CS 135I Winter 2024

# Week 3 Classwork

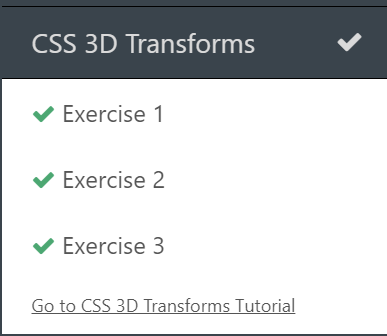
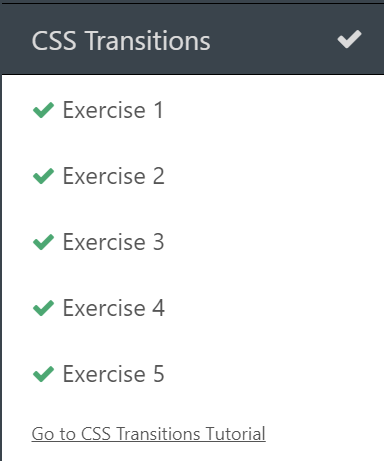
## Overview

We’re moving on to look at a new category of CSS techniques this week: transformation, transition, and animation. I’d recommend using the File menu to either download a Word version of this sheet or to create a copy on your own Google drive. Fill out the notes as we go along in class, and complete the activity so that you will be able to upload your proof of completion to Moodle at the end of class.

## Creating Movement with CSS

Concept: the CSS transform, transition, and animation properties can work together to add delightful movement and interactivity to your pages, with almost no download cost.

1. Build your knowledge of **transforms** by completing the [W3Schools transform exercise](https://www.w3schools.com/css/exercise.asp?filename=exercise_css3_2dtransforms1). Paste a screenshot of the exercise navigation bar expanded showing your 4 green check marks for transforms:  
   A screenshot of a computer

   Description automatically generated
2. Build your knowledge of **transitions** by completing the [W3Schools transition exercise](https://www.w3schools.com/css/exercise.asp?filename=exercise_css3_transitions1). Paste a screenshot of the exercise navigation bar expanded showing your 5 green check marks for transitions:  
   
3. Build your knowledge of **animations** by completing the [W3Schools animations exercise](https://www.w3schools.com/css/exercise.asp?filename=exercise_css3_animations1). Paste a screenshot of the exercise navigation bar expanded showing your 6 green check marks for animations:  
   A screenshot of a cell phone

   Description automatically generated

## Applying CSS Transforms, Transitions, and Animation Concepts to a Webpage

1. Download the *classwork03.zip* file from Moodle
2. Extract the contents, and examine both the existing *animationAndTransitions.html* file and the *primary.css* file.
3. Update the *primary.css* file using transform, transition, and animation properties so that:
   1. The image in the aside calls the spotLight animation when users hover over it. The animation should last 3 seconds.
   2. Create a rounded rectangular shape that appears BEFORE the header element. Its properties should be:
      1. It contains the empty string for content.
      2. It is absolutely positioned 0.75em above the top of the header element and 0.75em to the left of the header element.
      3. It is 2em wide and has a height of 120% of the header.
      4. It should have a background color of black and rounded corners. You may choose the radius you like, but something dramatic like 50% might be nice.
   3. When users hover over the header, change the BEFORE element so that it is transformed to 200 times its original width and 5 times its original height.
   4. This works, but isn’t very polished-looking. Add transition properties so that the transform change happens over 3 seconds, and uses the ease-out transition timing function.
      1. Change the transition timing functions to some of its other [values](https://www.w3schools.com/cssref/css3_pr_transition-timing-function.php). Can you see the difference? If not, increase the transition duration a bit.
   5. Create an animation keyframe named *showRedText*.
      1. The color of text should change from transparent at the start to crimson at the end.
      2. The font size of the text should change from 0 at the start to 200% at the end.
   6. Attach the *showRedText* animation to the first h3 element in the *ad* aside.
      1. Experiment with various animation durations to find a timing that you like.
      2. Experiment with various animation timing functions to find a value that you like.
   7. Repeat steps (e) and (f), but create a *showBlueText* animation that changes from transparent to navy and attach it to the last h3 element in the *ad* aside.
   8. Up the drama by making the last part of the ad headline appear to zoom in and out.
      1. Create a *zoomText* animation keyframe that:
         1. Sets the font-size to 200% at the 0% and 100% mark.
         2. Sets the font-size to 250% at the 50% mark.
      2. Apply the *zoomText* animation to the last h3 elements in the *ad* aside. Remember that the animation-name property can accept a comma-separated list of values.
         1. Don’t forget to update the animation-duration property to pick a length of time for the *zoomText* animation.
   9. Next, add even more drama by creating a cover reveal effect for the ad headline.
      1. Create a *blockIn* animation keyframe. It should:
         1. Change the width from 0 (at 0%) to 100% (at 100%)
      2. Create a *blockOut* animation keyframe. It should:
         1. Change the width from 100% (at 0%) to 0 (at 100%)
      3. Apply the *blockIn* **and** the *blockOut* animations to the rectangles after the first and last h3 elements in the *ad* aside.
         1. Each animation should take the same length of time to execute. Try 4 seconds to start, and then adjust as desired.
   10. It is very common to see a little motion used as attention-getters on buttons and links. Let’s create a “jiggle” effect for the links in our *ad* aside:
       1. Create a *buttonTilt* animation keyframe:
          1. At 0% and 100% the settings are:
             1. Box-shadow: 1px 2px 4px #888888
             2. Padding is 1em
             3. The element should be transformed to a rotation of 0 degrees
          2. At 25% and 75% the settings are:
             1. Box-shadow: 2px 4px 4px #666666
             2. Padding is 1.15em
             3. The element should be transformed to a rotation of 3 degrees
          3. At 50% the settings are:
             1. Box-shadow: 2px 4px 4px #555555
             2. Padding is 1.15em
             3. The element should be transformed to a rotation of -3 degrees
       2. Attach the *buttonTilt* animation to the hover pseudo-class for the list-item elements inside the *ad* aside’s nav element.
          1. Don’t forget to give it a (short!) duration.
   11. Creating events that happen on hover in CSS is pretty straightforward to do, as we’ve just seen. But what about changing things when users click on objects? This is a little trickier with *only* HTML and CSS. (It’s simple when you add JavaScript into the mix, but that’s a different class!). A common technique is known as the checkbox hack.  
         
       In essence, a checkbox and label are added just before the element to be animated, transformed, or transitioned. The checkbox is hidden, and its label styled as desired. When users click on the label, the checkbox (even hidden) switches to its checked state. In CSS, you can use the :checked pseudoclass and a next sibling operator to then apply changes to the element immediately after the checkbox.  
         
       Let’s try it!
       1. Your HTML page already contains a checkbox and label setup. Take a quick peek at the HTML code to refresh your memory on how this works.
          1. Note that the value of the *for* attribute in the *label* element **must** match the value of the *id* attribute in the *input* element.
       2. In the CSS file, use an id selector to target the *chkToggle* checkbox and change its display to *none*.
       3. Target the aside that is immediately after the checkbox using this selector:  
          #chkToggle:checked + aside
       4. Use the *visibility* property and a value of *hidden* to make the ad disappear, but retain a space for it on the page. (This will keep the page from “jumping”)
       5. Test the page in the browser window, noticing that we do not need to code an :unchecked state for the element. When the label, and thus the checkbox, is deselected the visibility of the aside will reset to its original value.
       6. To get the same effect with a transition, we would not be able to use the *visibility* property, since its values are specific keywords rather than numbers. What could we change in order to make the ad gradually disappear/reappear?
4. When finished, test your page one last time in the browser window. You should have
   1. A hover effect for the header element and the image in the ad section
   2. An automatic animation for the h3 elements in the ad section that involves both text effects and a color block reveal
   3. Animated hover effects for the *buy* and *sell* buttons
   4. A click effect for the “x” label that makes the ad disappear
5. Zip the folder and upload to the appropriate link in Moodle for your classwork points. Nice work!