Final Project's Documentation

A Raspberry Pi Project Websites

Max Humberto De Janón Fasano

Purdue Polytechnic

CIT 21200

Jacqui Kane

November 24th, 2024

External Styling

My external style css file has the main elements that all the pages have in common. I started with setting my font-face to the fonts files I wanted to use. I decided to use the files because I want to have control of the feel of my website.

I first set * to border, margin and padding of 0px and the font family set to Roboto, one of the fonts I want to use. I do this so I take care of all the spaces around the website so it has a better personal feel.

I have a wrapper that I named .container that I set to:

```
.container {
    width: 100vw;
    height: auto;
    display: flex;
    flex-wrap: nowrap;
    flex-direction: column;
}
```

This is the main wrapper and it keeps my header, nav, .conent and footer looking great no matter what. Flex is a main point on most of my project and it helped me a lot.

My header has a very simple styling itself, but h1 and h2 have specific styling and I wanted them to be specific to the header:

```
header h1 {
    margin-top: 25px;
    margin-left: 80px;
    line-height: 26px;
    font-family: Quicksand-Bold;
    color: black;
}
```

```
header h2 {
    margin-left: 80px;
    font-family: Quicksand;
    color: rgb(29, 29, 29);
}
```

I set them this way so this way only the h1 and h2 that are inside the header would have this styling.

My nav is also pretty standard, other than having a container for the list items and the list in the nav bar has been set to:

```
nav ul {
    list-style-type: none;
}

nav ul li {
    display: inline;
}
```

This makes the lit have no list decoration and be displayed in line. I used this in the nav for helping with website readers. Both containers inside the nav have been set to inline-blocks, mainly so they could be moved and only occupy a limited space instead of the entire row, the reason why is they are off center and if they were not set to this, they would overflow in my nav bar and create an empty space on the right of the page (I figured this out during debugging).

There are hover effects for the .navButton and for the anchor in the that container to switch colors on mouse overs. This is a style that gives visual aid to users to know they are hoover over something interactive, in this case a link.

The .content container is a flex container that spans the entire horizontal view port. This is set to row over 800px with. It contains only the article on the landing page but for the parts

and setup page it contains articles and asides. Under 800px wide it will switch to column and this will have the article set on top of the aside, both occupying the entire width at that point.

```css
.content {
    display: flex;
    flex-wrap: nowrap;
    flex-direction: row;
    width: 100vw;
    min-height: 90vh;
    max-height: auto;
}
```

```css
.content {
    flex-wrap: nowrap;
    flex-direction: column;
}
```

I set some margin values to different elements so they would look good inside the article:

```css
article ul  {
    margin-left: 6%;
    margin-right: 3%;
}

article ol  {
    margin-left: 6%;
    margin-right: 3%;
}

article h2 {
    margin-top: 3%;
    margin-left: 3%;
    margin-right: 3%;
}
```

```css
article p {
    margin-left: 3%;
    margin-right: 3%;
}
```

I decided to use % this time to try it out. I like how it gives some movement to the elements. These are also adjusted for under 800px wide in the media query so that in mobile devices the margins increase.

I have set anchors to text-decoration non and they have their colors set to #ce1e54 and on hover its set to change to #53ce63.

Landing Page

Each page has a set of internal styling. The three of them share an article but the article in the landing page is different because there is not aside in the landing page. In my landing page, the with is set to 100%. It will span the width of its container. This is the reason why I chose internal styling here. Even though internal styling is higher in order, I decided not to include external styling for the article just so I don't get confused later on.

All the margins are taken care of in the external styling but there is some inline styling for some of the elements. This cover all the types of styling that we talked about in class.

Inside the article there is a flex container for the specs of the Raspberry Pi that has a list and an image. This is set to two different styles depending on the width:

```
.specsContainer {
    display: flex;
    flex-wrap: nowrap;
    justify-content: center;
    background: rgb(73,173,75);
    background: linear-gradient(90
    width: 96%;
    height: 420px;
    padding: 2%
}
```

```
@media screen and (max-width: 800px) {
    .specsContainer {
        flex-direction: column;
        height: 830px
    }
```

It is is set to default row and set to column under 800px width and the height also changes. I wanted this to be set this way because I saw this looked good on mobile.

All of my flex elements are set to not enlarge and to shrink. I didn't' want images to get distorted but in mobile they needed a little help, and the shrinking doesn't look too awful.

The last section of the body has some pins! I wanted to include some regular pins and flip cards in this project. I used this opportunity to add some in line styling to the information inside the pins to look as centered as possible so I added some in line styling to the h3 tags in each individual pin so that the text inside would look good.

Parts Page

This page uses two external styling files. One is the external style file, and it uses a second style file for my flip cards.

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

The internal styling for the article here is different because it is going to have containers inside and I want them to stack perfectly, so I set it to:

```
article {
  display: flex;
  flex-direction: column;
  width: 80%;
  height: auto;
  background-color: #e6e6e6;
}
```

The width is also set to 80% because my aside is set to 20%. Since the content container is set to row, this means they will be side by side. Bit this content container has a an internal style twist in this page:

```
@media screen and (max-width: 800px) {
  .content {
    flex-wrap: nowrap;
    flex-direction: column;
  }
```

It is going to switch to column under 800px width setting the article on top of the aside:

```
article {
  order: 1;
  width: 100%;
}

aside {
  order: 2;
  width: 100%;
  font-size: 1em;
}
```

I used order 1 and order 2 to make sure the article was first and the aside was second.

The link buttons in the aside section have similar styling to the nav bar, on hover they change color, the anchors have no text decoration and the font color changes on hover as well.

I added borders to my flip cards, I used the color #53ce63 for the back of the flip card. The container will center them on the page and the flip cards have no expand or shrink options.

Setup Page

This page shares very similar styling from the parts page. They are pretty much identical. One thing that is evident here that might not be so evident on the first few pages is the color shift. Every section has an alternating color. I switch from #e6e6e6 that is the background of the article, to white that is set on every other steps container as inline styling.

```
<div class="steps">
  <h2>Raspberry Pi OS:</h2><br>
```

```
<div class="steps" style="background-color: #ffffff;">
  <h2>Initial Router Setup:</h2><br>
```

Every images section has its own flex container that is very similar. The reason why I made one for each section is if later one I decide to update the page and I change something for one section, it doesn't mess with the others. Here is the style for one section as an example:

```css
.imagerInstructions {
  display: flex;
  flex-wrap: wrap;
  width: 96%;
  height: auto;
  padding: 2%;
  justify-content: center;
}

.imagerInstructions .instructions {
  flex: 0 1 297px;
  height: 343px;
  margin: 20px;
  background-color: #c5caca;
  border: 1px solid #333333;
  border-radius: 20px;
}

.imagerInstructions .instructions img {
  width: 100%;
  height: 100%;
  border-radius: 20px;
}
```

They all use the instructions container BUT since there is a nesting rule, I can modify one section without affecting the others.

One last thing, I saw that in my phone I was getting overflow from a long string of text and the last inline styling I used on a <li> was <li style="word-wrap: break-word; overflow-

wrap: break-word;">, this class taught me how to find solutions to problems and think outside the box. Now my issue cleared on my iPhone.

Some last explanations

My inspiration for this website was the Raspberry Pi website. I could not find a section for their colors, like we were taught to look for for Purdue or IU colors. I used Gimp to find the hex colors from the Raspberry Pi logo and other elements in the page. I wanted to keep uniform colors in the website. I also looked online for the fonts they use and downloaded the fonts and used those in my website and used @font-face like we learned in class.

I used folders for my stylingsheets, my images and fonts to keep them organized. Even for my setup page images, I have a sub folder inside the images folder to store those images.