What is SASS?

SASS (Syntactically Awesome Stylesheet) is a CSS pre-processor which helps to reduce repetition with CSS and saves time. It is more stable and powerful CSS extension language that describes style of document cleanly and structurally.

Why to use SASS?

- It is pre-processing language which provides indented syntax (its own syntax) for CSS.
- It allows writing code more efficiently and easy to maintain.
- It is super set of CSS which contains all the features of CSS and is an open source pre-processor, coded in Ruby.
- It provides document style in good structure format than flat CSS.
- It uses re-usable methods, logic statements and some of the built in functions such as color manipulation, mathematics and parameter lists.

List out some features of SASS?

- It is more stable, powerful and compatible with versions of CSS.
- It is super set of CSS and is based on the JavaScript.
- It is known as syntactic sugar for CSS which means it makes easier way for user to read or express the things more clearly.
- It uses its own syntax and compiles to readable CSS.
- You can easily write CSS in less code within less time.
- It is an open source pre-processor which is interpreted into CSS.

Difference between SASS and SCSS?

SCSS stands for Sassy CSS, and it keeps all the traditional CSS syntax, but adds in the Sass superpowers.
The biggest difference between the two is Sass (using .sass file extension) uses indentation instead of curly braces and semi-colons.
SCSS introduced in CSS3 Advance topics and majorly these both terms use interchangeably. Not major difference.
For more on difference read these:
Difference
Testing SASS : Sassmiester
More example:https://www.javatpoint.com/sass-vs-scss

How to use SASS with our IDE'S:

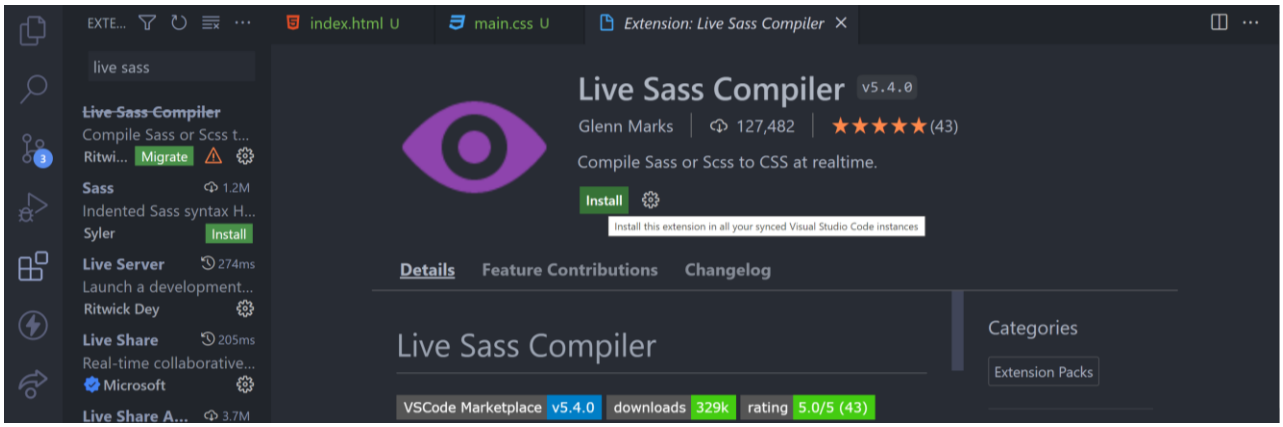
Many ways:

- By command line (mostly used)
- By ruby language
- By using SASS VS Code extension(I am following this)

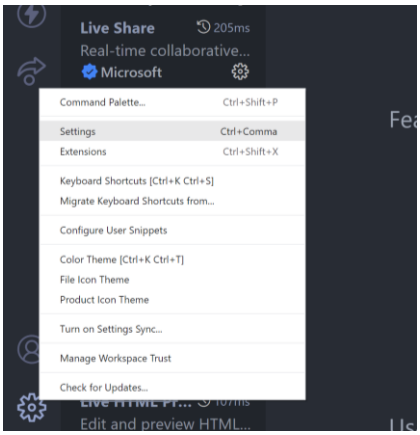
Check at freecodecamp also used same terminology and example website

1. Integrating Sass within Visual Studio Code

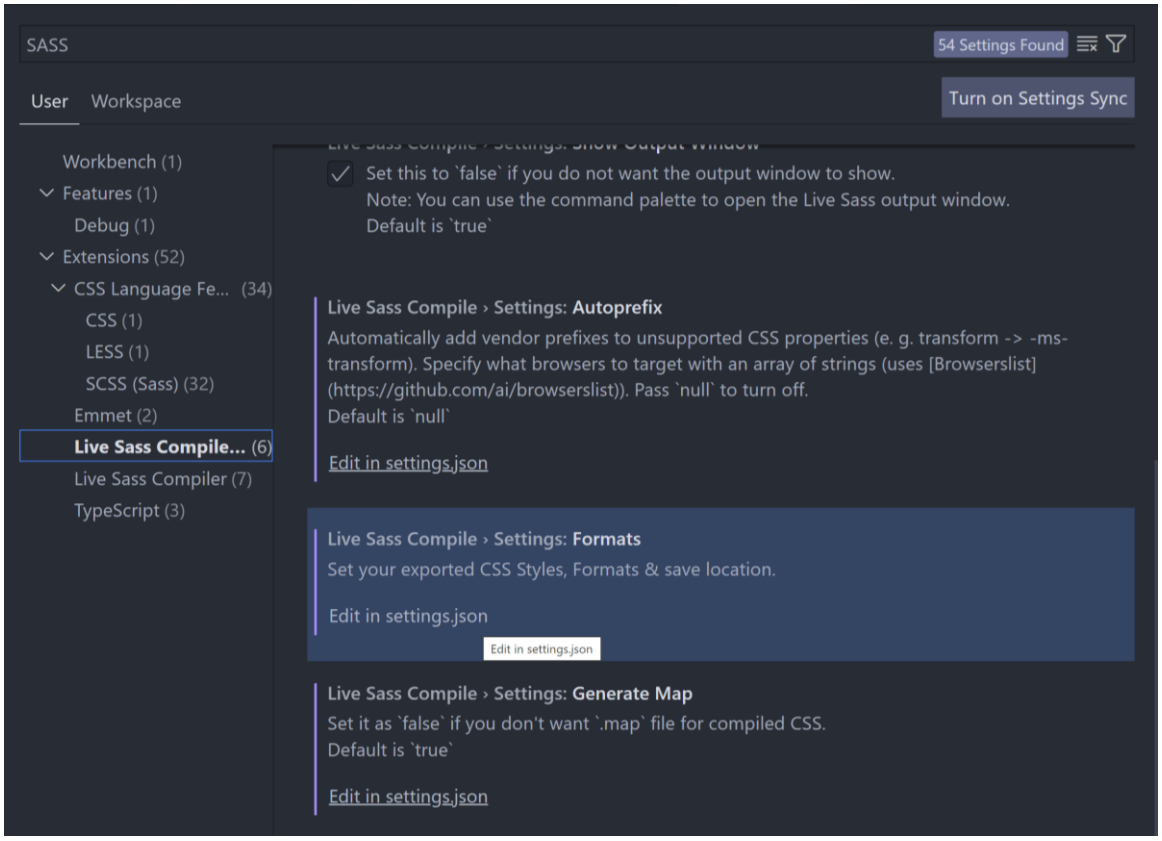
Open extension tabs and search: Live SASS Compiler



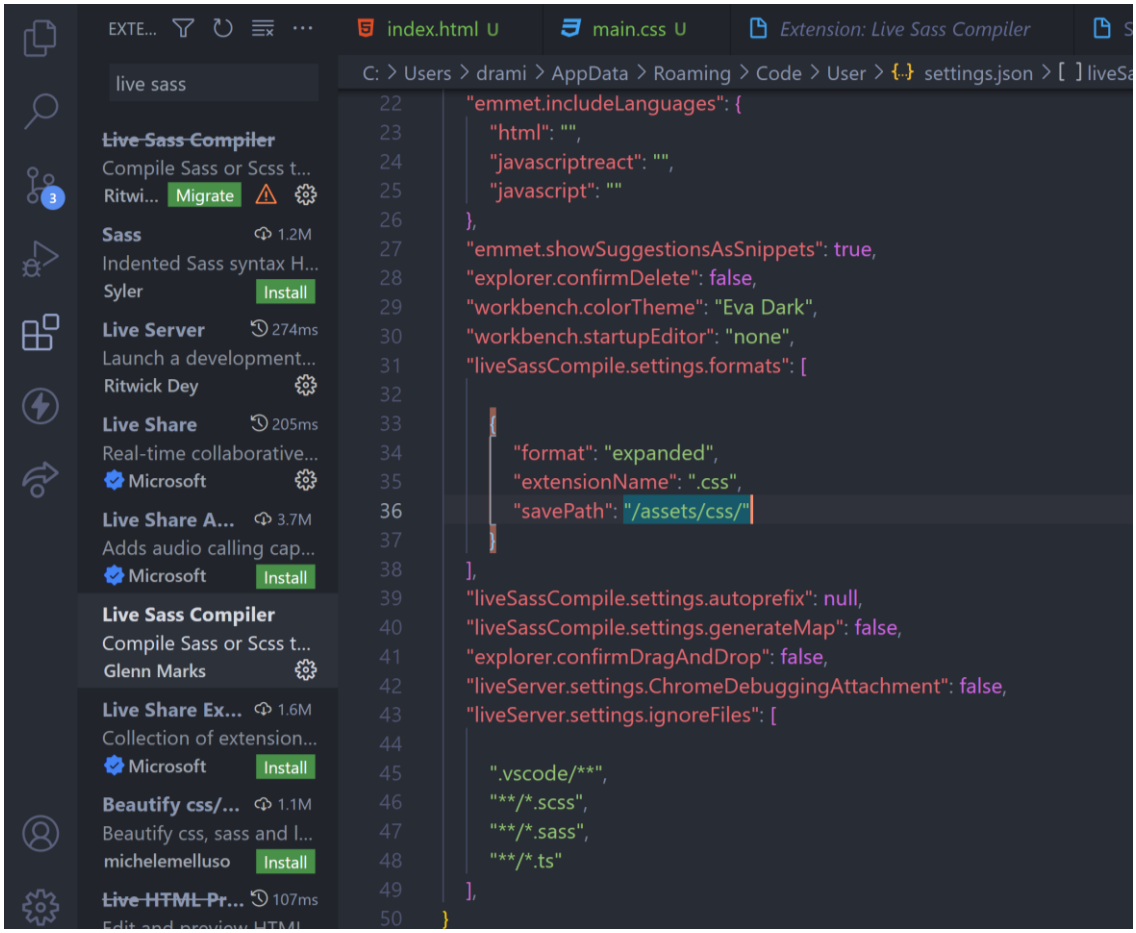
Go to Settings



Search SASS Compiler and then Format json file:

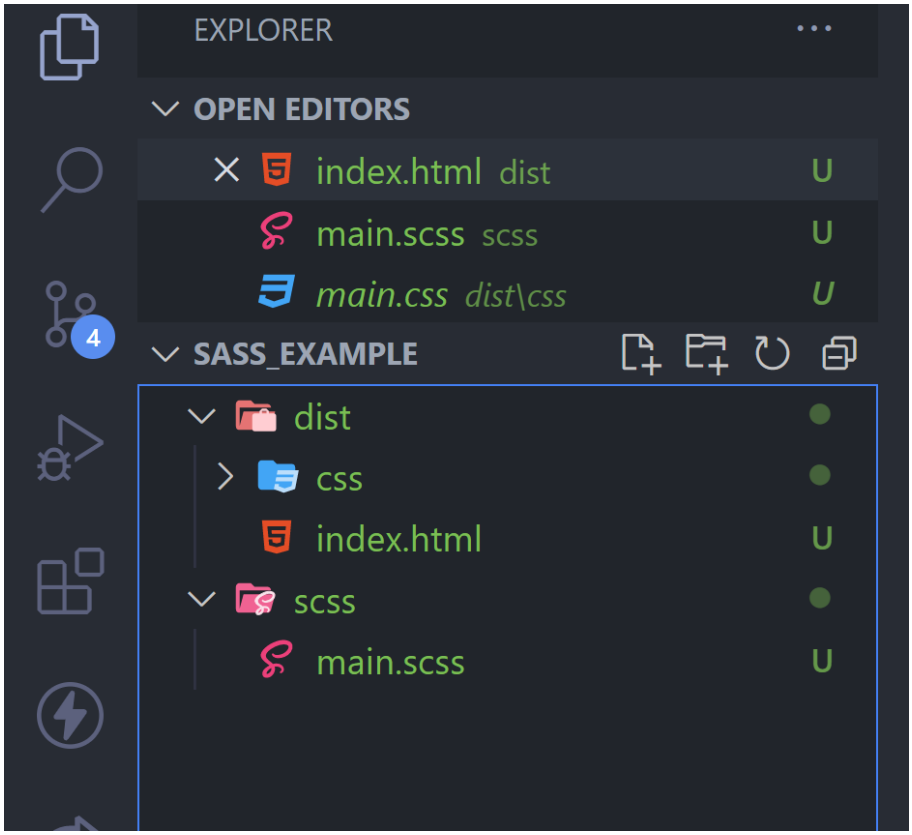


Change the default path settings:



```
//This is your change save path
      {
          "format": "expanded",
          "extensionName": ".css",
          "savePath": "/dist/css"
      },
```

Now make an empty folder with name and open it with VS Code and then create a folder dist ,css folder,index.html,scss with main.css file folder like this:



Now write in index.html file and link stylesheet : don't create css file it as it will be created automatically by scss compiler

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

Now look at two syntax of SASS Pre-processor:

SCSS SYNTAX

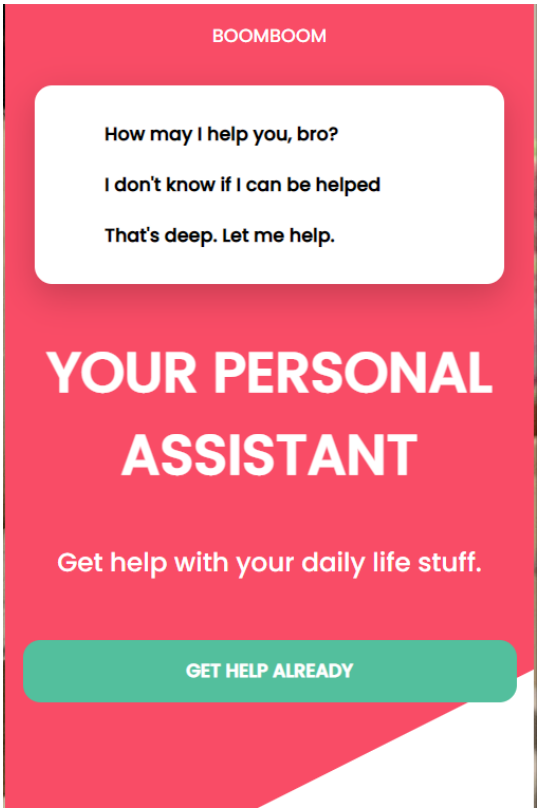
```
@mixin button-base()
{
@include typography(button);
display:inline-flex;
position:relative;
height:$button-height;
border:none;
vertical-align:middle;
&:hover{cursor:pointer;}
}
```

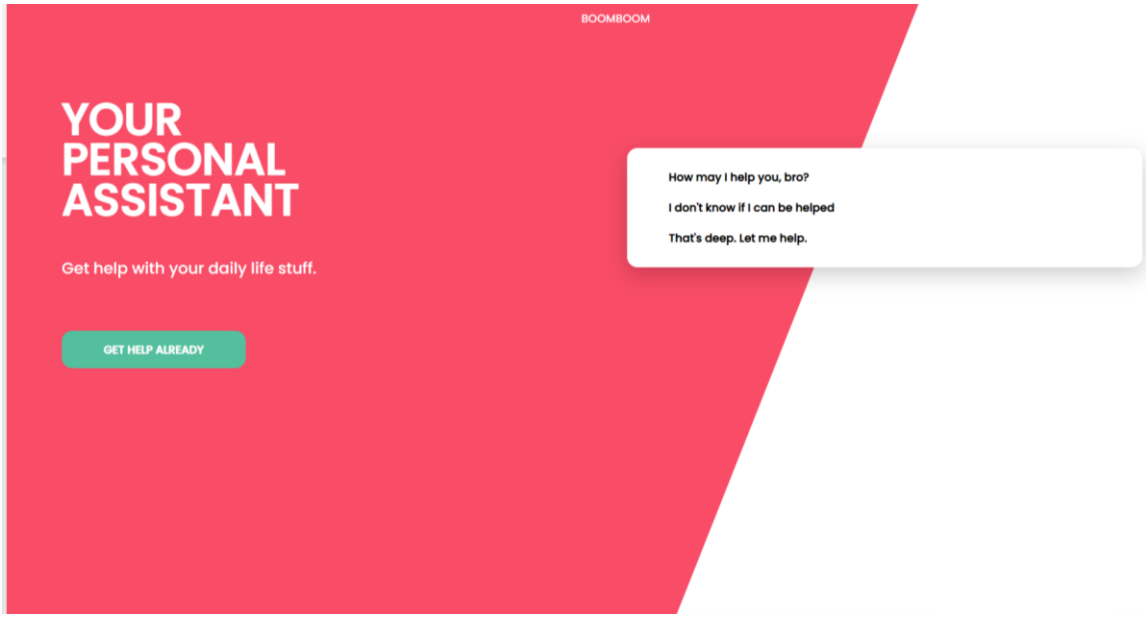
SASS ORIGINAL SYNTAX

```
INDENTED SYNTAX
@mixin button-base()
@include typography(button)
display:inline-flex
position:relative
height:$button-height
border:none
vertical-align:middle
&:hover
 cursor:pointer
```

Example:

Create the webpage like this by using SCSS pre-processors. First Create mobile version:





Colour palette:
https://colorhunt.co/palette/53bf9df94c66bd4291ffc54d

First write in index.html file:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>SASS Example</title>
    <link rel="stylesheet" href="css/main.css">
</head>
<body>

    <div id="bg"></div>

    <header>
        <a href="#">BoomBoom</a>
    </header>

    <main>
        <section id="card">
            <ul>
                <li>
                    <span></span>
                    <strong>How may I help you, bro?</strong>
                </li>
                <li>
```

```
                    <span></span>
                    <strong>I don't know if I can be helped</strong>
                </li>
                <li>
                    <span></span>
                    <strong>That's deep. Let me help.</strong>
                </li>
            </ul>
        </section>
        <section id="primary">
            <h1>Your Personal Assistant</h1>
            <p>Get help with your daily life stuff.</p>

            <a href="#">Get help already</a>
        </section>
    </main>

</body>
</html>
```

Apply now in main.scss:

SASS Variables:

```
$mycolor: #F94C66;


body{
    background-color:$mycolor;
}
```

You can create as many variables and pass to the properties.

SASS Maps: Helps in organising

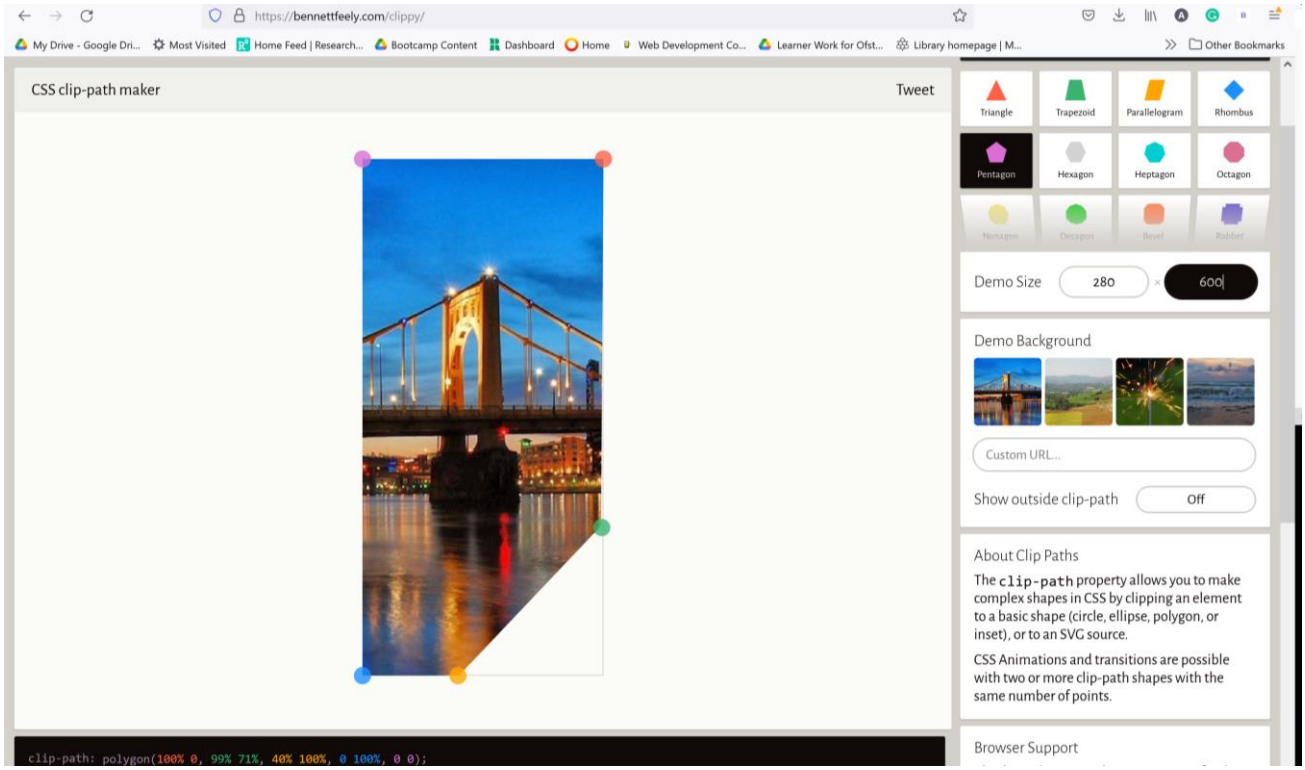
We want to create different type of colors so how it works:

```
$colors:(
   primary: #F94C66,

   accent:#FFF6BB
);
body{
   background-color:map-get($colors,primary);
}
```

Now in order to make mobile app style we need to use some properties :**Clip path (Very handy)**

Goto https://bennettfeely.com/ and search Clippy

Choose pentagon and change height to 600 and then create the shape:



Now go to main.scss file:

Add these:

```
@import
url('https://fonts.googleapis.com/css2?family=Mochiy+Pop+P+One&family=Nunito&f
amily=Poppins:wght@300;500&family=Roboto+Slab&family=Roboto:wght@300;700;900&d
isplay=swap');


body, html {
    height: 100%;
}

body {
    font-family: 'Poppins';
    margin: 0;

    #bg {
        clip-path: polygon(100% 0, 100% 82%, 45% 100%, 0 100%, 0 0);
```

```
        background-color: color(primary);
        width: 100%;
        height: 100%;
        position: absolute;
        z-index: -1;

    }
```

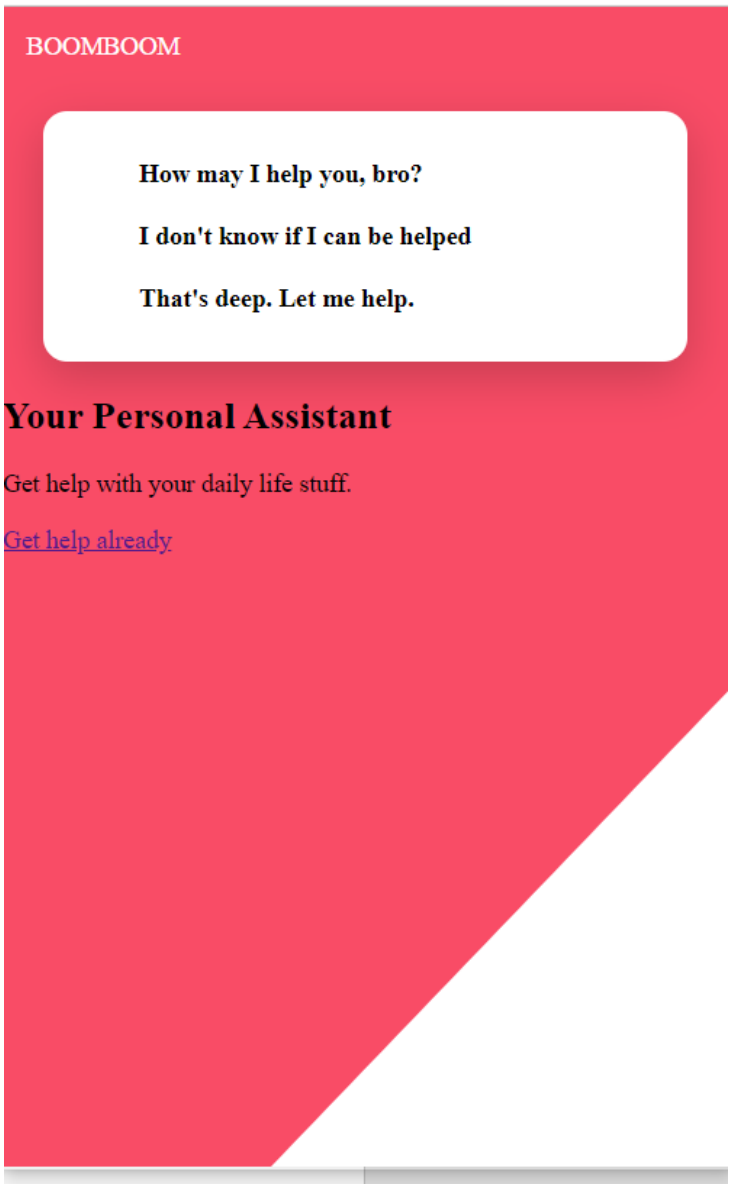
Next Part:

Now adding SCSS for ul ,li and span ,the SCSS file looks like this :

```
body {
  font-family: "Poppins";
  margin: 0;
}
body #bg {
  clip-path: polygon(100% 0, 100% 59%, 37% 100%, 0 100%, 0 0);
  background-color: #F94C66;
  width: 100%;
  height: 100%;
  position: absolute;
  z-index: -1;
}
body header a {
  color: #fff;
  text-decoration: none;
  padding: 15px;
  display: block;
  text-transform: uppercase;
}

main section#card {
  background: #fff;
  padding: 20px;
  margin: 1em auto;
  border-radius: 15px;
  box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2);
  width: 80%;
}
main section#card ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}
main section#card ul li {
  margin-bottom: 10px;
}
```

```
main section#card ul li span {
  position: absolute;
  width: 30px;
  height: 30px;
  background-color: color(primary-light);
  border-radius: 50%;
  margin-right: 10px;
}
main section#card ul li strong {
  display: inline-block;
  margin-left: 40px;
  margin-top: 10px;
}
```

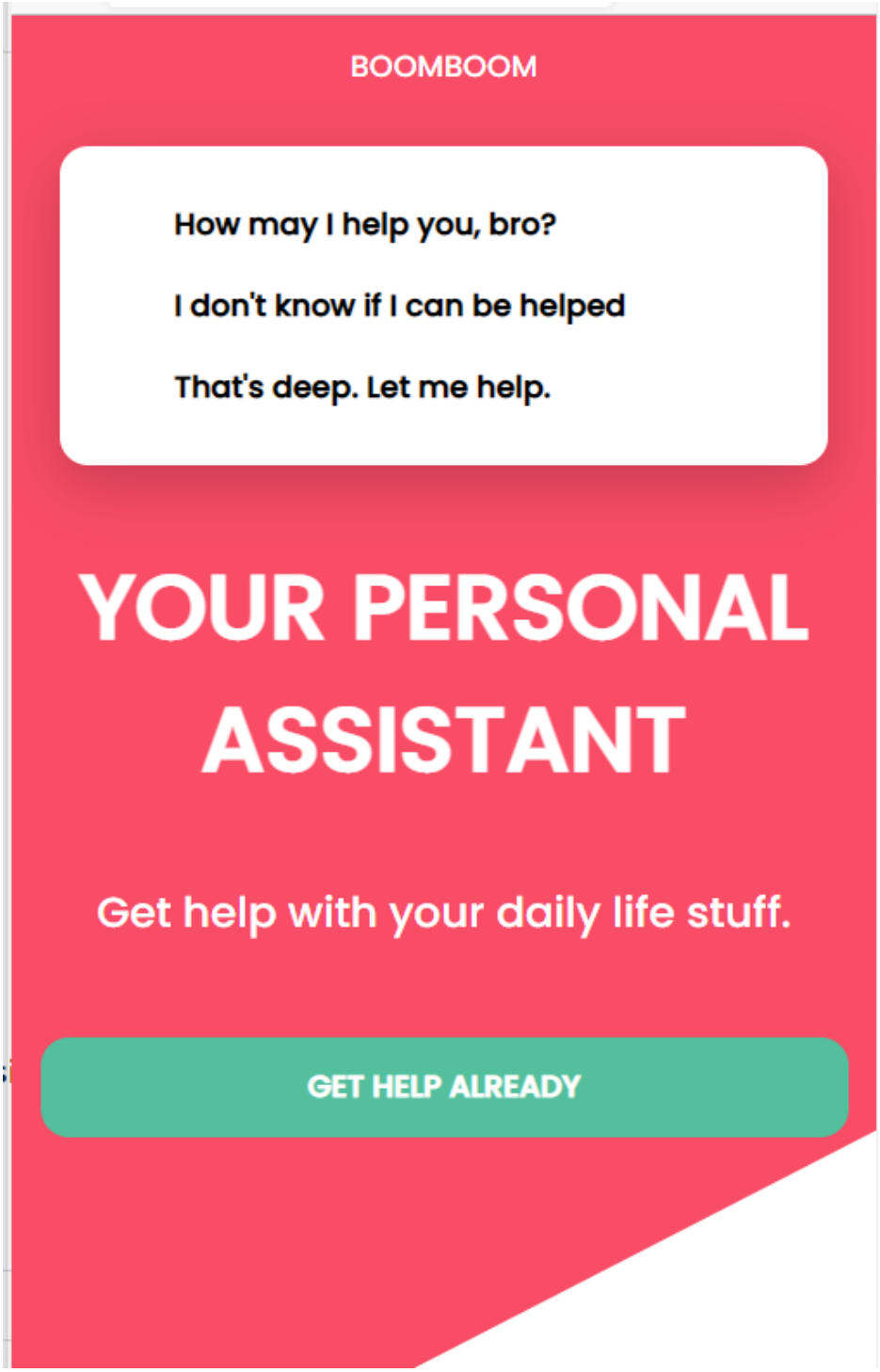
The output looks like this:



Now adding Section primary code :Insert inside main {}:Nesting Code : ✓Nesting - Sass lets you nest CSS selectors in the same way as HTML. Simple! Be careful not to over nest.

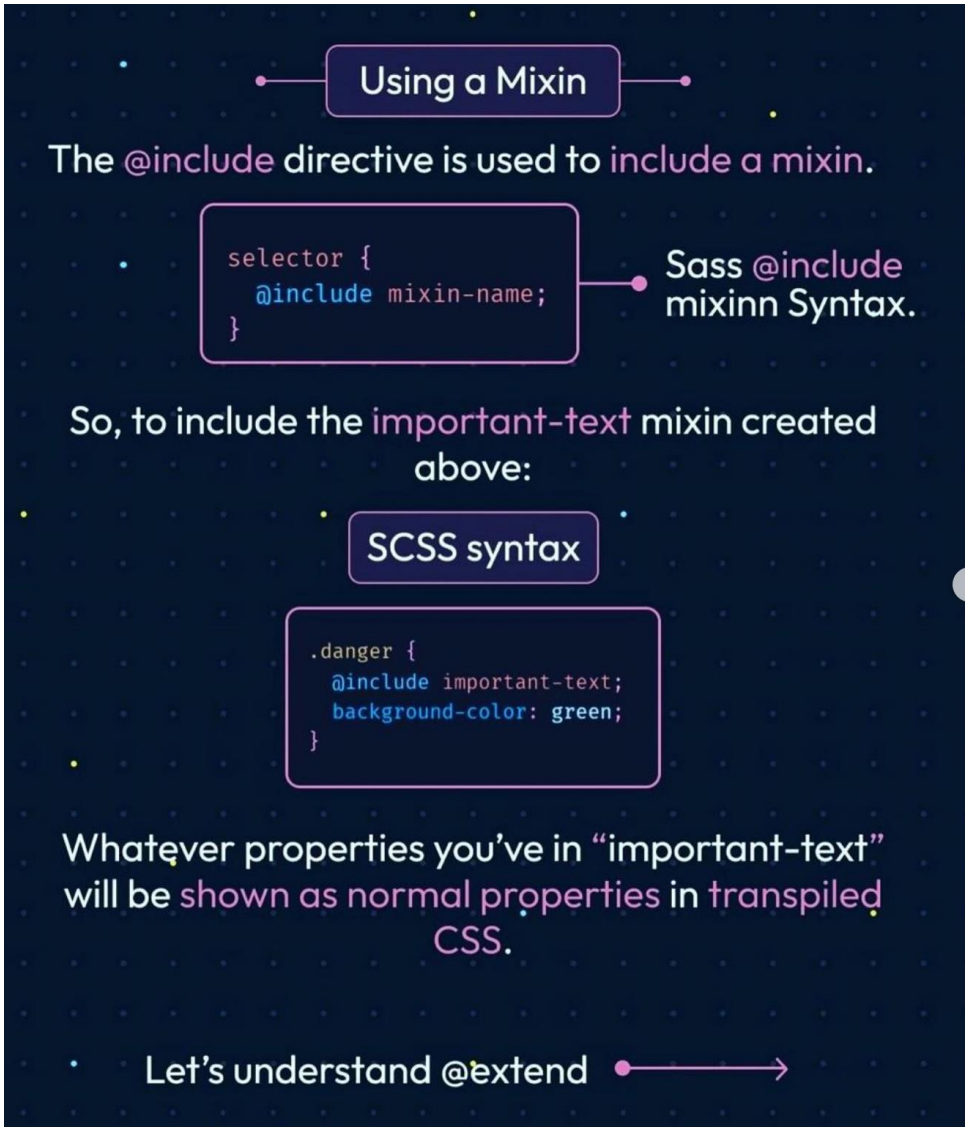
```
section#primary {
        color: #fff;
        padding: $padding;
        text-align: center;


        h1 {
            font-size: 3em;
            margin-top: 10px;
            text-transform: uppercase;


        }

        p {
            font-size: 1.4em;
        }

        a {
            color: color(primary-light);
            border-radius: $borders;
            text-decoration: none;
            text-transform: uppercase;
            font-weight: bold;
            background-color: color(accent);
            display: block;
            text-align: center;
            margin: 50px auto 0 auto;
            padding: $padding;


        }
    }
```

Also add the @import for google fonts



5. Now next topic in SCSS is about Mixin:

✓Mixins- lets you make groups of CSS declarations you want to reuse throughout your site. It also makes it easy to avoid using non-semantic classes.



We are using in our project by applying media queries:

We need to define at which point should the mobile view goes to the desktop view.

```
//Mixin helps in important selector sections

$desktop: 840px;

@mixin desktop {
    @media (min-width: #{$desktop}) {
        @content; //this specify whereever element we applying
this rule
    }
}
```

Now we have to apply it to the bg background that we need to change our clip-path to the desktop version. Go back to bennetfeely.com and now change canvas to 500w to 280h and change the points

```
@include desktop {
            clip-path: polygon(0 0, 75% 0, 55% 100%, 0% 100%);
        }
```

We have to use it at various places and totally up to us which breakpoints we want :

Add this to main :

```
   @include desktop {
        display: grid;
        grid-template-columns: 50% auto;
        grid-template-areas:
            "primary card";
    }
```

And also on section#card:

```
@include desktop {
            grid-area: card;
            height: fit-content;
            align-self: center;
            margin: 1em;
        }
```

The final SCSS Looks like this:

```
@import
url('https://fonts.googleapis.com/css2?family=Mochiy+Pop+P+One&family=Nunito&f
amily=Poppins:wght@300;500&family=Roboto+Slab&family=Roboto:wght@300;700;900&d
isplay=swap');
$colors: (
    primary: #F94C66,
    primary-light: lighten(#F94C66, 40%),
    primary-dark: darken(#F94C66, 40%),
    accent:#53BF9D
);

$padding: 15px;
$borders: 15px;
```

```scss
@function color($color-name) {
    @return map-get($colors, $color-name)
}

$desktop: 840px;

@mixin desktop {
    @media (min-width: #{$desktop}) {
        @content;
    }
}

body, html {
    height: 100%;
}

body {
    font-family: 'Poppins';
    margin: 0;

    #bg {
        clip-path: polygon(100% 0, 100% 82%, 45% 100%, 0 100%, 0 0);
        background-color: color(primary);
        width: 100%;
        height: 100%;
        position: absolute;
        z-index: -1;

        @include desktop {
            clip-path: polygon(0 0, 75% 0, 55% 100%, 0% 100%);
        }
    }

    header a {
        color: #fff;
        text-decoration: none;
        padding: $padding;
        display: block;
        text-transform: uppercase;
    }
}

main {

    @include desktop {
        display: grid;
        grid-template-columns: 50% auto;
        grid-template-areas:
```

```
        "primary card";
}

section#card {
    background: #fff;
    padding: 20px;
    margin: 1em auto;
    border-radius: $borders;
    box-shadow: 0 10px 30px rgba(0,0,0,.2);
    width: 80%;

    @include desktop {
        grid-area: card;
        height: fit-content;
        align-self: center;
        margin: 1em;
    }

    ul {
        list-style-type: none;
        margin: 0; padding: 0;

        li {
            margin-bottom: 10px;

            span {
                position: absolute;
                width: 30px;
                height: 30px;
                background-color: color(primary-light);
                border-radius: 50%;
                margin-right: 10px;
            }

            strong {
                display: inline-block;
                margin-left: max(40px);
                margin-top: 10px;
            }
        }
    }
}

section#primary {
    color: #fff;
    padding: $padding;
    text-align: center;
```

```
        @include desktop {
            grid-area: primary;
            text-align: left;
            margin: 4em 0 0 4em;
        }

        h1 {
            font-size: 3em;
            margin-top: 10px;
            text-transform: uppercase;
        }

        p {
            font-size: 1.4em;
        }

        a {
            color: color(primary-light);
            border-radius: $borders;
            text-decoration: none;
            text-transform: uppercase;
            font-weight: bold;
            background-color: color(accent);
            display: block;
            text-align: center;
            margin: 50px auto 0 auto;
            padding: $padding;

            @include desktop {
                display: inline-block;
                padding: $padding $padding * 4;
            }
        }
    }
}
```