Sass

The full form of SASS is **Syntactically Awesome Style Sheets**

Sass : Is CSS pre-processor or CSS with the super power .

First download : live sass server vs code extension .

Difference between sass and scss :

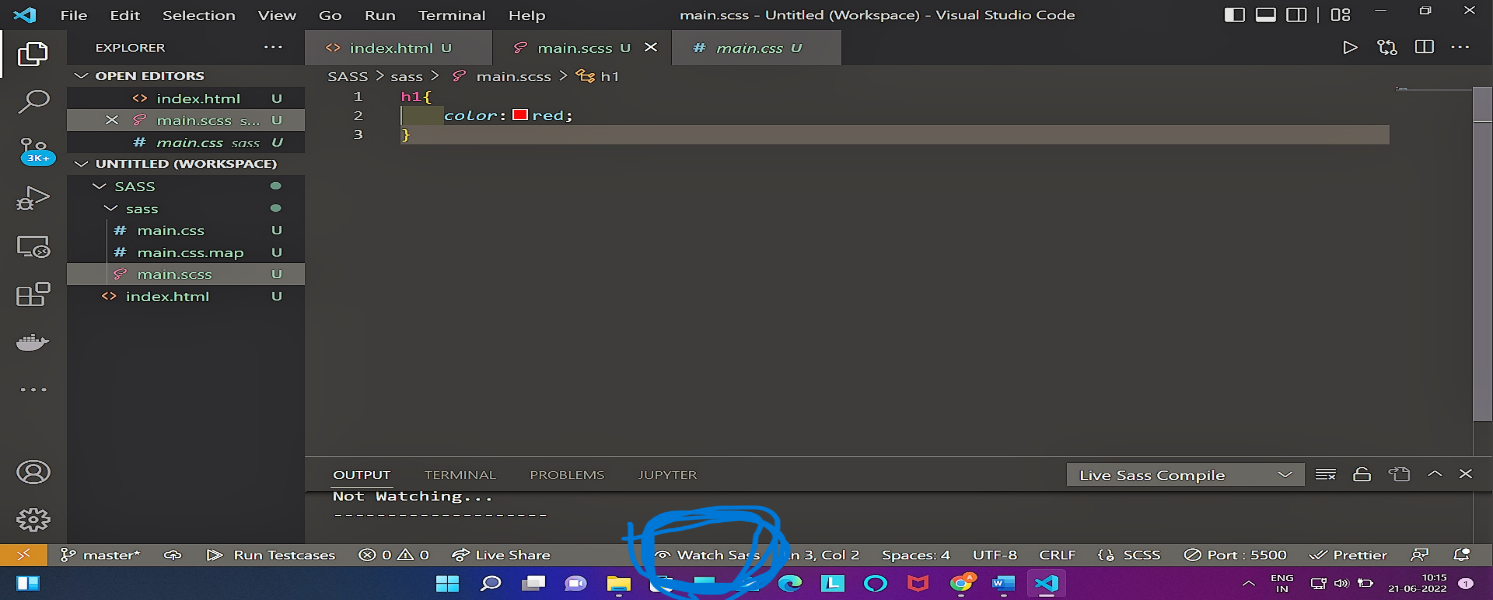
Scss : CSS 3 ;

Sass :css ;

Step for working in sass :

Setp 1 : create a sass folder in that folder , create a main.scss file and write all your code in it .

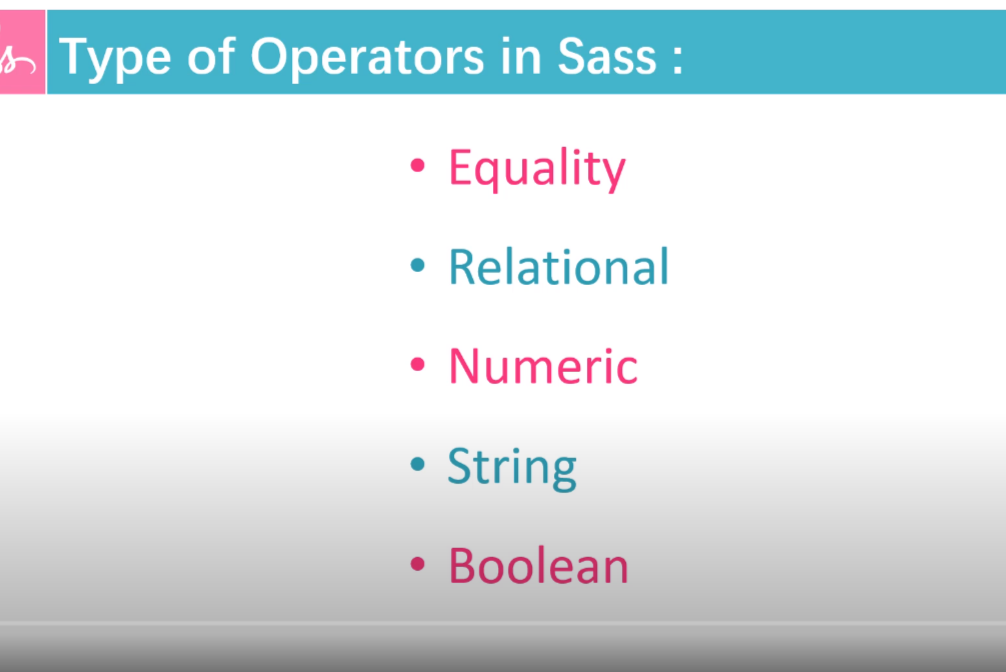
Step 2 : click on live sass that extension that we have downloaded .

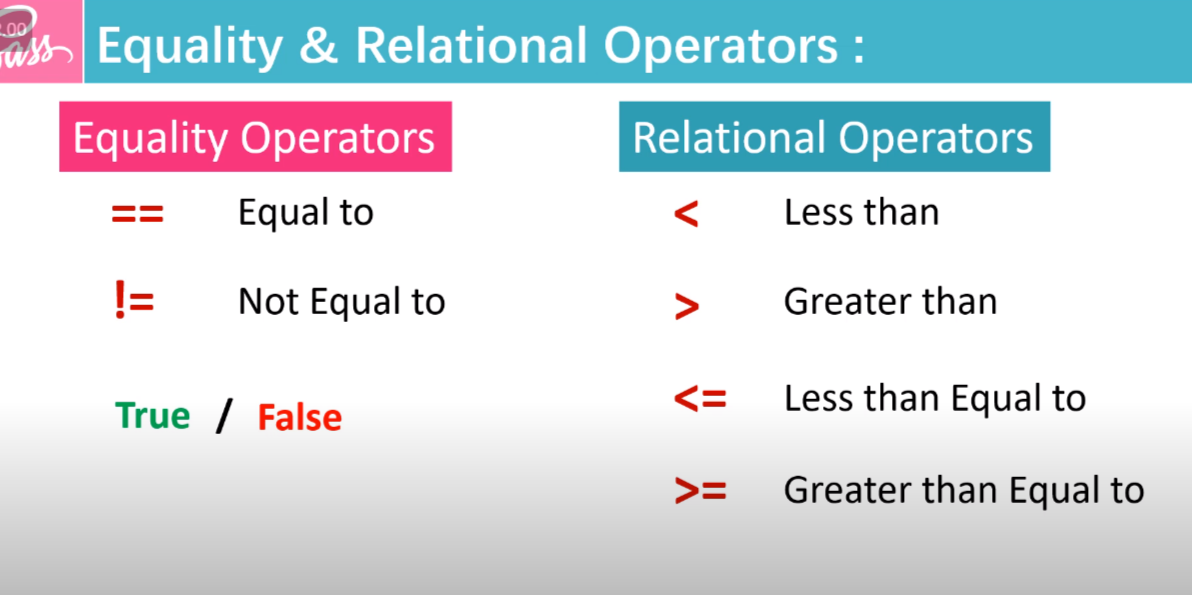


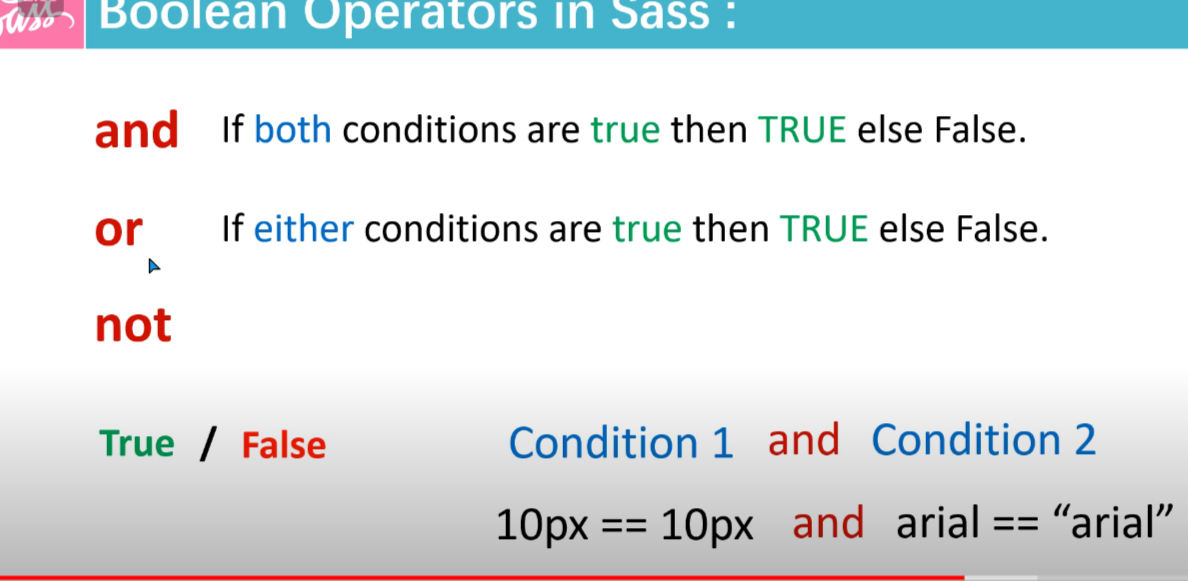
Step 3 : add main.css file in your html file .

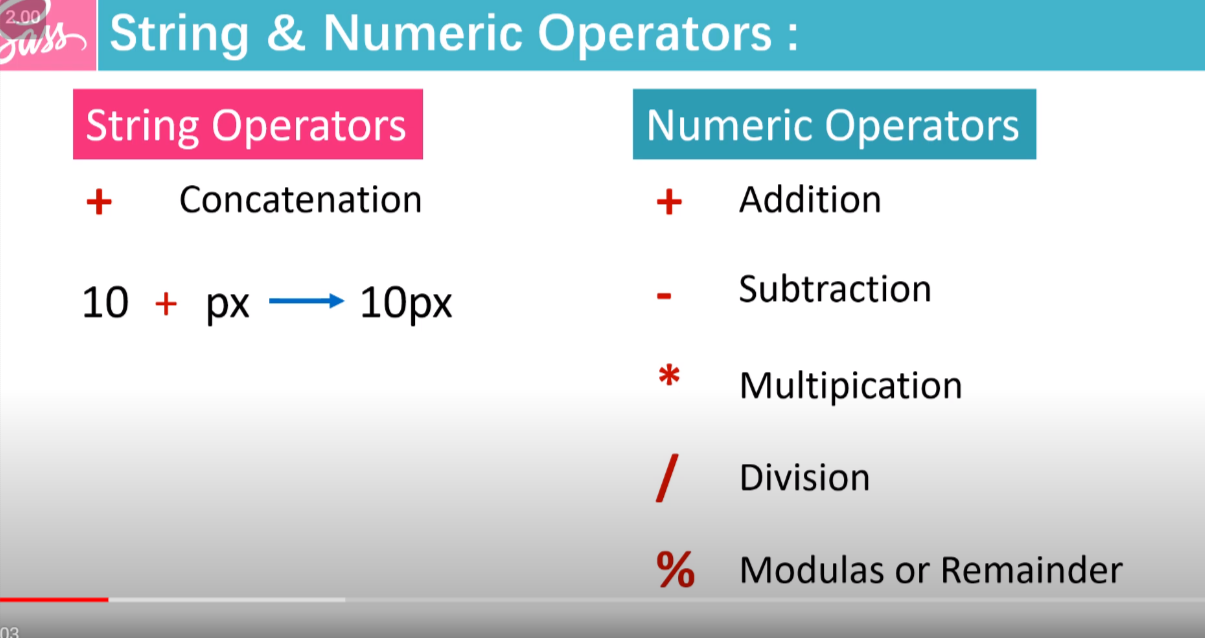
SCSS PROPERTY

1. OPERATOR :









We can do addition ,subtraction ,multiplication and all operation in scss file that we had made .

Code in main.scss

h1{

*color*:red;

*font-size*: 5rem +5rem;

}

Code in main.css

h1 {

*color*: red;

*font-size*: 10rem;

}

2. Variables

If I want to use a particular things various no of times so I can define

Variable with : $variable\_name :variable\_value ;

For example I want to use a particular colour for various no of time :

Defining :

$mycolor:#27ae60;

Using :

.main-header{

*background-color*: $mycolor;

}

3. nesting :

Html file :

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>SCSS Tutorial</title>

    <link rel="stylesheet" href="sass/main.css">

</head>

<body>

    <header class="main-header">

        <h1>Welcome to my channel</h1>

        <p>please like and subscribe</p>

    </header>

</body>

</html>

Scss file :

\*{

*padding*: 0;

*margin*: 0;

*box-sizing*: border-box;

}

html{

*font-size*: 62.5%;

}

$mycolor:#ecf0f1;

.main-header{

*width*: 100vw;

*height*: 100vh;

*display*: flex;

*justify-content*: center;

*align-items*: center;

*flex-direction*: column;

*background-color*:$mycolor;

    h1{

*color*:red;

*font-size*: 5rem +5rem;

*transition*: 0.3s ease;

        &:hover{

*color*: #9b59b6;

        }

    }

    p{

*color*: #16a085;

*font-size*: 5rem+5rem;

*transition*: 0.3s ease;

        &:hover{

*color*: #e84393;

        }

    }

}

4. Mixin:

We use this if we are a set of code again and again then we create a mixing variable :

E.g. :

Creating a mixin

@mixin flex{

*display*:flex;

*justify-content*: center;

*align-items*: center;

*flex-direction*: column;

}

Using mixin

.main-header{

    @include flex;

}

4.1 Mixin with parameter :

Some time we need to use certain code same for many classes but with different parameter so we can do with mixin as well :

@mixin flex($para1){

*display*:flex;

*justify-content*: center;

*align-items*: center;

*flex-direction*: column;

*letter-spacing*: $para1;

}

Partials:

We can create a different page for variable , mixin media query and then I can call them .

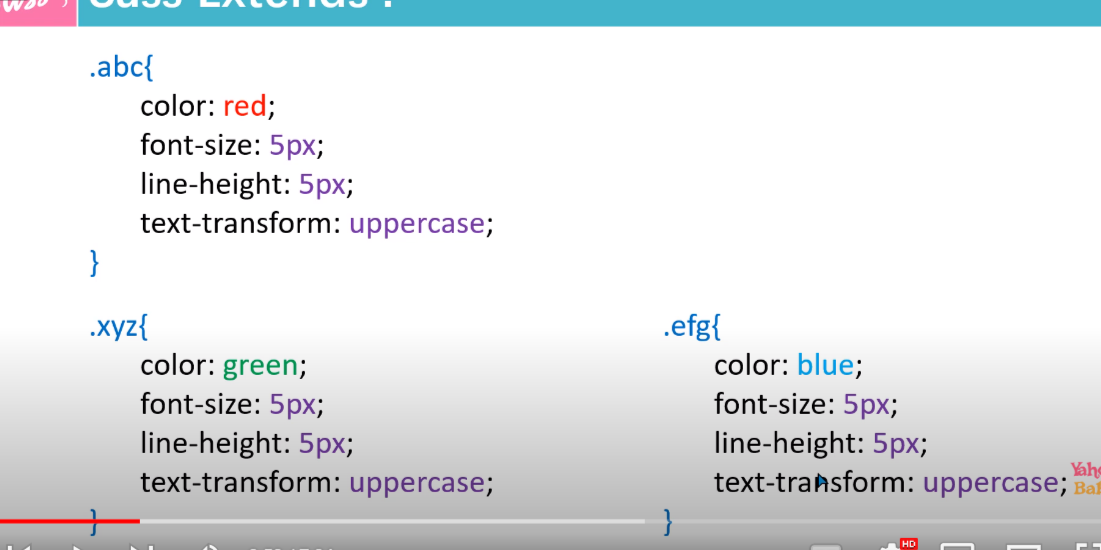
Note : for creating partials file it should be start with udersocre (\_\_) and with the extension .scss

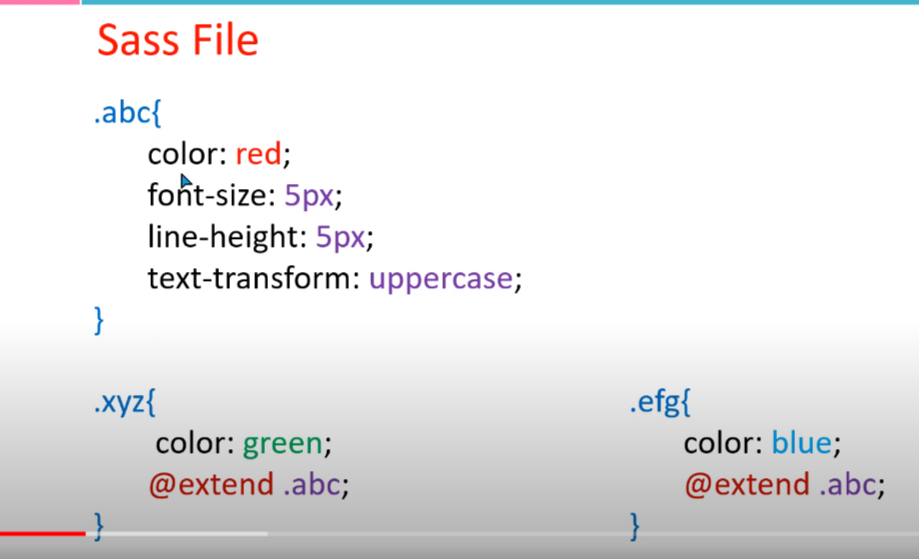
We need to import each seprate file in main.

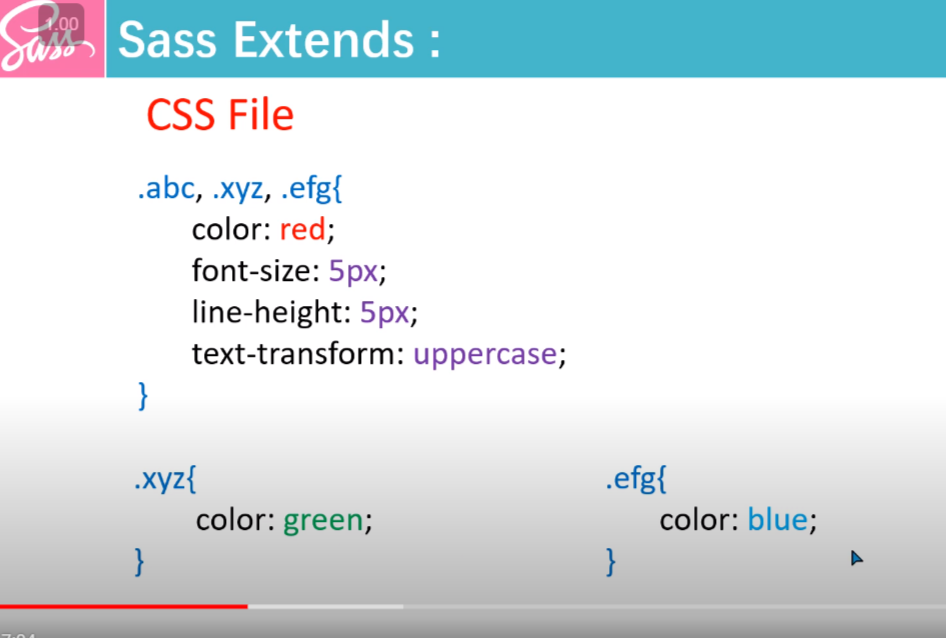
@import '/sass/mixin';

@import '/sass/variable';

6. extends

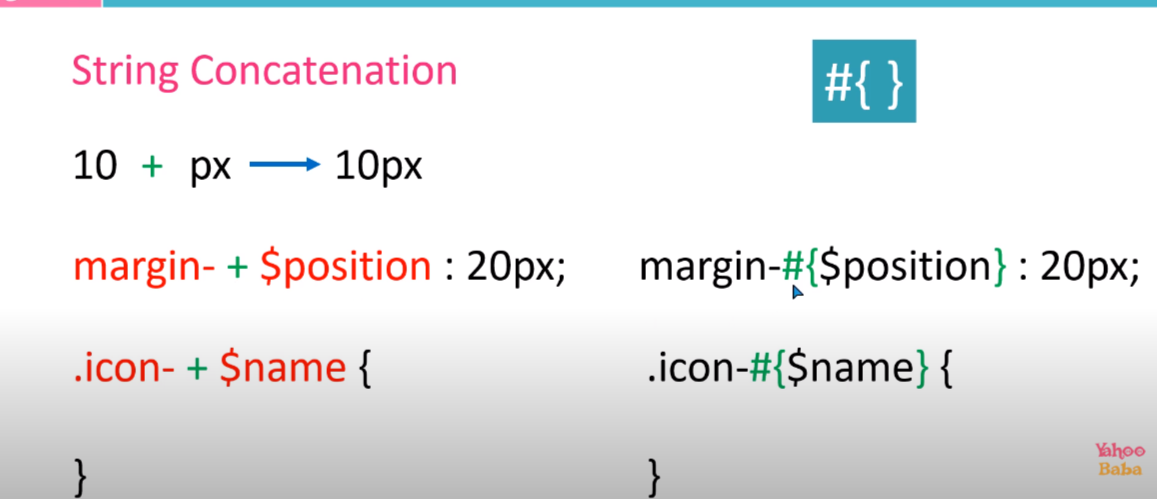






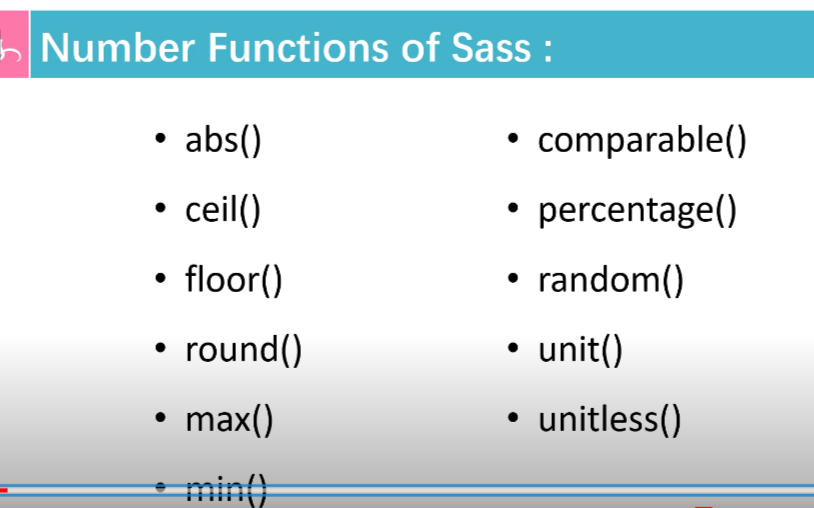
7.Interpolation

If we want to concat two property we can do this by the help pf interpolation.



8. Functions :

A . number function :



1.abs() going to give absolute value whether the input is negative or positive.

2.ceil () : it will give upper value of floating value

3.floor():it will give lower integer of a floating no .

4.round() : if the floating value is less than 0.5 it will give floor value other wise it will give ceil value .

5.max(num1,num2,num3….) : It will give maximum of the input

6.min(num1,num2,num3,…): it will give minimum of the input

7.comparable(value1,value2) :it will true/false checking if the value is comparable or not .

8.percentage(): it will give the percentage of what ever the input will be .

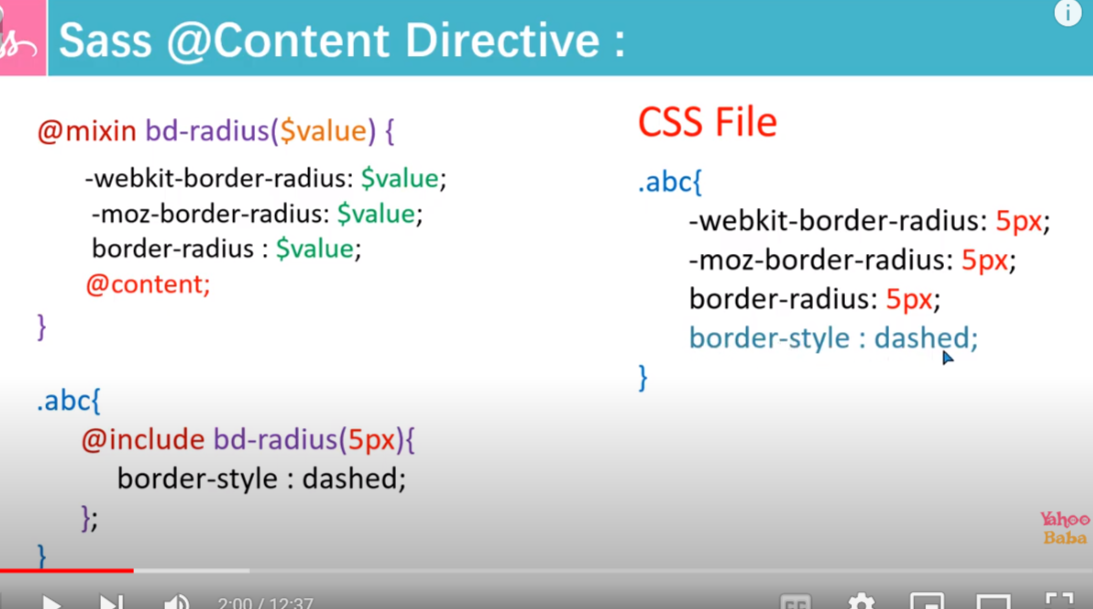
9.random():it will give the random value

10:unit():it will going to give the unit of input .

11unitless():check wheter the input Is unit less or not .

9. content-directive :

If we want to send property with the mixin the we can do that with the help of content directive .



10. at-root directive :

