**CSS**

**Installing SASS**

1. Install Ruby on “rubyinstaller.org”

2. open Start Command Prompt with Ruby

3. type “gem install sass” or “sudo gem install sass”

4. “sass --v”

5. “cd desktop/folderName”

**Compiling SASS**

6. sass --watch sassFolder/master.sass:master.css

**SASS Layout**

sassFolder

- 01 base

- 02 addons

- 03 layout

- 04 modules

master.sass

**Base Styles**

body

background: #fab818

margin: 0px

padding: 0px

overflow-y : scroll

font:

family: Arial

**Variables**

$yellow: #fab818

**Adding partial SASS files**

\_partial.sass

@import folderName/partial

**CSS Selectors**

Pseudo Selectors

a:link

a:visited

div:hover

a:active

input:focus

Attribute Selectors

a[target="\_blank"]  
 background-color: yellow;

input[type="text"] , input[type="password"]

margin: 5px

Nth Selectors

p:first-child

p:last-child  
p:last-of-type

p:nth-child(2)

p:nth-of-type(2)

**CSS Properties and Values**

**Background**

background-color : blue;

background-image : url("../background.jpg");

background-origin : border-box; - background-origin are for images

background-clip : border-box - background-clip are for colors

border-box - the background image starts from the upper left corner of the border

padding-box - the background image starts from the upper left corner of the padding edge

content-box - the background image starts from the upper left corner of the content

background-repeat : repeat;

repeat-x; repeat-y; no-repeat;

background-position : top;

bottom; top right; bottom left; left; center; 50% 10%; 30px 100px;

background-attachment : fixed;

scroll; local;

background-size : cover;

contain; 100px 50px;

**Background Multiple**

div

background-image: url(image1.jpg), url(image2.jpg);

background-position: right bottom, left top;

background-repeat: no-repeat, repeat;

**Box Shadow**

text-shadow: 2px 2px black

box-shadow: 0 0 8px 1px blue;

box-shadow: 0 0 0px 2px white, 0 0 0px 4px blue, 0 0 20px 6px blue;

**Border**

border : 2px solid black

border-style : solid; dotted; dashed; none; hidden;

border-width : 2px; 2px 4px 2px 4px;

border-color : blue; blue yellow blue yellow

border-radius : 2px; 2px 4px 2px 4px;

border-top-left-radius : 2px;  
 border-top-right-radius : 2px;  
 border-bottom-right-radius : 2px;  
 border-bottom-left-radius : 2px;

border-image: url(border.png) 30 round;

border-image: url(border.png) 30 stretch;

**Clearfix**

.clearfix

&:before

content: " "

display: table

&:after

content: " "

display: table

clear: both

.clearfix

overflow: auto

**Column**

column-count: 3;

column-gap: 40px;

column-rule: 1px solid lightblue;

column-rule-style: solid;

column-rule-width: 1px;

column-rule-color: lightblue;

column-span: all; 1;

column-count number of columns an element should be divided into

column-gap the gap between the columns

column-rule-style style of the rule between columns

column-rule-width width of the rule between columns

column-rule-color  color of the rule between columns

column-rule shorthand property for setting all the column-rule

column-span how many columns an element should span across

**Cursor**

cursor : default

text; pointer; wait; not-allowed

pointer-events:none

**Display**

display : block

inline; inline-block; none

**Float and Clear**

float : left

right

clear : left

right; both

If an element is taller than the element containing it, and it is floated, it will overflow outside of its container.

Then we can add overflow: auto; to the containing element to fix this problem:

.clearfix

&:before

content: " "

display: table

&:after

content: " "

display: table

clear: both

.clearfix

overflow: auto

**Font**

font-family : "Times New Roman", Times, serif;

font-style : normal

italic

font-weight : normal

bold

font-size : 20px

to import your own font, use the @font-face

@font-face

font-family: fontname;

src: url(fontname.ttf);

div

font-family: fontname;

**Form**

form

&:focus

outline: none

**Height and Width**

height : 50px; 20%; 100vh

width : 100px; 50%; 100wh

min-width : 50px

max-width : 100px

min-height : 100px

max-height: 200px

box-sizing: border-box;

use 100vh and 100wh to make the width and height equal the size of your screen

box-sizing property allows us to include the padding and border in an element's total width and height.

**Image Filters**

filter: blur(4px);

brightness(250%); (180%); grayscale(100%); hue-rotate(180deg); invert(100%);

opacity(50%); saturate(7); (100%); drop-shadow(8px 8px 10px green);

**Margin**

margin : 2px 5px 2px 5px

margin-top : 100px;  
margin-bottom : 100px;  
margin-right : 150px;  
margin-left : 80px;

margin : 0 auto

**Overflow**

overflow : visible

auto

overflow-y : scroll

overflow-x : hidden

**Padding**

padding : 2px 5px 2px 5px;

padding-top : 50px;  
 padding-right : 30px;  
 padding-bottom : 50px;  
 padding-left : 80px;

**Position**

position : static

relative; fixed; absolute

**Resize**

resize : horizontal; vertical; both

**Text**

color : blue

text-align : right

left

center

justify

text-decoration : none

overline

line-though

underline

text-transform : uppercase

lowercase

capitalize

text-indent : 50px

letter-spacing : 50px

word-spacing : 50px

line-height : 50px

text-direction : rtl

text-shadow : 2px 2px #FF0000

**Transition**

transition : width 2s , height 4s , transform 2s

all 0.2s ease-in-out

width 2s ease-in 1s, height 4s ease-out 2s

transition-property : width

height

transition-duration : 2s

transition-timing-function : ease-in-out

ease - specifies a transition effect with a slow start, then fast, then end slowly (this is default)

linear - specifies a transition effect with the same speed from start to end

ease-in - specifies a transition effect with a slow start

ease-out - specifies a transition effect with a slow end

ease-in-out - specifies a transition effect with a slow start and end

cubic-bezier(n,n,n,n) - lets you define your own values in a cubic-bezier function

transition-delay : 2s

**Z Index**

z-index : 3

**2D Transform**

transform : rotate( 2deg );

scale ( 1.2 ); scale ( 2,3 )

translate(50px,100px) - moves an element from its current position

rotate() - rotates an element clockwise or counter-clockwise

scale() - increases or decreases the size of an element

skewX(20deg) - skews an element along the X-axis by the given angle

skewY() - skews an element along the Y-axis by the given angle

skew(20deg, 10deg)

matrix()

transform-origin: bottom left

**3D Transform**

transform: rotateX(150deg)

rotateY(150deg); rotateZ(90deg)

rotateX() - rotates an element around its X-axis at a given degree

rotateY() - rotates an element around its Y-axis at a given degree

rotateZ() - rotates an element around its Z-axis, same as rotate