

0-60: Animating the Web

@jeremyckahn

Hi! Who am I?

Jeremy Kahn

Web Developer at
YouTube

I spend a lot of time
writing open source
JavaScript libraries!

* Opinions and views are my own



github.com/jeremyckahn

twitter.com/jeremyckahn

Animation is communication

History



JS

CSS

CSS 3

Frame Rate

60

Tweening

Keyframing

Tweens

vs

Keyframes

Let's see a
demo!


```
// jQuery-style tween animation
```

```
$e11
```

```
  .css({ left: -700 })
```

```
  .animate({ left: 0 }, 2500);
```

```
$e12
```

```
  .css({ left: -700 })
```

```
  .delay(750)
```

```
  .animate({ left: 0 }, 1750);
```

```
$e13
```

```
  .css({ left: -700 })
```

```
  .delay(1500)
```

```
  .animate({ left: 0 }, 1000);
```

```
// Keyframe animation
```

```
e11
```

```
.keyframe(0, { left: -700 })  
.keyframe(2500, { left: 0 });
```

```
e12
```

```
.keyframe(750, { left: -700 })  
.keyframe(2500, { left: 0 });
```

```
e13
```

```
.keyframe(1500, { left: -700 })  
.keyframe(2500, { left: 0 });
```

JavaScript

No run loop!

Only **one** thread

setInterval

Do **not** use
setInterval for
animations.

callback

callback

callback

callback

callback

callback

callback

callback

callback

callback

callback

callback

callback

callback

callback

callback

setTimeout

```
var el =  
    document.getElementById('my-el');  
el.style.position = 'absolute';  
var left = 0;  
  
function update () {  
    el.style.left = left + 'px';  
    left++;  
    setTimeout(update, 1000 / 60);  
}  
  
update();
```

requestAnimationFrame

```
var el =  
    document.getElementById( 'my-el' );  
el.style.position = 'absolute';  
var left = 0;  
  
function update () {  
    el.style.left = left + 'px';  
    left++;  
    requestAnimationFrame( update );  
}  
  
update();
```

Architecture

```
var el =  
    document.getElementById( 'my-el' );  
el.style.position = 'absolute';  
var left = 0;  
  
function update () {  
    el.style.left = left + 'px';  
    left++;  
    requestAnimationFrame( update );  
}  
  
update();
```


dt

State logic and
drawing logic must
not be tightly
coupled!

```
function update () {  
    draw(calculateState());  
    requestAnimationFrame(update);  
}
```

```
function calculateState () {  
    // Computes state based on dt  
}
```

```
function draw (state) {  
    // Renders state to the screen  
}
```

```
update();
```

```
function update () { // Controller
    draw(calculateState());
    requestAnimationFrame(update);
}
```

```
function calculateState () { // Model
    // Computes state based on dt
}
```

```
function draw (state) { // View
    // Renders state to the screen
}
```

```
update();
```

Model

View

Controller

Model - Maintains state

View - Renders Model visually

Controller - Controls Model

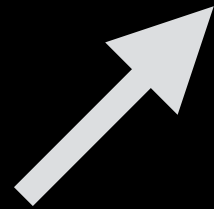
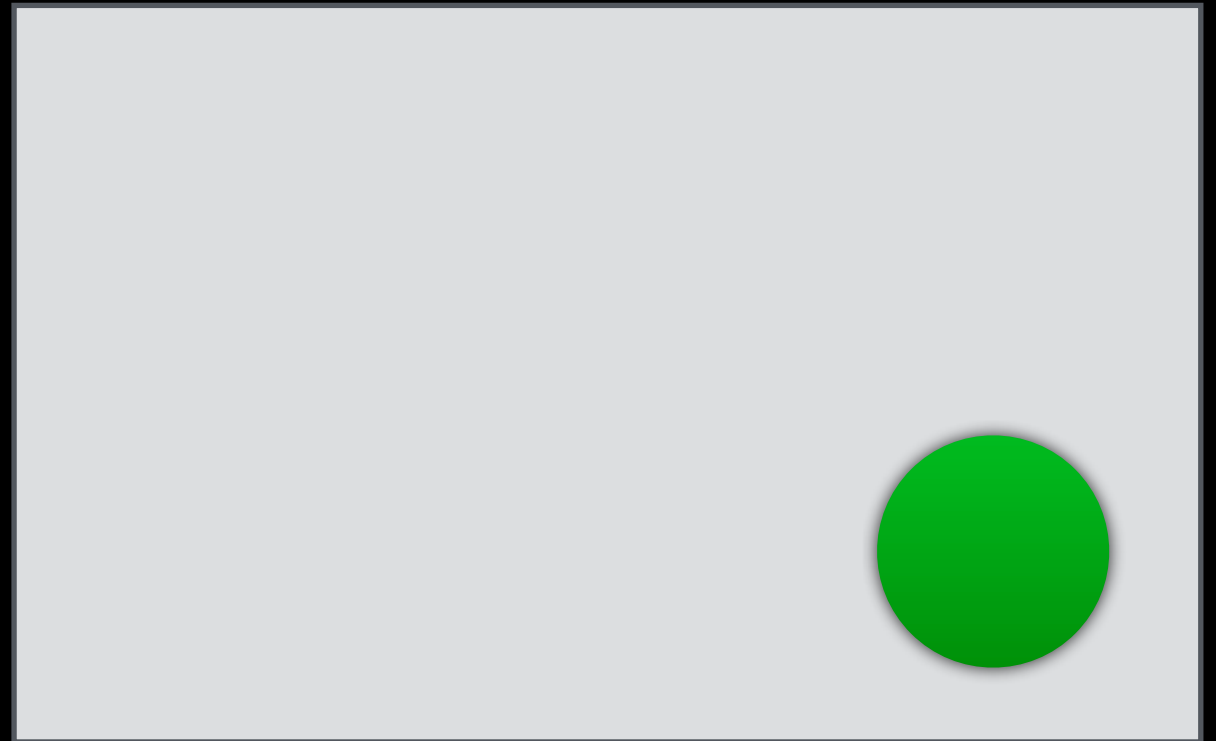
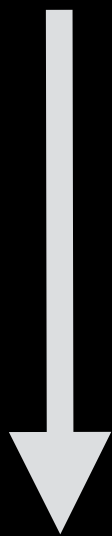
```
// Model  
{ x: 10, y: 10 }
```



```
// View  
function drawCircle (params) {  
  // magic...  
}
```

```
// Model
```

```
{ x: 300, y: 150 }
```



```
// View
```

```
function drawCircle (params) {  
  // magic...  
}
```


CSS

transitions

```
h2 {  
  font-size: 2.5em;  
  text-align: center;  
  -webkit-transition-property: all;  
  -webkit-transition-duration: 1000ms;  
  -webkit-transform: scale(.65);  
}  
  
h2:hover {  
  -webkit-transform: scale(1);  
}
```

@keyframes

```
h2 {  
  font-size: 6em;  
  text-align: center;  
  -webkit-transform: scale(.2);  
  -webkit-animation-duration: 2000ms;  
  -webkit-animation-fill-mode: both;  
  -webkit-animation-name: thirsty;  
}
```

```
@-webkit-keyframes thirsty {  
  0% {  
    -webkit-transform: scale(.2);  
  }  
  40% {  
    -webkit-transform: scale(.4);  
  }  
  65% {  
    -webkit-transform: scale(.4);  
  }  
  100% {  
    -webkit-transform: scale(.4, 1);  
  }  
}
```

Easing


```
transition-timing-function:  
  cubic-bezier(0.250, 0.250, 0.750, 0.750);
```

Combine easing formulas for curved motion

JavaScript

VS

CSS

JavaScript

Pros: Flexible, works everywhere.

Cons: Everything lives in the JavaScript thread.

CSS

Pros: Better performance.

Cons: Limited easing formula choices, not supported everywhere.

Performance

DOM

Minimize reflows

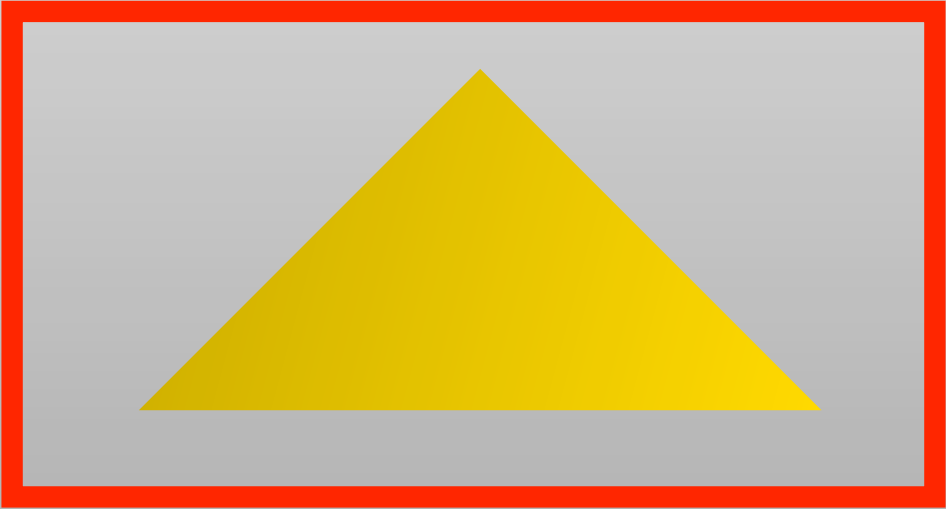
<canvas>

Pre-rendering

Multiple canvases







Quality

Sub-pixel rendering

Sub-pixel rendering

Scaling

Tools

CSS:

Ceaser

SASS mixins

JavaScript / Tweening:

jQuery

Zepto

Shifty

Keyframing:

Rekapi

Rekapi

A Model and Controller for
keyframe animations

Generates CSS `@keyframes`

Graphical tools

Sencha Animator

Adobe Edge

Styleie

The future...

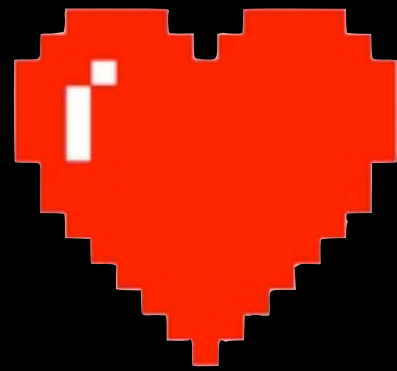
JavaScript

VS

CSS

JavaScript *and* CSS

JavaScript



CSS

Thanks!

Questions?

Resources

<http://ejohn.org/blog/how-javascript-timers-work/>

<http://matthewlein.com/ceaser/>

<http://creativejs.com/resources/requestanimationframe/>

<http://rekapi.com/>

<https://github.com/jeremyckahn/styleie>

<https://gist.github.com/1002116>