

Computer Science Principles

Web Programming

Web-Presentation Design with CSS

CHAPTER17: LAYOUT MANAGEMENT III (ANIMATION)

DR. ERIC CHOU

IEEE SENIOR MEMBER



CSS3

Chapter 11: CSS Hierarchy and Selectors

Chapter 12: Text, Image and Foreground (Contents)

Chapter 13: Color and Background (Contents)

Chapter 14: Box Model (Padding, Border, and Margin)

Chapter 15: Layout Management (Floating and Positioning: where should the Element go)

Chapter 16: Layout Management (Page Level Planning)

Chapter 17: Layout Management (Transition, Transforms, and Animation: space and time domain transformation)

Chapter 18: CSS Techniques (Put Everything Together)

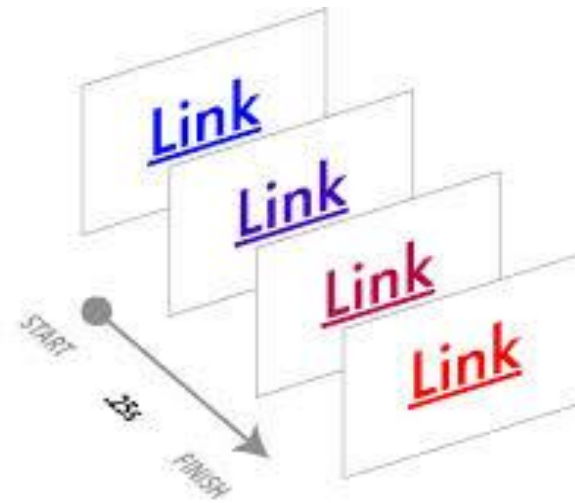
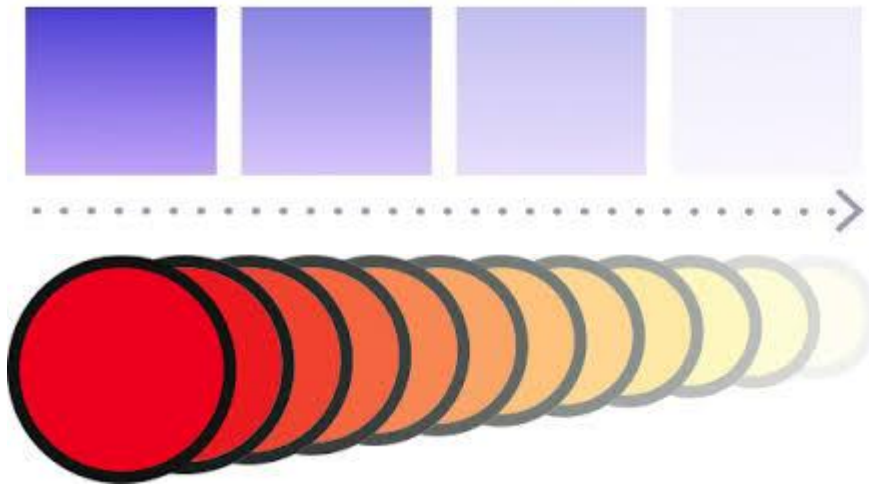


Transition

LECTURE 1

CSS Transitions

- Transition is used to change the status of some elements from one state to another.
- The CSS transition will smooth out the state changes from state to state over time by filling frames in between. Animators call that **tweening**.
- CSS transitions were originally developed by the **webkit** team for the safari browser, but they are now a Working Draft at the W3C.





Transition Basics

Shorthand `transition: property duration timing-function delay;`

Which CSS property to change:

`transition-property: property-name | all none`

How long it should take:

`transition-duration: time`

The manner in which the transition accelerates:

`transition-timing-function: ease | linear | ease-in | ease-out | ease-in-out | step-start | step-end | step | cubic-Bezier(#, #, #, #)`

Whether there should be a pause before it starts:

`transition-delay: time`

Transition Property and Duration Example

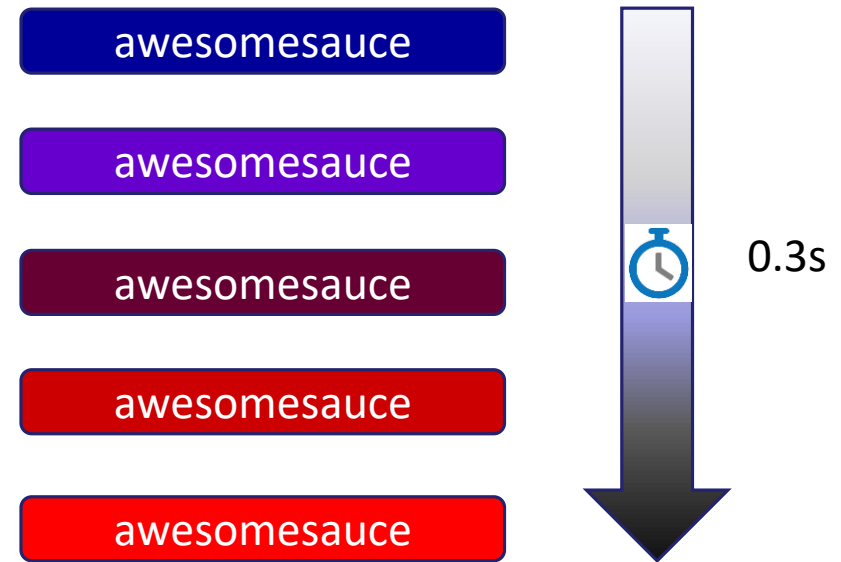
The markup:

```
<a href="" class="smooth">awesomesauce</a>
```

The styles:

```
a.smooth { /* base condition */
  display: block;
  text-decoration: none; text-align: center;
  padding: 1em 2em ;
  width: 10em;
  border-radius: 1.5em;
  color:#fff;
  background-color: mediumblue;
  transition-property: background-color;
  transition-duration: 0.3s; }
```

```
a.smooth:hover, a.smooth:focus{ ← # actions for transition
  background-color: red; }
```





Timing

LECTURE 2

Timing Functions

www.joelambert.co.uk/morf/

Source & Docs @ GitHub

Download morf.js

Download morf.min.js

Morf comes bundled with the fantastic [Shifty.js](#) for tweening regular CSS properties.

If you already use Shifty heres a version of Morf without Shifty pre-bundled. Morf requires at least Shifty 0.1.3.

morf.noshifty.js

morf.noshifty.min.js

Native

These are the natively supported easing functions, built into WebKit.

linear

ease

ease-in

ease-out

ease-in-out

Custom

These are custom easing functions (*thanks to Robert Penner & Thomas Fuchs*) that can produce much more interesting transitions.

sinusoidal

spring

easeTo

easeFrom

easeFromTo

bouncePast

bounce

swingTo

swingFrom

swingFromTo

elastic

easeInOutBack

easeOutBack

easeInBack

easeOutBounce

easeInOutCirc

easeOutCirc

easeInCirc

easeInOutExpo

easeOutExpo

easeInExpo

easeInOutSine

easeOutSine

easeInSine

easeInOutQuint

easeOutQuint

easeInQuint

easeInOutQuart

easeOutQuart

easeInQuart

easeInOutCubic

easeOutCubic

easeInCubic

easeInOutQuad

easeOutQuad

easeInQuad

linear

Timing Function Description

- **ease:** Starts slowly, accelerates quickly, and then slow down at the end. This is the default value and works just fine for most short transitions.
- **linear:** Speed stays consistent from the transition's beginning to end.
- **ease-in:** Start slowly, then speed up.
- **ease-out:** Start out fast, then slows down.
- **ease-in-out:** Starts slowly, speeds up, and then slow down again at the very end. It is similar to ease, but with less pronounced acceleration in the middle.

Timing Function Description

- **ease-in-out:** Starts slowly, speeds up, and then slows down again at the very end. It is similar to ease, but with less pronounced acceleration in the middle.
- **cubic-bezier(#, #, #, #):** This is a function for defining a Bezier curve that describes the transition acceleration. It's super math-y and I can't explain it all here.
- **steps(#, start | end):** Divides the transitions into a number of steps as defined by stepping function. The first value is the number of steps, and the start and end keywords define whether the change in state happens at the beginning (start) or end of each step.

Timing Function Description

- step-start:** The changes states in **one step**, at the beginning of the duration time (the same as steps(1, start)). The result is **a sudden state change**, the same as if no transition had applied at all.
- step-end:** This changes states in one step, **at the end** the duration time (the same as steps(1, end)).

Setting a delay

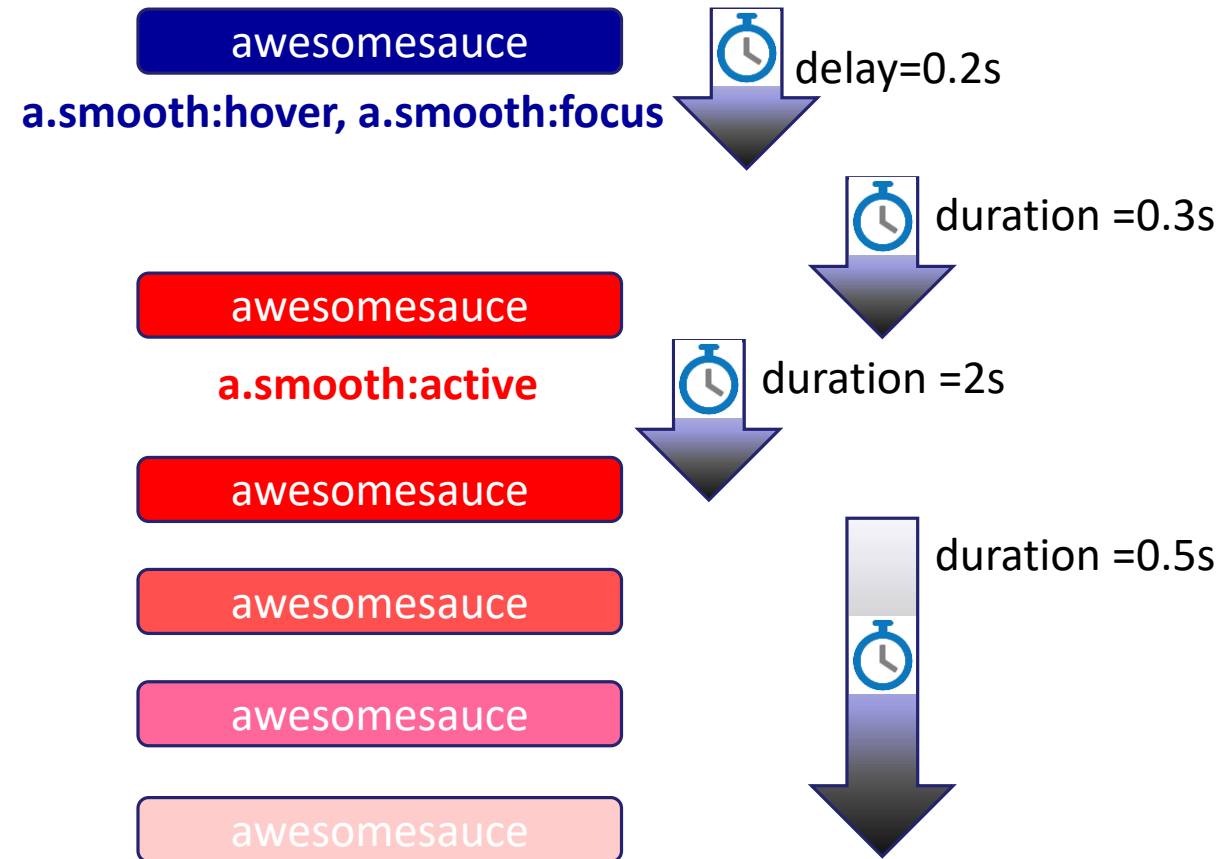
The transition-delay property delays the start of the animation by a specified amount of time.

```
a.smooth { ...
  transition-property: background-color;
  transition-duration: 0.3s;
  transition-timing-function: ease-in-out;
  transition-delay: 0.2s; }
```

```
a.smooth { ...
  transition-property: opacity;
  transition-duration: .05s;
  transition-timing-function: ease-out;
  transition-delay: 2s; }
```

```
a.smooth:hover, a.smooth:focus {
  background-color: red; }
```

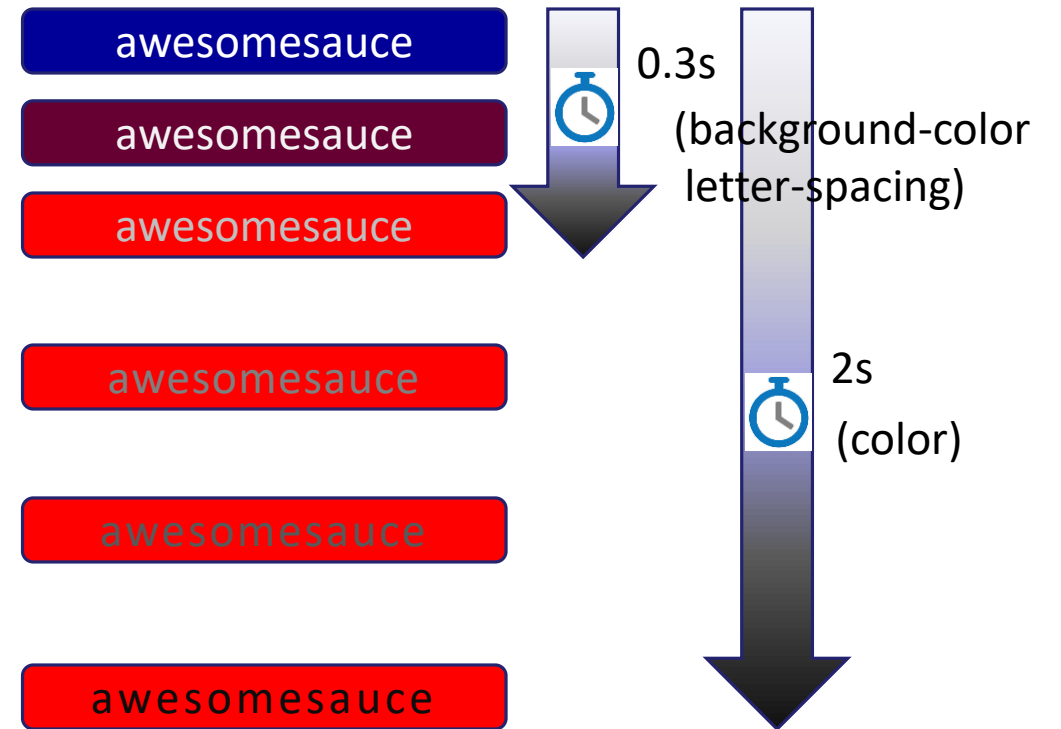
```
a.smooth:active { opacity:0; }
```



Multiple Transitions in One Action

```
a.smooth {
  ...
  transition-property: background-color, color, letter-spacing;
  transition-duration: 0.3s, 2s, 0.3s;
  transition-timing-function: ease-out, ease-in, ease-out;
}

a:hover, a:focus {
  background-color: red;
  letter-spacing: 3px;
  color: black;
}
```



Backgrounds	Border/Outlines	Color/Opacity	Font/text	Element Measurements	Position
background-color	border-bottom-color	Color	font-size	height	top
background-position	border-bottom-width	Opacity	font-weight	width	right
	border-left-color	visibility	letter-spacing	max-height	bottom
	border-left-width		line-height	max-width	left
	border-right-color		text-indent	min-height	z-index
	border-right-width		text-shadow	min-width	clip
	border-top-color		word-spacing	margin-bottom	
	border-top-width		vertical-align	margin-left	
	border-top=color			margin-right	
	border-top-width			margin-top	
	border-spacing			padding-bottom	
	outline-color			padding-left	
Animatable CSS Properties	outline-offset			padding-right	
	outline-width			padding-top	
				crop	



Transform

LECTURE 3



CSS Transforms

Scale

Translate

Rotate

Skew



skew



scale



translate



rotate



skew X



scale X



translate X



skew Y



skew Y



scale Y

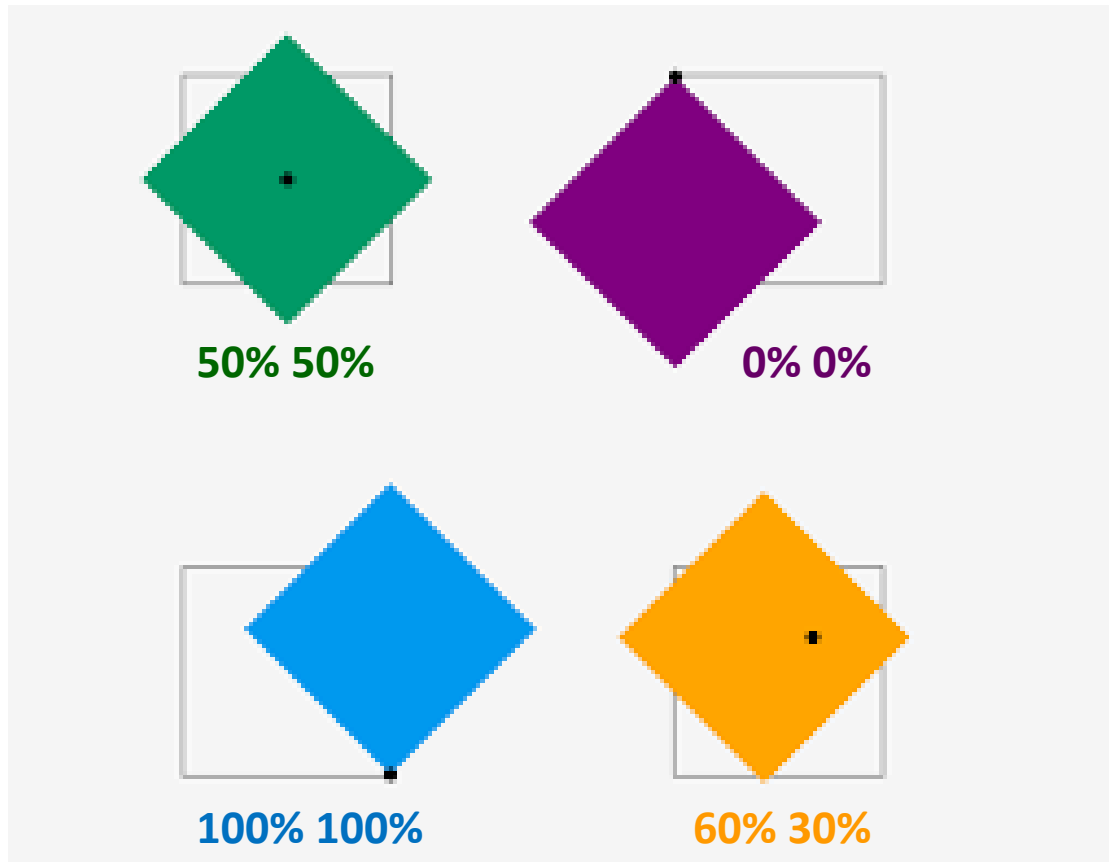


translate Y



flip Y

Transform Origin (Rotate)



`transform-origin: percentage | length | left | center | right | top | bottom`

`transform: rotate(-10deg); /* in degree */`

Transformable Elements:

- `img`, `canvas`, `form input`, and `embedded media`

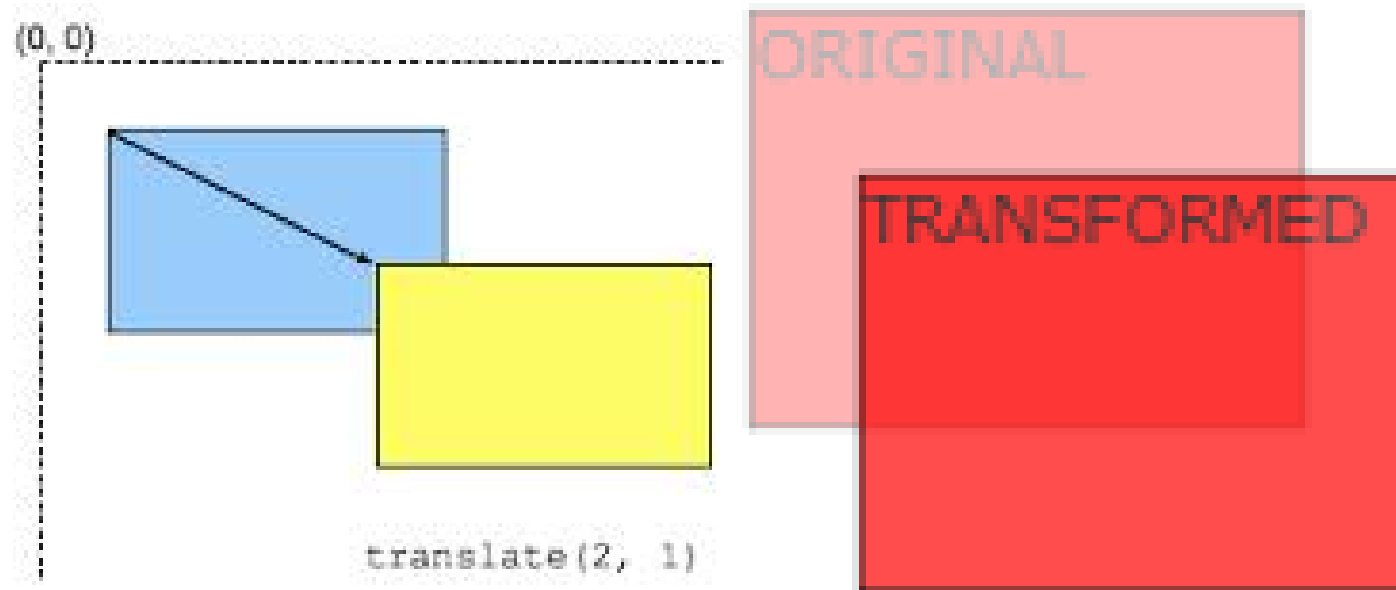


Transforming the position(translate)

`transform: translateX(X);` /* `transform: translateX(50px);` */

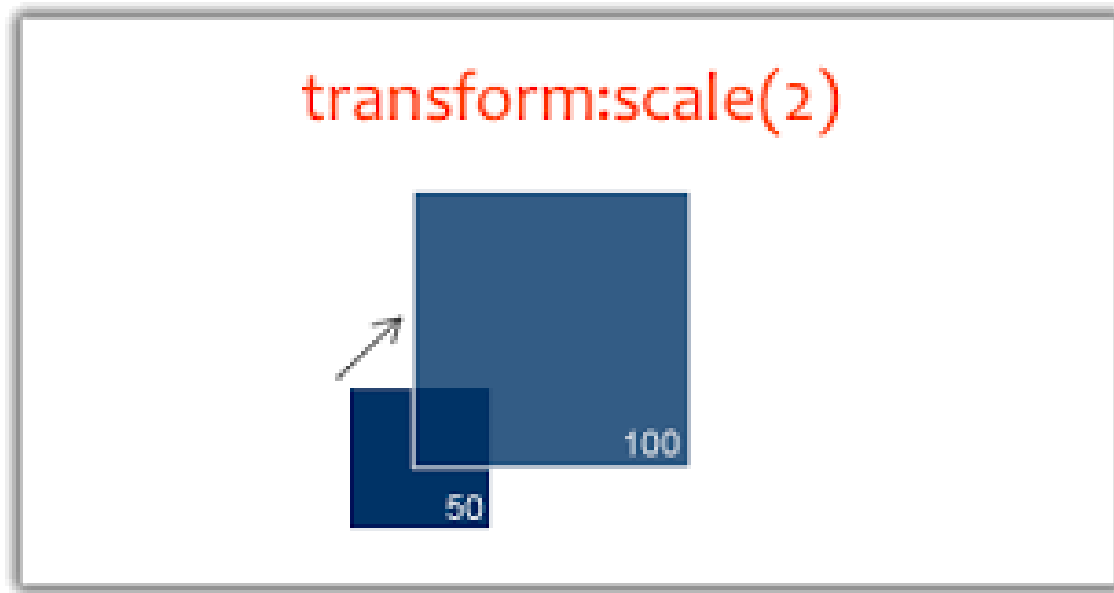
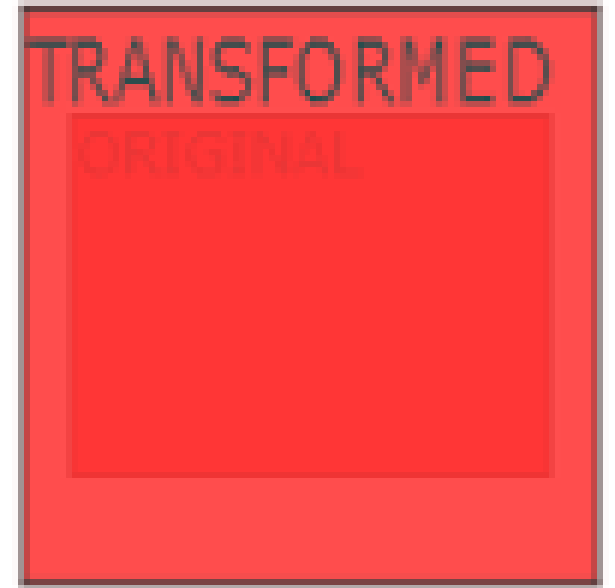
`transform: translateY(Y);` /* `transform: translateY(20px);` */

`transform: translate(X, Y);` /* `transform: translate(50px, 20px)` */



Transform the size (scale)

```
transform: scaleX(Xsize);           /* transform: scaleX(1.5); */
transform: scaleY(Ysize);           /* transform: scaleY(2);   */
transform: scale(Xsize, Ysize);     /* transform: scale(2, 1.5); */
```



Making it slantry (skew)

Transform: skew(Xangle);

```
/* transform: skewX(15deg); */
```

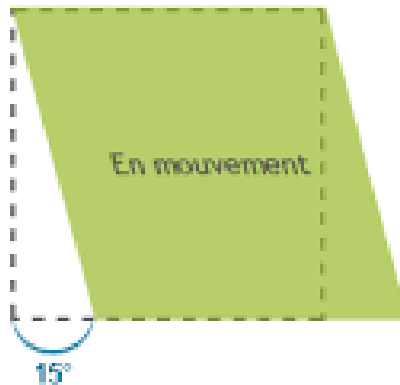
Transform: skew(Yangle);

```
/* transform: skewY(15deg); */
```

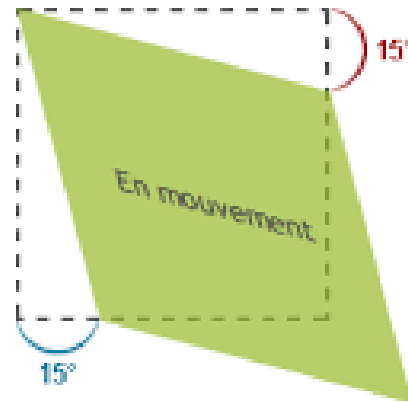
Transform: skew(Xangle, Yangle);

```
/* transform: skew(15deg, 15deg); */
```

skew(15deg, 0deg)



skew(15deg, 15deg)





3D Transform

LECTURE 4

3D Transforms Browser Support

-ms- -moz- -webkit-

CSS3 3D Transforms - Working Draft

*Usage stats:

Global

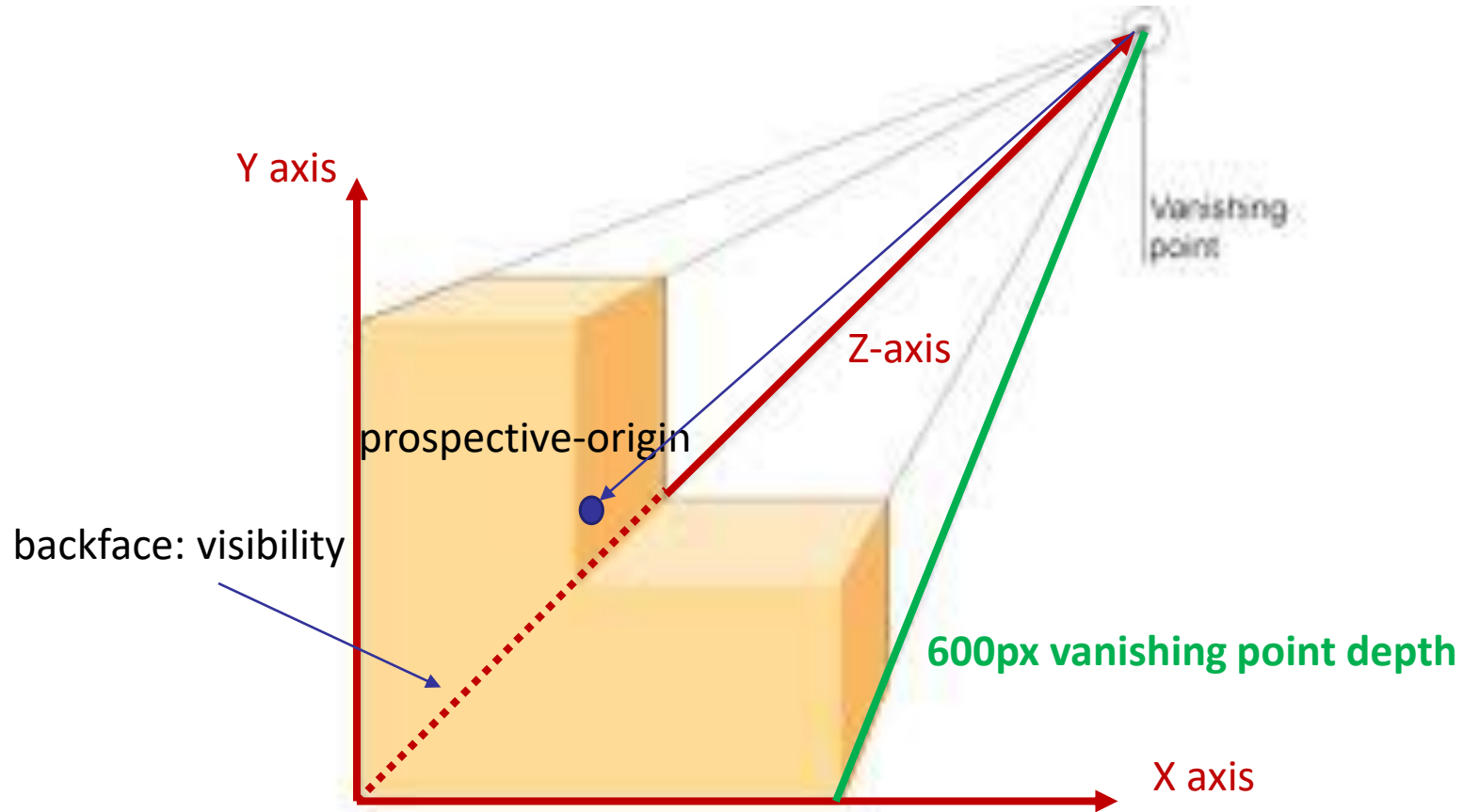
Support:

53.2%

Method of transforming an element in the third dimension

Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
									2.1
						3.2 -webkit-			2.2
	7.0	3.6				4.0-4.1 -webkit-		10.0	2.3
	8.0	12.0 -moz-				4.2-4.3 -webkit-		11.5	3.0 -webkit-
Current	9.0	13.0 -moz-	19.0 -webkit-	5.1 -webkit-	12.0	5.0 -webkit-	5.0-6.0	12.0	4.0 -webkit-
Near future	10.0 -ms-	14.0 -moz-	20.0 -webkit-	5.2 -webkit-					
Farther future		15.0 -moz-	21.0 -webkit-						

3D Model



Vanishing Point Projection on X-Y Plane:

prospective-origin: 50% 50%

Vanishing Point Depth:

prospective: 600px;



Set Up 3D Model

-webkit-perspective: 600px;	/* -webkit- browsers */
-webkit-perspective-origin: 50% 50%;	
-webkit-backface-visibility:30%;	
-moz-perspective: 600px;	/* -moz- Firefox browser */
-moz-perspective-origin: 50% 50%;	
-moz-backface-visibility:30%;	
-ms-perspective: 600px;	/* -ms- IE browsers */
-ms-perspective-origin: 50% 50%;	
-ms-backface-visibility:30%;	
perspective: 600px;	/* general browsers (unknown) */
perspective-origin: 50% 50%;	
backface-visibility:30%;	

3D Operators

translate3d, translate, scale3d, scaleZ, rotate3d, rotate, rotate, rotate, rotate, matrix3d

perspective 250

perspective Origin X 50

perspective Origin Y 50

3d matrix of div#inner


<input type="range" value="1"/>	<input type="range" value="0"/>	<input type="range" value="0"/>	<input type="range" value="0"/>
1	0	0	0
<input type="range" value="0"/>	<input type="range" value="1"/>	<input type="range" value="-0.58"/>	<input type="range" value="0"/>
0	1	-0.58	0
<input type="range" value="0"/>	<input type="range" value="-0.77"/>	<input type="range" value="1"/>	<input type="range" value="0"/>
0	-0.77	1	0
<input type="range" value="-58"/>	<input type="range" value="52"/>	<input type="range" value="-58"/>	<input type="range" value="1"/>
-58	52	-58	1

transform-Style ☒ preserve-3d ☐ flat

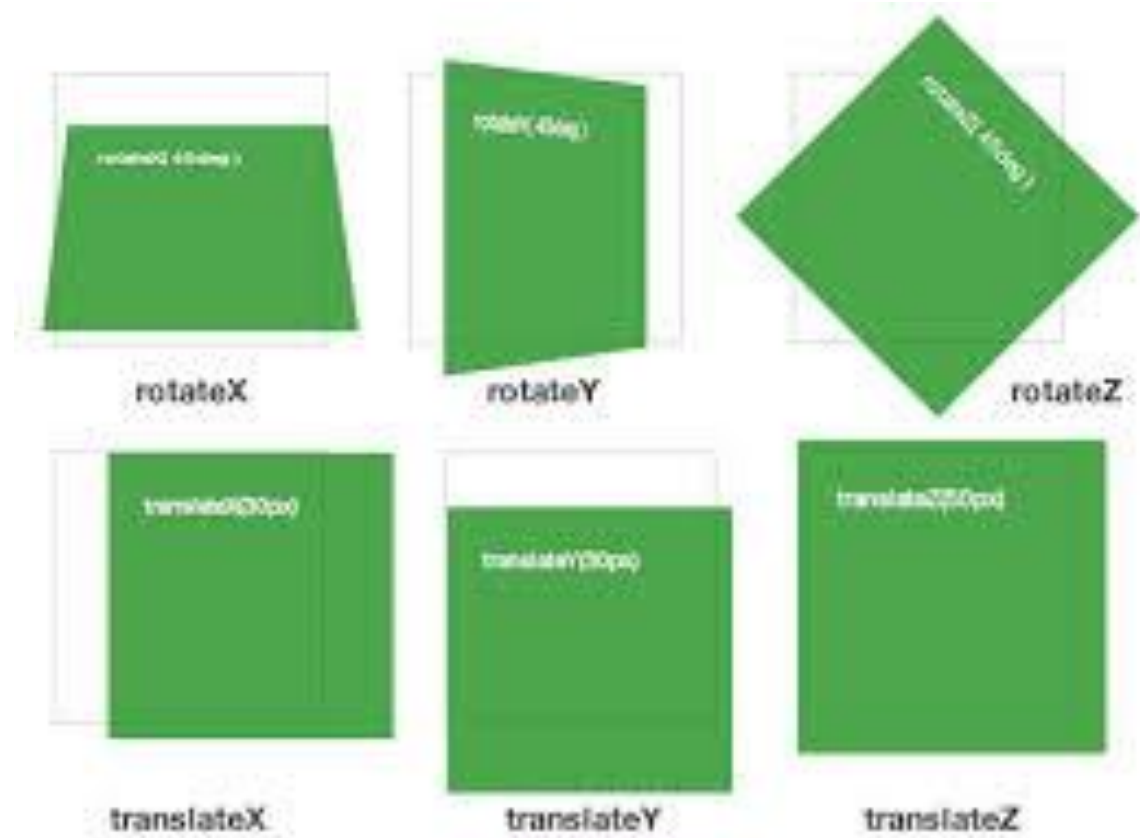
```

-webkit-transform: matrix3d(1, 0, 0, 0,
                           0, 1, -0.58, 0,
                           0, -0.77, 1, 0,
                           -58, 52, -58, 1);

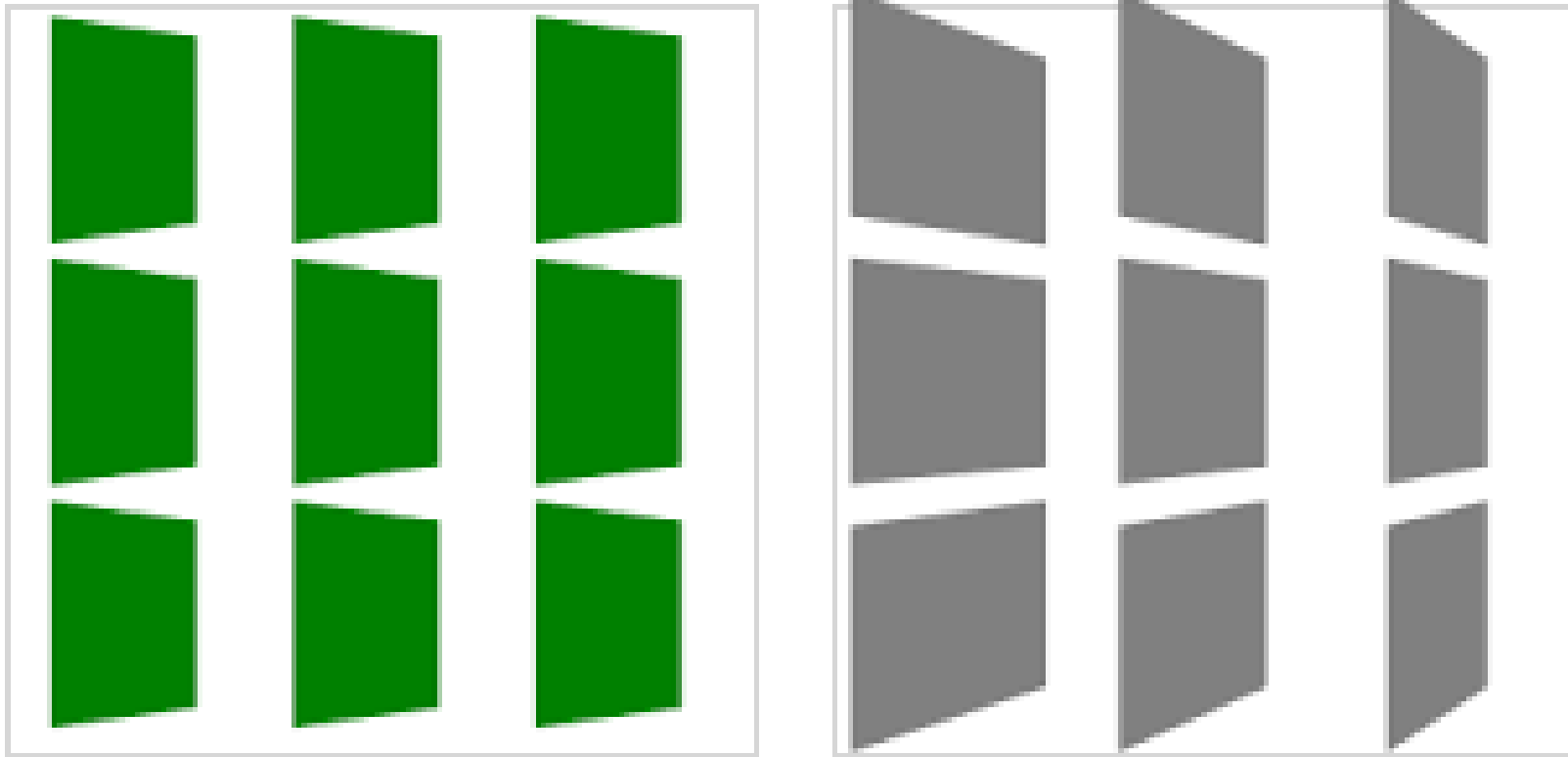
-webkit-transform-style: preserve-3d;
    
```



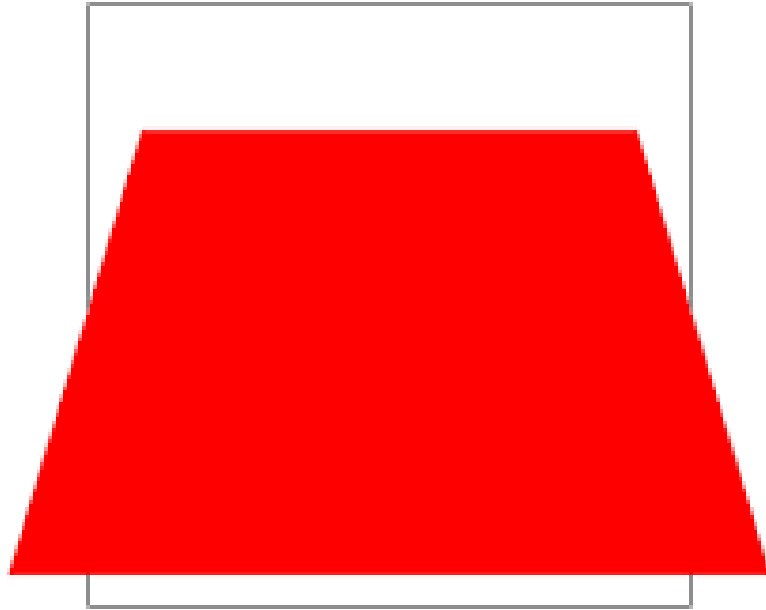
3D Rotation and Translate



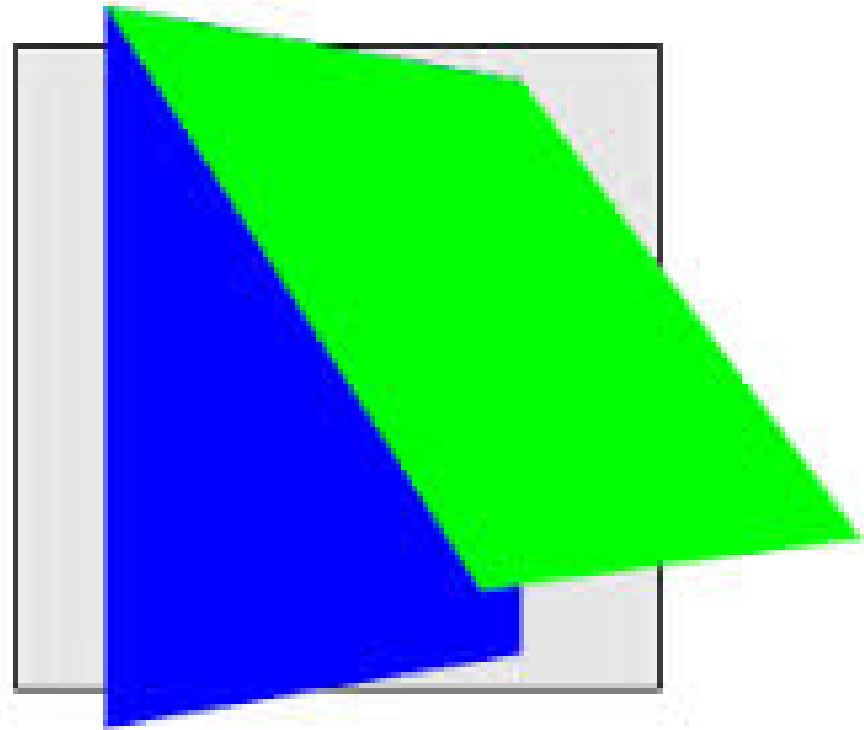
3D RotateY



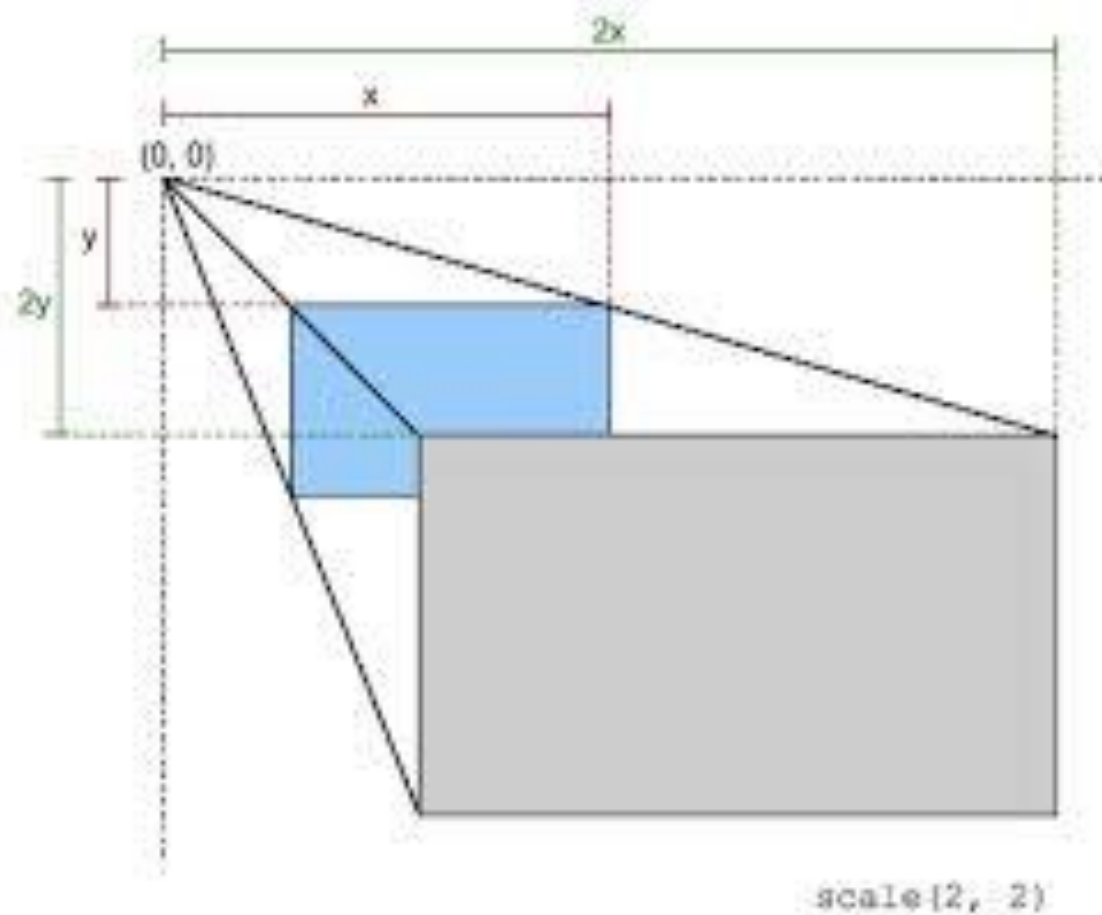
3D RotateX



3D RotateZ



3D Scale



3D Transformed Image





Animation

LECTURE 5

Keyframe Animation

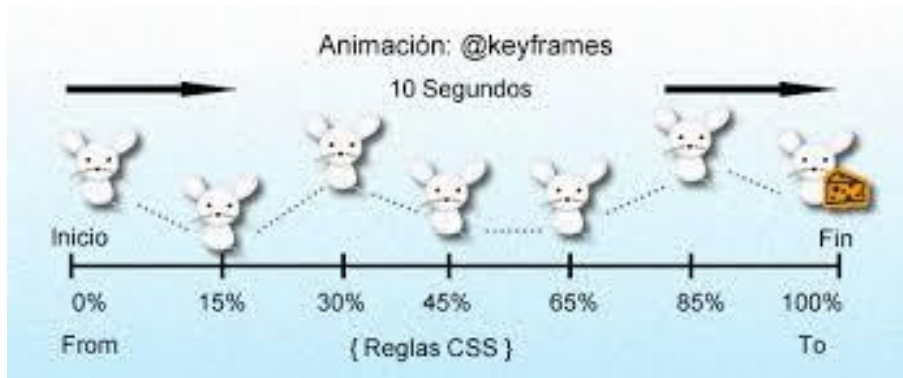
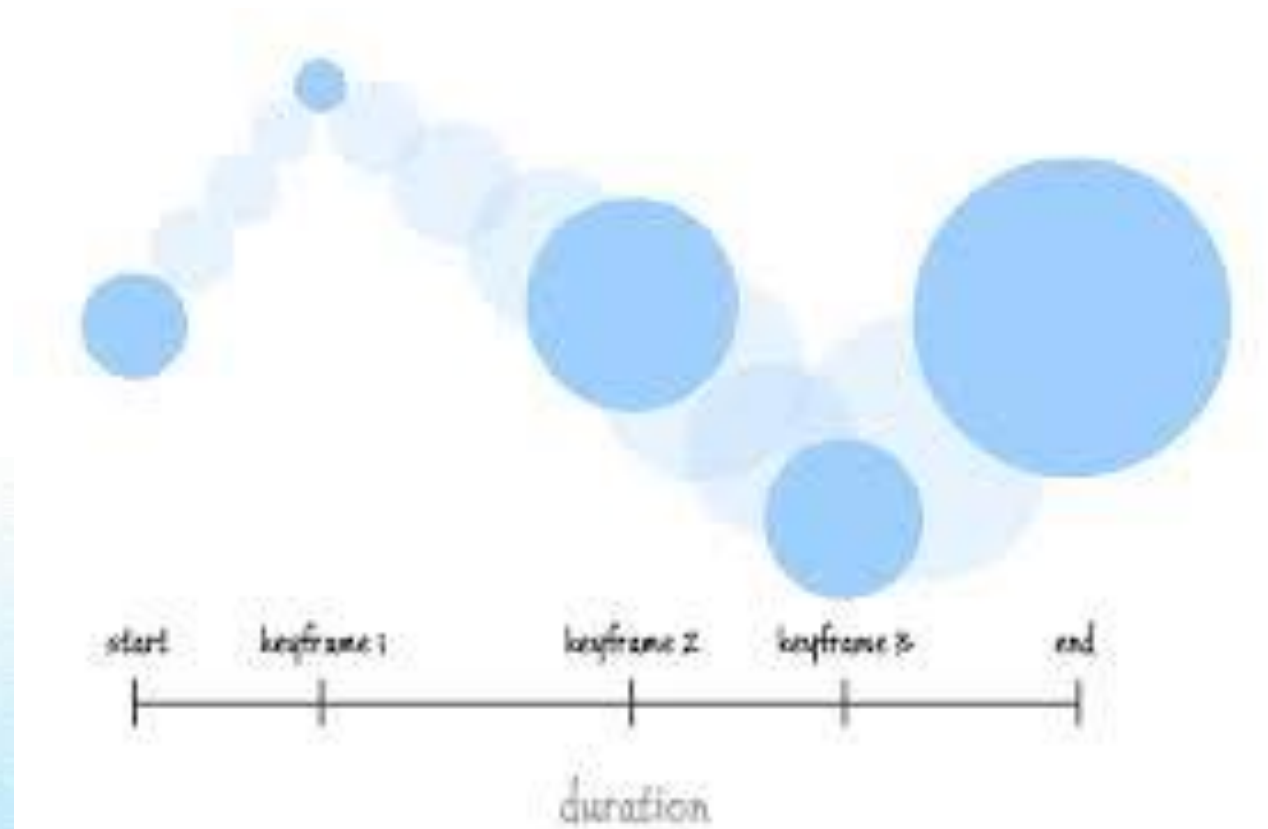
Animation Tool:

If you want to use CSS Animations but lack the wherewithal to learn to code it all yourself, there are tools that give you a timeline interface for creating your animations and generate the HTML and CSS for you. Here are a few as of this writing:

- Tumult Hype, <http://www.tumultco.com/hype/> (Mac Only)
- Sencha Animator, <http://www.sencha.com/products/animator/>
- Adobe Edge, <http://labs.adobe.com/technologies/edge/>

What is @keyframe

- Key frames are the frames that used to interpolate the other frames when used in video compression algorithm or animation.
- To be short, Key frames are the frame with real data. Other frames are created by these frame data.



Establishing the keyframes

- The animation process has two parts: first, establish the key frames with a `@keyframes` rule, and then add animation properties to the elements that will be animated.

```
@keyframes colors {  
  0% {background-color: red; }  
  20% {background-color: orange; }  
  40% {background-color: yellow; }  
  60% {background-color: green; }  
  80% {background-color: blue; }  
  100% {background-color: purple; }  
}
```

```
#magic {  
  ...  
  animation-name: colors;  
  animation-duration: 5s;  
  animation-timing-function: linear;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
}  
Or,  
#magic {  
  animation: colors 5s linear infinite alternate;  
}
```



@keyframes establishment and Adding animation properties

```
@keyframes animation-name: {  
    keyframe { property: value; } /* keyframe in terms of percentage */  
    keyframe { property: value; }  
}
```

Applies to <div id="magic">Magic !</div>

Establishment of animation:

animation-name: Which animation to use?

animation-duration: How long it should take?

animation-timing-function: The manner in which it should accelerate.

animation-delay: Whether to pause before it starts.

animation-iteration-count: How many times it should repeat?

animation-direction: Whether it plays forward, in reverse, or alternates back and forth.

animation-play-state: Whether it should be running or paused. The play-state can be toggled on and off with JavaScript or on hover.

animation-fill-mode: Which animation to use?

@keyframe example

