Documentation

CMYK Colors

```
.example-cmyk {

        color: cmyktorgb(13,100,100,4);
        background: cmyktorgb(0,0,0,100);
}

/* it's give cmyk to rgb color value */
```

Hexadecimal

```
.example-hexadecimal-check {
        color: 6hexa(eeeeee);
}

/* your color value does not exceed more than six characters    */
```

Type Scale

```
  @include typescale-rem(16,5,4);
```

it's gives 8 selector with font size values from extra large to captions
lg xxl xl md sm paragraphy sub-text captions

Vendor Prefix

```
p {
        @include vendors(text-align,center,2);
```

}

Monochromatic-Color

```
p   {
      @include mono-color(1,0,10,10);
}
```

1 means color property,
2 means background property

0 - hue value;

/* it's should be same number like 10,10 */

10 - control Monochromatic-color value;
10 - control Monochromatic-color value;

you can use monochromatic color value

Triadic Colors

```
p {
```

```
        @include triadic-color(1,30);
}
```

1 -   color property
2 -   background property
30 - hue value

/* It produces colors based on Triadic-color, so later you can decide which color you want to use and the remaining colors you can keep or delete that */

Tetradic Colors

```
p {
        @include Tetradic-color(1,30);
}
```

1 -   color property
2 - background property
30 - hue value

/* It produces colors based on Tetradic-color, so later you can decide which color you want to use and the remaining colors you can keep or delete that */

Analogous Colors

```
p {
        @include Analogous-color(1,30);
}
```

/* It produces colors based on Analogous-color, so later you can decide which color you want to use and the remaining colors you can keep or delete that */

Flexible Media Query

```
@include media(600,1) {
p     {
  color: red;
}
}
```

you have three options em, pt ,px :

   1 means 600 will convert to em values
   2 means 600 will convert to pt values
   3 means 600 will convert to px values

   /* it's got flexible values like em pt px   */

Grid System

`@include grid system(560,30,10);`

560 - width value
10 - column
30 - gutter

/* who is familiar with grid system will recognize this well */

Typescale-Rem

```
p {
     font-size: typpscale-rem(1.125,48,1);
}
```

/*   it will build type scale value from font-sizes largest to small values */

Button Round or Non-Round

```css
.button {
    border-radius:    border(1,1);
}
```

/* same value 1 == 1 it's give round */

```css
.button {
    border-radius:    border(1,2);
}
```

/* same value 1 != 2 it's give non-round */

Remove Margin and Padding

```css
* {
    margin:    rm(8);
    padding: rm(8);
}
```

/*   it will remove default margin and padding values in the body   */

Font-Weight 100 to 800

```css
h1 {
    font-weight: adjustbold(2);
}
```

/* 2 - 200 */

  /* generate font weight 100 to 800 */

Color Accessbility For   Black Color Value Only !!

```
p {
    color:    checkcolorasb(#000);
}
```

/* it's only accepting black color value passed under color accessibility */

Detect Layouts

@include detect-layouts;

Go to your index.html
in body tag at HTML,add the class name ".detect".
it will detect desktop and mobile screen layouts when you resize the browser and
after completing your project and remove the .detect class from the body tag in
index HTML