Next time

Keyframe animations
SVG animations

Web Animations with CSS Part 2

Keyframe Animations

- A list of what should happen over the course of the animation - which properties should change now and when.


w/
Transition
END


w/
keframe
0%
80%
90%
100%

Creating Keyframe Animations

- There are two parts to creating keyframe animations:

    1) Create the animation.

    2) Assign the animation.

Step 1: Keyframe Recipe

Declare your animation:

```
@keyframes <name-animation> {
  <step 1> {<property>: <value>;}
  <step 1> {<property>: <value>;}
}
```

Step 1: Keyframe Recipe

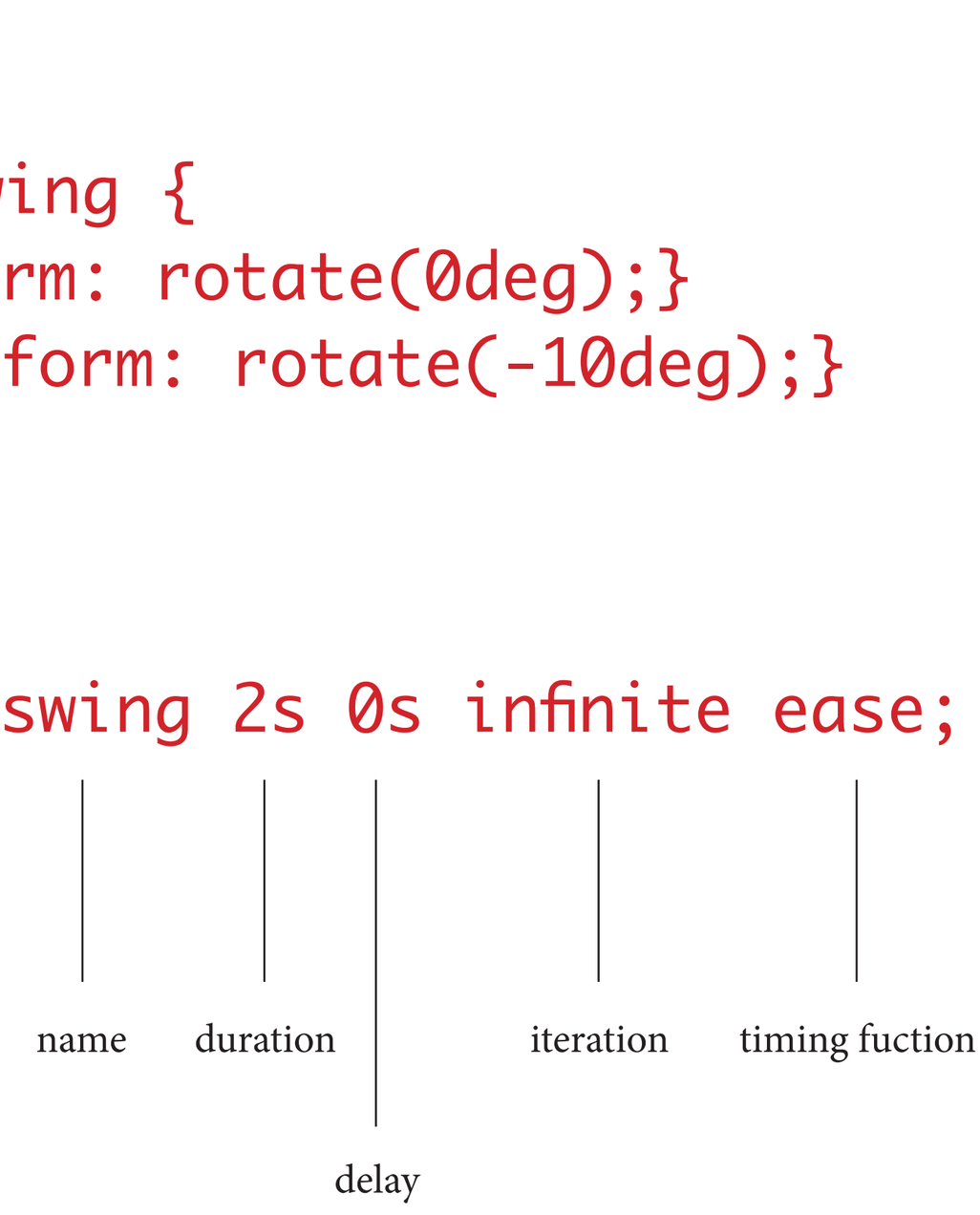
Example:

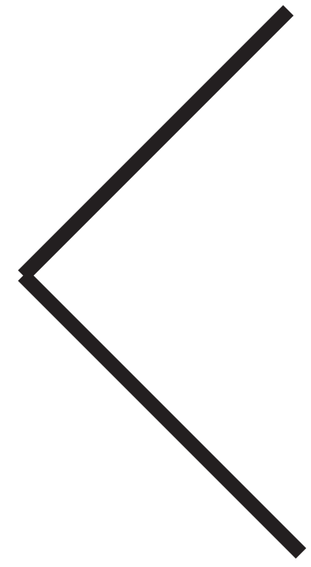
```
@keyframes swing {
  0% {transform: rotate(0deg);}
  100% {transform: rotate(-10deg);}
}
```

Step 2: Assign animation to an element

```
@keyframes swing {
  0% {transform: rotate(0deg);}
  100% {transform: rotate(-10deg);}
}

#left-arm {
  animation: swing 2s 0s infinite ease;
}
```

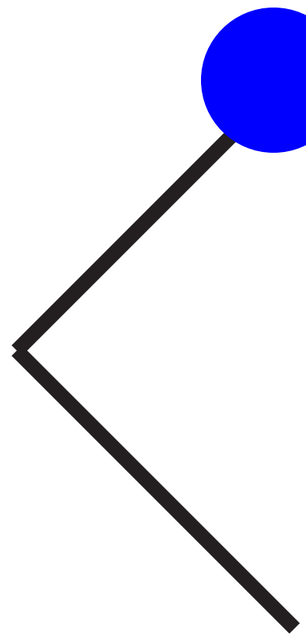




Step 2: Assign animation to an element

```
@keyframes swing {
  0% {transform: rotate(0deg);}
  100% {transform: rotate(-10deg);}
}

#left-arm {
  transfrom-origin: top center;
  animation: swing 2s 0s infinite ease;
}
```



Multi-Step Keyframes

```
@keyframes swing {
  0% {transform: rotate(0deg);}
  20% {transform: rotate(-10deg);}
  40% {transform: rotate(0deg);}
  60% {transform: rotate(10deg);}
  80% {transform: rotate(0deg);}
  100% {transform: rotate(-10deg);}
}

#left-arm {
  transfrom-origin: top center;
  animation: swing 2s 0s infinite ease;
}
```

Combining Keyframes

```
@keyframes swing {
  0%, 40%, 80% {transform: rotate(0deg);}
  20%, 100% {transform: rotate(-10deg);}
  60% {transform: rotate(10deg);}
}
```

```
#left-arm {
  transfrom-origin: top center;
  animation: swing 2s 0s infinite ease;
}
```

‘Forward’ Keyword

- Lets say we are fading in a button. The code might look something like this:

```
@keyframes fadeIn {
  from {
    opacity:0;
    visibility:hidden;
  }
  to {
    opacity:1;
    visibility:visible;
  }
}
```

‘Forward’ Keyword

```
.btn {
  animation: fadeIn .25s;
}
```

Will retrun to the start state!

‘Forward’ Keyword

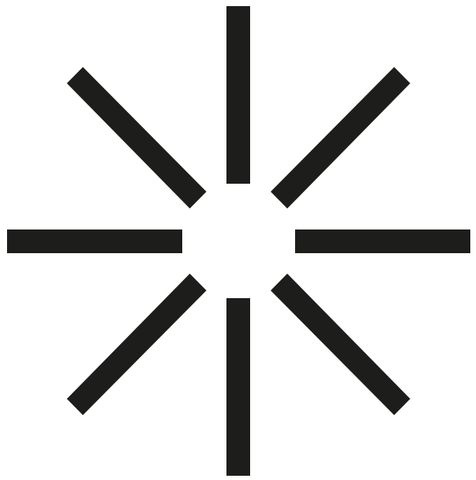
```
.btn {
  animation: fadeIn .25s forwards;
}
```

Keeps the animation
at the final visible step

Animating SVGs with CSS

- This is to be used when you have a ton of different features you want to work with. A more complex graphic...

An SVG image:



Group 2.svg

Group 2.svg

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg width="220px" height="223px" viewBox="0 0 220 223" version="1.1" xmlns="http://www.
w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
    <!-- Generator: Sketch 3.7.1 (28215) - http://www.bohemiancoding.com/sketch -->
    <title>Group 2</title>
    <desc>Created with Sketch.</desc>
    <defs></defs>
    <g id="Exhibition-Pages" stroke="none" stroke-width="1" fill="none" fill-rule="
    evenodd">
        <g id="Group-2" fill="#1D1D1B">
            <polygon id="Path" points="104.470915 223 115.525861 223 115.525861
            138.972977 104.470915 138.972977"></polygon>
            <polygon id="Path" points="133.151988 126.971535 125.261562 134.978139
            183.942955 194.189768 191.574542 186.454"></polygon>
            <polygon id="Path" points="28.4222341 186.454 36.0454714 194.189768
            94.735214 134.978139 86.836438 126.971535"></polygon>
            <polygon id="Path" points="137.101376 117.10293 219.996776 117.10293
            219.996776 105.89707 137.101376 105.89707"></polygon>
            <polygon id="Path" points="0 117.10293 82.8953997 117.10293 82.8953997
            105.89707 0 105.89707"></polygon>
            <polygon id="Path" points="125.261562 88.0303249 133.151988 96.0284652
            191.574542 36.5459997 183.942955 28.8102323"></polygon>
            <polygon id="Path" points="36.0454714 28.8102323 28.4222341 36.5459997
            86.836438 96.0284652 94.735214 88.0303249"></polygon>
            <polygon id="Path" points="104.470915 84.0270229 115.525861 84.0270229
            115.525861 0 104.470915 0"></polygon>
        </g>
    </g>
</svg>
```

Line 1, Column 1 | Spaces: 4 | XML

An SVG image:

- SVG is a file format that contains vector-base images.
  SVG is written in XML, another tag-based language.
  SVGs are always crip. Render perfectly on retina displays.

Accessing Elements in SVGs

Assign classes to the parts

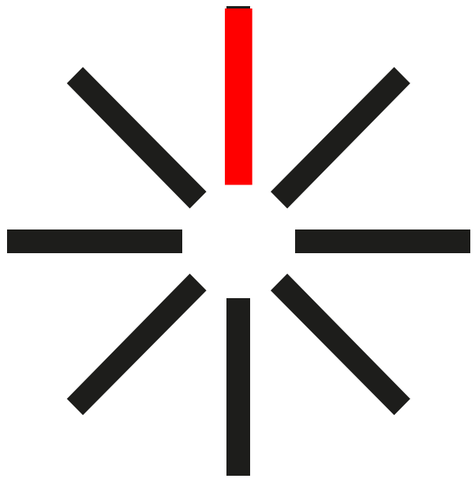
Group 2.svg

Group 2.svg

```
1  <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2  <svg width="220px" height="223px" viewBox="0 0 220 223" version="1.1" xmlns="http://www.
   w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
3      <!-- Generator: Sketch 3.7.1 (28215) - http://www.bohemiancoding.com/sketch -->
4      <title>Group 2</title>
5      <desc>Created with Sketch.</desc>
6      <defs></defs>
7      <g id="Exhibition-Pages" stroke="none" stroke-width="1" fill="none" fill-rule="
       evenodd">
8          <g id="Group-2" fill="#1D1D1B">
9              <polygon id="Path" class="line_1" points="104.470915 223 115.525861 223
               115.525861 138.972977 104.470915 138.972977"></polygon>
10             <polygon id="Path" class="astrix_line_1" points="133.151988 126.971535
               125.261562 134.978139 183.942955 194.189768 191.574542 186.454"></polygon>
11             <polygon id="Path" class="astrix_line_1" points="28.4222341 186.454 36.
               0454714 194.189768 94.735214 134.978139 86.836438 126.971535"></polygon>
12             <polygon id="Path" class="astrix_line_1" points="137.101376 117.10293
               219.996776 117.10293 219.996776 105.89707 137.101376 105.89707"></polygon>
13             <polygon id="Path" class="astrix_line_1" points="0 117.10293 82.8953997
               117.10293 82.8953997 105.89707 0 105.89707"></polygon>
14             <polygon id="Path" class="astrix_line_1" points="125.261562 88.0303249
               133.151988 96.0284652 191.574542 36.5459997 183.942955 28.8102323"></polygon>
15             <polygon id="Path" class="astrix_line_1" points="36.0454714 28.8102323
               28.4222341 36.5459997 86.836438 96.0284652 94.735214 88.0303249"></polygon>
16             <polygon id="Path" class="astrix_line_1" points="104.470915 84.0270229
               115.525861 84.0270229 115.525861 0 104.470915 0"></polygon>
17         </g>
18     </g>
19 </svg>
```

Line 10, Column 52 | Spaces: 4 | XML

Accessed Elements

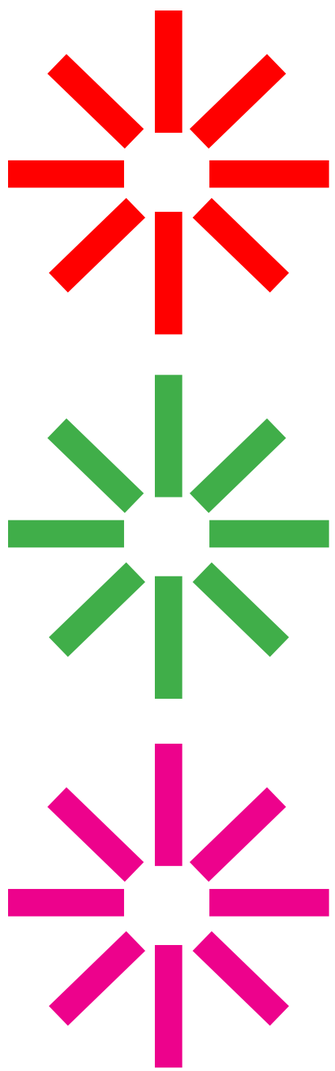


```
.line_1 {
  fill: red;
}
```

Group 2.svg

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg width="220px" height="223px" viewBox="0 0 220 223" version="1.1" xmlns="http://www.
w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
    <!-- Generator: Sketch 3.7.1 (28215) - http://www.bohemiancoding.com/sketch -->
    <title>Group 2</title>
    <desc>Created with Sketch.</desc>
    <defs></defs>
    <g id="Exhibition-Pages" stroke="none" stroke-width="1" fill="none" fill-rule="
    evenodd">
        <g id="Group-2" fill="#1D1D1B">
            <polygon id="Path" class="line_1" points="104.470915 223 115.525861 223
            115.525861 138.972977 104.470915 138.972977"></polygon>
            <polygon id="Path" class="astrix_line_1" points="133.151988 126.971535
            125.261562 134.978139 183.942955 194.189768 191.574542 186.454"></polygon>
            <polygon id="Path" class="astrix_line_1" points="28.4222341 186.454 36.
            0454714 194.189768 94.735214 134.978139 86.836438 126.971535"></polygon>
            <polygon id="Path" class="astrix_line_1" points="137.101376 117.10293
            219.996776 117.10293 219.996776 105.89707 137.101376 105.89707"></polygon>
            <polygon id="Path" class="astrix_line_1" points="0 117.10293 82.8953997
            117.10293 82.8953997 105.89707 0 105.89707"></polygon>
            <polygon id="Path" class="astrix_line_1" points="125.261562 88.0303249
```

Animating the Fills



```
@keyframe colorChange {
  0%   {fill: red}
  50%  {fill: green}
  100% {fill: magenta}
}

.line_1, .line_2, .line_3,
.line_4, .line_5, .line_6,
.line_7, .line_8 {
  animation: colorChange 3s infinate;
}
```

Unique Properties for Styling SVGs

enable-background
fill
fill-opacity
filter
mask
stroke
stroke-dasharray
stroke-dashoffset
viewport-fill

Checkout MDN for the full list of SVG properties.

Unique Properties for Styling SVGs

enable-background
fill
fill-opacity
filter
mask
stroke
stroke-dasharray
stroke-dashoffset
viewport-fill

Checkout MDN for the full list of SVG properties.