408- CSS 3 TRANSFORMS AND TRANSITIONS

transform - translate, rotate, scale, and/or skew

transform functions

TRANSLATION - move element left, right, up, down

transform: translate(x, y) translate(45px, -45px);

transform: translateX(45px); translateY(45px);

(vendor prefixing for IE9 and pre :058, Android 4.4.3

-webkit-transform
-ms-transform

Transforms do not work on in-line elements.
Add 'display: inline-block;'

Scaling - scale(x,y) ← x = Horizontal y = vertical
- if only a single value, used for both x & y

Do not declare a new transform because of the cascade, a second would override the first.

Scaling and translation does not impact document flow.

There will not be any accomodation or reflowing when scaling an in-line-block.

Rotation - rotate() rotates an element around a point of origin. default point of origin is center

degrees, positive is clockwise
(can be grads, radians, or turns)

Skew - skew() skews an element around a point of origin. Settings are similar to rotate

CHANGING ORIGIN OF TRANSFORM

transform-origin has same syntax as background-position. defaulting to the center of the object.

W08 - TRANSFORMS AND TRANSITIONS

CHOOSE ORDERING CAREFULLY... IT MATTERS

If you rotate before translate, the translate will be on the rotate axis.

IE8 and earlier - css3 transforms are unsupported before IE9 but they can be mimicked through use of position: relative, and top and left values

```
.translate {
    position: relative;
    top: 200px;
    left: 200px;
}
```

scale by altering width and height or changing font size.

can also use filters:

http://www.useragentman.com/IETransformsTranslator/

TRANSITIONS - animation using native css transitions require less client-side processing than Javascript.

css transitions are declared along with regular

simple transition using CSS

1. Declare original state of the element in default style declaration
2. Declare final state of the transitioned element; for example, a:hover
3. Include transition functions in the default style declaration.
   transition-property, transition-duration, transition-timing-function, and transition-delay

Transition is declared in the default or originating state

-webkit- vendor prefix is still needed for older mobile devices (iOS 6.1, BlackBerry 10, Android 4.3 and UC Browser)

##W08 - TRANSFORMS AND TRANSITIONS #A

Transition-property - defines the CSS properties of the element that should be transitioned. 'all' is default.

- ANY PROPERTY CHANGING FROM ONE VALUE TO ANOTHER THAT ALSO HAS A MIDPOINT CAN BE TRANSITIONED.

1px red border → 15px blue border

↑ Transition color and width

- The midpoint of 1px and 15px is 8px
- The midpoint of red to blue is done numerically

border-style solid → border-style dashed will not transition

Although visible and hidden do not have a midpoint, they can be transitioned

- Include a pre-state and a post-state
- transition-property: transform doesn't do anything by itself. other definitions must be included.

Transition-duration - sets amount of time the transition will take. can be specified is seconds(s) or milliseconds (ms)

200ms is considered optimum. goldilocks between too slow and too fast.
Reverse transition happens by default.

Transition-timing-function - control pace of the transition in a more granular detail.

most common:

- ease - linear - ease-in - ease-in-out - ease-out

can customize the timing using a cubic-bezier function

http://cubic-bezier.com

W08 - TRANSFORMATIONS AND TRANSITIONS

transition-delay - introduces a delay before transition begins.

negative delays cause the transition to start immediately but begin further into the animation

tansition Shorthand - properties can be included in any order except duration comes before delay. delay requires a duration to come first.

Multiple Transitions - can call multiple transitions in a single call.

```
transition-property: transform, color;
transition-duration: 0.2s, 0.1s;
transition-timing-function: ease-out, linear;
transition-delay: 50ms;
```

to have all transitions together, only include a single property; otherwise, use comma-separated lists in the same order as the transition-property

Shorthand:

```
transition: transform 0.2s ease-out 50ms, color 0.1s linear 50ms;
```

'all' keyword can be used to transition all properties at same rate

```
transition: all 0.2s ease-out 50ms;
```

transitionend event fires when the tranition is complete. It will fire multiple times when there are multiple transitions with different durations.

W08 - TRANSFORMS AND TRANSITIONS

ANIMATIONS - control each step of an animation via keyframes.

keyframe - snapshot definition of a starting and ending point of any smooth transition.

- CSS transitions define a first and last keyframe
- Animations allow any number of keyframes to guide the animation in more complex ways.

Generally, it is best to use CSS for simple-state changes, but it is better to use JavaScript for intricate, stateful UIs.

KEYFRAMES

- FIRST, CREATE NAMED ANIMATION
- THEN, ATTACH TO AN ELEMENT'S PROPERTY DECLARATION BLOCK

@keyframes rule for IE10+ and FF16+

@-webkit-keyframes for all Webkit implementations

```
@-webkit-keyframes myAnimation {
    /* put animation keyframes here */
}
@keyframes myAnimation {
    /* put animation key frames here */
}
```

Do not quote animation name

Each key frame looks like its own nested CSS declaration block. Percentage value instead of a selector. Keyword: from and to inside each keyframe include properties to animate. any order - percentage values carry the transition order.

W08 - TRANSFORMS AND TRANSITIONS

simple animations:

```
@keyframes moveRight {
  from {
    tranform: translateX(-50%);
  }
  to {
    transform: translateX(50%);
  }
}

@keyframes appearDisappear {
    0%, 100% {
      opacity: 0;
    }
    20%, 80% {
      opacity: 1;
    }
}
@keyframes bgMove {
  100% {
    background-position: 120% 0;
  }
}
```

with these animation defined, an element must have an animation name for there to be an animation

W08 - TRANSFORMS AND TRANSITIONS

ANIMATION PROPERTIES

animation-name · property used to attach an animati to an element.

animation-name: appearDisappear; no quotes around animation name.

animation-duration - property defines length of time an animation takes to complete.

animation-duration: 300ms; ← ms or s

animation-timing-function - determines how the animatio will progress over the duration.
Same option with addition of step-start, step-stop, and steps(number, direction)

animation-timing-function: linear;
animation-timing-function: steps(4, end)

Using two animations, one to move the image within the frame, and one to move the frame.

using the steps(number, direction) we divide the animation into 4 steps and move in the direction in quick succession

start in position 0,0, -90px, 0, -180px, 0, and -270px, 0.
if direction is start, we see 25%, 50%, 75%, 100
if direction is end, we see 0%, 25%, 50%, 75%

animation-iteration-count · defines how many times the animation will play.

animation-direction - normal iteration → 0% → 100%
reverse 100 → 0%
alternate 0% → 100 → 0%

animation-direction: alternate;

W08 - TRANSFORMS AND TRANSITIONS

Animation-Delay - define how long to wait before animation begins in seconds or milliseconds

Animation-fill-mode - define what happens before first animation.
none, backwards, forwards, both

animation-fill-mode: both;

Fill mode	page load	1s	2s	after 2s
none	green	red	blue	green
backwards	red	red	blue	green
forwards	green	red	blue	blue
both	red	red	blue	blue

Animation-play-state - running or paused.

Shorthand - space-separated list of values