

Unit 6G4Z2101: Introduction to Web Design and Development

Worksheet 4: CSS3 Transforms, transitions and animations

Workshop topic

This worksheet gives you the code for some of the CSS3 properties, for you to try out, but there is much more that you can do with CSS3. References are provided to relevant resources.

NOTE:

For some CSS transforms, transitions and animations you still need to use browser prefixes (as discussed in the lecture). The first piece of code has the browser prefixes included in the worksheet, but after that you should add them yourself, even if they are not shown in the worksheet.

Create a box with a drop-shadow effect

1. In Notepad++ create a new html document and save it as css3demo.htm.
2. Create a new css document, and save it as css3.css, then link it to your html document.
3. Add a style for **body** and set the **margins** and **padding** to **zero**. Set the font-family as **Arial, Helvetica, sans-serif**.
4. In your HTML document add the following to create a box:

```
<div id="box">
<p>This is a box using css rounded corners</p>
</div>
```

5. In your CSS document add a style and set the following:
 - a. width and height of 200px
 - b. padding of 10px
 - c. a top and left margin to give it some space
 - d. a 2px solid border with a colour (see http://www.w3schools.com/cssref/css_colornames.asp)
6. Save both files and open in a browser to check that you can see a box.
7. In your css document add the following to your **#box** style to create rounded corners. Add a comment to say what it does.

-moz-border-radius: 25px;

```
-ms-border-radius: 25px;  
-webkit-border-radius: 25px;  
-o-border-radius: 25px;  
border-radius: 25px;
```

8. Experiment with changing the pixel value to see what effect it has on the corners.
9. Add the following to the **#box** style, to give the box a drop-shadow.

box-shadow: 6px 6px 8px #colourcode; *replace colourcode with the colour you used for the border*

10. Save your CSS file and refresh the browser window.
11. Experiment with changing the pixel values to see how they each affect the shadow. The first pixel value will affect the horizontal offset, the second affects the vertical offset and the third affects the blur distance. If there is no blur distance the shadow edge will be solid.
12. Add a fourth value (try 8px) before the colour code, to create shadow spread. This creates a shadow around all four sides of the box.
13. Change the two 6px values to negative values -6px. Check in the browser and note that the shadow is now at the top left. So you can change where the drop shadow is applied by using positive or negative values.

Create text shadow

Text shadow works in the same way as box shadow; having pixel values for horizontal offset, vertical offset, blur distance and a colour.

1. Add a heading to your HTML document, set as <h1>.
2. Add a style for your heading as shown, choosing one colour for the text, and another for the drop-shadow (try using a lighter shade of the main colour):

```
h1 {  
    font-size: 3em;  
    color: #colour1;  
    text-shadow: 3px 3px 0px #colour2;  
}
```

You can add more than one shadow to text by adding a comma followed by a second set of values to the same statement. This works better with bigger text.

3. Change the **h1** font size to **4em**, and after **#colour2** add a **comma followed by 6px 6px 0px and a third colour**. Try another shade of the main colour. It can work well having a strong colour for

the main text, a much lighter shade for the first shadow and shade in between for the second shadow (see below).

heading

Apply CSS Transforms

Transforms allow you to rotate, scale, skew and translate (move) elements. The use 'transform' as the property, and the values vary depending on the effect you want to achieve.

1. Add the following statement to your style for #box and check in a browser:

```
transform: rotate(45deg);
```

Transforms can be activated by mouse hover. You will change the rotate to be hover-activated, and add a second transform to scale the box up. More than one transform can be added to the same statement.

2. Add a new style with the selector **#box:hover** and move the transform statement to here. Now the box will rotate as you hover the mouse pointer over it.
3. After (45deg) and before the semi-colon (;) add

```
scale(1.2, 1.2)
```

4. Try your page in a browser to check that the transform is triggered by mouse hover. The scale values are for the x-direction and the y-direction respectively.
5. Replace 'rotate(45deg)' with `translateY(6em)` and how the box should move instead of rotating. You can also use `translateX` or combine them into a single `translate` statement.
6. Add a `rotateY` transform, and make the box rotate 180 degrees.

Refer to the following web page to see other transform settings.

http://www.w3schools.com/cssref/css3_pr_transform.asp

Transforms are often used for things like photo galleries



Apply CSS Transitions

CSS transitions allow you to animate property changes (e.g. resizing an element or altering background colour), over a given period of time.

1. In your HTML document create a second div called **#box2**
2. Give **#box2** the same styling as **#box**.

When using transitions, you include a timing function, defining the duration over which the transition will take place.

To make a box scale, for example, you set an initial size first. Your box already has values for width and height.

Now you need to define the transition. You will alter the box's width over a duration of 2 seconds.

3. Add the following code to the style for **#box2**:

```
transition: width 2s;
```

Now you can add a hover statement in which you declare the end value for the property you are transitioning; in this case a new width for **#box2**.

4. Add the following code to your stylesheet to set the new width for the box:

```
#box2:hover {width: 300px;}
```

5. Test in a browser and when you hover over the box it should smoothly increase in width.

You can also add a transform to a transition. So you could control the transform speed of the first box by using a transition.

6. Add the following code to the style for #box (not to the hover style):

```
transition: transform 2s;
```

7. When you test the page in a browser, the transform should now happen much more smoothly.

You can add several transition actions to a single statement, as with transforms. An important difference is that with transitions you put a comma between each new transition. See example below.

```
transition: width 2s, height 2s, transform 2s;
```

Note that 'transition: width 2s' is actually shorthand for declaring a transition-property and transition-duration. The longhand for the statement above is shown below.

```
transition-property: width, height, transform;  
transition-duration: 2s;
```

You can also add ease-in, ease-out and ease-in-out functions to a slow a transition at the beginning or end.

You can find a list of the animatable properties at
http://www.w3schools.com/cssref/css_animatable.asp

Keyframe Animation

Keyframes can give you much more control over animation effects. If you want to move a box across the screen, for example, you could use keyframes to say what position the box should be in at given percentages of time, through whatever duration has been set for the complete animation.

1. Create a new html file, and add a box.
2. Create a stylesheet and attach it to your html document.
3. Add a style to make the box the same dimensions as the ones you worked with earlier.
4. Give the box a solid border and a background colour.
5. You will add a CSS animation to make the box move across the screen. Because you are going to move the box, you need to give it a starting position.
6. Set the box's position as **relative** with a **left** position of **0** and a **top** position of **60px**.
7. Add an animation function to the box as follows:

```
animation: moveBox 6s;
```

This tells the browser we are creating an animation called 'moveBox' (you create your own animation name) which will run over a 6-second duration:

8. Now add the keyframe information for 'moveBox' as follows:

```
@keyframes moveBox
{
  0% {left:0px;}
  30% {left:200px;}
  50% {left:400px;}
  75% {left:0px;}
  100% {left:0px;}
}
```

@keyframes is a standard selector, and you decide on the percentages and values to use. This code makes the box move more slowly at first as it moves 200px over 30% of the 6s duration. Then it moves another 200px in only 20% of the time. Therefore the second movement is faster.

Try it on your own

Create a text effect

CSS can be used for creating effects that previously would have been created with Flash, such as animated banners.

Advertising banners will often use text effects, such as fading text in. Have a go at creating a banner-type text effect, where three lines of text fade in at different intervals.

1. Create a new html document and add three paragraphs of short text (e.g. CSS3 transitions/can animate/your page).
2. Give the text a large font size and centre it using **text-align**.
3. Create a transition which will make each line of text fade in from white to a colour. Using **animation-delay**, make the second paragraph appear a couple of seconds after the first, and the third one a couple of seconds after that.

HINTS

- You will need to have a way of targeting each paragraph individually
- You will need to set starting colour values
- The colour is the property being transitioned
- Since the text is fading in from white, you can't see it to hover on it. You can make the transition respond to hovering over the body of the document, as in the example below.

```
body:hover .p3 {color: black;}
```

Create more animation effects

1. Develop the code for the animated box so that it moves round the screen in a square formation (left, down, right, up), landing back in its starting position.
2. Develop the code further so that the box changes colour as it moves (use **background** not background-color).

Create an animated banner

1. Create a banner which contains an advertising slogan.
2. Animate both the background and the text. For example, the background could move in from the side, top or bottom. The text could fade in, move in, etc.
3. Try adding an animated 360 degree rotation to one of the elements.

In this workshop you have learned a range of techniques for adding a variety of effects to your html content. Whilst these techniques should not be overused, they do add an extra dimension to what you can achieve with just html and css. See the following example of an animated banner which uses images <http://tympanus.net/Tutorials/AnimatedWebBanners/>

There are lots of other examples online of what you can do with css animations. When searching look for “pure css3 animations” to find examples that do not rely on javascript.