|  |  |
| --- | --- |
| Index.html CSS | Explanation |
| @media screen and (max-width: 1050px){      p {          font-size: 25px;      }  } | Once the screen reaches a max width of 1050 pixels or less, the font size of my paragraph changes to 25 pixels, so they fit on the screen. |
| @media screen and (max-width: 800px){      p {          font-size: 30px;      }      } | Once the screen reaches a max width of 800 pixels or less, the font size of my paragraph changes to 30 pixels, so it covers more of my screen. |

|  |  |
| --- | --- |
| About.html CSS | Explanation |
| article {          height: 1200px;      } | I’m changing the height of the article to 1200 pixels because I need this webpage to be longer than the others, as it has 10 different descriptions in a list. |
| aside {          height: 1200px;          display: flex;          flex-wrap: wrap;          flex-direction: row;      } | I’m changing the height of the aside to be 1200 pixels because, once again, I need this webpage to be longer than the others, as the aside has 10 different images that need to fit.  I’m adding a flex display for the images so that I can neatly stack them.  I’m adding flex wrap so that they wrap around when the aside can’t carry the images in one row.  I’m adding a flex direction of row so that my images will lay in a row, until they wrap around. |
| .box {          height: 95%;      } | I’m changing the height of the box in the article to have a height of 95% because there is more height in the article, which means I need to increase the height of the box that is in the article, to make it look even. |
| @media screen and (max-width: 1300px){          aside {          height: 1375px;          }          article {              height: 1375px;          }  } | Once the screen reaches a max width of 1300 pixels or less, the size of the aside changes to 1375 pixels because I need more room for all of my pictures to fit without it ending up outside of my aside.  For the article, once the screen reaches a max width of 1300 pixels or less, the size of it will change to 1375 pixels as well, that way it matches the size of the aside and fits the list that I have. |
| @media screen and (max-width: 800px){          aside {          height: 1400px;          }          article {              height: 1200px;              overflow: hidden;          }      } | Once the screen reaches a max width of 800 pixels or less, the size of the aside stays at 1400 pixels because it still needs to be able to contain all of the photos of the movies.  However, once the screen reaches a max width of 800 pixels or less, the size of the article changes to 1200 pixels, as there is less room needed because the aside will then be below the article, meaning all of my text stretches out. |
| <p style=”font-size: 18px;”> Here is my list below, ranked from best to worst, in my opinion:</p> | For this paragraph, I changed the font size to 18 pixels, because the original font size of 30 pixels in the css file is too large. |
| <li style="font-weight: bold;">The Movie Name (year)</li> | For each numbered list where it says a movie name, I made the font bold as a way to make it stand out over the description that is below it. |
| <li><p style=”font-size: 13px;”>Movie Description</p></li> | For each unordered list where it has the description of the movie, I made it so that the paragraph size changes to 13 pixels so that way all of the descriptions and the list fits on the screen without making it too small to read. |

|  |  |
| --- | --- |
| Contact.html CSS | Explanation |
| @media screen and (max-width: 800px){          article {          height: 450px;          }          .box {              height: 400px;  }  p {  font-size: 28px;  }      } | Once the screen reaches a max width of 800 pixels or less, the size of the article changes to 450 pixels because I only have a paragraph in there so it doesn’t need much room to fit.  Once the screen reaches a max width of 800 pixels or less, the size of the box container also changes to 400 pixels because it is inside the article, so I made it even by fixing both the size of the article and the box.  I also changed the size of the p tag to 28 pixels whenever the screen reaches a size of 800 pixels or less so that it fits into the article tag nicely. |
| <p style=”font-size: 25px;”> | For the paragraph in my contact file, I changed the size of it to 25 pixels, instead of the original 30 pixels in the css file, so that it would all fit in the article and the box container without overflowing. |

|  |  |
| --- | --- |
| Final.css CSS | Explanation |
| .wrapper {      width: 100%;      height: 100%;      position: absolute;      display: flex;      flex-wrap: wrap;  } | The wrapper, which is the container around everything, I made have a width and height of 100% because I want it to affect the entire webpage.  The position is absolute because that is what I want each element to be in my webpage unless otherwise noted.  The display is flex so the pages can easily move when inspecting or going to a smaller screen.  I made the flex display have a flex wrap so that way each flex element in the wrapper container knows to wrap around if there isn’t space for it when the size of the page moves. |
| header {      background-color: #a48e75;      width: 100%;      height: 60px;      text-align: center;  } | I made the background color based off the Adobe color wheel.  I made the width of the header 100% so that it would spread across the whole page.  I made the height only 60 pixels so that it would be large enough to fit the title of each page.  I made the text in the header align in the center so that it would look nice. |
| nav {      position: relative;      background-color: #BFA688;      width: 100%;  } | I made the nav have a relative position so that it would work off of the page and go right after the element before it.  I made the background color based off the Adobe color wheel.  I made the width 100% so that it would spread across the whole page. |
| nav li {      float: left;      list-style-type: none;      margin: 0;      padding: 0;  } | For the nav list, this is the home, about, and contact.  I made the list float left so that way the list elements would be next to each other instead of below each other like a normal list.  I made the list style type to be none so that way the bullet points would go away.  I made the margin and padding 0 so that the list would start at the very beginning of the nav. |
| nav li a{      display: block;      color: white;      padding: 14px 16px;      text-decoration: none;  } | I made the a reference of the nav list have a display of block so that way the default behavior changes to where it takes up the full width available.  I made the color white for the actual text as it matches with the rest of my webpage.  I made the padding of the top/bottom14 pixels and then the padding of left/right 16 pixels so that way the words wouldn’t be touching each other and would look better in the nav bar.  I made the text decoration be equal to none so that you wouldn’t be able to tell it’s a link. |
| nav li a:hover {      background-color: brown;  } | When you hover over the words in the list in the nav bar, the background color around that word changes to brown. |
| article {      background-color: #D9BFA0;      height: 500px;      width: 50%;  } | I made the background color of the article based off the Adobe color wheel.  I made the height be 500 pixels so that everything in the article fit without any overflow.  I made the width 50% of the screen so I could put the aside right next to it. |
| .box {      width: 90%;      height: 455px;      background-color: antiquewhite;      margin-left: 5%;      margin-top: 20px;  } | My box element is inside my article so I only made the width of it 90% so that it would have some dimensions.  Therefore, I also only made my height 455 pixels compared to the actual article height of 500 pixels, so it would have some dimensions. It’s also going to be the box that contains all my text.  I made the background color antique white because it fit with the rest of the colors I got off of the Adobe color wheel.  I made the margin left be at 5% so that it was in the center of the article in terms of width.  I made the margin top be 20 pixels so that it was in the center of the article in terms of height. |
| ol {      color: brown;  } | I made the font color of my ordered list, which is my movie titles and descriptions, brown to match with my Adobe color wheel theme. |
| ol li {      font-size: 15px;  } | I made the font size of the organized list to be 15 pixels, which just affects the movie titles. |
| ul li p{      text-align: left;  } | I made the paragraph in the unorganized list aligned to the left so that it would be right against the bullet point. |
| aside {      background-color: #866c50;      height: 500px;      width: 50%;  } | I made the aside have a background color according to the Adobe color wheel.  I made the height of the aside 500 pixels so that it would match the height of the article.  I made the width of the aside 50% so that it would next to the article. |
| h3 {      position: absolute;      width: 150px;      background-color:#a48e75;      text-align: center;  } | The h3 is the movie names in the aside in the about.html file.  I made the position of it to be absolute so that it would follow the styling before it, which was the images.  I made the width match the image width so that it would look nice.  I made the background color according to the Adobe color wheel and so that it would stand out in the aside.  I made the text align in the center so that it would all look organized underneath the image. |
| .imageOutline {      margin-left: 5%;      margin-top: 20px;      height: 90%;      width: 90%;      border: 2px solid red;  } | The image outline is for the image I have in my 1st and 3rd webpage.  I made the margin left at 5% so that it would be in the middle of the aside according to the width.  I made the margin top at 20 pixels so that it would be in the middle according to the height.  I also made the height and width only 90% of the aside as to add dimensions.  I added a border of 2 pixels solid red which aligns with the colors of the rest of my webpage. |
| img {      width: 100%;      height: 100%;  } | I made the images fit 100% of their div container in terms of height and width. |
| footer {      background-color: #edd7bc;      width: 100%;      color: brown;  } | The background color aligns with the Adobe color wheel.  I made the width of the footer 100% so it would span across the whole page.  I also made the color of the text in the footer brown as it fits with the rest of the color theme. |
| p {      color: brown;      padding-left: 10px;      font-size: 30px;      padding-right: 10px;      text-align: center;  padding-top: 8px;  } | The color of the paragraph tag is brown as it fits with the color theme of the pages.  I made the padding left and right to be at 10 pixels so that the text wasn’t right against the box container.  I also made the font size to be 30 pixels so that the text would fit nicely in the box container without overflowing.  I made the text align at center as well to have it look more organized.  I added padding at the top at 8 pixels so that the p tag wouldn’t be stuck to the top of the container. |
| h1 {      color: white;  } | I made the color of the text in the h1 tag to be white so it would fit with the color theme of the rest of the page. |
| h4 {      padding-left: 5px;  } | I made the h4 tag, which is in the footer, to have a padding of 5 pixels on its left so that it wouldn’t be right against the beginning of the footer. |
| .card {      width: 150px;      height: 190px;      border: 1px solid black;      margin-left: 8%;      margin-top: 8%;  } | I made the width of the cards be 150 pixels so I could fit at least 3 images in a row in a full-sized webpage view without the images looking proportionally off.  I also made the height be 190 pixels as that is what I thought the best fit was for it proportionally.  I made the cards have a border of 1 pixel solid black as a way to make them look more organized.  For the cards, I made them have a margin top and margin left of 8% so they wouldn’t be right against the top or side of the aside container and it looked even from the top/bottom and side/side. |
| .warning {      position: fixed;      width: 100%;      height: 20px;      top: 114px;      background-color: crimson;  } | I made a sliding warning text that had a fixed position so you could see it in the contact.html file no matter where you scrolled.  I made the width be 100% so it would span across the entire width of the page.  I made the height be 20 pixels so it would be large enough to fit the text inside.  Since it has a fixed position, I moved it down 114 pixels on the page so it would reside between my nav bar and article and aside bar when you first open the page.  I made the background color of it crimson as it fit with the rest of the color theme of the page and it also fit with the warning text, as if it was dangerous. |
| @media screen and (max-width: 800px){      article {          width: 100%;      }      aside {          width: 100%;      }  } | When the screen reaches a max width of 800 pixels or less, the article’s width is going to take 100% of the screen, instead of it’s original 50% so that it will be on top of the aside.  I also did the same with the aside and changed the width from 50% to 100% when it reaches 800 pixels or less, so that it will stack below the article at that size. |