Contents

[General 12-06-2019 2](#_Toc11222663)

# General 12-06-2019

I found styled-components to be useful. You can have all the styles along with JS code.

Install by npm install styled-components

Import styled from ‘styled-components’;

* Just like scss you can define nested selectors such as follows

const StyledMenuItems = styled.ul`

display:flex;

flex-direction:column;

width: 60%;

a {

text-decoration:none;

display:block; // to make whole area clickable

margin-left:1.3rem;

margin-top:0.3rem;

margin-bottom:0.3rem;

font-size:1.3rem;

color:#fff;

}

li {

list-style-type: none;

border: 1px solid white;

border-collapse: collapse;

}

@media only screen and (min-width: 993px) {

        flex-direction:row;

align-items:center;

li {

border:0px;

}

    }

`

* You can use any html tag as styled.ul as I have done in above code. Another styled component can be nested in a styled component.

const StyledName = styled.span`

color:red;

margin-right:0.5rem;

`

const StyledItem = styled.div`

display:flex;

flex-direction:column;

font-size: 1.1rem;

${StyledName}{

font-size:4rem;

}

`

* React props can be used in passing parameters to a styled component. For that you need to use Interface if you are using TypeScript.
* Also you can extend an already created styled component. Here Button2 is extended from Button1

interface **Props** {

    myColor: string;

  }

const padding = '3em'

const Button1 = styled.**button**<**Props**>`

background-color: magenta;

padding: ${padding};

color: ${(props:any) => props.myColor};

`

const Button2 = **styled**(Button1)`

padding: 1em;

`

export {Button1, Button2}

<**Button1** *myColor*='green'>Click</**Button1**>

<**Button2** *myColor*='black'>Another button</**Button2**>

Styled component can be extended from another styled component: