CSS Transitions

**How transitions work, breakdown:**  
.element {  
 transition: [property] [duration] [ease] [delay];  
}

**Example:**  
.element {  
 transition: opacity 300ms ease-in-out 1s; background-color 300ms ease-in-out 1s;  
}

**Animatable properties with transitions**Font-size:   
background-color:  
width:  
left:

<http://oli.jp/2010/css-animatable-properties/>

**Performant properties, what you should only animate otherwise the frame rate will be slower than 60fps**  
position:  
Scale:  
rotation:  
opacity:  
  
<http://www.html5rocks.com/en/tutorials/speed/high-performance-animations/>

**Two ways to perform a transition**Hover pseudo classClass changes

**The best way of creating the smoothest hover animation is to create a parent and child element, and animate the child. This ensures that there is no jittering in the hover state… example below.**

<!doctype html>

<html>

<head>  
 <style>  
 .trigger {  
 Width: 200px;  
 height: 200px: border: 20px solid #ccc;  
 }

.box {  
 display: inline-block;  
 Background-color: pink;  
 width: 200px;  
 height: 200px;  
 transition: transform 300ms ease-in-out;  
 pointer-events: none; **//use this to stop clicks in areas not required** }

.trigger:hover .box {  
 transform: translate(200px, 150px) rotate(20deg);  
 }

</style>  
</head>

<body>  
 <div class="trigger">  
 <div class="box">sss</div>  
</div>

</body>

</html>

**Alternatively use JQuery to perform a class swap**  
$(“.trigger”).on(“click”, function(){  
$(this).toggleClass(‘whatever’);

});

**And apply the class to the css**.trigger.whatever .box {  
 transform: translate(200px, 150px) rotate(20deg);

}

**You can change the easing on the animation by inspecting the element and adding the code to the css**cubic-bezier(0.46, 0.03, 0.52, 0.96);

CSS Animation – keyframes

**Keyframes are used to define specific changes in an animation, they can be set up as “from - to” or written in % “0% - 100%” or “0% 20% 50% 80% 100%” etc..**

**The name relates to the animation name that is added to the element to be animated.**

@keyframes [name] {

From {  
 [styles]

}

To {  
 [styles]

}

}

@keyframes potato {

0% {  
 width: 200px;  
 background-color: orange;  
 }

50% {  
 width: 200px;  
 background-color: red;  
 }

100% {  
 width: 400px;  
 background-color: green;  
 }

}

Animation   
  
**How animations work, breakdown**  
.element {  
animation: [name] [duration] [timing-function] [delay] [iteration-count] [direction] [fill-mode] [play-state];   
}  
  
**example**  
.element {  
animation: myframes 2s ease-in-out 0s infinite normal forwards paused;  
}

**The long-hand approach to animation in css is written as follows:**

animation-name: myframes;  
animation-duration: 2s;  
animation-timing-function: linear/ease/ease-in/ease-out/ease-in-out/cubic-bezier; //how the animation runs  
animation-delay: 0s; // how long before the animation starts  
animation-iteration-count: 3; //animation cycles  
animation-direction: normal/reverse/alternate; //alternate relates to back and forth  
animation-fill-mode: normal/forwards; //select forward to finish the animation as intended  
animation-play-state: running/paused; //you could move animation-play-state to a class, set it to paused then run a toggleClass command in jQuery to pause the animation on click.