Week 08 – Chap 8 and 12 HTML5 and CSS3 for the real-world notes

**Transform** - A CSS3 property that lets you translate, rotate, scale, and/or skew any element on the page with *transform functions*.

Transform Functions:

**Translation** - allow you to move elements left, right, up or down

Translate (x, y) // X from the left, Y from the top

Ex. Transform: translate(45px, -45px);

You can also use:

translateX(45px) to move element along the X axes

translateY(45px) to move element along the Y axes

\*\* Transforms don’t work on inline elements, so just add display: inline-block; \*\*

**Scale** - scales an element by the defined factors horizontally then vertically (if you put just one value it will use for both X and Y

Scale (x,y) //X will scale, Y will scale

Ex. Transform: scale(1, 2);

This will keep the X to its original proportion, and it will double y

To double the proportion

You can also use:

scaleX(x) to scale only horizontal dimensions

scaleY(y) to scale only vertical dimensions

\*\* Don’t declare a new transform because of the cascade, just declare the second function in the first \*\*

**Rotation** - rotates an element around the point of origin by a specified angle value

Rotate(deg) //will rotate in degrees

Ex. Transform: rotate(5deg)

This will rotate the element 5 degrees clockwise. Use a negative value for counter

**Skew** - specifies a skew along the X and Y axes

Ex. Transform: skew(15deg, 4deg);

You can omit the Y parameter and the skew will only occur on the X axis

There are also axis-specific versions

skewX() and skewY()

**Transitions** - Allow the values of CSS properties to change over time, essentially providing simple animations.

**Transition-property** - defines the CSS properties of the element that should be transitioned, with **all** for all properties being the default.

Any property changing from one value to another for which you can find a valid midpoint can be transitioned.

Ex. Color, width, border, visibility (this one is an exemption), etc.

Items such as border-style: dashed to solid cannot be transitioned.

If you want to transition more than one property, but not all, comma-separate them

Ex. Transition-property: transform, color;

**Transition-duration -** sets how long the transition will take: the duration of time it takes to go from the default state to the transitioned state. Specify this in either seconds (s) or milliseconds (ms).

0.2s (200ms) is considered the optimum time for a transition. Anything slower will make the website seem slow, and anything faster may be too subtle.

**Transition-timing** - Lets you control the pace of the transition in even more granular detail. It can start off slow but then finish faster, or vise-versa.

Key terms:

Ease, linear, ease-in, ease-out, ease-in-out

Apart from using key terms you can use the **cubic-bezier** function that accepts four numeric parameters.

Ex. Cubic-bezier(0, 0, 1, 1)

**Steps** is another way. You define the number of steps and the direction of either start or end, where either the first step happens at the animation start, or the last step happens at the animation end respectively.

Ex. Steps(5, start) would jump through the steps of 0%, 20%, 40%, 60%, and 80%

Steps(5, end) would jump through the steps of 20%, 40%, 60%, 80%, and 100%

**Transition-delay** - delays the transition before it begins. Values are seconds (s) or milliseconds (ms)

Ex. Transition-delay: 50ms;

**Transition shorthand** - Makes it possible to combine all transition properties into one

Ex. Transitions without shorthand:

Transition-property: transform;

Transition-duration: 0.2s;

Transition-timing-function: ease-out;

Transition-delay: 50ms;

Ex. Transition with shorthand:

Transition: transform 0.2s ease-out 50ms;

\*Properties in shorthand syntax can be in any order, but if a delay is included, you must also include a duration, and the duration must precede the delay.

**Multiple transitions** - you can pick different elements to transition in the same *transition-property* but give them different transition effects

Ex. Using a comma with separate the transition effect of each element in their respective order

transition-property: transform, color;

transition-duration: 0.2s, 0.1s;

transition-timing-function: ease-out, linear;

transition-delay: 50ms;

Transform will have a duration of 0.2s, a timing of ease-out, and a delay of 50ms

Color will have a duration of 0.1s, a timing of linear, and a delay of 50ms

Ex. This is the shorthand version of the same example above:

Transition: transform 0.2s ease-out 50ms, color 0.1s linear 50ms

**Chapter 12**

**Canvas** - an element drawn in javascript. Add a <canvas></canvas> element to the HTML page and the rest you do in Javascript. Adding text inside the html element will only show up if the element is not supported by the visitor's element.

Canvas take width and height attributes to determine how large the canvas' coordinate system is. If width and height isn't specified, it will default to 300Wx150H

Ex.

<canvas id="myCanvas" class="myCanvas" width="200" height="200">

Sorry! Your browser doesn’t support Canvas.

</canvas>

You can add a border in CSS to visually distinguish it on the page.