Much Element, Very Animate

Sam Thorogood ≈ Nov 2014

tl;dr: Element.animate deserves a place in your toolbox

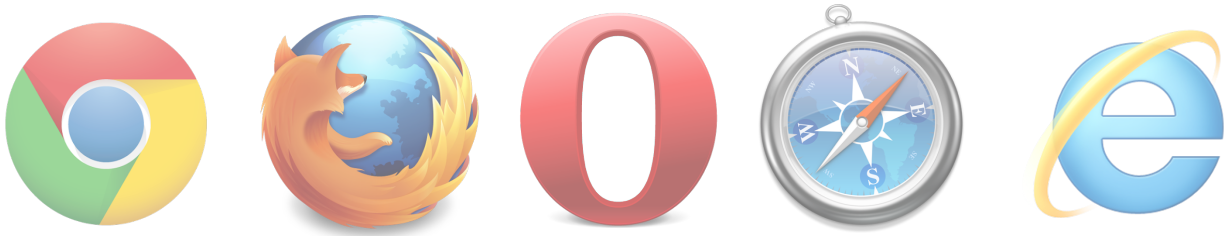
Along with CSS transitions, requestAnimationFrame, ...

```
var duration = 5000;
var player = cloud.animate([
  {transform: 'translate(' + start + 'px, 0px)'},
  {transform: 'translate(' + end + 'px, 0px)'}
], duration);

player.onfinish = function() {
  cloud.parentNode.removeChild(cloud);
};

// later
player.pause();

// scrub player
player.currentTime = 200;
```

Or for something completely different

Not your average animation!


```
var progress = ...;
var player = progress.animate([
  {width: '0', background: 'blue'},
  {width: '100%', background: 'red'}
], {duration: 100, fill: 'both'});

player.pause();

var onProgressUpdate = function(bytes, max) {
  player.currentTime = bytes / max * 100;

  if (bytes >= max) {
    var effect = [{opacity: 1}, {opacity: 0}];
    var timing = {duration: 500, fill: 'forwards'};
    player.animate(effect, timing);
  }
};
```



In the beginning there was window.setTimeout

And it was pretty rubbish

Manual callbacks

"Best"


```
var start = +new Date();
var end = start + 5000;
var duration = end - start;

(function updateAnimation() {
  var now = +new Date();
  var ratio = (now - start) / duration;

  elem.style.transform = 'translate(' + ratio * 100 + 'px, 0px)';

  if (now < end) {
    window.setTimeout(updateAnimation, 1000 / 60);
  }
}());
```

Manual callbacks #2

Worse...

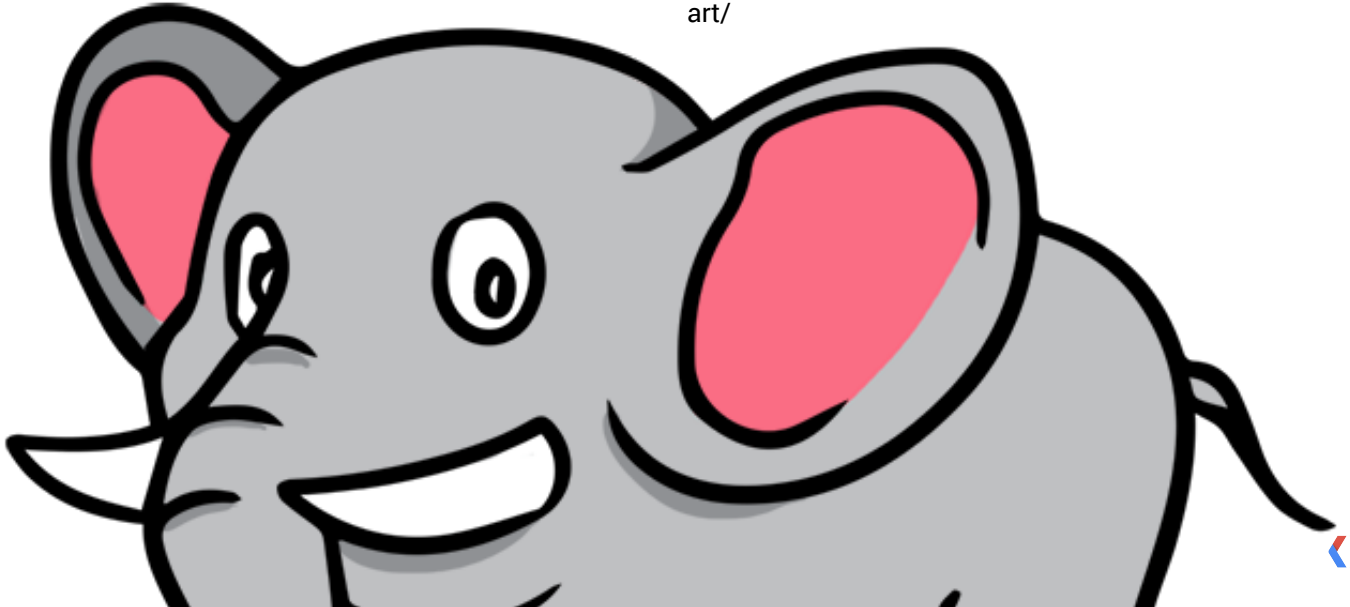
```
var start = +new Date();
var end = start + 5000;
var duration = end - start;

var interval = window.setInterval(function() {
  var now = +new Date();
  var ratio = (now - start) / duration;

  elem.style.transform = 'translate(' + ratio * 100 + 'px, 0px)';

  if (now >= end) {
    window.clearInterval(interval);
  }
}, 1000 / 60);
```

Some culprits

Okay, just one

But there is a polyfill.

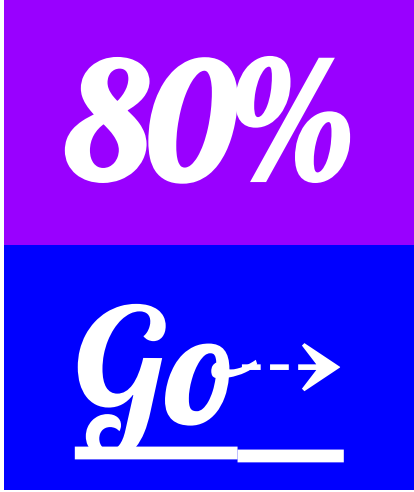
image: http://www.clipartlord.com/category/animals-clip-art/elephant-clip-art/



window.requestAnimationFrame (spec 2011, enabled 2013+?)

Using requestAnimationFrame

Basically like using setTimeout!




```
var start = +new Date();
var end = start + 5000;
var duration = end - start;

(function updateAnimation() {
  var now = +new Date();
  var ratio = (now - start) / duration;

  elem.style.transform = 'translate(' + ratio * 100 + 'px, 0px)';

  if (now < end) {
    window.requestAnimationFrame(updateAnimation);
  }
}());
```

Great!

But...

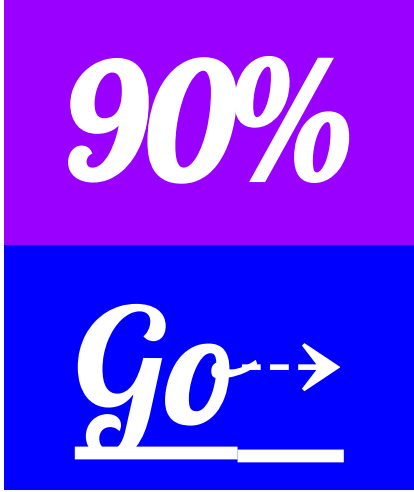
- ★ Runs neatly on frame boundaries
- ★ Only when the window is visible
- ★ But lots of imperative overhead
- ★ Still uses the main thread: target of 16ms!

CSS Transitions

Declarative, known and loved

Pretty standard use cases

```
.menu {
  transition: all 0.25s;
  background: white;
}
.menu:hover {
  background: gray;
}

@keyframe moveanim {
  from: {transform: translate(0, 0) rotate(0);}
  to: {transform: translate(400px, 0) rotate(10deg);}
}
.dancing {
  animation: moveanim 5s linear infinite;
}
```



More awkward for inlining

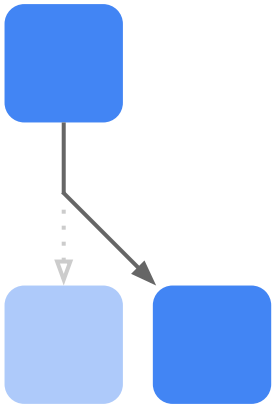
Hacks to set initial state

```
snowFlake.style.transform = 'translate(' + snowLeft + 'px, -100%)';

// wait for snowFlake to settle
window.setTimeout(function() {
  snowFlake.style.transitionProperty = 'transform';
  snowFlake.style.transitionDuration = '1500ms';
  snowFlake.style.transform = 'translate(' + snowLeft + 'px, ' +
      window.innerHeight + 'px)';
}, 0);
```

Animation drift

Sometimes, you don't really know the destination. Or where you started.

```
var s = elem.style;

s.transitionProperty = 'transform';
s.transitionDuration = '1500ms';
s.transform = 'translate(0, 100px)';

window.setTimeout(function() {
  s.transform = 'translate(50px, 100px)';
} 750);
```



Web Animations

Web Animations aren't trying to replace CSS Transitions

Shared animation engine

At least in Chrome.

@keyframes	element.animate(**effect**, timing)
animation-delay	timing.delay
animation-direction	timing.playbackRate*
animation-duration	timing.duration
animation-fill-mode	timing.fill
animation-iteration-count	timing.iterations
animation-timing-function	timing.easing

Clouds

Let's revisit the toolbox!


```
var player = cloud.animate([
  {transform: 'translate(' + start + 'px, ' + y + 'px)'},
  {transform: 'translate(' + end + 'px, ' + y + 'px)'}
], {
  duration: 10000,
  delay: i * 200,
  easing: 'ease-in-out',
  fill: 'both'
});
```

Let's use AnimationPlayer for fine motor control

Scrubbable Clouds


```
var player = cloud.animate({ … });
players.push(player);
// ...
var startEvent, startTime;
document.addEventListener('mousedown', function(event) {
  startEvent = event;
  startTime = players[0].currentTime;
  players.forEach(function(p) { p.pause(); });
});
document.addEventListener('mousemove', function(event) {
  if (!startEvent) return;
  var delta = startEvent.offsetX - event.offsetX;
  var updateTime = startTime + delta;
  players.forEach(function(p) { p.currentTime = updateTime; });
});
```

UI snapping

Impress your friends!


```
var targets = [0, 500, 1000];

document.addEventListener('mouseup', function(event) {
  var startTime = player.currentTime;
  var targetTime = findNearest(targets, player.currentTime);
  var duration = Math.abs(targetTime - startTime);
  var startAt = +new Date();
  (function drift() {
    var ratio = (+new Date() - startAt) / duration;
    ratio = Math.min(1.0, ratio);
    player.currentTime = ratio * (startTime + targetTime);
    if (ratio < 1.0) requestAnimationFrame(drift);
  }());
});
```

Now & Future

Phase 1	Spec	Future
Element.animate	Constructors	Compositive, additive effects
AnimationPlayer	Animation{Group,Sequence}	???
getCurrentPlayers	Custom Effects	Profit!

Chrome 39 + polyfill

web-animations-next polyfill*

Future spec syntax

```
// Constructor
var anim = new Animation(target, [{ color: 'blue' }, { color: 'red' }]);
document.timeline.play(anim);

// Groups
var groupAnimation = new AnimationGroup([anim, anim2]);

// Sequences
var sequence = new AnimationSequence([
  new Animation(firstTarget, [ … ]),
  groupAnimation
]);
```

Addendum

Things to think about

- **Avoid animating left, top, width, height etc**

Forces browser to perform layout. Transform is defined not to affect other elements.

- **Hardware-accelerated surfaces**

~~transform: translateZ(0);~~

will-change: transform;

thank you

Appendix

google.com/+SamThorogood

http://web-animations.github.io/web-animations-demos

https://github.com/web-animations/web-animations-js

http://www.w3.org/TR/web-animations (July) & http://w3c.github.io/web-animations (Nov)