

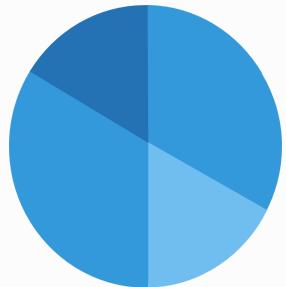
# ***ANIMATE THE WEB WITH EMBERJS***

Jessica Jordan



jjordan\_dev

***WHAT I DO FOR WORK***



simplabs

# **WHAT I DO AFTER WORK**





***ANIMATIONS***

# ***ANIMATIONS MAY REFER TO...***

Interactions on UI Elements

:hover

# ***ANIMATIONS MAY REFER TO...***

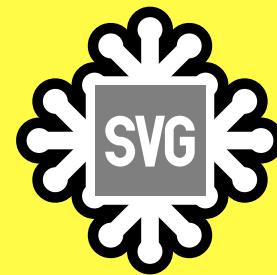
Page Transitions



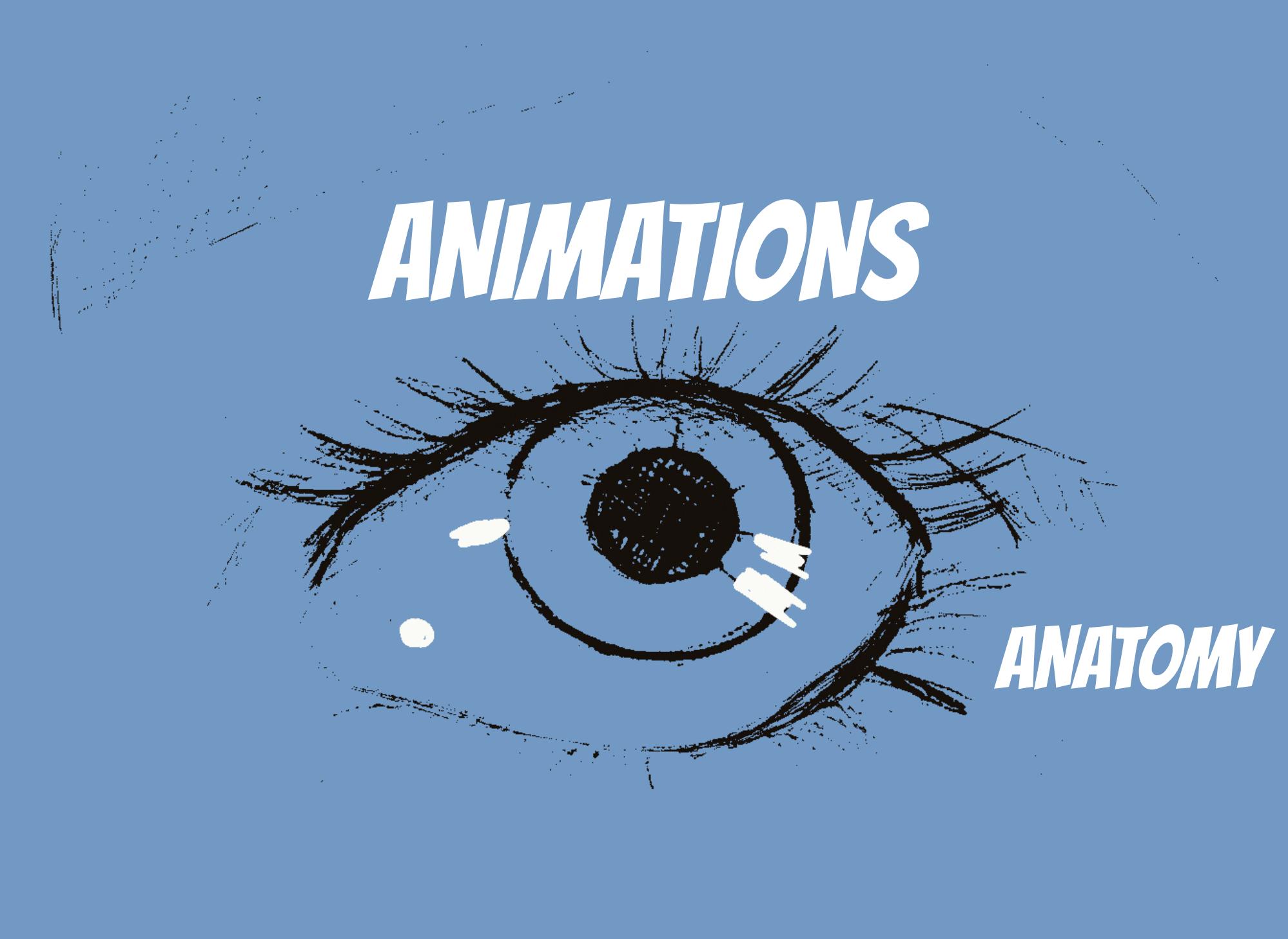
with e.g Ember Liquid Fire

***ANIMATIONS MAY REFER TO...***

Data Visualization



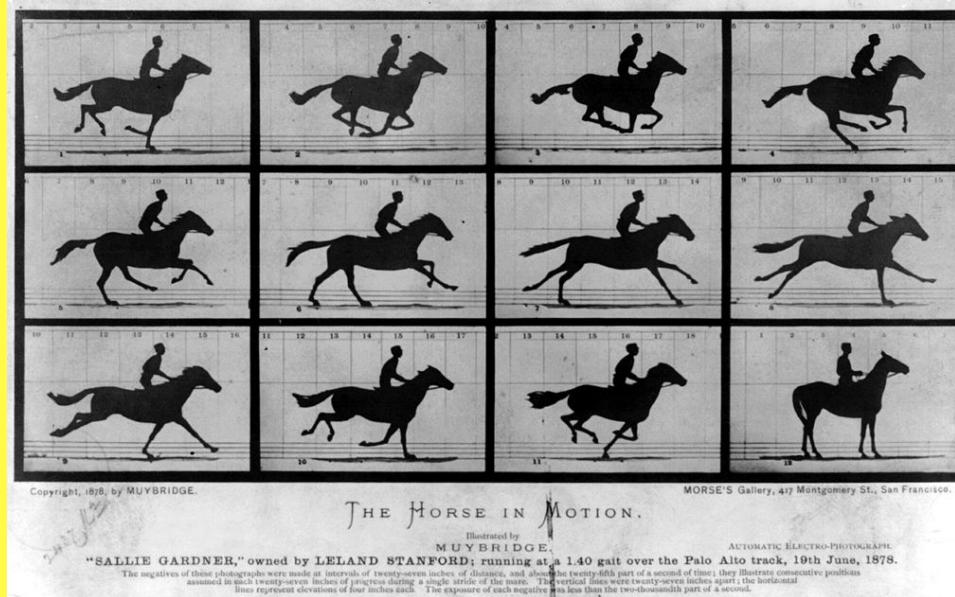
# ***STORYTELLING THROUGH ANIMATIONS***



**ANIMATIONS**

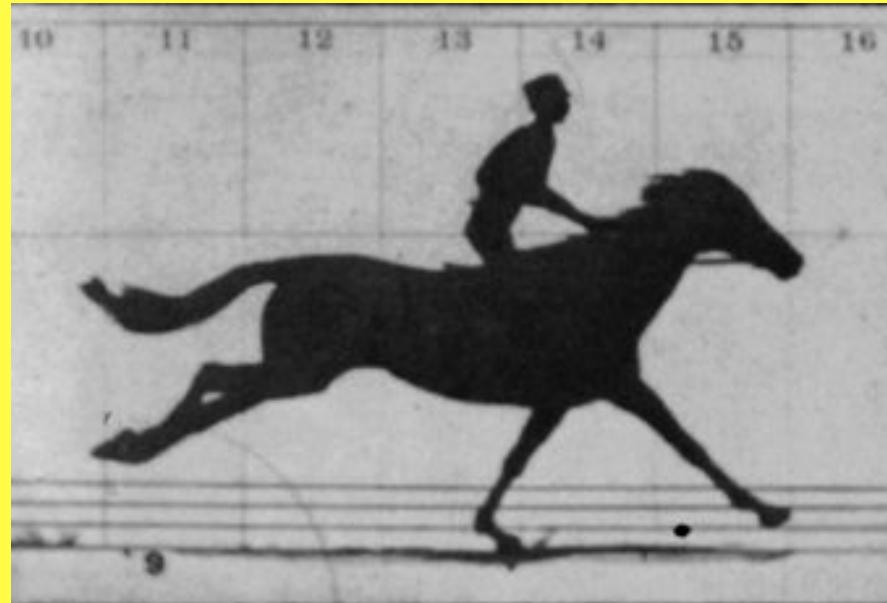
**ANATOMY**

# FRAME-BY-FRAME ANIMATIONS



Muybridge - The Horse In Motion, 1878

# **FRAME-BY-FRAME ANIMATIONS**



Muybridge - The Horse In Motion, 1878

# **EARLY BEGINNINGS OF ANIMATIONS**



19th century animation devices: Phenakistoscope

***...LEADING TO FIRST  
CINEMATOGRAPHY***



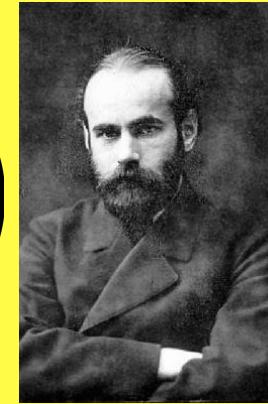
Roundhay Garden Scene 1888 - Louis Le Prince

# ***ANIMATIONS - A DEFINITION***

Image Sequence + Display Frequency = Illusion of motion

# **SMOOTH ANIMATIONS POWERED BY YOUR BRAIN**

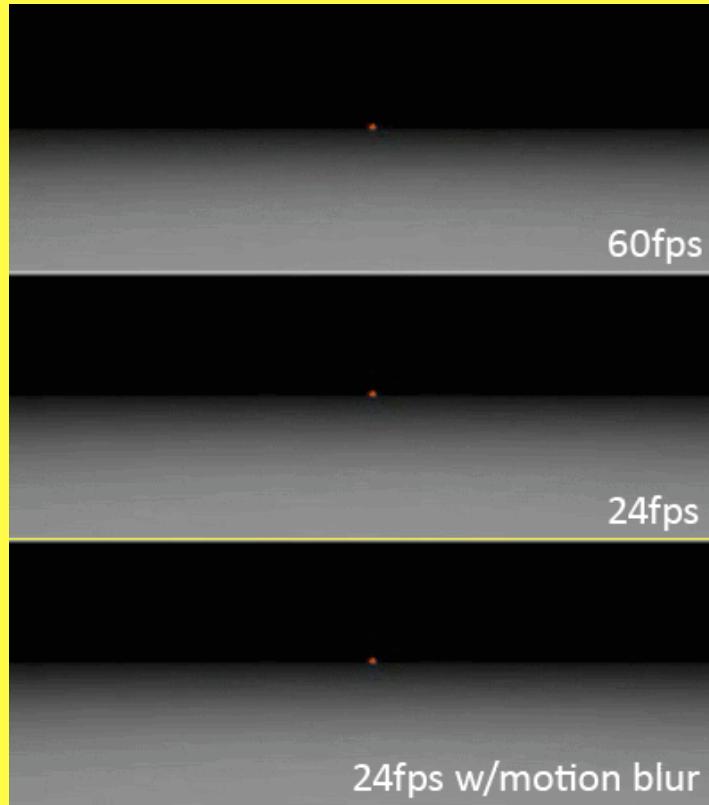
Φ



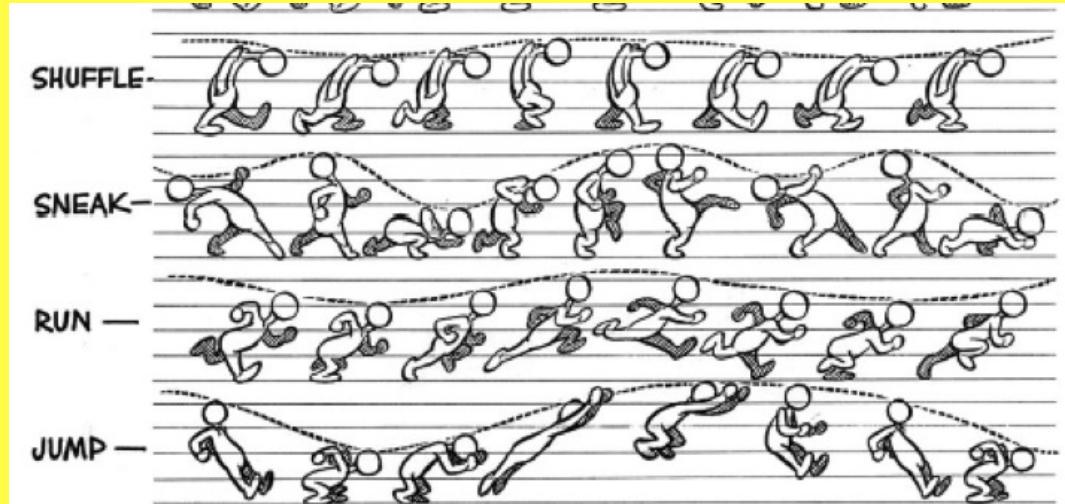
## Phi Phenomenon

Gestalt Psychology by Max Wertheimer (1912)

# **EFFECTS OF FRAME RATES**



# **FRAME-BY-FRAME ANIMATIONS IN CARTOONS**



Alla Gadassik - "The Animated Line: Performing and Generating Movement in Early Animation"

***THE STORY OF ANIMATIONS CONTINUES  
UNTIL TODAY***



***ANIMATIONS ON THE WEB***



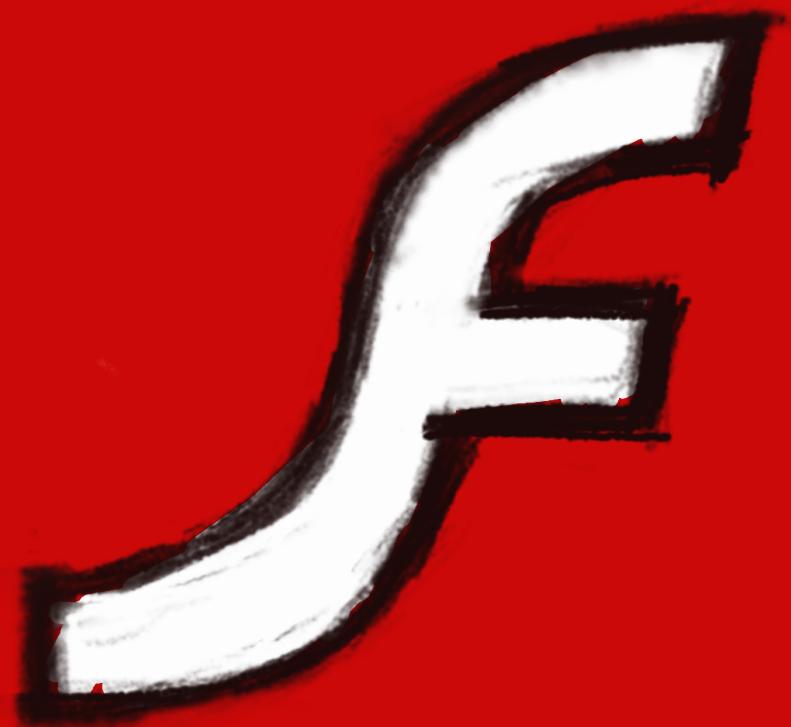
# ***ANIMATIONS***

# ***TOOLS AND WEB APIs***

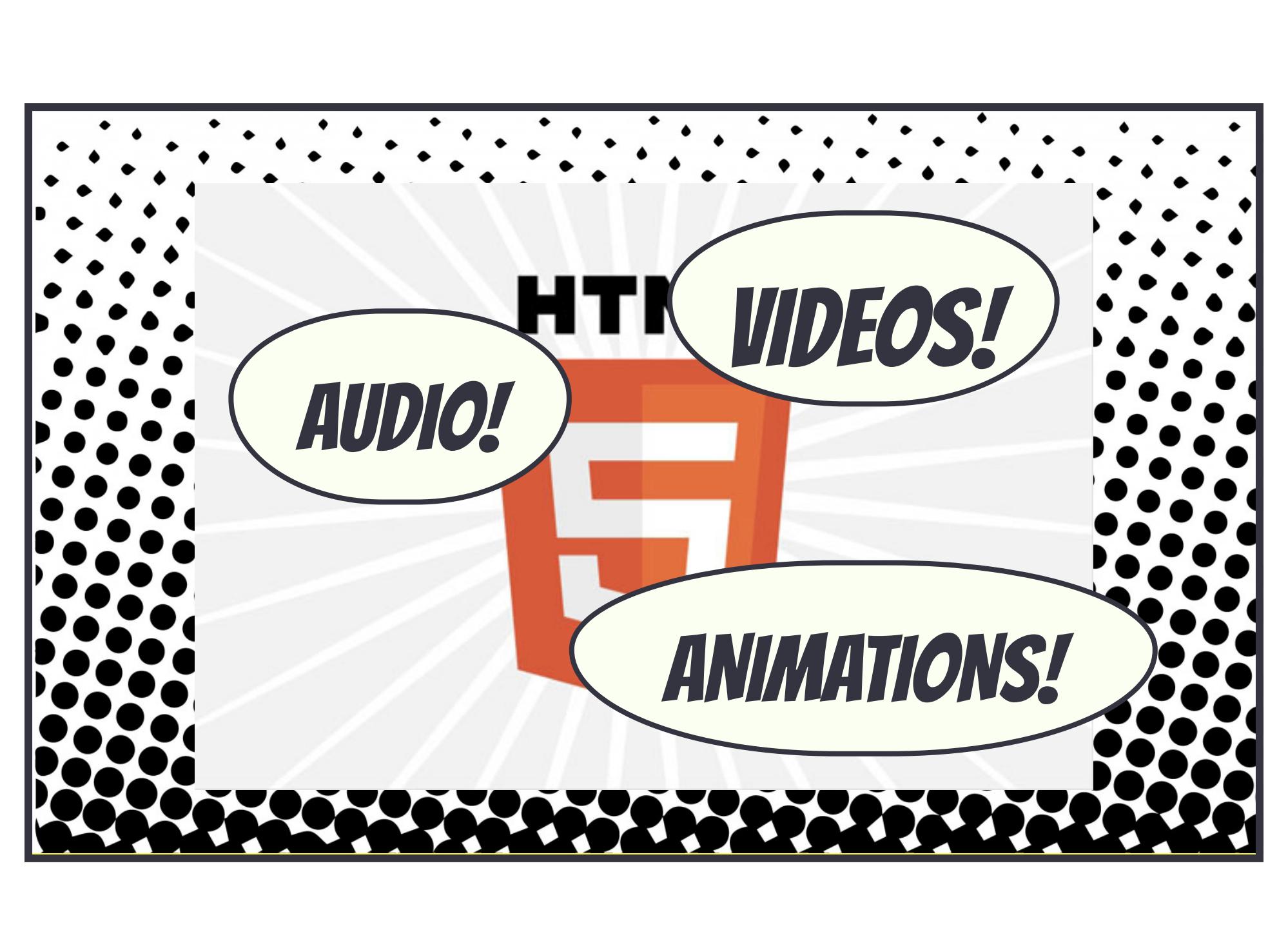
# ***WEB ANIMATIONS 20 YEARS AGO?***

**(modem dials)**

**FLASH**



**SINCE 1996**



**AUDIO!**

**VIDEOS!**

**ANIMATIONS!**

# ***"HTML5" ANIMATIONS***

**HTML**



**CSS**



**JS**



...and SVG, WebGL, XML and other Web APIs

# **WHY OPEN WEB STANDARDS?**

open  
consistent  
available



What about the creator experience?

# **THE INFINITE CANVAS**

Rachel Nabors, Keynote "Storytelling on the Shoulders of Giants" - OSCON 2014

# ***ANIMATIONS DEMO***

***EMBER & HTML5 CANVAS***

# ***HTML5 CANVAS***

**<canvas></canvas>**

Powerful Web API which can be leveraged for  
animations

# *CANVAS CONTEXT OBJECT*

```
<!-- index.html -->
<canvas id="comic-panel"></canvas>

// drawing.js
const canvas = document.getElementById('comic-panel');

if (canvas.getContext) {
  var ctx = canvas.getContext('2d');

} else {
  throw `Everything's fresh, but there's
  no context object here,
  so time for some polyfilling:
  bower install --save canvas-5-polyfill`;
}
```

# **HTML5 CANVAS & EMBER COMPONENTS**

Let's build it!



# ***CREATING THE COMPONENT***

```
{ {comic-panel width=width height=height} }

// app/components/comic-panel.js
import Ember from 'ember';

export default Ember.Component.extend({
  tagName: 'canvas'
})
```

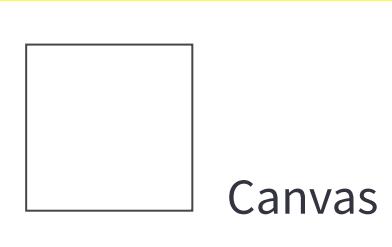
# **DRAWING TO THE CANVAS**

sx, sy: origin of source image

sWidth, sHeight: width and height of source image

dx, dy: origin of canvas destination

dWidth, dHeight: width and height of canvas destination



Canvas

# **DRAWING TO THE CANVAS**

```
// app/components/comic-panel.js
export default Ember.Component.extend({
  Ember.on('init', function() {
    const image = new Image();
    image.onload = () => {
      this.set('naturalHeight', image.height);
      this.set('naturalWidth', image.width);
      this.set('pseudoImg', image);
    };
    image.src = this.get('imgSrc');
  })
})
```

# **DRAWING TO THE CANVAS**

```
// app/components/comic-panel.js
export default Ember.Component.extend({
  draw() {
    // ...
    ctx.clearRect(0, 0, canvasWidth, canvasHeight);

    ctx.drawImage(img, sx, sy, sWidth, sHeight,
                  0, 0, dWidth, canvasHeight);
  }
})

/* app/components/comic-panel.js */
setup: Ember.observer('pseudoImg', function() {
  const ctx = this.get('element').getContext('2d');
  this.set('ctx', ctx);
  this.draw();
})
```

# ***GETTING ACTIONS HOOKED IN***



**FRAME 0**

**FRAME +**

---

**PLAY**

# **RUN ANIMATIONS**

```
// app/components/comic-panel.js
loop() {
  this.draw();
  this.nextFrame();
  Ember.run.later(this, this.loop, 100);
}
```

# **RUN AND CANCEL ANIMATIONS**

```
const loop = Ember.run.later(this, this.loop, 100);  
this.set('currentLoop', loop);
```

```
if (!this.get('runAnimation')) {  
  Ember.run.cancel(currentLoop);  
  this.set('currentLoop', null);  
})
```

# **DEMO FINAL**



**FRAME 0**

**FRAME +**

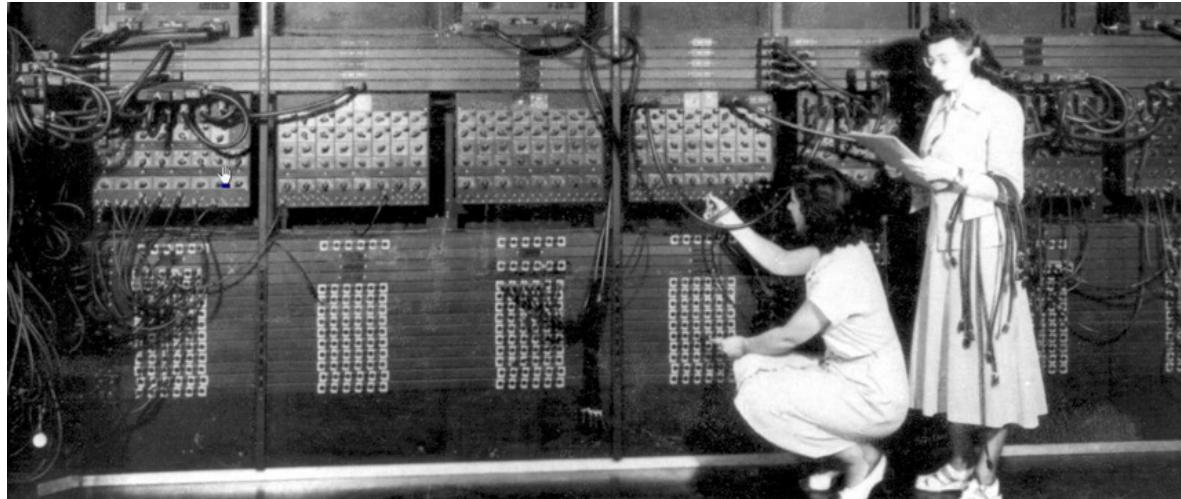
---

**PLAY**

# ***HTML5 CANVAS - GOOD TO KNOW***

- ♥ Performance
- ♥ Developer Control
- ? Creator Experience
- ? Accessibility

# ***HTML5 REQUIRES LOTS OF WIRING UP...***



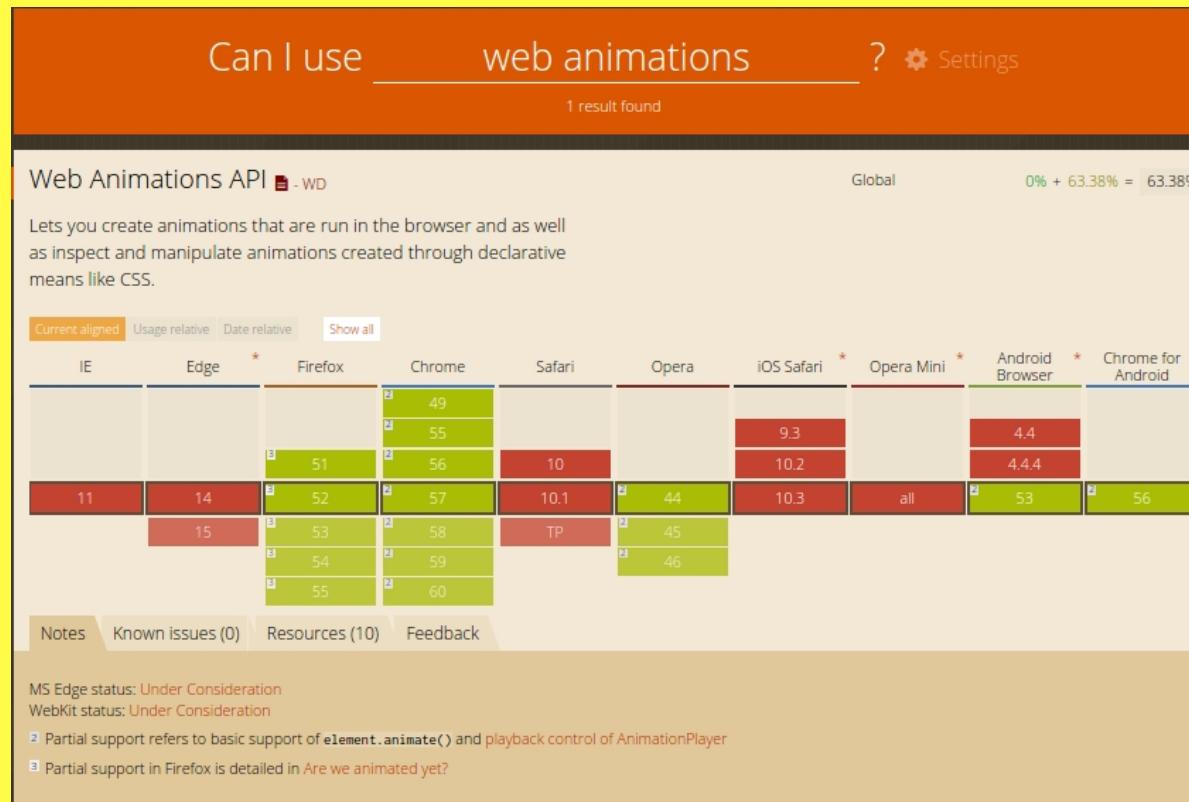
# ***ANIMATIONS DEMO***



***EMBER & WEB ANIMATIONS API***

# ***WEB ANIMATIONS API - ANIMATION OF THE FUTURE***

# ...BUT THE BROWSER SUPPORT?



That's a "No", right?



# ***WEB ANIMATIONS JS POLYFILL***

Build WAAPI powered animations today!

# **BUT OF COURSE...**

...there's an addon for that!



```
ember install ember-web-animations-next-polyfill
```

# *CSS3 ANIMATIONS GO JAVASCRIPT*

```
@keyframes rotating {  
  0% {  
    transform: rotate(0) translate3D(-50%, -50%, 0);  
    color: #000;  
  },  
  33% {  
    color: #431236;  
    offset: 0.333;  
  },  
  66% {  
    transform: rotate(360deg) translate3D(-50%, -50%, 0);  
    color: #000:  
  }  
}
```

CSS3

# **KEYFRAME EFFECTS**

```
// KeyFrameEffect Objects:  
var rotating = [  
  { transform: `rotate(0) translate3D(-50%, -50%, 0)`,  
    color: '#000'  
  },  
  { color: '#431236',  
    offset: 0.333  
  },  
  { transform: 'rotate(360deg) translate3D(-50%, -50%, 0)',  
    color: '#000'  
  }  
];
```

JS

# **DEMO: ANIMATION COMPONENT IN EMBER BASED ON WAAPI**



{ { comic-panel } }

# **CREATING KEYFRAME EFFECTS**

# ***CREATING KEYFRAME EFFECTS***

```
// app/my-route/controller.js
let characterObject = {
    ....,
    keyFrames: [
        { backgroundPosition: '0 0' },
        { backgroundPosition: `0 100%` }
    ],
    ....
};
```

# ***CREATING KEYFRAME EFFECTS***

```
let characterObject = {  
    .......,  
    animationOptions: {  
        duration: 500,  
        easing: `steps(5)`,  
        iterations: 'Infinity'  
    }  
    ......  
};  
  
{ {comic-panel comicLayer=characterObject} }
```

# ***CREATING AND STARTING THE ANIMATION***

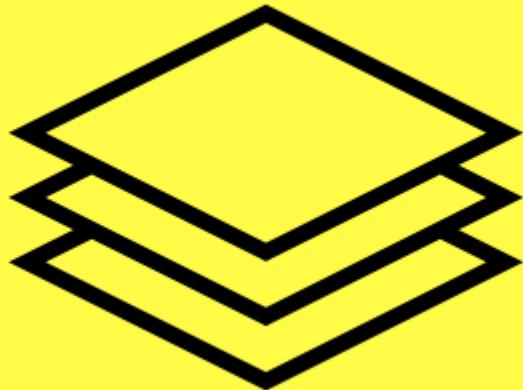
```
/* app/components/comic-panel/component.js */
export default Ember.Component.extend({
  ...
  startAnimation: Ember.on('didRender', function() {
    let keyFrames = this.get('keyFrames');
    let animationOptions = this.get('animationOptions');
    this.$()[0].animate(keyFrames, animationOptions);
  }),
  ...
})
```

# ***LET'S MAKE IT REAL***



**PLAY**

# ***MULTI-LAYERED ANIMATIONS***



# **EMBEDDING COMIC LAYER SUB COMPONENTS**

```
/* app/components/comic-panel/template.hbs */
{{#each comicLayerList as |layer|}}
  {{comic-layer keyFrames=layer.keyFrames
                animationOptions=layer.animationOptions
                frameAction=(action "setKeyFrames"))}}
{{/each}}

// app/components/comic-panel-layer.js
createKeyFrames: Ember.on('didRender', function() {
  // ...
  const layer = this.get('element');
  const keyFrame = new KeyframeEffect(
    layer,
    this.get('keyFrames'),
    this.get('animationOptions')
  );
  this.sendAction('frameAction', keyFrame);
})
```

# ***SYNCHRONIZE THE ANIMATION LAYERS***

```
/* app/components/comic-panel/component.js */
setupAnimation() {
  const timeline = this.get('timeline');
  const keyFrameEffects = this.get('keyFrameEffects');

  const group = new GroupEffect(keyFrameEffects);

  const animation = new Animation(group, timeline);

  animation.pause();

  this.set('animation');
},
```

# ***SYNCHRONIZE THE ANIMATION LAYERS***

```
/* app/components/comic-panel/component.js */
export default Ember.Component.extend({
  // ...
  actions: {
    play() {
      this.get('animation').play();
    }
    pause() {
      this.get('animation').pause();
    }
  }
  // ...
})
```

# **DEMO FINAL**



**PLAY**

# ***WEB ANIMATIONS API - GOOD TO KNOW***

- ♥ Creator Experience
- ♥ Developer Control
- ♥ Accessibility
- ? Performance





# ***ANIMATIONS***

# ***EPILOGUE***

**THE STORY ABOUT ANIMATIONS CONTINUES  
UNTIL TODAY**

**WITH ❤ EMBER & OPEN WEB STANDARDS**

***THE WEB IS YOURS TO BE ANIMATED***

# **RESOURCES**

Tomster and Zoey Illustration by Lindsey Wilson

"Inifinite Canvas" quote from Rachel Nabor's  
Keynote @ OSCON 2014: "Storytelling On The  
Shoulders of Giants"

Learn more about the Web Animations API with  
the excellent How To Guide at MDN

Full source code for the animation demos



**THANK YOU!**

Jessica Jordan



jjordan\_dev