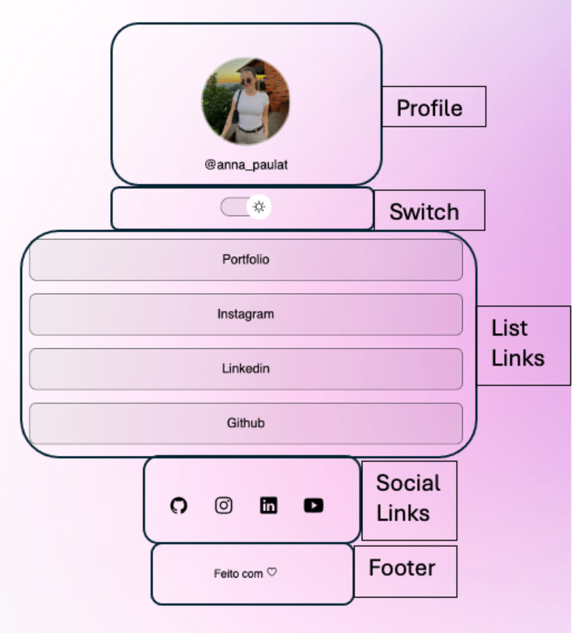
**DISCOVER - PORTFOLIO**

I divided my project into a container containing 5 parts:



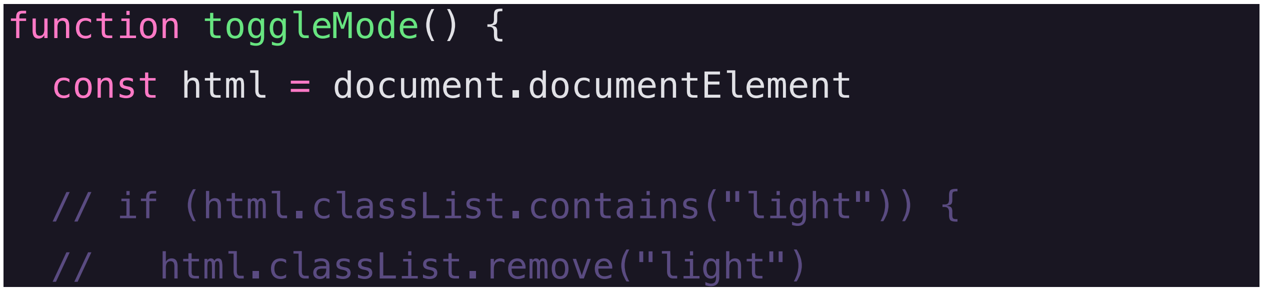
## 1. Profile

The profile only contains an image with my full name underneath.

## 2. Switch

For the switch, I created an onClick event called toggleMode in the button tag. So that when clicked, it changes between light/dark mode.

With the toggle mode function, I learned that instead of doing a whole if/else to add or remove the light class, we have the toggle option that does this:

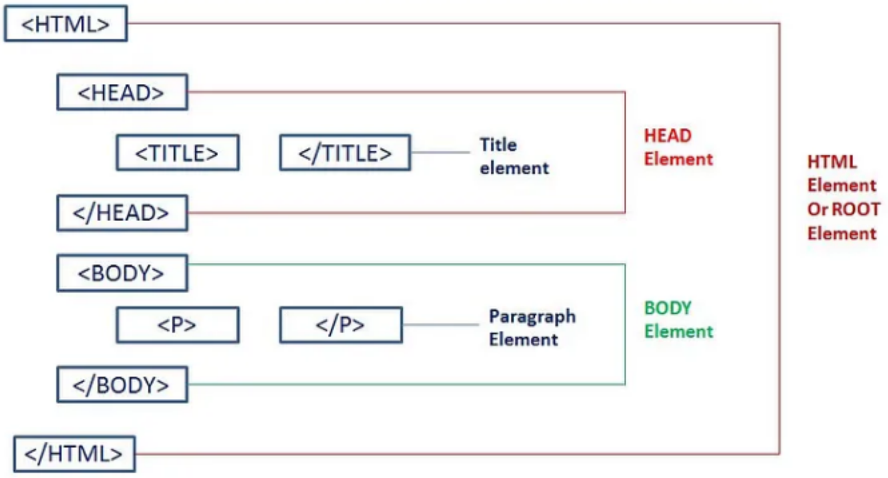


```
function toggleMode() {
  const html = document.documentElement

  // if (html.classList.contains("light")) {
  //   html.classList.remove("light")
  // } else {
  //   html.classList.add("light")
  // }

  // instead doing an if/else, we have a toggle method to help us:

  html.classList.toggle("light")
```

Furthermore, since I put the “light” class inside the html tag:

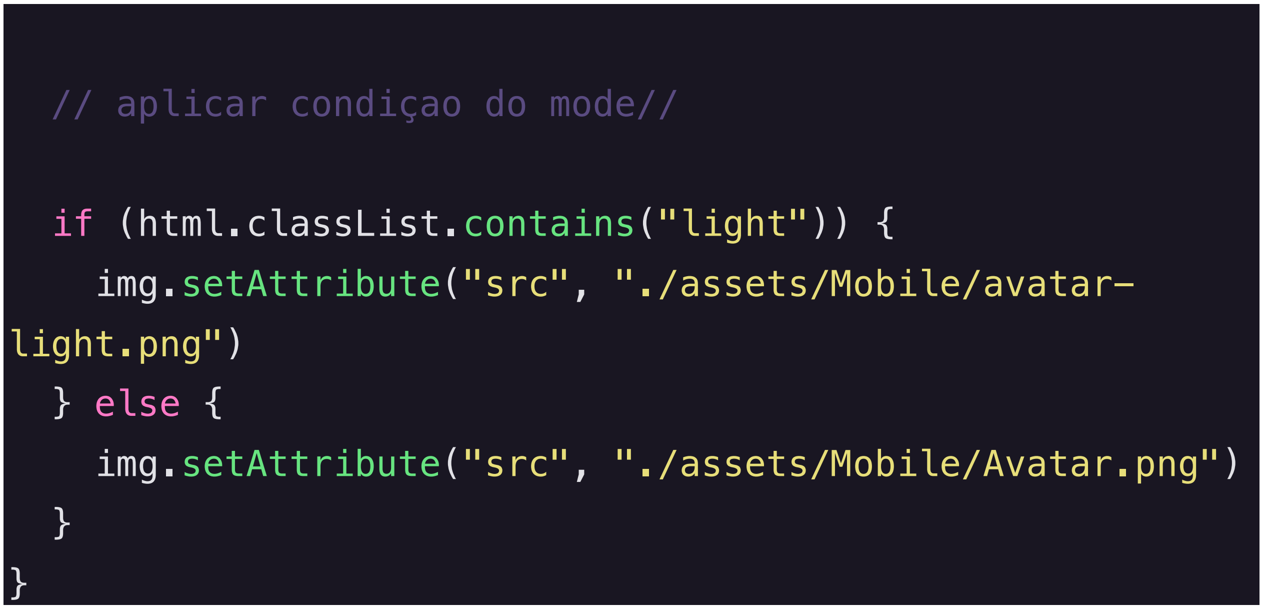
```
<html lang="pt-br" class="light">
  <head>
```

To check if the light class exists (constains) in the class list of the html tag, I use “document.documentElement”. But, if I had placed the class inside the body, it would be “document.documentBody”, and if it were in the header, it would be “document.documentHeader”. According to the image below:



I also learned that to change an attribute, I can use the “setAttribute. For example, to change the image, I called the setAttibute method for “img”. As a parameter, I passed what I want to change (src) with the value I want to set:

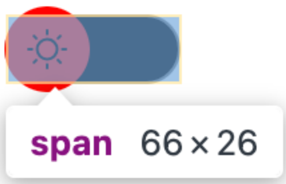
```
const img = document.querySelector("#profile img")
```

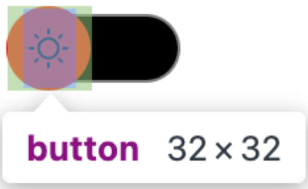


```
  // aplicar condiçao do mode//

  if (html.classList.contains("light")) {
    img.setAttribute("src", "./assets/Mobile/avatar-
light.png")
  } else {
    img.setAttribute("src", "./assets/Mobile/Avatar.png")
  }
}
```

For the switch customization part:

Inside the html, in the switch div, I created a button and a span. I turned the button into a circle, and added an image url on it. I transformed the span into a rectangle with rounded edges:

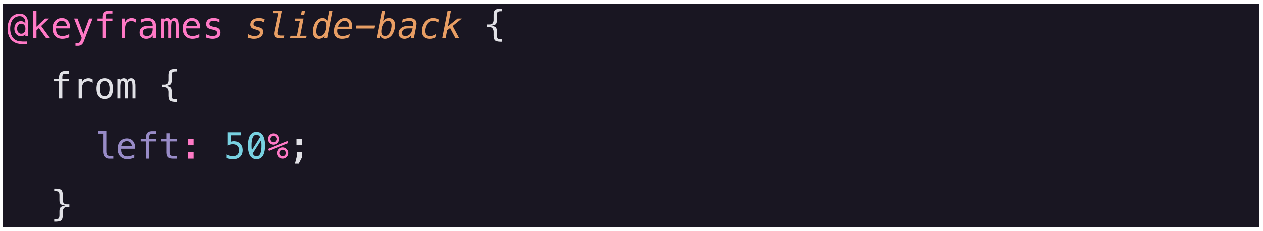




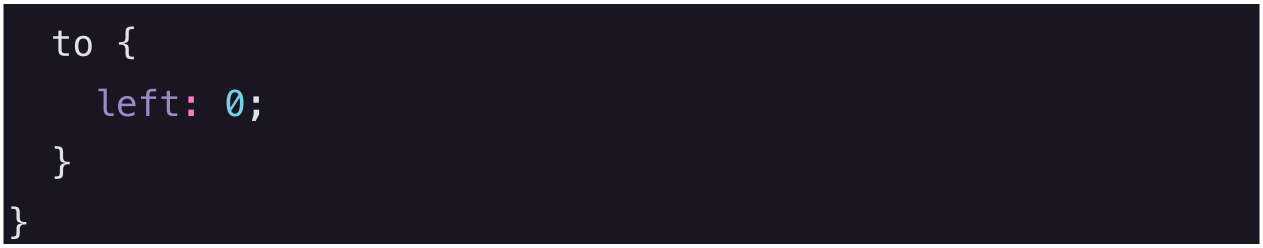
Furthermore, I learned that we can use the “cursor: pointer” option to make the “hand” appear when we position the mouse over the button

To build this button, I also learned that we have a website https://caniuse.com/ where I can check the compatibility of some CSS customizations with browsers.

Furthermore, I also learned how to make animations. For example, to slide the button from one side of the span to the other smoothly, I created the following animations:

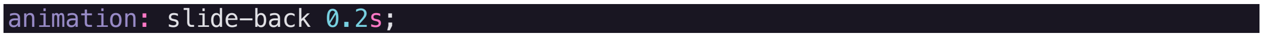


```
@keyframes slide-back {
  from {
    left: 50%;
  }
```



```
  to {
    left: 0;
  }
}
```

I called this animation inside the #switch button tag with a time of 0.2s:



```
animation: slide-back 0.2s;
```

## 3. List-Links

For the links, I created a UL list where I used display flex in column. And I created a customization using hover to change some CSS attributes when the mouse is over.

## 4. Social-Links

For social links, I learned a website where we can find icons and use in our code: https://ionic.io/ionicons

To use the icons, we need to add the following scripts at the end of the body tag:

```
<script type="module" src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.esm.js"></script>
<script nomodule src="https://unpkg.com/ionicons@7.1.0/dist/ionicons/ionicons.js"></script>
```

After adding the script at the end of the body, we can search for the icon we want to use and include it in the html file.

## 5. Footer

For the footer, I just created a message and used an ion-icon.