**CSS Active Elements**

**transitions, transforms, animations**

# [Transforms](https://github.com/nycda-staff/nycda-curriculum/blob/master/lectures/wdi-css-animation/wdi-css-animation.md)

## 2d transforms

source: https://www.w3schools.com/css/css3\_2dtransforms.asp

allows translate, rotate, scale, and skew effects to be applied to elements

transform types

translate() move element along x and/or y axes

transform: translate(50px, 100px);

rotate() rotate element by specified degrees

transform: rotate(20deg);

transform: rotate(-20deg);

scale() expand and/or shrink element width and/or height

transform: scale(2, 3);

skewX() skews an element along the X-axis by the given angle

transform: skewX(20deg);

skewY() skews an element along the Y-axis by the given angle

transform: skewY(20deg);

skew() skews an element along the X and Y-axes by the given angles

transform: skew(20deg, 10deg);

matrix() combines all the 2D transform methods into one

matrix(scaleX(),skewY(),skewX(),scaleY(),translateX(),translateY())

transform: matrix(1, -0.3, 0, 1, 0, 0);

http://www.useragentman.com/blog/2011/01/07/css3-matrix-transform-for-the-mathematically-challenged/

## overlays

assigning class on hover

container element (example: box5)

holds initial state and overlay element

must have a specified position property (e.g. position: relative;)

initial overlay element properties (example: slidein)

has position: absolute; property (to hold initial location values)

initial values for width, height, left, right, top, bottom

triggered overlay element properties

final values for width, height, left, right, top, bottom

index.html

<div id="box5">

<div class="slidein">

<div class="text">Slide-in</div>

</div>

</div>

styles.css

.slidein {

position: absolute;

width: 0;

height: 100%;

bottom: 0;

left: 100%;

right: 0;

overflow: hidden;

background-color: #008CBA;

transition: 500ms ease;

}

#box5:hover .slidein {

width: 100%;

left: 0;

}

## prefixes

different browsers respond differently to transition attributes

prefixes assure that user's browser can process animation code

-webkit-transition-duration: 500ms;

-moz-transition-duration: 500ms;

-ms-transition-duration: 500ms;

-o-transition-duration: 500ms;

transition-duration: 500ms;

# [Transitions](https://github.com/nycda-staff/nycda-curriculum/blob/master/lectures/wdi-css-animation/wdi-css-animation.md)

## sources

https://www.w3schools.com/css/css\_pseudo\_classes.asp

https://www.w3schools.com/css/css3\_transitions.asp

https://www.w3schools.com/css/css3\_intro.asp (menu: transitions, animations)

## pseudo classes

defines a special state (temporary class) of an element

syntax

selector:pseudo-class {

property:value;

}

example

.menu-item:hover {

background-color: blue;

}

## hover selector

triggered by mouseover

can be used on any element (not just anchor tags)

must have <!DOCTYPE> declaration in index.html for non-anchor elements to work

## transitions

changes css property values smoothly from one value to another over a given duration

specifying transitions -- must specify the property to transition and the duration of transition

property to add an effect to

duration of the effect

s seconds

ms milliseconds

example:

/\* ======= start values ======= \*/

#box1 { // initial conditions (before transition applied)

background-color: thistle;

color: black;

}

/\* ======= hover values ======= \*/

#box1:hover { // triggers start of transition

background-color: purple;

color: white;

}

/\* ======= transition values ======= \*/

#box1 {

transition: width; // which property to apply transition to

transition-duration: 100ms; // duration of transition (milliseconds)

}

transition options

transition used for shorthand specification of transitions

transition-delay specifies a delay (in seconds) for the transition effect

transition-duration how many seconds or milliseconds a transition effect takes to complete

transition-property name of the CSS property the transition effect is for

transition-timing-function speed curve of the transition effect

shorthand examples:

transition: width 2s;

transition: width 2s, height 4s;

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

transition-timing-functions

specifies a speed curve for transition period

transition timing options

ease - specifies a transition effect with a slow start, then fast, then end slowly (this is default)

linear - specifies a transition effect with the same speed from start to end

ease-in - specifies a transition effect with a slow start

ease-out - specifies a transition effect with a slow end

ease-in-out - specifies a transition effect with a slow start and end

cubic-bezier(n,n,n,n) - lets you define your own values in a cubic-bezier function

examples:

#div1 {

transition-timing-function: linear;

}

#div2 {

transition-timing-function: ease;

}

# Animations

## Animation Basics

Animations in CSS are handled in two major ways:

• Transitions - for simple animations

• KeyFrames - for more complex animations

## Transitions

simple animations created by changing CSS properties from one state to another

Transition Properties

• transition-property - the name or names of the CSS properties to which transitions should be applied

• transition-duration - the amount of time a transition lasts

• transition-delay - the amount of time to wait before initiating a transition

• transition-timing-function - the animation function to use to initiate the transition

Transition Example

changes width, height, background-color over 1 second

/\* starting specifications \*/

.box {

background-color: #eeffee;

width: 100px;

height: 100px;

transition-property: width, height, background-color;

transition-duration: 1s;

}

/\* target specifications triggered on hover \*/

.box:hover {

background-color: #ffcccc;

width: 200px;

height: 200px;

}

## Keyframes

allow more control over how animations take place

define the start state, end state, and intermediate states of an animation

the browser executes transitions according to keyframe specifications

Primary Keyframe Properties

• animation-name - the name you chose for the animation

• animation-duration - the amount of time an animation lasts

• animation-delay - the amount of time to wait before initiating an animation

Keyframes Examples

/\* sets the keyframe animation and the duration \*/

p {

animation-name: slidein;

animation-duration: 3s;

}

.moving-div:hover {

animation-name: multiProps;

animation-duration: 5s;

}

/\* defines the keyframe animation \*/

@keyframes slidein {

/\* starting property values \*/

from {

margin-left: 100%;

width: 300%;

}

/\* ending property values \*/

to {

margin-left: 0%;

width: 100%;

}

}

@keyframes multiProps {

0% {

color: white;

background-color: red;

}

25% {

width: 80px;

height: 45px;

line-height: 45px;

color: blue; // text color changes to blue

background-color: yellow; // bg color changes to yellow

}

50% {

width: 200px; // width expands from 80px to 200px

height: 200px; // height expands from 45px to 200px

margin-left: 10%; // margin-left increases by 10%

color: yellow; // text color changes to yellow

background-color: blue; // bg color changes to blue

}

100% {

color: lawngreen; // text color changes to lawngreen

background-color: green; // bg color changes to green

}

}

## terminology

properties

transition-property - CSS properties to which transitions should be applied

transition-duration - the amount of time a transition lasts

transition-delay - the amount of time to wait before initiating a transition

transition-timing-function - initiate/complete (e.g easeOutCubic, easeInSine)

properties/description

@keyframes Specifies the name you chose for the animation

animation Shorthand property for setting all the animation properties

animation-name Specifies the name of the @keyframes animation

animation-delay Specifies a delay for the start of an animation

animation-duration Specifies seconds or milliseconds to complete one cycle

animation-direction Specifies reverse direction or alternate cycles

animation-fill-mode Specifies styles when animation is delayed or is finished

animation-play-state Specifies whether the animation is running or paused

animation-iteration-count Specifies the number of times an animation should be played

animation-timing-function Specifies the speed curve of the animation