CREATING BEAUTIFUL CSS ANIMATIONS

@RAMISAYAR

FITC Amsterdam 2016

AGENDA

- History of Animation on the Web
- Introducing CSS3 Animations
- Browser Compatibility
- CSS3 Animation Libraries
- Principles for Beautiful Animations

ASSUMPTIONS

- You know what the web is…
- Working knowledge of HTML5 & CSS3.
- Slides will be made available on SlideShare.net/RamiSayar

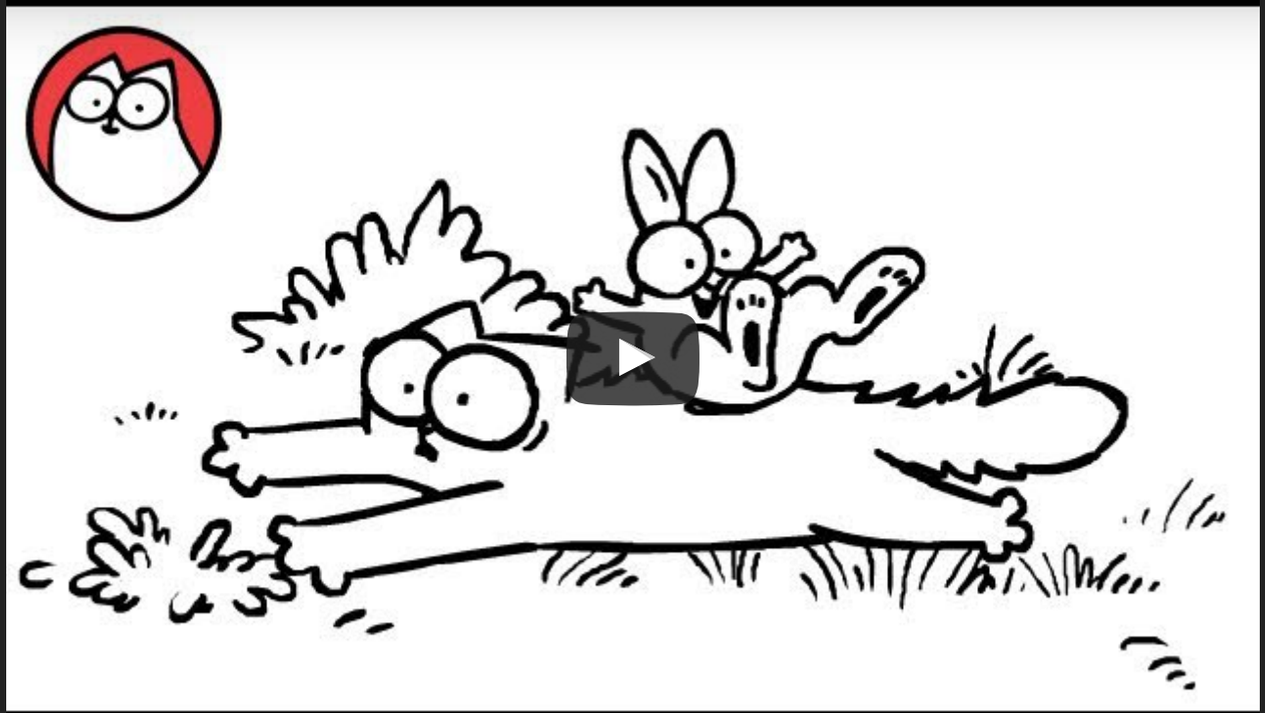
CAMERON'S WORLD

AND THEN THERE WAS FLASH

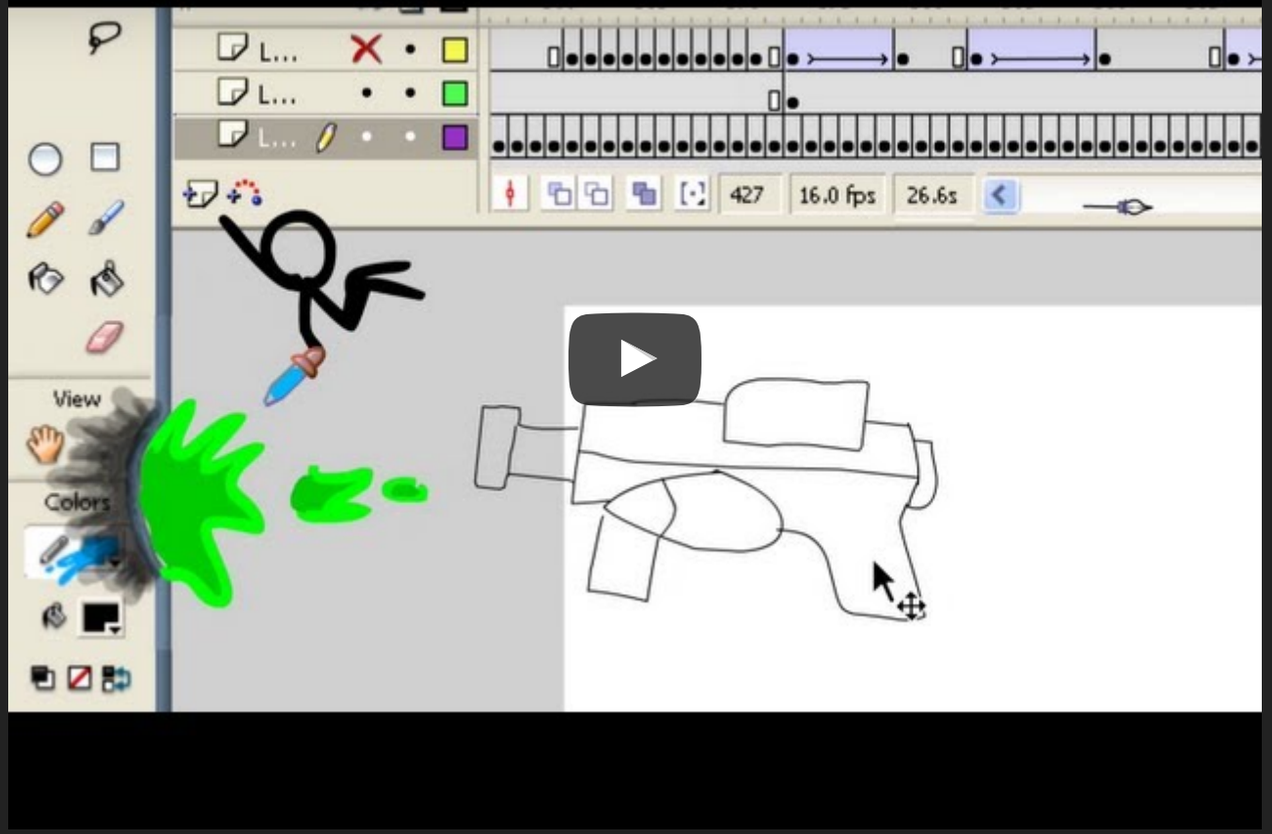
IT WAS A VERY DARK TIME FOR THE WEB. I CAN'T BELIEVE THIS STILL EXISTS...

CLASSIC FLASH ANIMATIONS

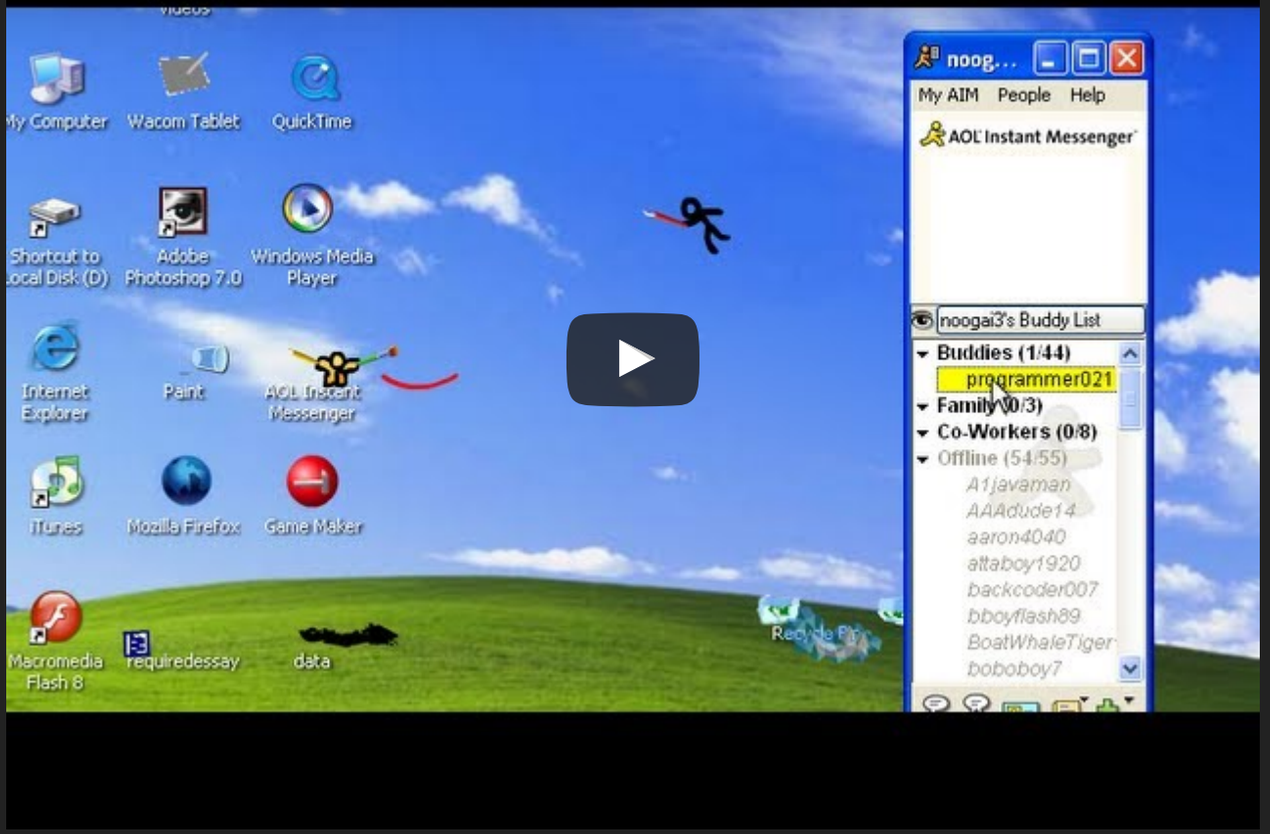
Hop It - Simon's Cat



Animator vs. Animation (original)



Animator vs. Animation II (original)



THINGS DIDN'T END WELL...

APRIL 2010, STEVE JOBS WRITES "THOUGHTS ON FLASH"

Thoughts on Flash

Apple has a long relationship with Adobe. In fact, we met Adobe's founders when they were in their proverbial garage. Apple was their first big customer, adopting their Postscript language for our new Laserwriter printer. Apple invested in Adobe and owned around 20% of the company for many years. The two companies worked closely together to pioneer desktop publishing and there were many good times. Since that golden era, the companies have grown apart. Apple went through its near death experience, and Adobe was drawn to the corporate market with their Acrobat products. Today the two companies still work together to serve their joint creative customers – Mac users buy around half of Adobe's Creative Suite products – but beyond that there are few joint interests.

I wanted to jot down some of our thoughts on Adobe's Flash products so that customers and critics may better understand why we do not allow Flash on iPhones, iPods and iPads. Adobe has characterized our decision as being primarily business driven – they say we want to protect our App Store – but in reality it is based on technology issues. Adobe claims that we are a closed system, and that Flash is open, but in fact the opposite is true. Let me explain.

First, there's "Open".

Adobe's Flash products are 100% proprietary. They are only available from Adobe, and Adobe has sole authority as to their future enhancement, pricing, etc. While Adobe's Flash products are widely available, this does not mean they are open, since they are controlled entirely by Adobe and available only from Adobe. By almost any definition, Flash is a closed system.

Apple has many proprietary products too. Though the operating system for the iPhone, iPod and iPad is proprietary, we strongly believe that all standards pertaining to the web should be open. Rather than use Flash, Apple has adopted HTML5, CSS and JavaScript – all open standards. Apple's mobile devices all ship with high performance, low power implementations of these open standards. HTML5, the new web standard that has been


RIP FLASH. 2015
OCCUPY
FLASH

BUT ONE MORE FOR OLD TIMES SAKE

WHILE FLASH EXISTED, WE HAD JAVASCRIPT & DHTML

DHTML IS THE COMBINATION OF HTML, CSS AND JAVASCRIPT.

- Animate text and images in their document, independently moving each element from any starting point to any ending point, following a predetermined path or one chosen by the user.
- Embed a ticker that automatically refreshes its content with the latest news, stock quotes, or other data.
- Use a form to capture user input, and then process, verify and respond to that data without having to send data back to the server.
- Include rollover buttons or drop-down menus.

DRUNK HARLEQUIN TEXT

DHTML CODE SAMPLE

```
function splat(text){   // randomize text color, size, and weight
  text=text.substring(0,text.length);
  var fs=.5
  for(i=0;text.length > i;i++){
    f=Math.random();
    if (Math.abs(f-fs) < .11) {f=Math.random()};
    if (Math.abs(f-fs) < .11) {f=Math.random()};
    if (f<.1) {c='00ffff'};
    if (f>=.1 && f<.2) {c='7cfc00'};
    if (f>=.2 && f<.3) {c='6495ed'};
    if (f>=.3 && f<.4) {c='ffd700'};
    if (f>=.4 && f<.5) {c='ff7f50'};
    if (f>=.5 && f<.6) {c='ff00ff'};
    if (f>=.6 && f<.7) {c='ccff00'};
    if (f>=.7 && f<.8) {c='ff6964'};
    if (f>=.8 && f<.9) {c='ff4500'};
```

JAVASCRIPT AND HTML5 CANVAS

EXAMPLES:

- Tearable Cloth
- Free Rider HD

Canvas is great for big, complex animations but it absolutely kills accessibility on your website. Not to mention that the JavaScript code quickly becomes complicated.

HISTORY OF ANIMATIONS ON THE WEB

- 1987 – Initial release of Graphics Interchange Format
- 1996 – Initial release of Adobe Flash Player
- 1997 – Netscape releases the “layers” feature in Netscape 4.
  - Layers was a precursor to div. Subsequently, dropped for divs.
  - JavaScript animations popular, part of a set of techniques called Dynamic HTML. Standards movement contributed to death of DHTML.
- 2004 – HTML5 Canvas Introduced by Apple in WebKit.
- 2008 – Initial release of Animated Portable Network Graphics (apng)
- 2009 – Initial Draft of CSS Animations (CSS3 Module)
- 2012-2016 – Resurgence of GIF in popularity???

INTRODUCING CSS3 ANIMATIONS

INTRODUCING CSS3 ANIMATIONS

- CSS3 Animations can be performed with 3 different techniques:
  - Transitions
  - Transforms
  - Animations (Narrow Definition)

CSS3 TRANSITIONS

CSS3 TRANSITIONS

```
transition: [ <transition-property> || <transition-duration>
|| <transition-timing-function> || <transition-delay> ]
```

transition-property includes color, width, height, background-color, padding, margin, top, right, bottom, left, font-size, font-weight, etc.

See W3C Spec.

CSS3 TRANSITIONS

```
transition: [ <transition-property> || <transition-duration>
|| <transition-timing-function> || <transition-delay> ]
```

transition-duration is how long the animation takes in seconds.

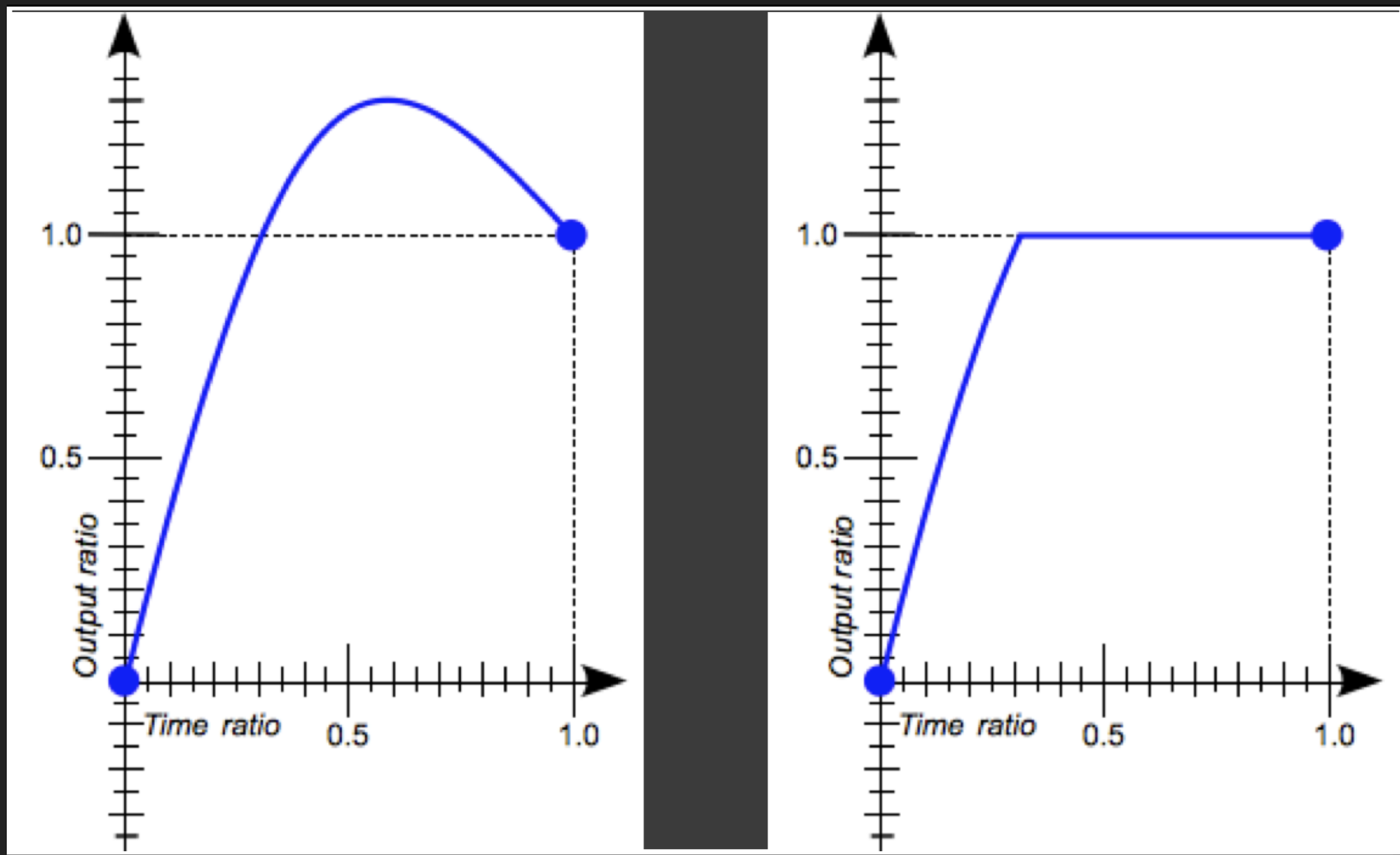
transition-delay is how long the animation waits before changing the transition-property.

CSS3 TRANSITIONS

```
transition: [ <transition-property> || <transition-duration>
|| <transition-timing-function> || <transition-delay> ]
```

transition-timing-function determines how intermediate values of the transition are calculated. Only two classes of timing functions: cubic-bezier() and steps() functions.

CSS3 TRANSITIONS – TIMING FUNCTIONS



Reference MDN

CSS3 TRANSITIONS – CUBIC-BEZIER()

- cubic-bezier() defines a cubic Bézier curve
- Curves are continuous, sometimes called easing functions.

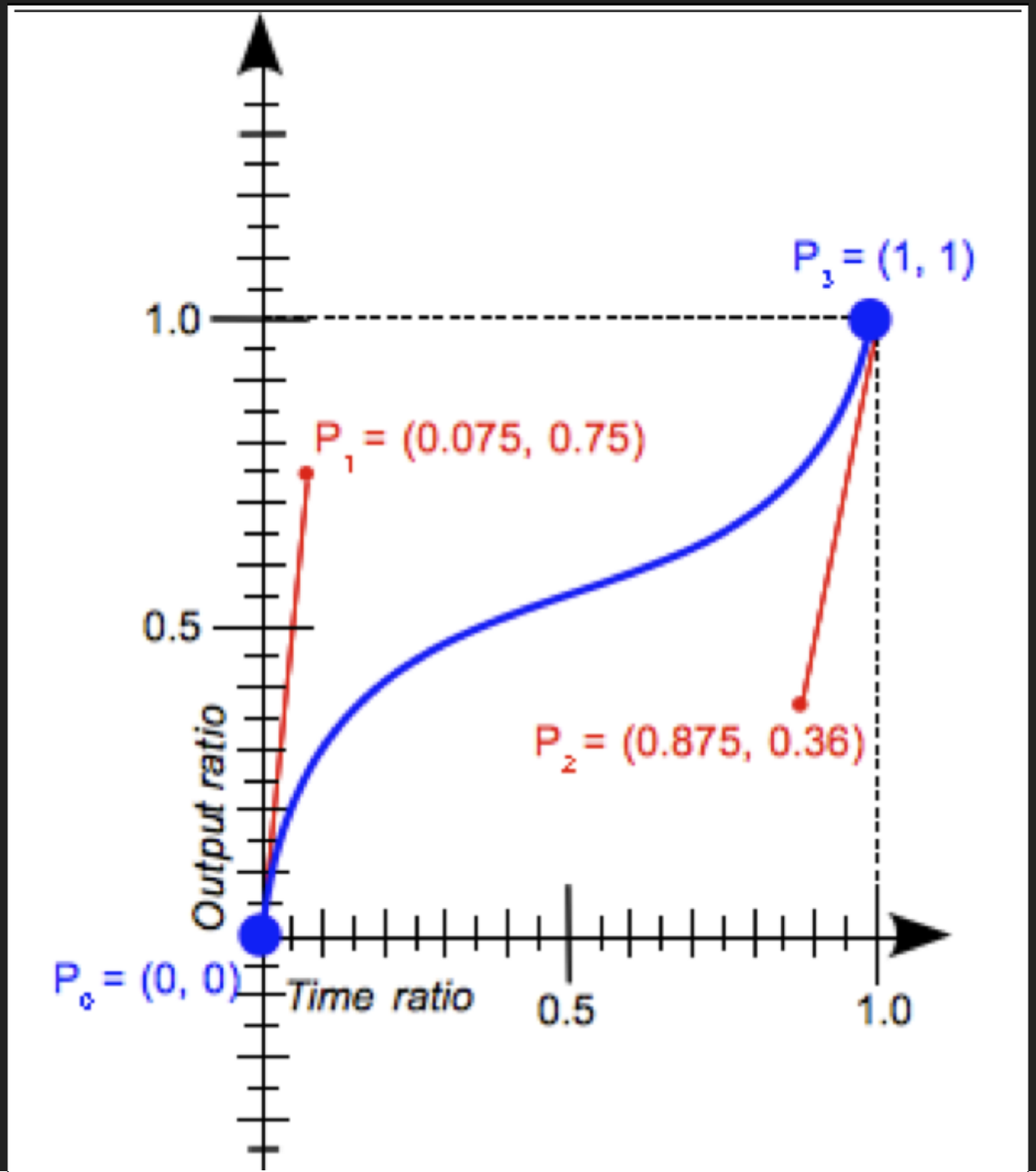
Reference MDN

CSS3 TRANSITIONS – CUBIC-BEZIER()

- Defined by four points P0, P1, P2, and P3. P0 and P3 are the start and the end of the curve.
- In CSS3, P0 is always set to (0, 0) and represents the initial time and the initial state, P3 is (1, 1) and represents the final time and the final state.

Reference MDN

CSS3 TRANSITIONS – CUBIC-BEZIER()



Reference MDN

CSS3 TRANSITIONS – CUBIC-BEZIER()

- Invalid curves are possible and cause the entire property to be ignored.
- P1 or P2 ordinate outside the [0, 1] range may generate bouncing effects.

```
cubic-bezier(x1, y1, x2, y2);
```

CSS3 TRANSITIONS – CUBIC-BEZIER()

- Common cubic bezier are named in CSS3.
- Cheat Sheet easings.net

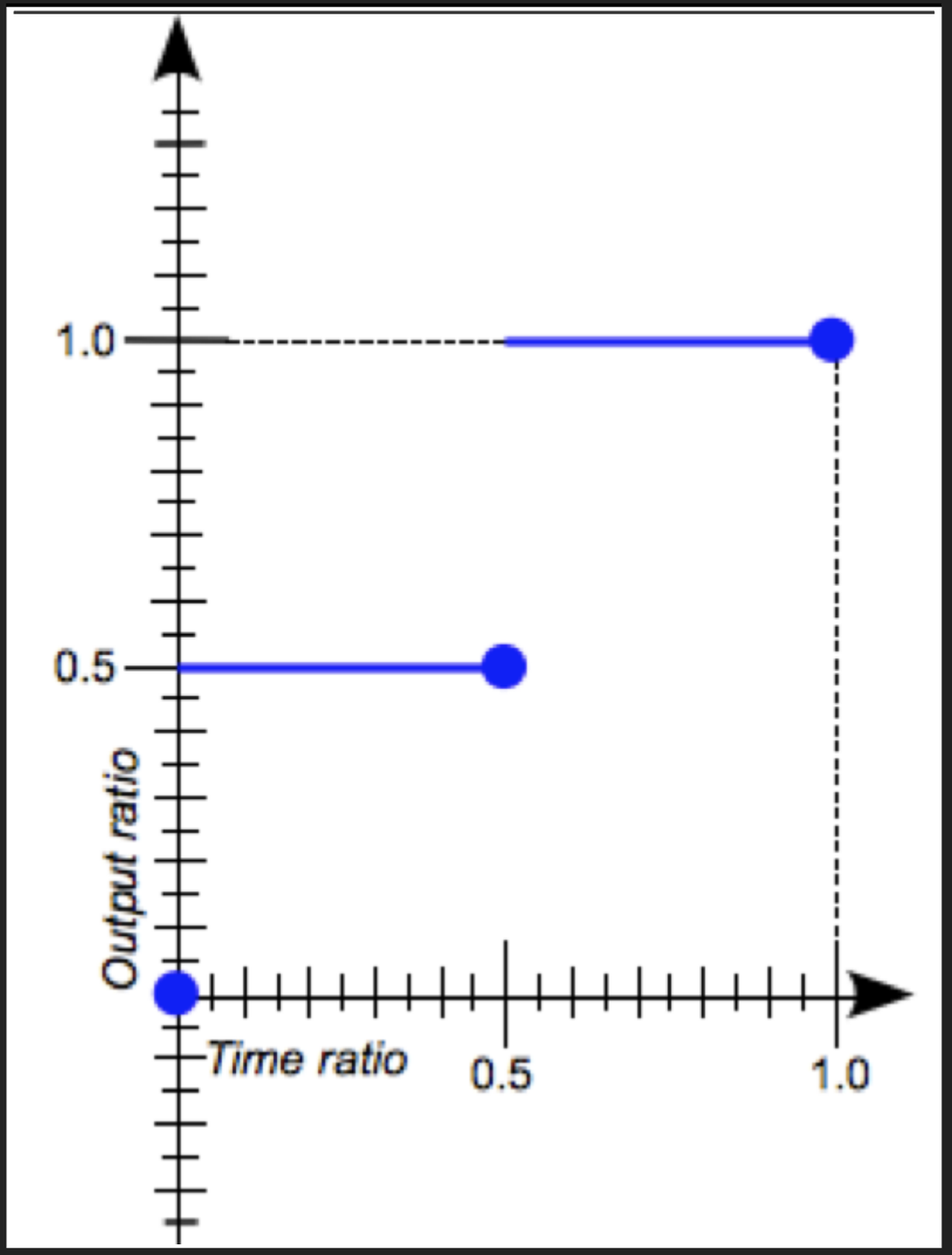
CSS3 TRANSITIONS – STEPS()

- steps(number_of_steps, direction) defines a step function with the direction indicating if it’s left/right-continuous.
- Direction is specified with start||end.

Reference MDN

CSS3 TRANSITIONS – STEPS()

```
steps(2, start);
```



Reference MDN

CSS3 TRANSFORMS

2D TRANSFORMS:

```
transform: skew(x, y | <angle>)
  || scale(x, y | <length> or <percentage>) || rotate(angle | <angle>)
  || translate(x, y | <length> or <percentage>)
```

CSS3 2D TRANSFORMS CODE SAMPLE


```
#twodtrans-1{
  transform: skew(10deg, 10deg);
}
#twodtrans-2{
  transform: scale(2, 1);
}
#twodtrans-3{
  transform: rotate(60deg);
}
#twodtrans-4{
  transform: translate(-10px, 50px);
}
```

CSS3 TRANSFORMS

3D TRANSFORMS:

```
transform: translate3d(x, y, z | <length> or <percentage>)
 /* Except z=<length> */
  || scale3d(x, y, z | <number>)
  || rotateX(<angle>)
  || rotateY(<angle>)
  || rotateZ(<angle>)
```

Perspective and perspective-origin set user’s perspective and vanishing point.

There are other properties like: transform-box, transform-origin, transform-style, backface-visibility.

CSS3 3D TRANSFORMS CODE SAMPLE



```
#threedtrans-1{
  transform: rotateX(180deg);
}
#threedtrans-2{
  transform: rotateY(180deg);
}
#threedtrans-3{
  transform: rotateZ(180deg);
}
/* perspective: 800px; perspective-origin: 50% 100px; */
```

CSS3 ANIMATIONS - @KEYFRAMES

- @keyframes define what an animation looks like at each stage.
- @keyframes are composed of three parts:
  - name - name of the animation
  - stage – a percentage used to define the properties at that stage. 0% (also can use from) and 100% (also can use to) have to be defined.
  - properties – CSS properties

CSS3 ANIMATIONS - @KEYFRAMES

```
@keyframes name {
  0% {/*can use from */
    top: 0;
  }
  50% {
    top: 30px;
  }
  100% {/* can use to*/
    top: 0;
  }
}
```

CSS3 ANIMATIONS - ANIMATION

CSS3 ANIMATIONS - ANIMATION

- Animation does two things:
  - Assign the @keyframes to the elements.
  - Define how it is animated.

```
#div {
  animation-name: keyframe-name;
  animation-duration: 1s;
  animation-timing-function: ease;
}
```

CSS3 ANIMATIONS - ANIMATION PROPERTIES

- Animation Properties:
  - animation-name: name (Default: none)
  - animation-duration: time of animation in seconds (Default: 0s)
  - animation-timing-function: timing function, remember cubic bezier curves (Default: ease)
  - animation-delay: delay to start of animation (Default: 0s)

CSS3 ANIMATIONS - ANIMATION PROPERTIES

- Animation Properties:
  - animation-iteration-count: times to run the animation (Default: 1)
  - animation-direction: direction to play the animation, normal || reverse || alternate || alternate-reverse (Default: normal)
  - animation-fill-mode: how should CSS apply styles, none || forwards || backwards || both (Default: none)
  - animation-play-state: animation running || paused (Default: running)

CSS3 ANIMATIONS - ANIMATION PROPERTIES

Animation Shorthand Syntax:

```
animation: [animation-name]
  [animation-duration]
  [animation-timing-function]
  [animation-delay]
  [animation-iteration-count];
```

CSS3 ANIMATION CODE SAMPLE


```
@keyframes fadeOut {
  0% {opacity: 1;}
  100% {opacity: 0;}
}
@keyframes pulse {
    0% { transform: scale(1); }
    50% { transform: scale(1.2); }
    100% { transform: scale(1); }
}
#css3animation{
  animation-name: pulse, fadeOut;
  animation-duration: 1s, 5s;
  animation-delay: 0s, 5s;
  animation-iteration-count: infinite, infinite;
  animation-play-state: paused;
}
```

BEAUTIFUL CSS ANIMATION EXAMPLES

- CSS 3D Solar System
- Pure CSS3 AT-AT Walker
- Stretchy Cat

BROWSER COMPATIBILITY

CSS3 Transitions - WD

Global 91%

unprefixed: 80.28%

Simple method of animating certain properties of an element, with ability to define property, duration, delay and timing function.

Current aligned | Usage relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8			45					4.3	
9			46					4.4	
10		43	47			8.4		4.4.4	
11	13	44	48	9	34	9.2	8	47	47
	14	45	49	9.1	35	9.3			
		46	50		36				
		47	51						

Notes | Known issues (5) | Resources (6) | Feedback

Support listed is for `transition` properties as well as the `transitionend` event. The prefixed name in WebKit browsers is `webkitTransitionEnd`

CSS3 2D Transforms - WD

Global 91.85%

unprefixed: 74.52%

Method of transforming an element including rotating, scaling, etc. Includes support for `transform` as well as `transform-origin` properties.

Current aligned | Usage relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8			45					4.3	
9			46					4.4	
10		43	47			8.4		4.4.4	
11	13	44	48	9	34	9.2	8	47	47
	14	45	49	9.1	35	9.3			
		46	50		36				
		47	51						

Notes | Known issues (4) | Resources (10) | Feedback

The scale transform can be emulated in IE < 9 using Microsoft's "zoom" extension, others are (not easily) possible using the MS Matrix filter

CSS3 3D Transforms - WD

Global 82.61% + 7.98% = 90.59%

unprefixed: 66.37% + 7.98% = 74.35%

Method of transforming an element in the third dimension using the `transform` property. Includes support for the `perspective` property to set the perspective in z-space and the `backface-visibility` property to toggle display of the reverse side of a 3D-transformed element.

Current aligned | Usage relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8			45					4.3	
9			46					4.4	
10 [1]		43	47			8.4		4.4.4	
11 [1]	13	44	48	9 [2]	34	9.2	8	47	47
	14	45	49	9.1 [2]	35	9.3			
		46	50		36				
		47	51						

Notes | Known issues (5) | Resources (10) | Feedback

[1] Partial support in IE refers to not supporting the transform-style: preserve-3d property. This prevents nesting 3D transformed elements.

[2] Safari 9 is reported to still require a prefix for the related `backface-visibility` property.

CSS Animation - WD

Global 90.94% + 0.03% = 90.97%

unprefixed: 73.5%

Complex method of animating certain properties of an element

Current aligned | Usage relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
8			45					4.3	
9			46					4.4	
10		43	47			8.4		4.4.4	
11	**13**	**44**	**48**	**9**	**34**	**9.2**	**8**	**47**	**47**
	14	45	49	9.1	35	9.3			
		46	50		36				
		47	51						

Notes | Known issues (6) | Resources (4) | Feedback

1. 'animation-fill-mode' property is not supported in Android browser below 2.3.
2. iOS 6.1 and below do not support animation on pseudo-elements. iOS 7 and higher are reported to have buggy behavior with animating pseudo-elements.
3. @keyframes not supported in an inline or scoped stylesheet in Firefox (bug 830056)
4. In Chrome `animation-fill-mode backwards` is wrong if `steps(x, start)` is used see example.
5. IE10 and IE11 do not support CSS animations inside media queries.
6. IE10 and IE11 on Windows 7 have a bug where translate transform values are always interpreted as pixels when used in animations test case

HANDLING BROWSER PREFIXES SANELY – AUTOPREFIXER

- PostCSS plugin to parse CSS and add vendor prefixes to CSS rules using values from Can I Use.
- Can add to Grunt and Gulp.

```
@keyframes name {
  0% {
    top: 0;
  }
  50% {
    top: 30px;
  }
  100% {
    top: 0;
  }
}
```

BECOMES

```
@-webkit-keyframes name {
  0% {
    top: 0;
  }
  50% {
    top: 30px;
  }
  100% {
    top: 0; }
  }
@keyframes name {
  0% {
    top: 0;
  }
  50% {
    top: 30px;
```

HANDLING BROWSER PREFIXES SANELY – BOURBON.IO

- Simple mixins library for Sass. Using @include to output prefixes.

```
box:hover {
  @include animation(scale 1.0s ease-in, slide 2.0s ease);
}
```

CSS3 ANIMATION LIBRARIES

animate.css is a bunch of cool, fun, and cross-browser animations for you to use in your projects. Great for emphasis, home pages, sliders, and general just-add-water-awesomeness.

Demo

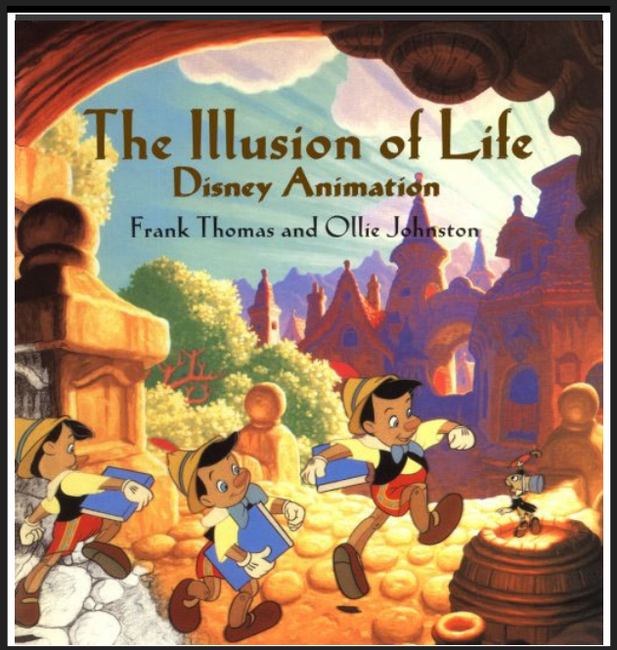
Magic.css is a bunch of CSS3 Animations with special effects.

Demo

THE GEOCITIES-IZER

PRINCIPLES FOR BEAUTIFUL ANIMATIONS

DISNEY ANIMATION: THE ILLUSION OF LIFE



Book by Ollie Johnston and Frank Thomas, two of the animation masters at Disney's during the Golden Age of animation, referred by Walt Disney as his "Nine Old Men".

PRINCIPLE 1: SQUASH AND STRETCH

Give a sense of weight and flexibility to drawn objects.

PRINCIPLE 2: ANTICIPATION

Prepare the audience for an action, and to make the action appear more realistic. Ex. A dancer jumping off the floor has to bend his knees first.

PRINCIPLE 3: STAGING

Direct the audience's attention and make it clear what is important.

PRINCIPLE 4: STRAIGHT AHEAD ACTION AND POSE TO POSE

"Straight ahead action" means drawing out a scene frame by frame from beginning to end, while "pose to pose" involves starting with drawing a few key frames, and then filling in the intervals later.

PRINCIPLE 5: FOLLOW THROUGH AND OVERLAPPING ACTION

Give the impression that characters follow the laws of physics, including the principle of inertia.

PRINCIPLE 6: SLOW IN AND SLOW OUT

Accelerate and Decelerate.

PRINCIPLE 7: FOLLOW ARCS

PRINCIPLE 8: SECONDARY ACTION

PRINCIPLE 9: TIMING (AKA FRAMES/SECOND)

PRINCIPLE 10: EXAGGERATION

PRINCIPLE 11: SOLID DRAWING

PRINCIPLE 12: APPEAL AKA CHARISMA

PURE CSS FAIL WHALE

WHAT DID WE LEARN?

- History of Animation on the Web
- CSS3 Animations
  - Transitions
  - Transforms
  - @keyframes & Animations
- Browser Compatibility
  - Autoprefixer
  - Bourbon.io
- CSS3 Animation Libraries
  - Animate.css

THANK YOU!

QUESTIONS?

ALSO, I WOULD LOVE TO SPEAK MORE IN EUROPE ;)

TW: @RAMISAYAR | GH: @SAYAR

SLIDESHARE.NET/RAMISAYAR

RESOURCES, REFERENCES, LINKS

- http://dailyjs.com/2010/06/27/167-history-of-javascript/
- http://www.nytimes.com/2013/02/14/fashion/common-on-early-i
  files-make-comeback.html?_r=1&
- https://hg.csswg.org/drafts/log/2579c1842a7c/css3-
  animations/Overview.src.html
- http://www.peterbe.com/plog/worst-flash-site-of-the-year-2010
- https://www.kirupa.com/html5/creating_simple_html5_canvas_a
- http://stackoverflow.com/questions/11182478/performance-css3
  vs-html5-canvas

RESOURCES, REFERENCES, LINKS

- http://css3.bradshawenterprises.com/transitions/#delays
- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions
- https://developer.mozilla.org/en-US/docs/Web/CSS/timing-function
- https://robots.thoughtbot.com/css-animation-for-beginners
- https://www.smashingmagazine.com/2011/09/the-guide-to-css-animation-principles-and-examples/
- https://ihatetomatoes.net/the-guide-to-css-animations-for-the-web/
- http://www.creativebloq.com/css3/animation-with-css3-712437