → add to screen

Using style tag

<style></style>

Using inline

style attribute

DOM

(Document object

mode)

→ tree of objects

(tree like structure)

<div style="background-color: red;">

CSS Selectors

Used to select HTML elements for styling.

for grouping elements with same properties use class,

e.g. h1, h2, h3
color: white;

body {

color: red;

element selector

background-color: blue;

↓
used to select element on the basis of tag name

ID

Selector

Class

Selector

Used to select an element with given id

Used to select an element with given class

#first {

color: blue;

background-color: red;

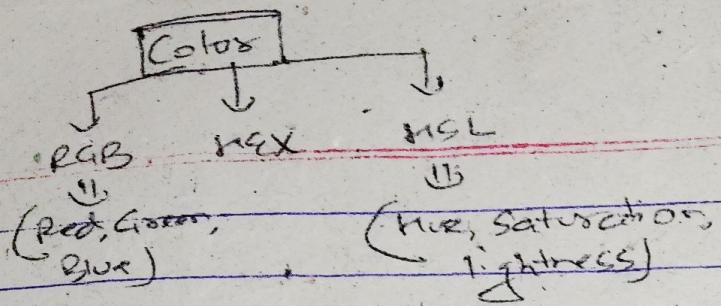
* {
margin: 0;

padding: 0;

}

is used to select the properties for all elements

Comments: /* --- */
ctrl + /



Background-ing Property

Body of

`background-image: url("img.jpg")`

{

The image will by default repeat in x-
and y direction.

`repeat-x` → repeat in x direction

`repeat-y` → " " y "

`no-repeat` ⇒ image not repeat

for ex :- body {

`background-image: url("img.jpg")`

`background-repeat: repeat-x;`

}

Background-size Property

`background-size: contain` (will fit in the img)

`background-size: cover` (will compromise the
: Auto img, and cover the
same img
size)

Background-position property

`background-position: top / left / right / top bottom
top centre / ;`

Background attachment property

Background-attachment: fixed

" " : scroll

" " : local

Background fixed

XXX background: red url("img.jpg") no-repeat fixed center;

↓ ↓ ↓ ↓
 color img repeat attachment
 Position

KIRAP

CSS Box Model



total height of element = height + Top/Bottom padding

+ Top/Bottom borders
+ Top/Bottom margin

margin/padding:

box { margin-bottom }

box {

margin: 3px

padding: 4px

margin: 4px 3px 4px 1px } ls

top right

left

sets margin & padding from all directions

Setting Borders

box {

border-width: 3px;

border-style: solid;

border-color: red;

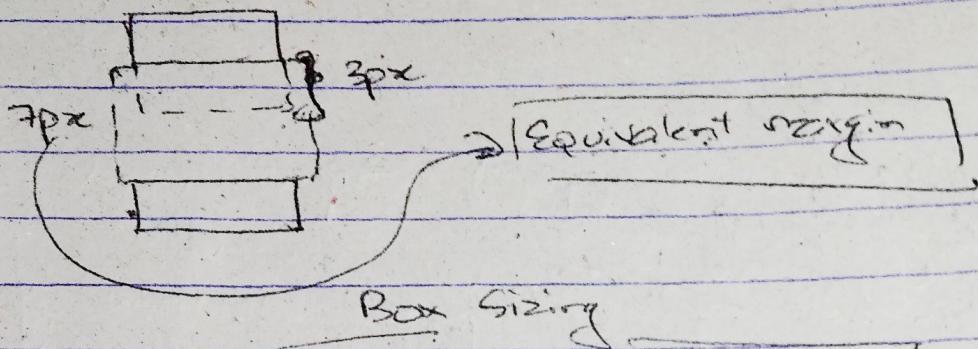
OR

box { top-bottom }

margin: 4px 2px left-right

~~MARGIN COLLAPSE~~

when two margins from different elements overlaps,
the equivalent margin is the greater of the two.



Box Sizing : Border Box

↳ includes

padding &
height & border in width

Box Sizing : Content Box

↳ doesn't include
padding & borders
in width

↳ refers to width of
content

DISPLAY & FONTS PROPERTIES

Display: inline;

→ it takes width
only how much is
required

→ Top margin / Top Padding
Bottom Bottom
is not allowed.

Text: Align;

horizontal movement of Text in container/div.

Text & decoration:

used to decorate text, underline, overline, etc.

Text & transform:

capitalise, uppercase, lowercase

line-height:

used to specify space b/w lines.

Font-family:

use multiple fonts in this property, any one will be rendered depending
upon the availability in user system.

Font-family: Basic, ABC, Rohit Ssi;

* * font-size, font-weight, font-style,
 ↓
 Boldness italic, normal, etc

Relative lengths ⇒ Size, Position & Lists

px → Pixel

vh → viewport height

vw → viewport width

em, rem

Parent font size

for ex: 3px → Parent font size

3em → 3x4 px.

The height/width

on any screen (at 3vw)

3% of visual width.

min-max-h-width

min-width: 3px;

max-width: 4px;

min-height: 3px;

max-height: 4px;

Position

static

→ default position

→ top/bottom/left/right/z-index
won't work

relative

→ top/bottom/left/right/
z-index will work

Absolute

→ Element is removed from
DOM

→ Position changes
with respect to
first non static parent
element

fixed

→ positioning
wrt to border
window

→ fixes the
position

sticky

→ Positioning
remain same
even if
scrolling

* z-index: in order to keep one
element above other (if they
species stack order
of an element are overlapping) just increase the
z-index for that element.

FLOAT PROPERTY

moves element to right/left.

float: left;
float: right;

FLEX BOX

Cross axis

1

better way to layout align and distribute space among items in container.

main axis

- container {

justify-content: center

horizontally
centers

• flex-direction: row;

• flex-direction: column,
 column-reverse,
 row-reverse, column-reverse

• flex-wrap: wrap, nowrap, wrap-reverse

display: flex;

align-items: center

vertically
centers

- order
- align-self
- flex-grow
- flex-shrink

align-content: center

removes spaces
b/w Boxes

grid-column: 1/5;
defines boundary
columns are
spanned

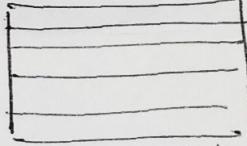
GRID

display: grid;

grid-row-gap: 10px;

grid-column-gap: 20px;

grid-template-columns: 24px



10px Auto
takes width
for columns.

grid-gap: 2px 9px;

↓
row column

grid-template-rows: 2px 10px Auto;

↑ takes ht. for rows

media-queries

if some condition reaches then new
css property triggers:

@media screen and (max-width: 800px) {

body {

background-color: red;

}

Transform

transform-origin: scale;

transform: rotate(x,y);

 " : Scale(x);

 " : Skew();