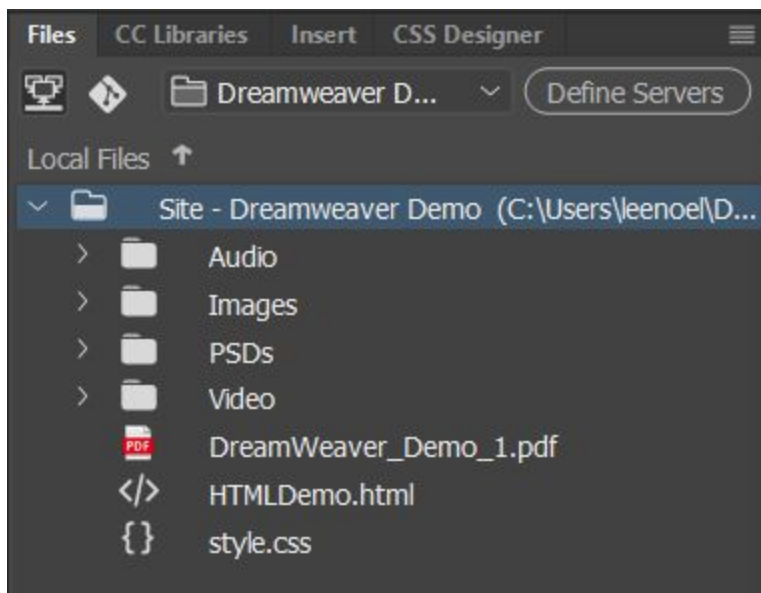


This demo is a continuation of the previous part completed in class. The previous demo ended with us importing and placing the edited Right Eye of the girl so that it will be placed into the correct part of the web page.

If you have not yet done so, import and place the Left Eye, Smile, and Closed Mouth png images into the web page -

1. Open Dreamweaver and open the previous demo so we can continue in this one. You may have to repeat the 'Manage Sites' process we went through in the previous demo if you are not able to view the files associated with your project in File structure on the right side of the DW window.



2. Let's begin to make changes to transform this composition into an interactive webpage! One of the most common type of interaction is the clickable link, let's enable this functionality in the Left Eye. In you HTML you should have a link defining the image as follows

```

```

3. To do this we will use a hyperlink which will live in an [<a>](#) tag. This tag is meant to enclose the element which will be used as a hyperlink as such

```
<a href="">  
    
</a>
```

4. Find a link to a website and place it between the quotations marks in href=""
This link could also be to a local file on the hard drive of your computer! This will let you link multiple html documents together into a functional website
5. To make the link open in a new tab when clicked you can add **target="_blank"** following the href:

```
<a href="https://www.google.com" target="_blank">
```

6. With the <a> tag and href inserted the left eye of the girl will now function as a link to another webpage and the first interaction has been built!
7. The next interactive element I would like to include is for the right eye to look to the right side when hovered over with the cursor. We will look at two different ways of accomplishing this goal in the coming steps
8. The first method we will utilize functionality called **OnMouseOver** and **OnMouseOut**. These are known as [Events](#) and can be included withing the tag line as such
OnMouseOver="this.src=' '
OnMouseOut="this.src=' '
9. Let's add this to the tag fo the Right eye. Your code should look like this

10. Let's examine this code briefly after seeing it in action. The image is given a source called **src** where it is defined initially. We then issued the command to redefine the source **this.src** using the **OnMouseOver** command and then used **OnMouseOut** to again redefine it back to its initial state when the mouse is not over the specified element!
11. However, notice the effect is opposite what I initially wanted. The Right Eye is looking right *until* it is hovered over by the cursor. To do this we could simply alter the source logic in the tag for the right eye. But let's try a second way of achieving this effect.
12. First though, let's disable the code we just wrote, but do so without deleting it. This will be useful if we need to reference the code structure in the future, or if we choose to re-enable the code. This is called a [Comment](#) and it is formatted like this

<!-- This is a comment -->
13. Anything contained within the brackets <!-- → will be counted as a comment and will not be executed in the code! So take this opportunity to place your tag for the Right

Eye in a comment. Then Let's add a fresh tag to re-establish the Right eye image without the OnMouseOver and OnMouseOut events. Your code should look something like this

```
<!-- 
-->

```

14. The second method will take place in the css of this webpage, so open the **style.css** tab in DW. We will alter the css to make the image of the right eye to transition between 0% and 100% opacity when we hover over it, and give the illusion that it is appearing and disappearing
15. For this we will be utilizing [Transition](#) and [Hover](#). Transition will define which style attribute of the class should be changed when hovered over, and Hover will define the new style that will be implemented. Let's see how this looks when implemented

```
.Reye{
    left: 899px;
    top: 900px;
    opacity: 0;
    transition: opacity 0s;
}

.Reye:hover {
    opacity: 1;
}
```

You can see the **.Reye** class we defined previously and the **.Reye:hover** class we added after it. Also note that the **Opacity** attribute in the first class is set to 0 (meaning transparent), but in the second **:hover** class it is set to 1 (meaning not-transparent). The **transition** property is what specifies which property should be changed when hovered over and how long it should take in seconds (0s means instantaneous).

16. This method can be used to transition any property (left, top, width, height, etc.) that is specified in a class. Take a moment to implement this method so the closed mouth image moves downward to reveal the smiling mouth underneath. Think of which property needs to be altered and how in order to achieve this effect!

17. Your new Css should looks something like this

```
.Mouth{
    left: 577px;
    top: 1309px;
    transition: top 5s;
}

.Mouth:hover {
    top:1509px;
}
```

I chose for the transition to take 5 seconds, but this is a matter of preference.

18. Reference the [Transition](#) entry at W3Schools to see more implementations of the property and other ways in which it can be used!

19. With this we have a functioning web page! You could potentially build an entire project using just the information given in these two demos! We will spend the rest of class going over some extra pieces of useful code that may interest you. Before ending this demo let's look at a popular one and add a music player to this web page! Download the file called "BackgroundAudio.mp3" so we can begin

20. Return to the html document we are working on. We are going to add an <audio> tag to the page (I typically put it at the top underneath "<body>"). Start a new line and type <audio> </audio>. Then put a few lines between the beginning and end of the tag so you have room to insert your audio file.

21. The formatting for the audio files is very similar to the img files and will look like this
<source src=" " type="audio/mpeg" >

The difference is this does not require an **alt=""** element, but instead needs a specification of the file type. BackgroundAudio.mp3 is an mpeg file, but most other types such as .wav can be used.

22. Your code should now look like this

```
<audio>
    <source src="Audio/BackgroundAudio.mp3" type="audio/mpeg" >
</audio>
```

However when you refresh your page you will not be able to play the audio! This is because we have simply told the software that the file is present, but have given it no way of playing, which we will do now.

23. This is done simply by adding "**controls**" to your <audio> tag as such
<audio controls>

</audio>

24. If you save and refresh you will now see an audio player at the top of your page. Hit the play button to hear your song!

25. This final step will introduce a useful element called a [<div>](https://www.w3schools.com/tags/tag_div.ASP) tag. This can be used to define custom sections of HTML that can be altered together as a group. This is a very powerful tool and I recommend looking at the entry on W3Schools to learn more https://www.w3schools.com/tags/tag_div.ASP

26. We will use this to reposition the audio player since it cannot be formatted using css on its own. We will use <div> and define a new class called "player" to achieve this.

27. Start new empty lines before and after your <audio></audio> element so you can contain it within a <div> </div> element. Doing so will define a new "space" for the audio player to occupy that we can edit instead of the player itself. Also give a class called class="player" to the div. It should look like this

```
<div class="player">
  <audio controls>
    <source
      src="Audio/BackgroundAudio.mp3"
      type="audio/mpeg" >
    </audio>
  </div>
```

28. Now go to the css file and add a new class like is shown below

```
.player{  
    position: absolute;  
    left: 600px;  
    top: 10px;  
    z-index: 10;  
}
```

For this class I am introducing a property called [z-index](#) which is used to change the order, from front to back in which images appear on a webpage. Setting it to 10 is simply to move it ahead of the other image elements so it will certainly be visible on top. Some designers will set ridiculous values such as 10000000000000000000 for this purpose.

And with this I will release you to create projects! I will introduce some other interesting codes, but they are typically very specific and not necessary to learn for all students.