Built with no framework and no use of bootstrap at all. Use only HTMl and CSS.

1. Setup
2. Requirement

Node JS

* Sass node >>> <https://github.com/sass/node-sass>
* Git
* Shell command

Extensions

* Live server
* Bracket Pair colorizer
* Prettier code formatter – set the editor.format to save in the settings

1. Initiate npm >>> npm init

* Install node-sass >>> npm i node-sass; it will create a node\_modules folder, the grocery
* Configure package.json >>> change the “test” to “sass” = “node-sass -w scss/ -o dist/css/ --recursive”
* Npm run sass >>> to start

1. Create a git repository

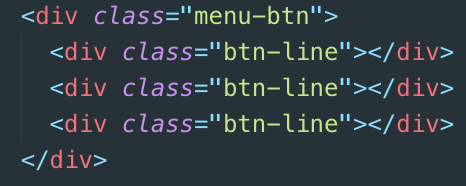
* Touch .gitignore >>> add node\_modules to be ignore
* Git init
* Git add .
* Git commit -m “”

1. Sources

* Images >>> pexels.com

1. Creation of the Main Page

* 3 horizontal lines



* Menu branding



* Home Page (Home, about, work and contact)
* Insert the font awesome cdn >>> <script src="https://kit.fontawesome.com/0d5cd337e9.js" crossorigin="anonymous"></script>



1. Setting up the SCSS

* Core style

Box-sizing; border-box; wrap the whole page so we can inject some padding that will not affect the height or width of the page.

&.lg-heading = nested css.

Margin-bottom is to push the elements beneath down

Padding: 2rem 1rem; means 2rem top and bottom and 1rem left and right. This is to give a light background in the sm-heading

Default html rem = 16px. Each rem is multiplied by 16

Background: rgba(lighten($primary-color, 2), 0.5) >>> lighten is a sass property. 2, is how much to lighten the original color which is the $primary-color. 0.5 is the opacity of the rgba to see the image background underneath.

Headings setup



* Setup the a’s

Text-decoration: none; will remove the underline like in each icon



* Setup the header

Position: fixed; >>> is to make the menu buttons always on top even if the user scroll

z-index: 2 = this is how to look like the element closer to the user



* Setup the $secondary-color for the text-secondary.
* Setup the main

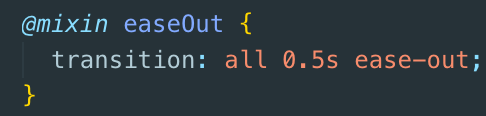
Padding: 4rem; will create 4rem space around each element inside the main. Main is the main page

@include easeOut – is the transition. It’s a @mixin variable that can be use anywhere

Overflow: hidden; no scroll bar in the homepage

Margin-top: 20vh; is to move the headings down. Vh is another unit. Whole page = 100vh, thus 20vh means 20 slices out fro 100

Vw is same as vh

* Setup the background image

Setup the variables:

$show-home-image: true;

$home-image: url(/img/background.jpg);

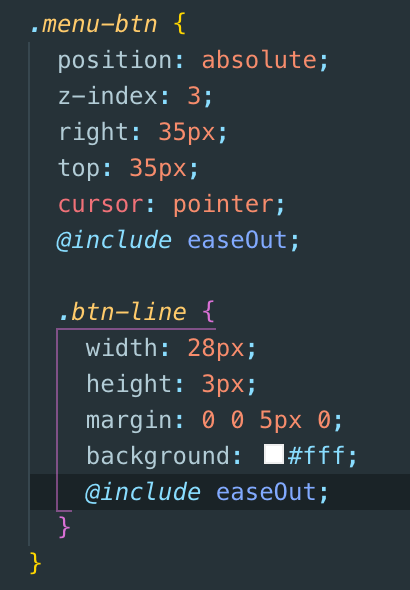
$background-opacity: 0.5;

And create the @mixin background so as not to clutter the body {}

.overlay – the function is to expose more the text and dig down the image background. Something like a protective screen protector

1. Position: absolute; to cover the whole body
2. Top, right, width and height makes the overlay to cover the whole screen
3. Baclground: rgba($primary-color, $background-opacity) --- color of the ovewrlay
4. Z-index: -1; >>> to dig down the overlay and moves up the text
5. .overlay and &:after, content:’’; is just the same. &:after made it cleaner as we don’t need anymore html div., a sudo element feature of sass.
6. Setting up the menu button JavaScript

* Create the new .scss, \_menu.scss



* The 3 mini horizontal lines should now appear.
* Setting up the javaScript @ main.js for the e horizontal lines
* Use document.querySelector to select all the elements in the index.html
* Menu-btn = 3 tiny horizontal lines
* Nav-item = for the home, about, work, contact
* toggleMenu – is the function where in the menu will be displayed or not displayed when the 3 tiny horizontal lines are toggled.
* 
* No change in the UI yet. Above JS defines first the behavior of the menu-btn
* Sass will handle the DOM behavior

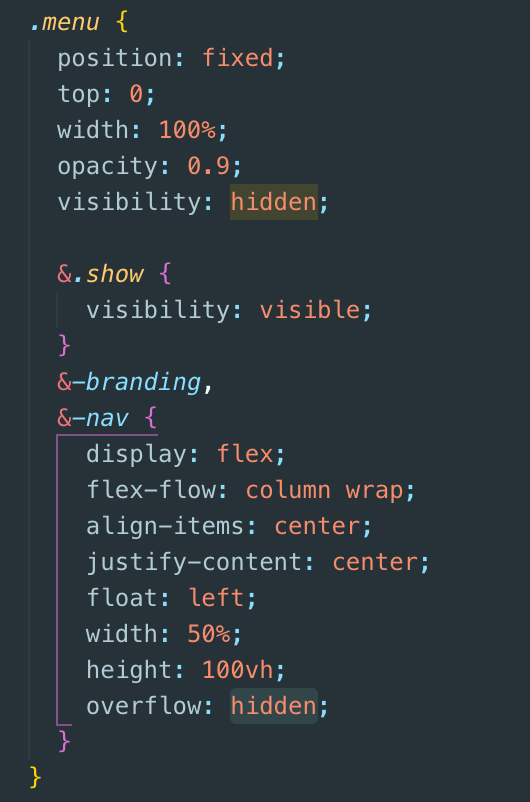
1. Setting up the UI behavior via above JavaScript
2. Transform the three horizontal lines to “X” by means of transform and translate in CSS

* CSS effects, transform, translate– “<https://developer.mozilla.org/en-US/docs/Web/CSS/transform>”
* $.close{} is from the javascript main.js



1. Work on the overlay that will display the photo and the nav items when the menu button is press.

* Visibility: hidden; to hide the items
* &.show{} is from main.js
* &-branding {} is similar to .menu-branding



* **Continue overlay**

1. @media for webpage responsiveness

@mixin mediaSM = is for the css to be apply if the screen is 500px; and so on.

Determine the size in the \_cofig.scss

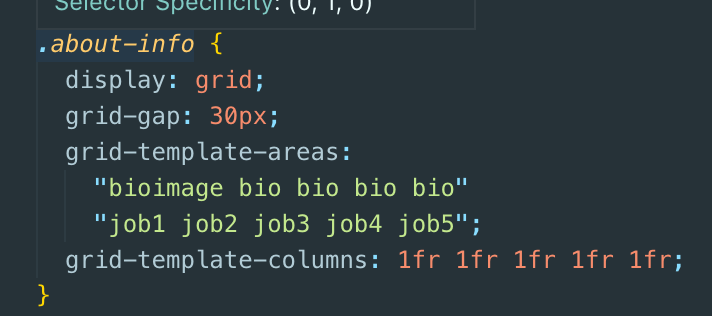
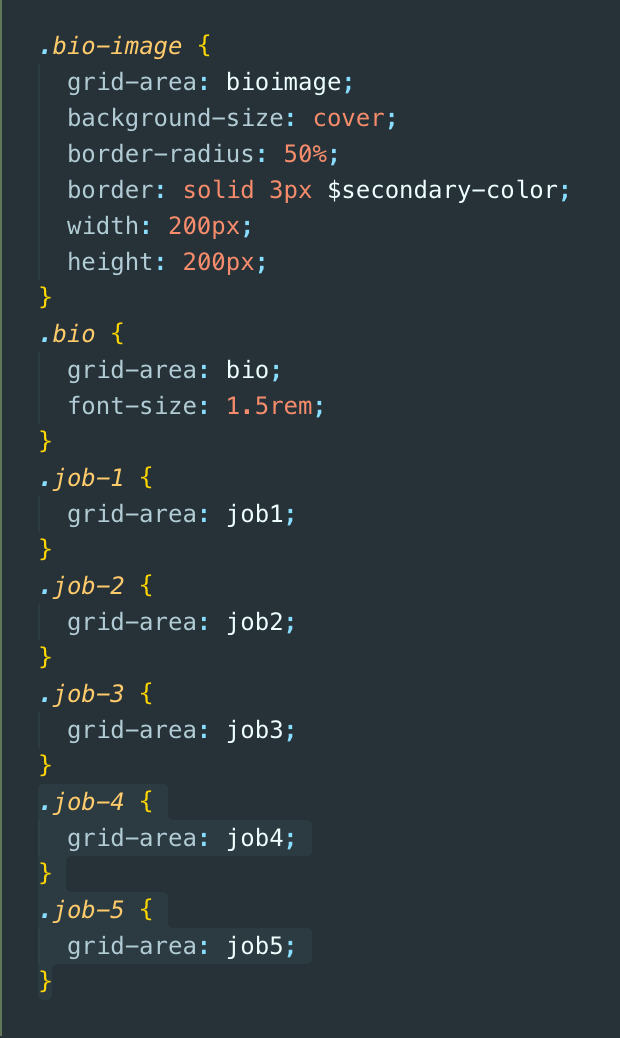
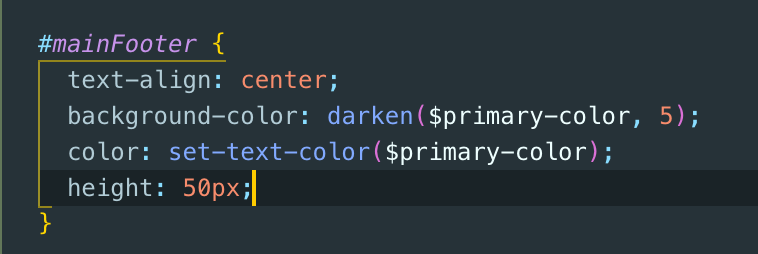
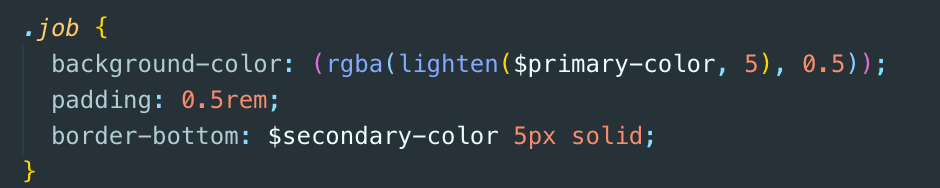


1. **Set text color depending on the theme color**

* The function accepts a value thru $color and test its lightness
* Then assign the function in the color: set0text-color($primary-color) and pass the $primary-color in the main.css also in .btn-line and nav-link of the \_menu.scss.



1. Create the about page in dist

* Change the title, main id to about. Transfer the “current to about to highlight about in the nav-link
* Remove the icons
* Add the footer
* Remove the background image bg-img
* Put a div .about-info, under h2. This will include all the contents of the about in grid display
* 1st feature is the image
* Bio
* Job-1 to job-3
* **Align using grid at main.scss**
* **Setup the about-info**
* 
* ‘bio-image bio bio bio bio’ etc is to visually display the layout of the grid
*  this is the definition of each grid area
* **Setup the #mainFooter**
* 
* **Create a sticky footer via calc in main {}**
* Adjust the height from 100% to min-height: calc(100vh – 60px)
* 
* Setup the job frames
* 
* Adjust the responsiveness

1. Work Page

* copy the about page as a new work.html
* set it up for work related inputs.
* Create the single format html and re-create this by 5.
* **Set the scss**
* Display as grid. 1fr 1fr 1fr and hover effect
* **Set the buttons**
* Create the mani .btn {}, so its reusable for both .btn-dark and btn-light
* Set the responsiveness
* For Xl – make it 4, 1fr
* Lg – 3, 1fr
* Md – 2, 1fr
* Sm – 1fr so that all will just stack at the center

1. Create the contact page by using flex box

* Copy the about html and change the inputs related to contact to set the contact html
* Set the main class as boxes
* Set the style in main.scss
* Display as flex, this will make it arrange horizontally
* &:hover is really good

1. Deployment

* Create a .ignore and ignore node\_module
* Git add .
* Git commit -m “”
* Git remote add origin [git@github.com:shernof/shernof.git](mailto:git@github.com:shernof/shernof.git)
* Git push -u origin master
* Npm install gh-pages
* Create the following in package.json
* "homepage": <https://shernof.github.io/shernof>
* "deploy": "gh-pages -d dist"
* Npm run deploy

Explore

<https://www.namecheap.com/>

<https://www.inmotionhosting.com/>