**SASS Course**

* Go to extinction in visual studio and add live sass then make in bar watch sass
* Create 2 files one call main.css and main.scss when you write in scss it will compile in css
* To create file.scss and don’t want to compile use before name \_
* \_color.scss
* To identify variables:
* $main:blue;
* $alt:red;
* $new:black;
* To call scss file use
* @import './sass/layout/pages/Global\_Rolles' ;
* @import './sass/variables/color';
* Then use it after call it in main file scss like

first {

    background-color: $main;

}

.foot {

    // local varable

    background-color: $alt;

}

.copy {

    // to overide the the global varaebel

    $back-colorr: white !global;

    background-color: $main;

}

h1{

    color: white;

    background-color: $new;}

Nestin you can use it easly

.parent{

    font-size: 50px;

.child{

    font-size: 30px;

    .grandchild{

        font-size: 20px;

    }

}

}

Or grouping like:

parant-one,

 .parant-two{

       padding: 20px;

        color:red;

    }

* To use shorthand like:

  .box{

        font: {

            size:20px;

            weight:bold;

        }

        padding: 10px;

        margin: auto{

            top:10px;

            bottom: 10px;

        }

    }

* To extend some property for element to another
* ..main-box {
* font-size: large;
* color: red;
* }
* .ads{
* @extend .main-box;
* padding: 10px;
* }

If there no element only you identify class for use ,use placeholder %:

 %main-box {

        font-size: large;

        color: red;

    }

    .ads{

        @extend %main-box;

        padding: 10px;

    }

* If condtion used like:
* $them:'light';
* .page {
* @if($them=='dark'){
* background-color: aqua;
* color: black;
* }@else{
* background-color: black;
* color: aqua;
* }
* }

Or more like

$round:true;

    .condtion {

        margin-top: 20px;

        @extend %main-box;

        background-color: green;

        border-radius: if($round,10px,null);

    }

* Interpolation is used to add the value to the name or property we should use ( # and {} )like:
* $company:'elzero';
* $postion:'left';
* .ad-#{$company}{
* font-size: 20px;
* background-image: url('image/#{$company}.png');// her we use the value
* #{$postion}:0; //her we use the property
* }

* To use function for example unique to create unique id:
* .ad-#{$company}-#{unique-id}{
* font-size: 20px;
* background-image: url('image/#{$company}.png');// her we use the value
* #{$postion}:0; //her we use the property
* }
* @Mixin use like function in the language programming and called with qinclude:

@mixin circle-100($dimintion){

 border-radius: 50%;

 width: $dimintion;

 height: $dimintion;

}

@mixin pinting($font) {

    font-size: $font;

    font-family: 'Courier New', Courier, monospace;

    text-align: center;

}

.page {

    margin: auto auto;

    @include circle-100(200px);

    @include pinting(30px);

      }

* For loop used like :
* $dimintion:0;
* @for $i from 1 through 10 {
* .circle-#{$dimintion + 100}
* {
* width:$dimintion + $i;
* height:$dimintion + $i;
* }
* $dimintion:$dimintion + 100;
* }
* @each and map used like array or object when you want to use more colour in website like:

$themes:red,blue,green;

@each $them in $themes {

    .#{$them}-them{

        .product{

            color: white;

            border-bottom: $them;

            .title{

                background-color: $them;

                font-weight: bold;

            }

        }

    }

}

To use object and key like this:

$socials:('facebook':blue,

          'youtube':red,

          'google':white );

     @each $name,$color in $socials {

         .#{$name}{

             background-color: $color;

             &::before{

                content: $name;

            }

         }

     }

$classes:'one' 20px red,'two' 20px blue,'three' 20px green;

@each $name,$font,$color in $classes {

    .#{$name}{

        font-size: $font;

        background-color: $color;

    }

}

* Function used like in another language like:
* @function half($size){
* $size:($size/2);
* @return $size;
* }
* @function culculate($cals...){
* $total:0;
* @each $cal in $cals {
* $total:$total+$cal;
* }
* @return $total;

And call it like:

 $width:200px;

.element{

width: $width;

height: half(200px);

top: culculate(200px,100px,50px);

can man send content like:

      @mixin keyF($anim-name){

         @-webkit-key-frames#{$anim-name} {

           @content;

         }

        @keyframes #{$anim-name}{

            @content;

        }

      }

      @include keyF(fade-in){

        from {

            opacity:0;

       }

       to{

           opacity: 1;

       }

      }

      @include keyF(fade-on){

        from {

            opacity:0;

       }

       to{

           opacity: 1;

       }

      }

* There are some string functions in w3school like:

append((a b c), d)  
Result: a b c d  
append((a b c), (d), comma)  
Result: a, b, c, d

and number:

abs(15)  
Result: 15  
abs(-15)  
Result: 15