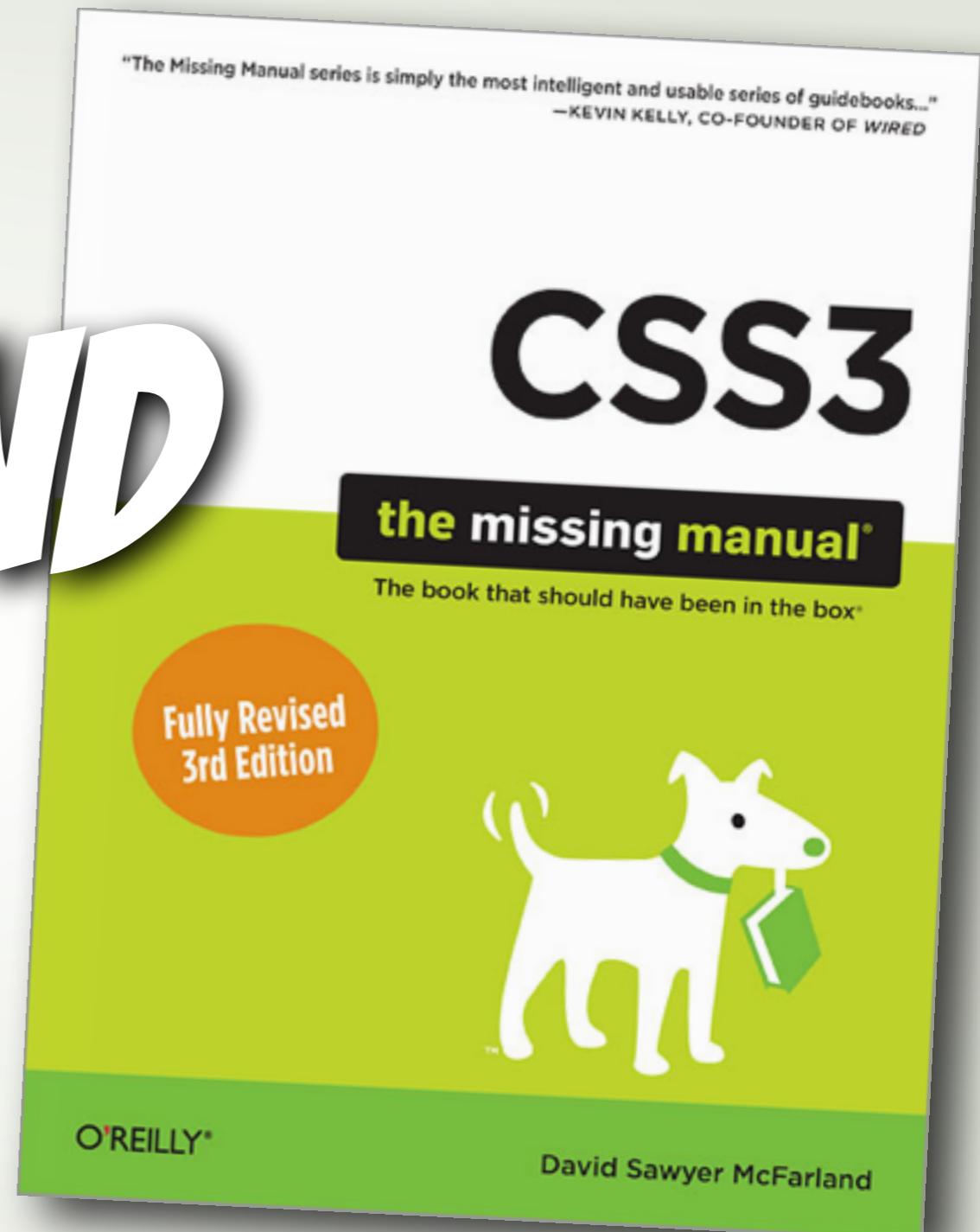
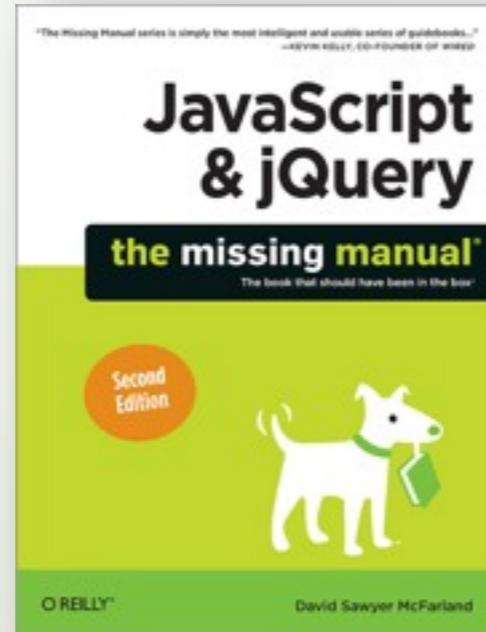
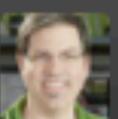


# **Building User Interfaces with CSS3**

# DAVE MCFARLAND



[Picks](#) [Popular](#) [Recent](#) [Tags](#)[Older Pens →](#)

Search...



Alexey

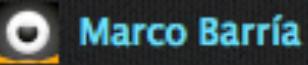
0

311

1

## MAZE - CSS game

Ingres desde la izquierda hasta la barra roja. // Enter from the left to the red bar.  
Haz clic sobre los numero hasta que logres pasar // Click on the numbers until you can pass



Marco Barría

3

399

4



Davilious

3

242

6



sid

0

303

6

©2013 CodePen *Demo or it didn't happen.*

Download button

[Store](#) [About](#) [Support](#) [Docs](#) [Blog](#) [Stats](#) [@CodePen](#)<http://codepen.io>

# **basics**

## **building you CSS3 tool kit**

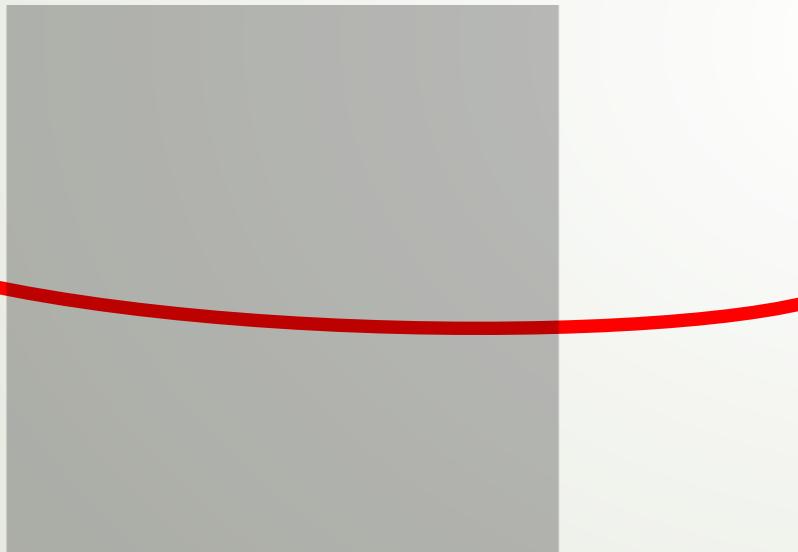
# RGBA

transparency FTW!



# rgba

```
1 .box {  
2   background-color: rgb(144,144,144);  
3   background-color: rgba(0,0,0,.25);  
4 }
```



*Compatibility tables for support of HTML5, CSS3, SVG and more in desktop and mobile browsers.*

Latest update: Three new features added: Strict Transport Security, CSS outline property and download attribute (December 3, 2012)

**Ads by Google**[HTML5](#)[HTML5 CSS 3](#)[HTML5 Web Browser](#)[HTML5 Canvas Demo](#)**Index****Tables**

Search:

border-radius, WebGL, woff, (

[Import stats](#)[FAQ](#)[Resources](#)[Embed](#)

## CSS

- @font-face Web fonts
- calc() as CSS unit value
- 2.1 selectors
- Counters
- Feature Queries
- Filter Effects
- Generated content
- Gradients
- Grid Layout
- Hyphenation
- inline-block
- Masks
- min/max-width/height
- outline
- position:fixed
- Regions

## HTML5

- Audio element
- Canvas (basic support)
- Color input type
- contenteditable attribute (basic support)
- Datalist element
- dataset & data-\* attributes
- Date/time input types
- Details & Summary elements
- Download attribute
- Drag and Drop
- Form validation
- HTML5 form features
- input placeholder attribute
- New semantic elements
- Number input type

## SVG

- Inline SVG in HTML5
- SVG (basic support)
- SVG effects for HTML
- SVG filters
- SVG fonts
- SVG fragment identifiers
- SVG in CSS backgrounds
- SVG in HTML img element
- SVG SMIL animation
- **All SVG features**

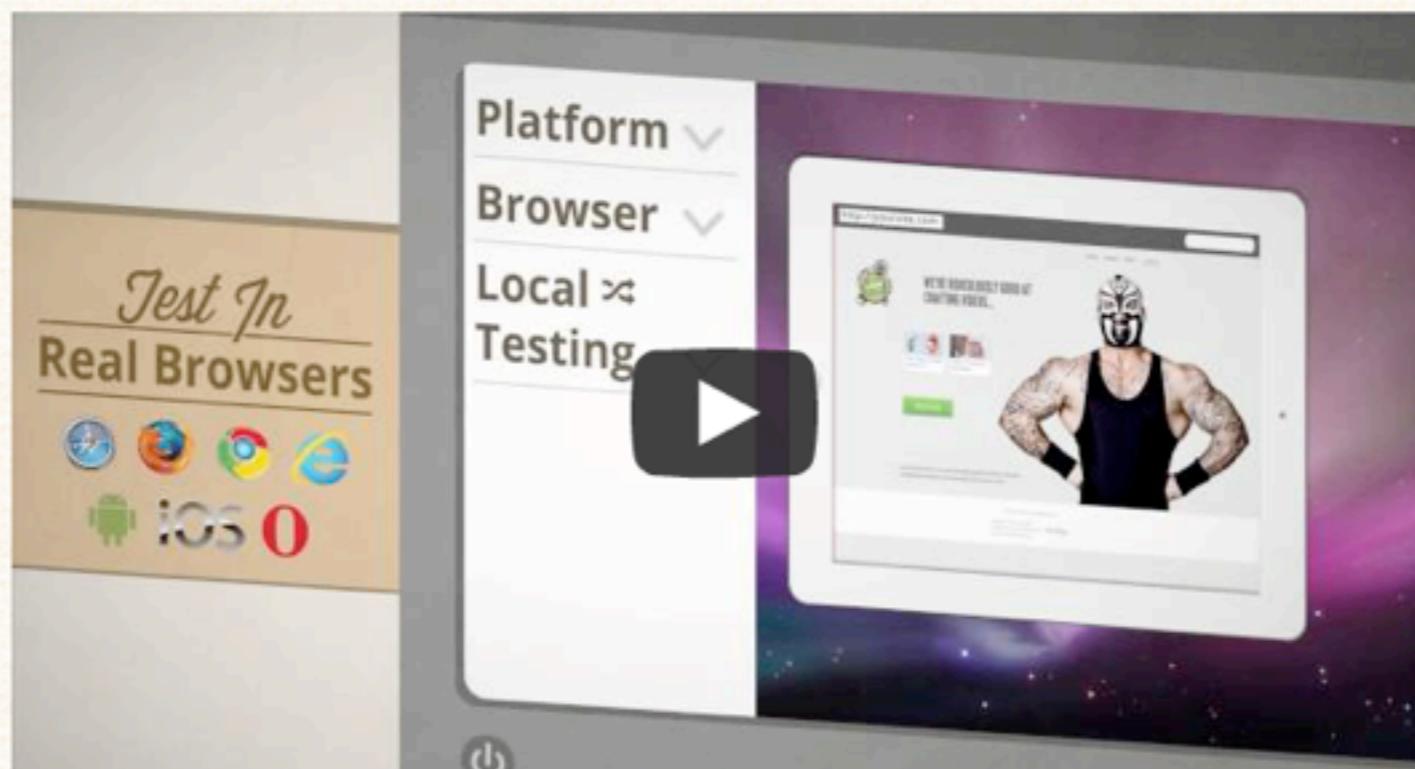
Other

<http://caniuse.com>

- async attribute for external scripts
- clipboard (DOM clipboard)

# Live, Web-Based Browser Testing

Instant access to all desktop and mobile browsers.  
Say goodbye to your setup of virtual machines and devices.



Free Trial

No credit card required. We promise.

 Full Name

 Email

 Password

Sign up for free

Get instant access to 200+ desktop and mobile browsers on different Windows, Mac & mobile OS flavors.

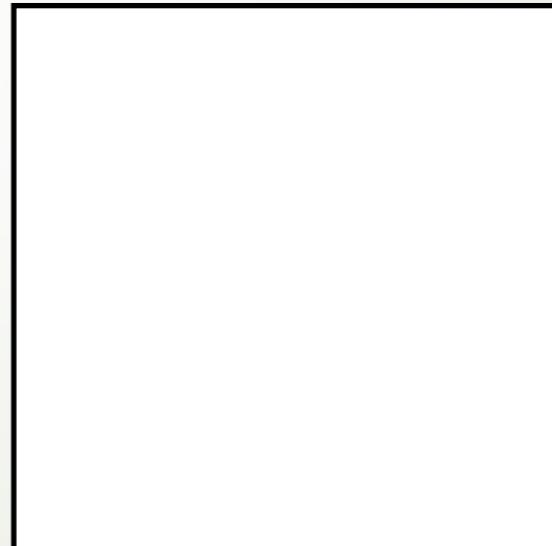
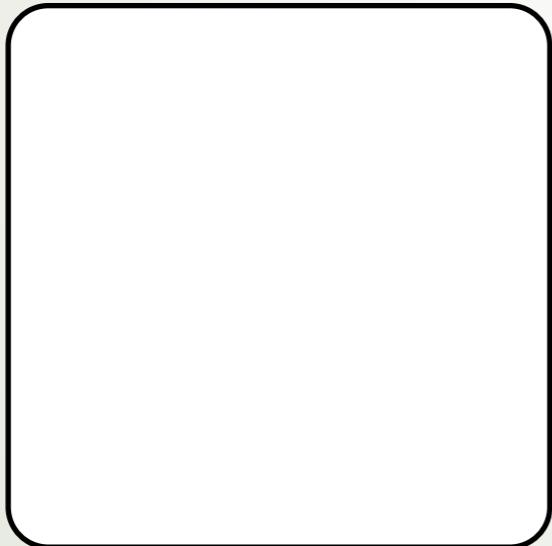
<http://www.browserstack.com>

# Border Radius



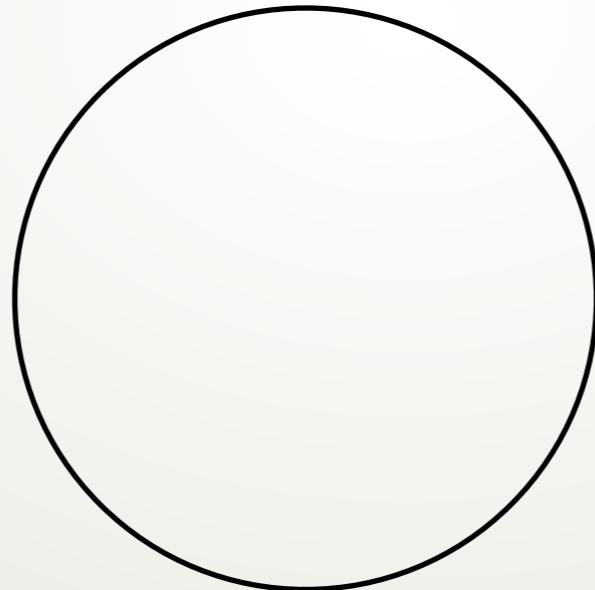
# Border Radius

```
1 .box {  
2   border-radius: 10px;  
3   border: 1px solid black;  
4 }
```



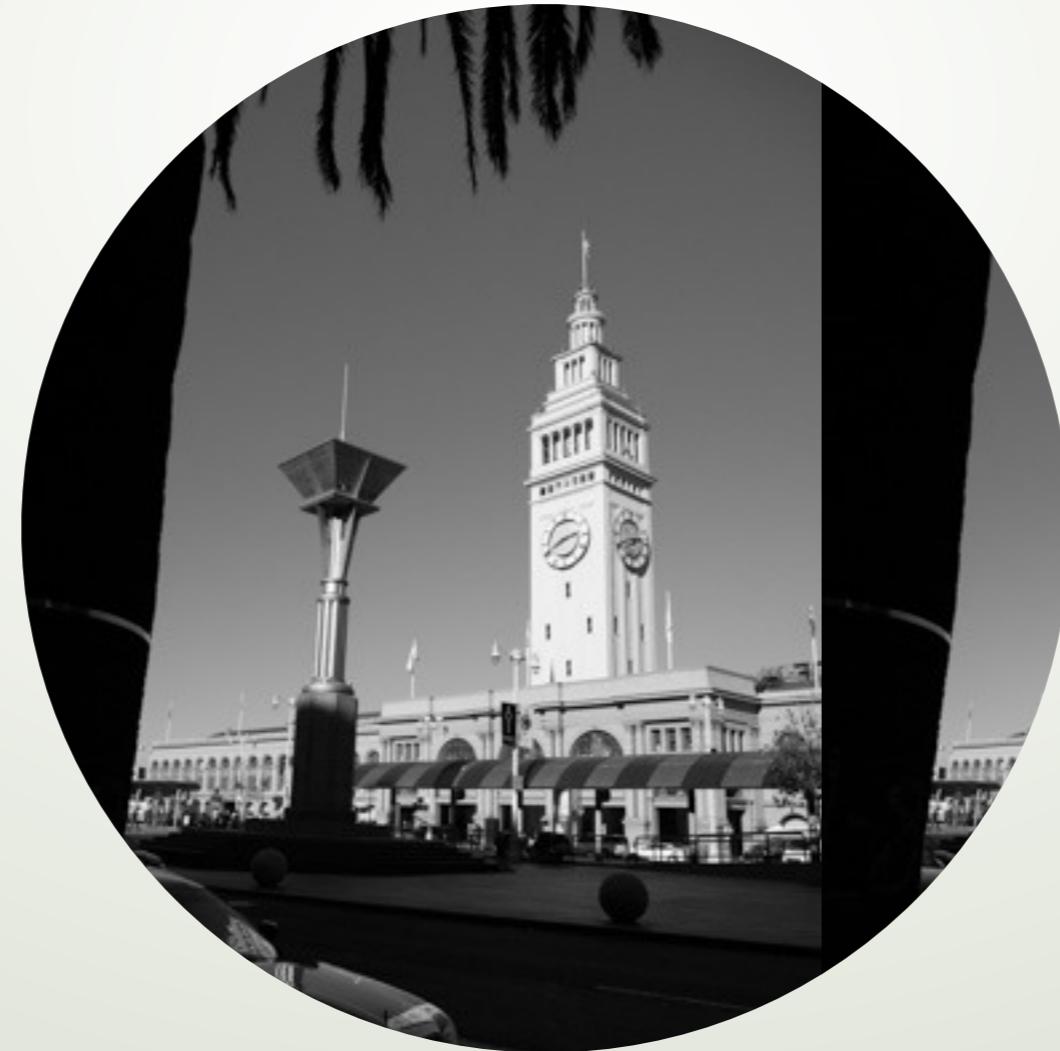
# Border Radius

```
1 .box {  
2   border-radius: 50%;  
3   border: 1px solid black;  
4 }
```



# Border Radius

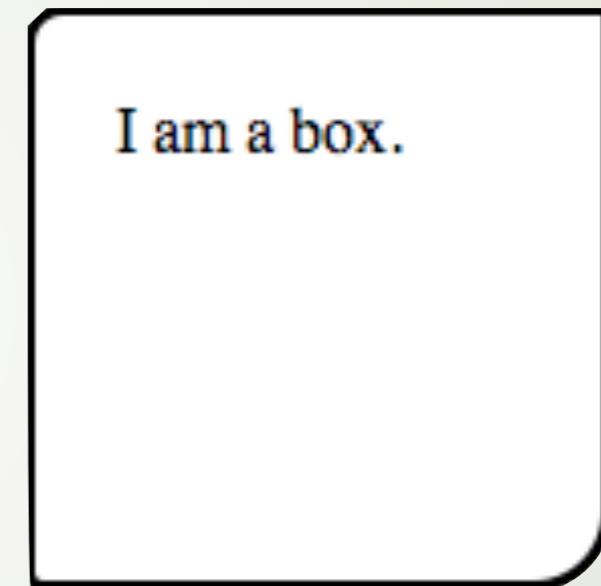
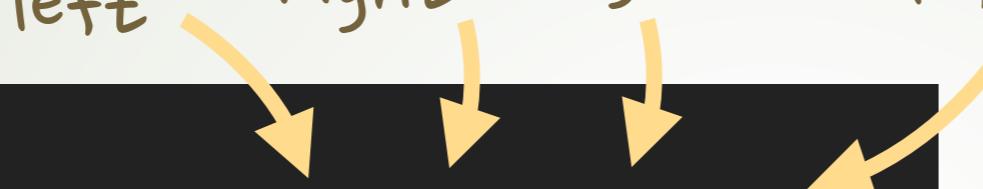
```
1 img {  
2   border-radius: 50%;  
3 }
```



# Border Radius

```
1 .box {  
2   border-radius: 10px 0 20px 5px;  
3   border: 3px solid black;  
4 }
```

top left      top right      bottom right      bottom left



<http://codepen.io/sawmac/pen/HLmeC>

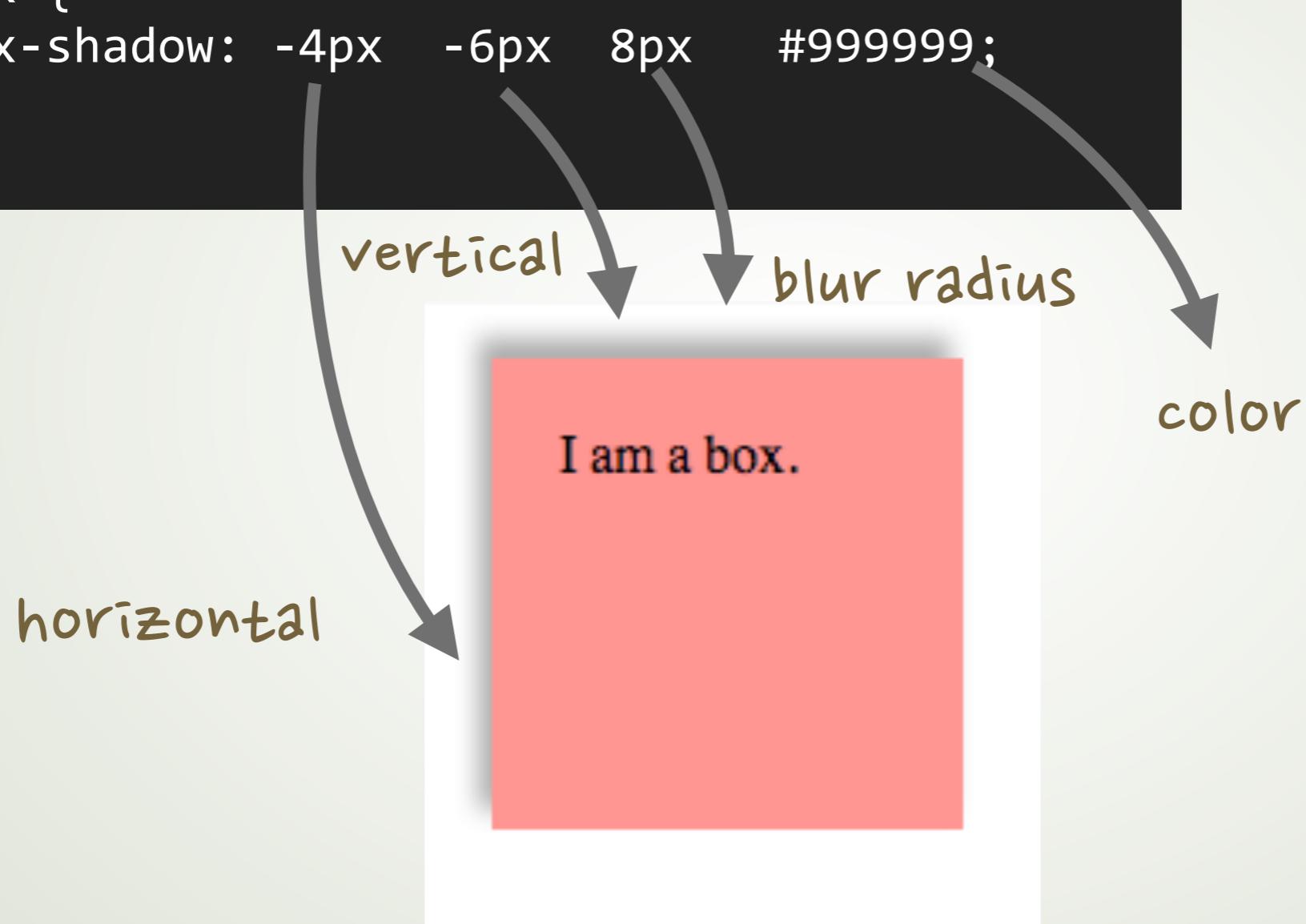
# **Box Shadow**

**Adding depth to your divs**



# Box Shadow

```
1 .box {  
2   box-shadow: -4px -6px 8px #999999;  
3 }
```



<http://codepen.io/sawmac/pen/CmlJJI>

# Box Shadow

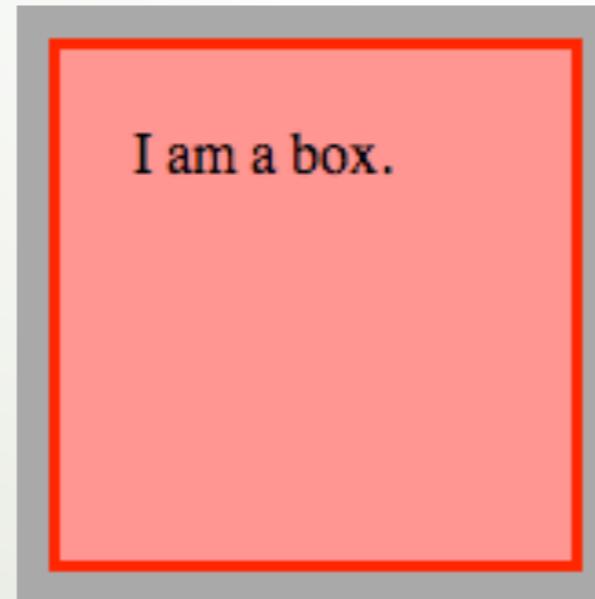
```
1 .box {  
2   box-shadow: -4px 6px 8px 10px #999999;  
3 }
```



<http://codepen.io/sawmac/pen/jvhKL>

# Multiple borders

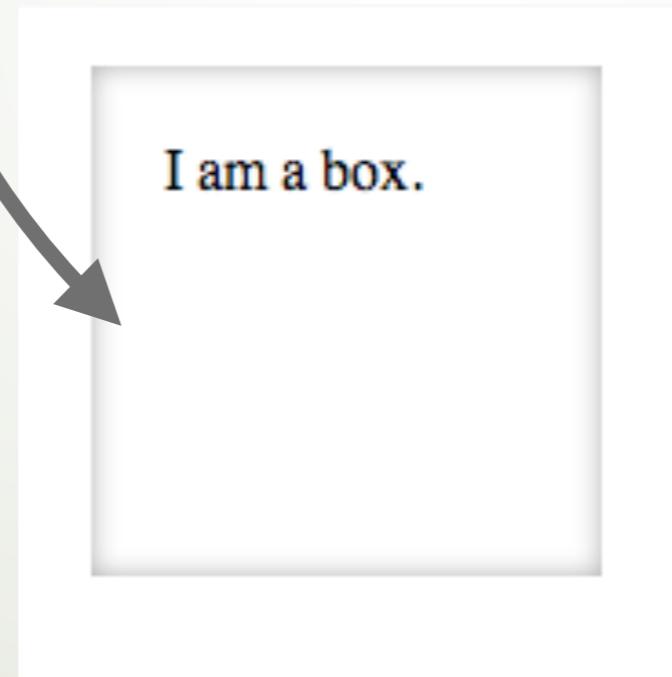
```
1 .box {  
2   box-shadow: 0 0 0 10px #999999;  
3   border: 2px solid red;  
4 }
```



<http://codepen.io/sawmac/pen/Cuhne>

# Box Shadow

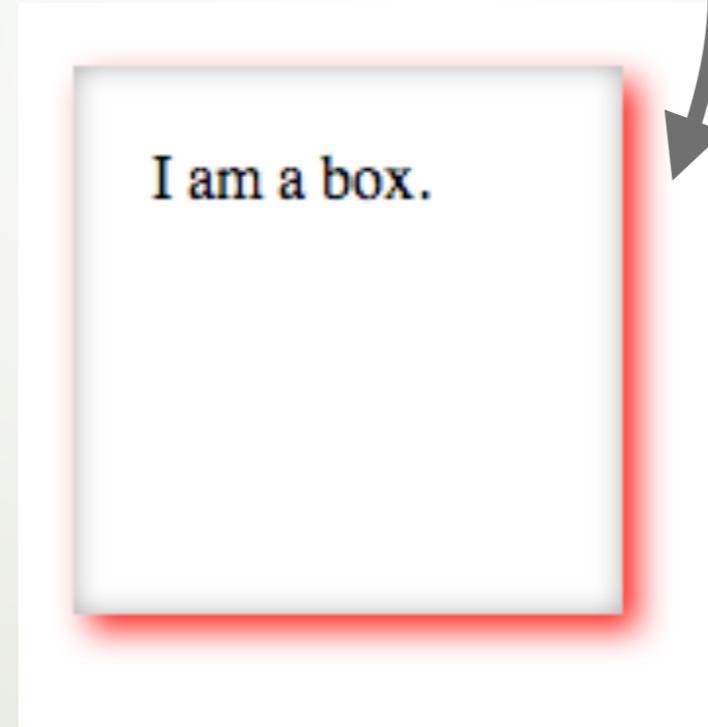
```
1 .box {  
2   box-shadow: inset 0 0 8px #999999;  
3 }
```



<http://codepen.io/sawmac/pen/sgbAd>

# Box Shadow

```
1 .box {  
2   box-shadow: inset 0 0 8px #999999,  
3   4px 4px 10px #FF0000;  
4 }
```

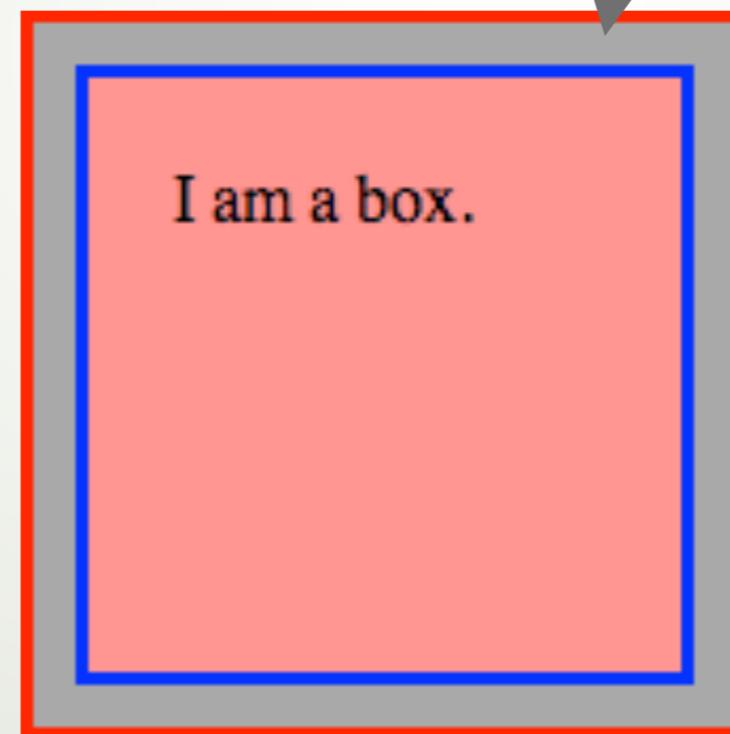


<http://codepen.io/sawmac/pen/mbFwk>

# Multiple, multiple borders

```
1 .box {  
2   box-shadow: 0 0 0 10px #999999,  
3                 0 0 0 14px #FF0000;  
4   border: 3px solid blue;  
5 }
```

first shadow on top



last shadow on bottom

<http://codepen.io/sawmac/pen/CkJjm>

# Text Shadow



“An indispensable tool.”

— Bruce Bowman, *Adobe BrowserLab Product Manager*

## Download Modernizr 2.6.2

Use the [Development version](#) to develop with and learn from. Then, when you're ready for production, use the build tool below to pick only the tests you need.

### CSS3

- @font-face
- background-size
- border-image
- border-radius
- box-shadow
- Flexible Box Model (flexbox)
- Flexbox Legacy
- hsla()
- multiple backgrounds
- opacity
- rgba()
- text-shadow
- CSS Animations
- CSS Columns
- CSS Generated Content  
(:before/:after)
- CSS Gradients
- CSS Reflections
- CSS 2D Transforms
- CSS 3D Transforms
- CSS Transitions

[TOGGLE](#)

### HTML5

- applicationCache
- Canvas
- Canvas Text
- Drag 'n Drop
- hashchange
- History (pushState)
- HTML5 Audio
- HTML5 Video
- IndexedDB
- Input Attributes
  - Note: does not add classes
- Input Types
  - Note: does not add classes
- localStorage
- postMessage
- sessionStorage
- Web Sockets
- Web SQL Database
- Web Workers

[TOGGLE](#)

### Misc.

- Geolocation API
- Inline SVG
- SMIL
- SVG
- SVG clip paths
- Touch Events
- WebGL

[TOGGLE](#)

### Extra

- html5shiv v3.6
  - html5shiv v3.6 w/ printshiv
  - Modernizr.load  
([yepnope.js](#))
  - Media Queries
  - Add CSS Classes
- className prefix:

► Extensibility

 GENERATE!

► Non-core detects

<http://modernizr.com/download/>



Follow us on Twitter



Contribute on GitHub



Subscribe with RSS



Donate to Modernizr

### Latest news

January 7th, 2013

[Modernizr Policy on Browser Bugs and Feature Detects](#)

Read our policy on bugs and false positives.

#### PREVIOUS

September 4th, 2012

[Modernizr 2.6.2 released](#)

July 19th, 2012

[Modernizr 2.6 released](#)

February 6th, 2012

[Modernizr 2.5: Supercharged for 2012](#)

# Modernizr

- ▶ Add <script> to web page

```
<script src="_js/modernizr.custom.94746.js"></script>
```

- ▶ Add class to <html>

```
<html class="no-js">
```

# Modernizr

IE 7	<html class="js no-multiplebgs no-opacity no-textshadow no-boxshadow no-cssanimations no-cssgradients no-csstransforms no-csstransforms3d no-csstransitions">
IE 8	<html class="js no-multiplebgs no-opacity no-textshadow no-boxshadow multiplebgs opacity no-cssanimations no-cssgradients no-csstransforms no-csstransforms3d no-csstransitions">
IE 9	<html class="js multiplebgs no-textshadow boxshadow opacity no-cssanimations no-cssgradients csstransforms no-csstransforms3d no-csstransitions">
IE 10	<html class="js multiplebgs textshadow boxshadow opacity cssanimations cssgradients csstransforms csstransforms3d csstransitions">
chrome	
safari	
firefox	

# Modernizr

- ▶ For current browsers

```
.textshadow h1 {
```

```
}
```

- ▶ For older browsers

```
.no-textshadow h1 {
```

```
}
```

# Transforms

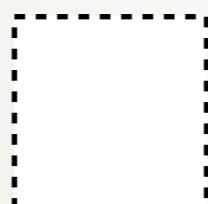
► Scale



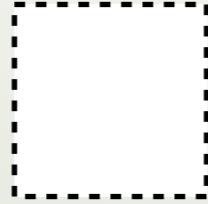
► Rotate



► Skew



► Translate



# Scale

```
1 .box {  
2   transform: scale(2);  
3 }
```

0-1 shrinks the element

1 is the same size

>1 enlarges the element

# Scale

```
1 .box {  
2   -moz-transform: scale(2);  
3   -o-transform: scale(2);  
4   -ms-transform: scale(2);  
5   -webkit-transform: scale(2);  
6   transform: scale(2);  
7 }
```

requires vendor prefixes



+ codepen.io/sawmac/pen/IEFax

↻ Reader ◀ ▶ &lt;/&gt; □/□ □/□

≡ myTinyTodo Demo

incontrol examples ▾

↔ Resizer

Get lat/long

Katana

film sites ▾

lego league ▾

&gt;&gt; +



Save

Fork

Info

Share

Editor

Details

Professor

## HTML

```
1 <div class="container">
2   <div class="transform"></div>
3   <div class="original"></div>
4 </div>
```

361



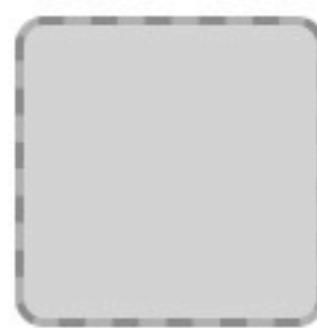
## CSS (SCSS)

362



## JS

```
6 .transform {
7   position: relative;
8   width: 100px;
9   height: 100px;
10  border-radius: 10px;
11  border: 3px dashed rgba(0,0,0,.25);
12  /* add transforms here */
13 }
14
15
16 .original {
17   position: absolute;
18   left: 0;
19   top: 0;
20   width: 100px;
21   height: 100px;
22 }
```



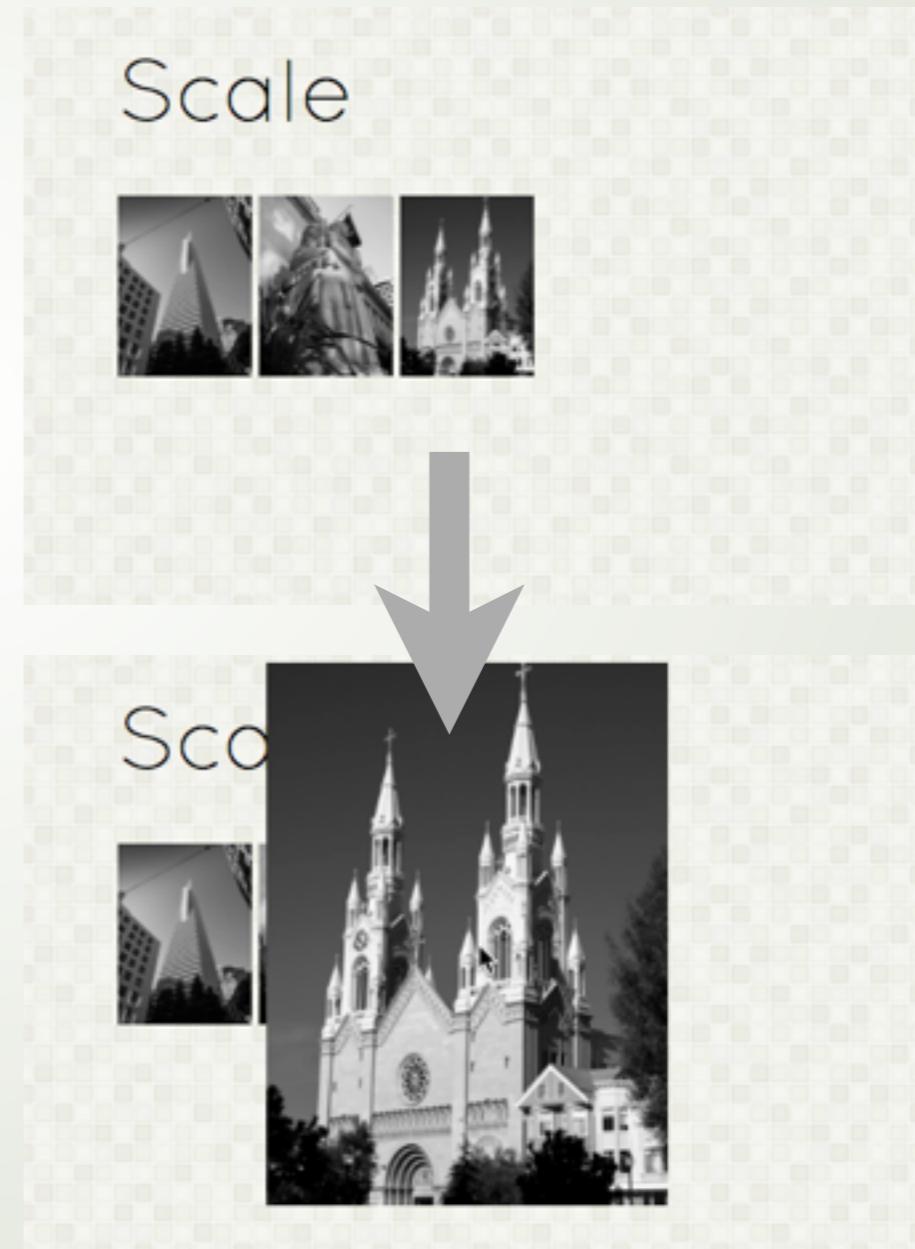
+ Add to Collection

<http://codepen.io/sawmac/pen/IEFax>

Keyboard

# Scale

```
1 img:hover {  
2     -moz-transform: scale(3);  
3     -o-transform: scale(3);  
4     -ms-transform: scale(3);  
5     -webkit-transform: scale(3);  
6     transform: scale(3);  
7 }
```



# Scale

```
1 .box {  
2   transform: scale(2,1);  
3 }
```

two separate values

first is horizontal

second is vertical

# Scale

```
1 img:hover {  
2     -moz-transform: scale(5,2);  
3     -o-transform: scale(5,2);  
4     -ms-transform: scale(5,2);  
5     -webkit-transform: scale(5,2);  
6     transform: scale(5,2);  
7 }
```



# Scale

negative values flip the element

```
1 .box {  
2   transform: scale(-1);  
3 }
```

flip both horizontal and vertically

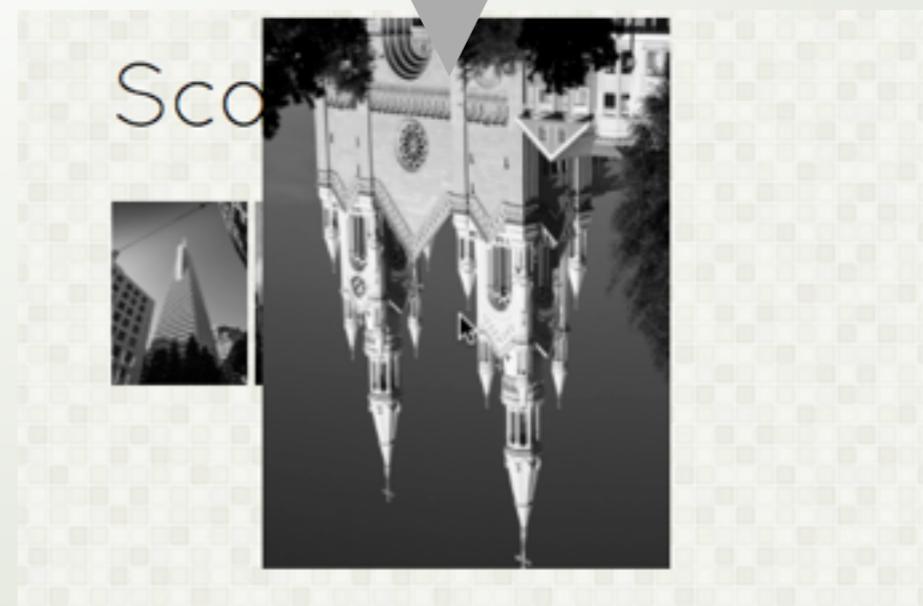
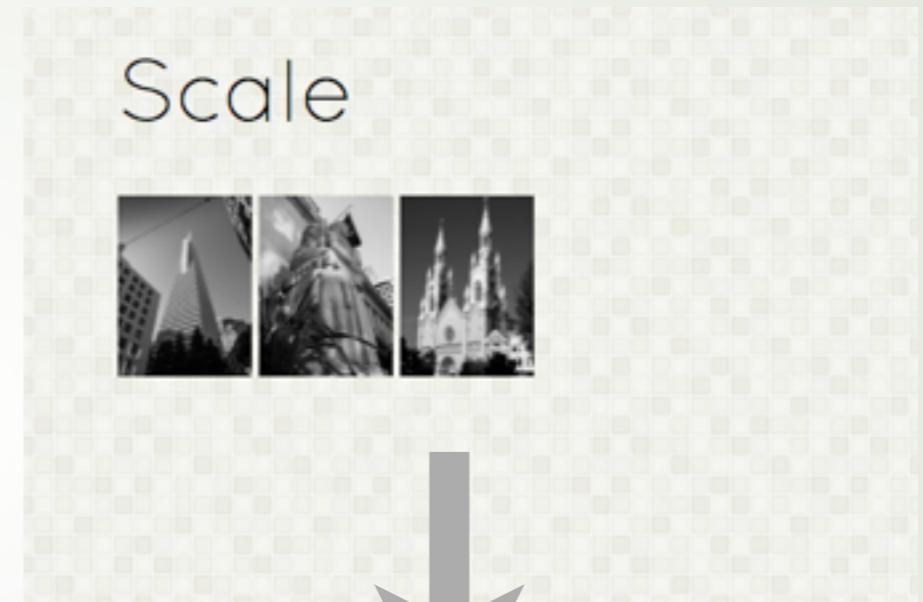
```
1 .box {  
2   transform: scale(-1,1);  
3 }
```

flip both horizontally only

# Scale

negative values flip the element

```
1 img:hover {  
2   -moz-transform: scale(3,-3);  
3   -o-transform: scale(3,-3);  
4   -ms-transform: scale(3,-3);  
5   -webkit-transform: scale(3,-3);  
6   transform: scale(3,-3);  
7 }
```



# Scale

there's also a scaleX

```
1 image:hover {  
2   transform: scaleX(2);  
3 }
```

...and scaleY

```
1 image:hover {  
2   transform: scaleY(.5);  
3 }
```

# Rotate

```
1 .box {  
2   transform: rotate(45deg);  
3 }
```

provide a degree value

0deg is no rotation

360deg is one rotation

-45deg is 45 degree rotation counterclockwise

# Rotate

```
1 h1 {  
2     -moz-transform: rotate(-3deg);  
3     -o-transform: rotate(-3deg);  
4     -ms-transform: rotate(-3deg);  
5     -webkit-transform: rotate(-3deg);  
6     transform: rotate(-3deg);  
7 }
```

Headline

# Rotate

## Headline

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor eu feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies est. Mauris placerat eleifend leo.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor eu feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies est. Mauris placerat eleifend leo.

Beware: transformed elements don't affect the flow of the page

# Skew

```
1 .box {  
2   transform: skew(10deg,45deg);  
3 }
```

x axis

y axis

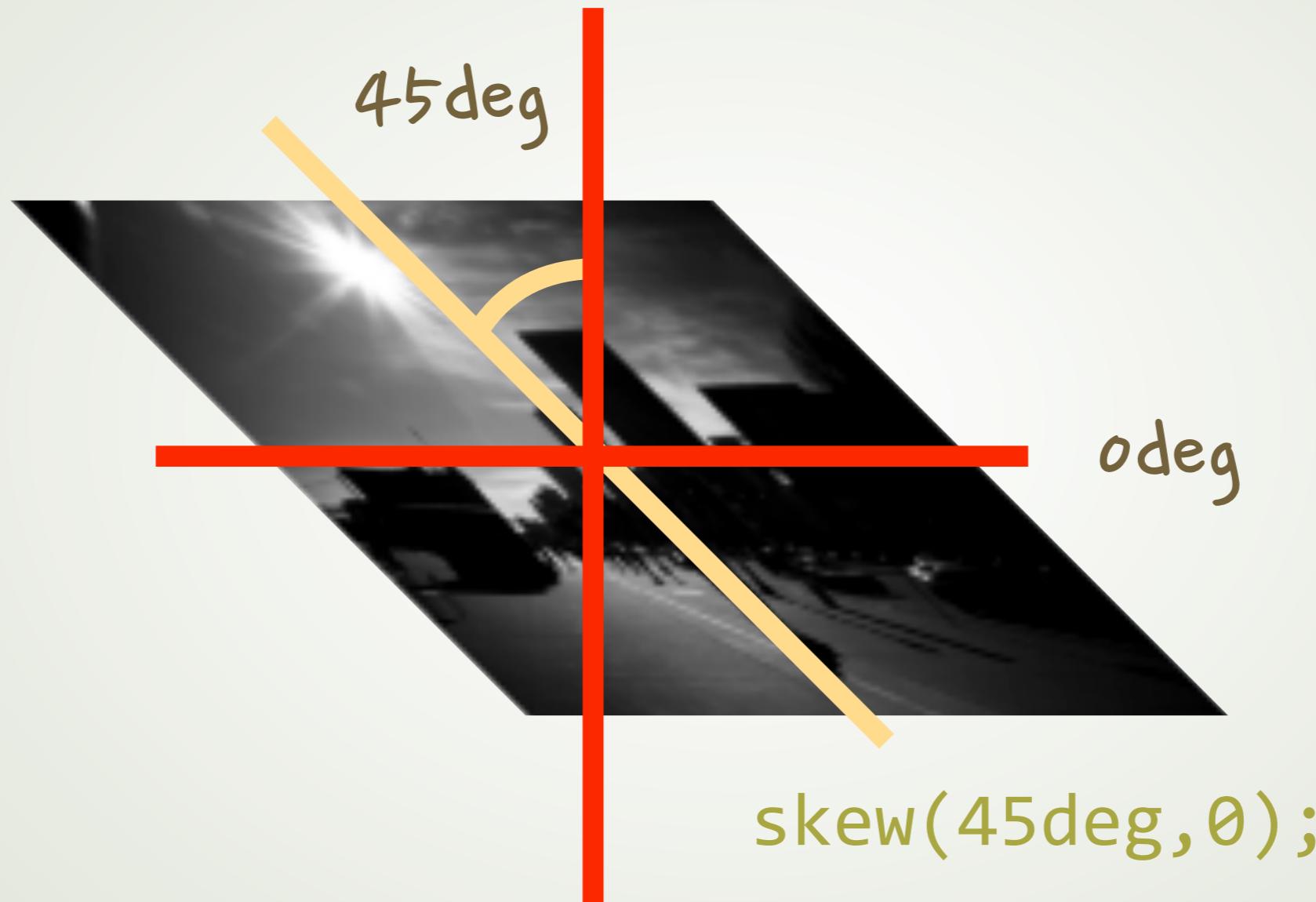
lets you slant an element on its  
x and y axes

# Skew

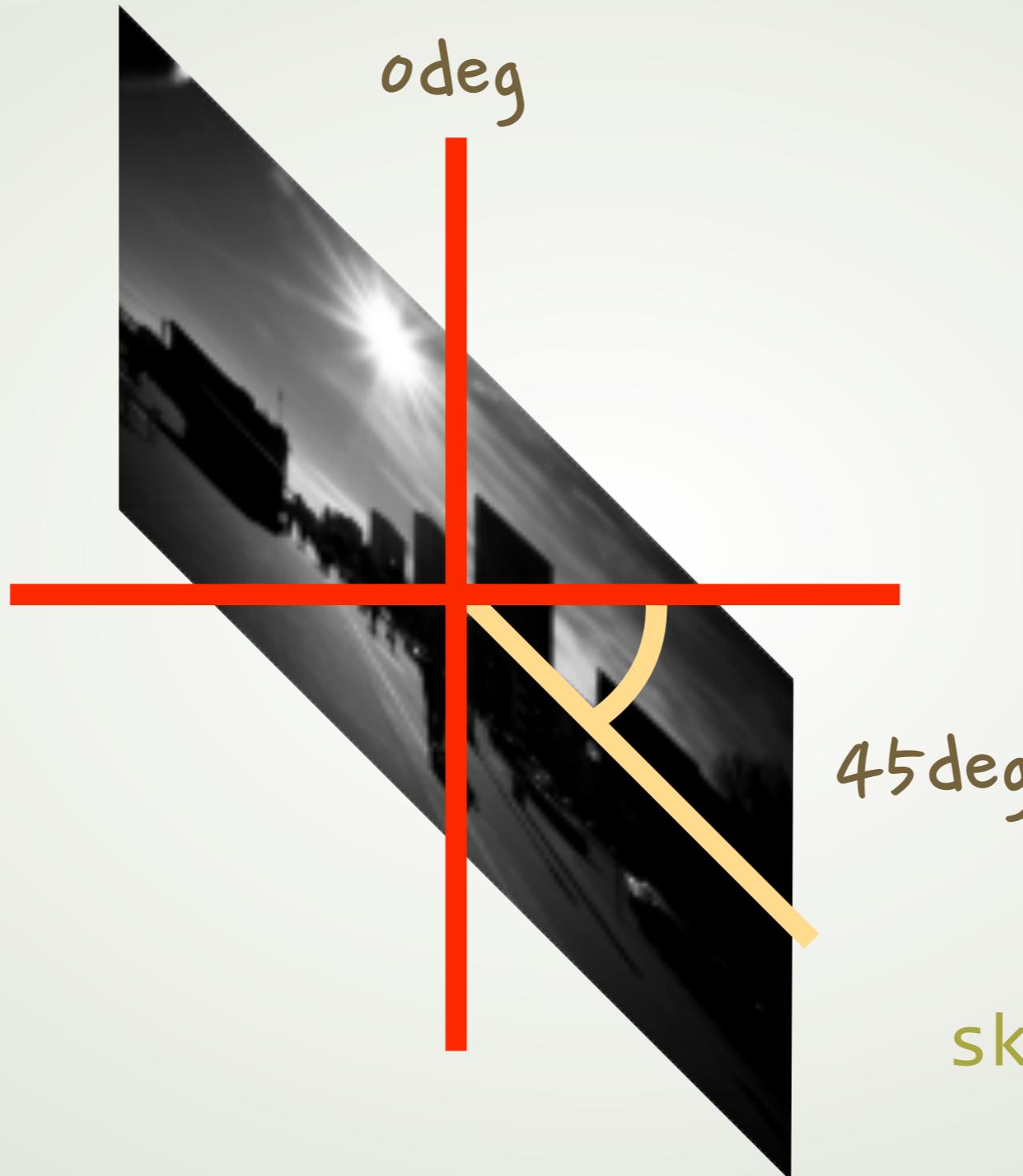
```
1 img {  
2   -moz-transform: skew(45deg,0);  
3   -o-transform: skew(45deg,0);  
4   -ms-transform: skew(45deg,0);  
5   -webkit-transform: skew(45deg,0);  
6   transform: skew(45deg,0);  
7 }
```



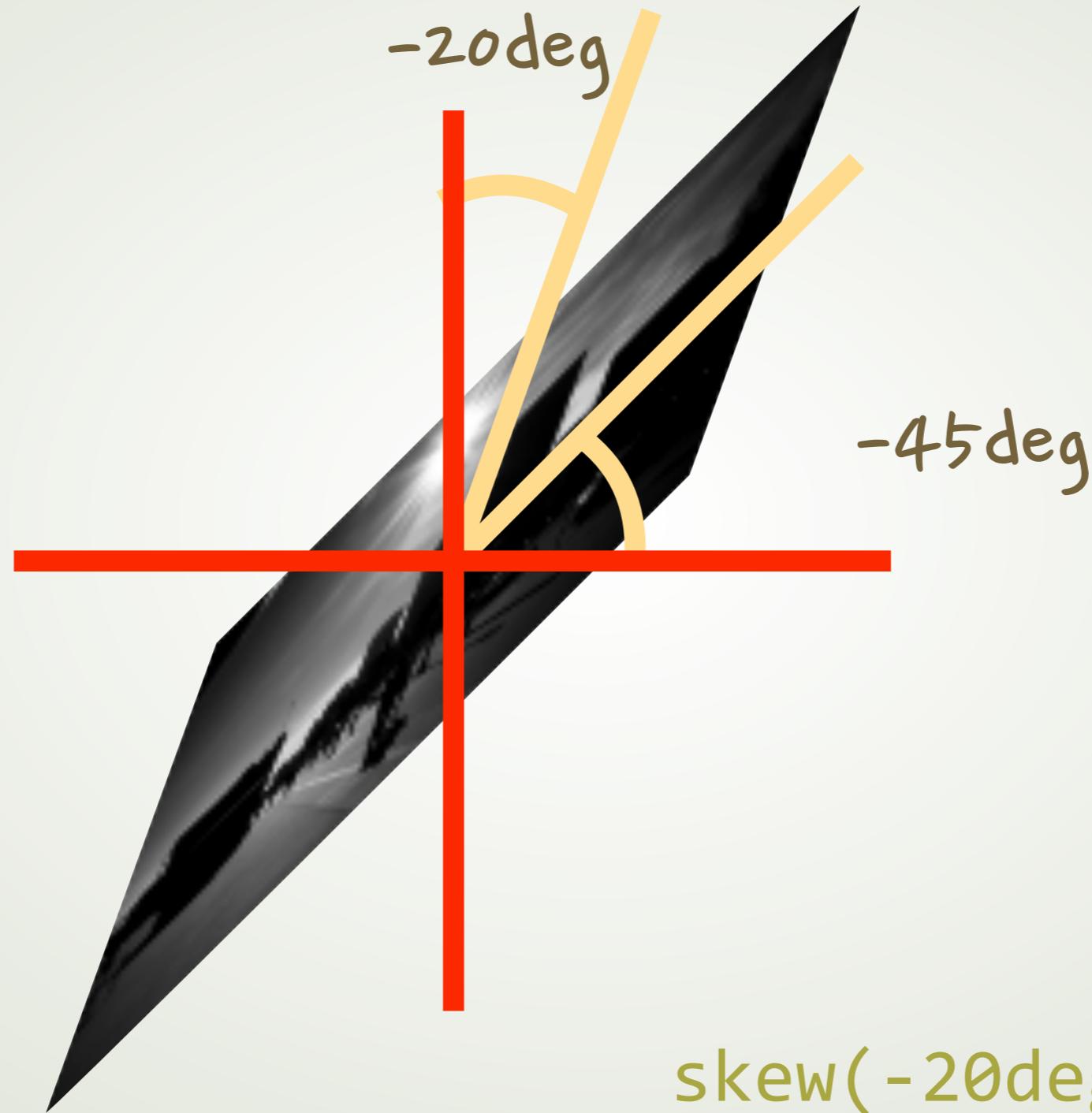
# Skew



# Skew



# Skew



# Skew

there's also a skewX

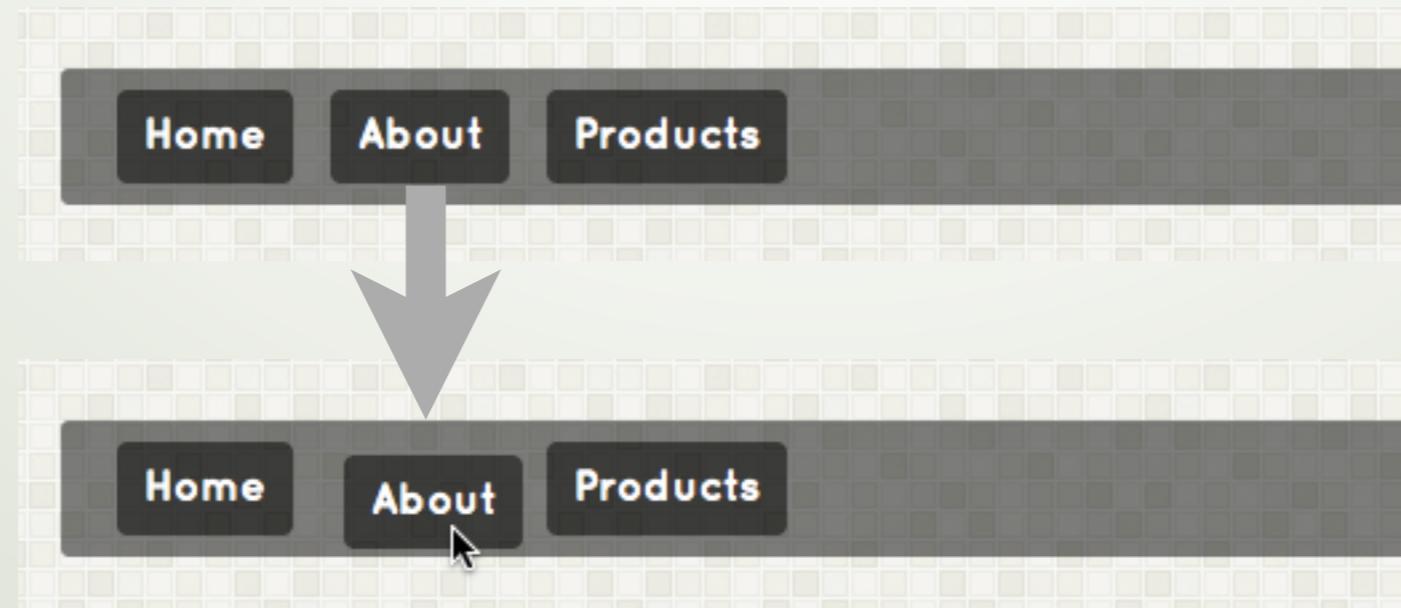
```
1 .box {  
2   transform: skewX(45deg);  
3 }
```

...and skewY

```
1 .box {  
2   transform: skewY(45deg);  
3 }
```

# Translate

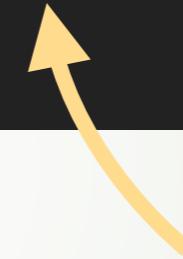
```
1 a:active {  
2     -moz-transform: translate(5px,5px);  
3     -o-transform: translate(5px,5px);  
4     -ms-transform: translate(5px,5px);  
5     -webkit-transform: translate(5px,5px);  
6     transform: translate(5px,5px);  
7 }
```



# Translate

horizontally

```
1 a:active {  
2   transform: translate(1px,1px);  
3 }
```



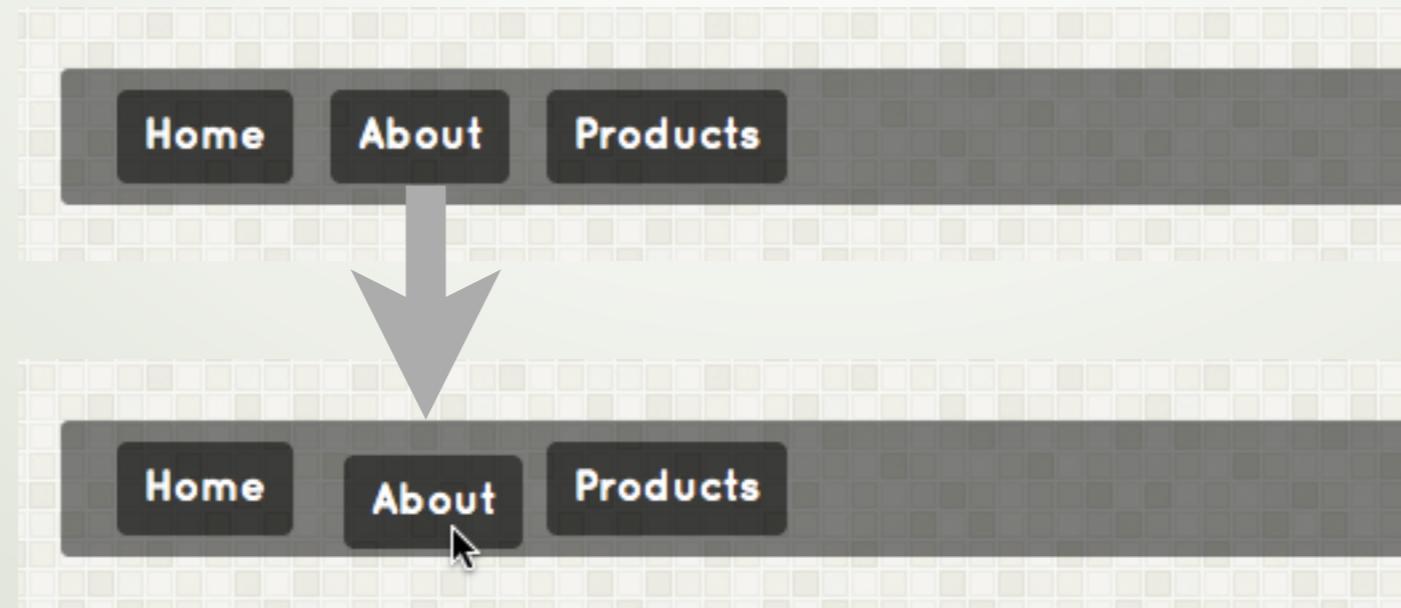
vertically

moves an element specified amount  
horizontally and vertically

can use px, em, %

# Translate

```
1 a:active {  
2     -moz-transform: translate(5px,5px);  
3     -o-transform: translate(5px,5px);  
4     -ms-transform: translate(5px,5px);  
5     -webkit-transform: translate(5px,5px);  
6     transform: translate(5px,5px);  
7 }
```



# Translate

there's also a `translateX`

```
1 a:active {  
2   transform: translateX(.5em);  
3 }
```

...and `translateY`

```
1 a:active {  
2   transform: translateY(.5em);  
3 }
```

# Multiple Transforms

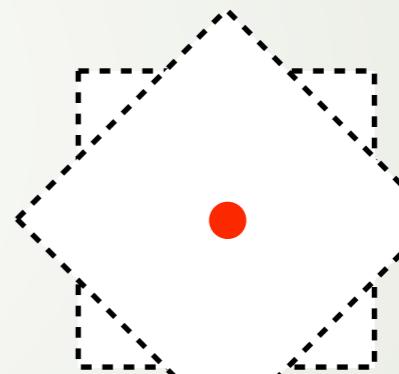
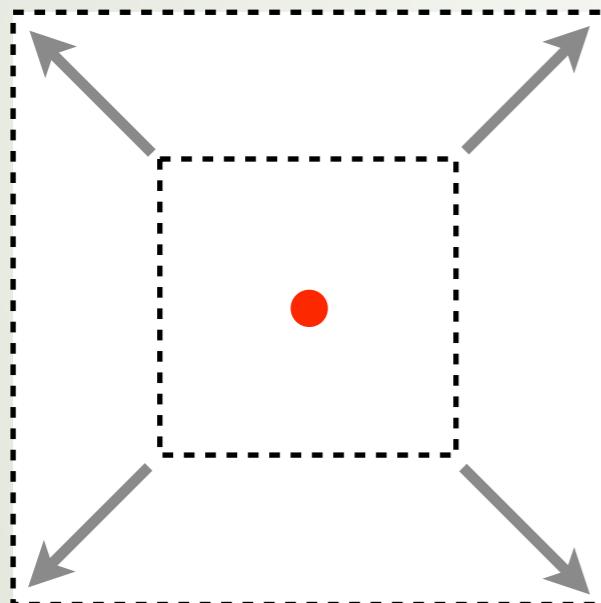
```
1 .box {  
2   transform: scale(2) rotate(45deg) translate(1em,2em);  
3 }
```

space-separated list of transform options

no commas!

# Transform Origin

normally transformations are applied  
to center of element

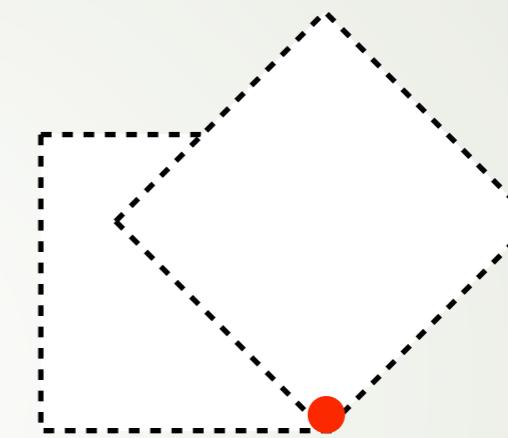
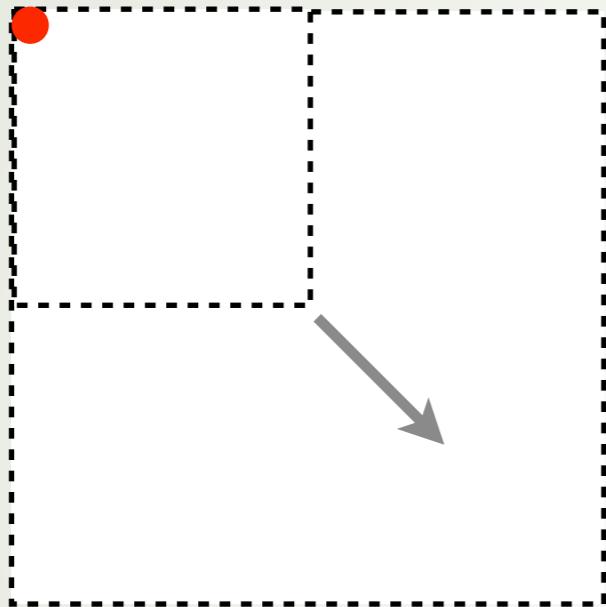


```
1 .box {  
2   transform: scale(2);  
3 }
```

```
1 .box {  
2   transform: rotate(45deg);  
3 }
```

# Transform Origin

but it doesn't have to be that way



```
1 .box {  
2   transform: scale(2);  
3   transform-origin: left top;  
4 }
```

```
1 .box {  
2   transform: rotate(45deg);  
3   transform-origin: 100% 100%;  
4 }
```

same values used for background-position property

# CSS Preprocessors



# Benefits

write less code

Automagically handles vendor prefixes

Makes CSS code management easier

minifies CSS files for faster downloads

# The Options

 SASS  
<http://sass-lang.com>

Less  
<http://lesscss.org>

Stylus  
<http://learnboost.github.com/stylus/>

# Hurdles

Ruby, node, egads

command-line setup

new syntax

(don't we have enough to learn already?)

# The Setup

codekit (mac)  
 <http://incident57.com/codekit/>

Live Reload (mac + win)  
<http://livereload.com>

the way of the geek  
<http://sass-lang.com/download.html>

# What does SASS do?

accepts standard CSS

imports (combine separate CSS files)

variables

mixins

...and a whole lot more\*

\*[http://sass-lang.com/docs/yardoc/file.SASS\\_REFERENCE.html](http://sass-lang.com/docs/yardoc/file.SASS_REFERENCE.html)

# A simple example

ScSS

```
1 a {  
2   color:red;  
3 }
```



cSS

```
1 a {  
2   color:red;  
3 }
```

styles.scss

styles.css

use ScSS

<http://codepen.io/sawmac/pen/Hkipb>

# #1 Reason to Use SASS

main SASS file

```
1 @import(reset);  
2 @import(type);  
3 @import(grid);  
4 @import(forms);
```

styles.scss

separate files

\_reset.scss  
\_type.scss  
\_grid.scss  
\_forms.scss

# Variables

ScSS

```
1 $brandColor: red;  
2 a {  
3   color:$brandColor;  
4 }
```



cSS

```
1 a {  
2   color:red;  
3 }
```

styles.scss

styles.css

<http://codepen.io/sawmac/pen/epIkB>

# Nested Rules

ScSS

```
1 nav {  
2   background-color: blue;  
3  
4   li {  
5     float: left;  
6   }  
7  
8   a {  
9     display: block;  
10    border: 1px solid black;  
11    text-decoration: none;  
12    padding: 2px;  
13  
14  }  
15 }
```

styles.scss

cSS

```
1 nav {  
2   background-color: blue;  
3 }  
4 nav li {  
5   float: left;  
6 }  
7 nav a {  
8   display: block;  
9   border: 1px solid black;  
10  text-decoration: none;  
11  padding: 2px;  
12 }
```

styles.css

<http://codepen.io/sawmac/pen/nbGuA>

# Nested Rules

ScSS

```
1 .special {  
2   color: red;  
3   &:visited {  
4     color: yellow;  
5   }  
6   &:hover {  
7     color: blue;  
8   }  
9 }
```

cSS

```
1 .special {  
2   color: red;  
3 }  
4 .special:visited {  
5   color: yellow;  
6 }  
7 .special:hover {  
8   color: blue;  
9 }
```

& refers to parent

<http://codepen.io/sawmac/pen/EKLgo>

# Mixins

ScSS

```
1 @mixin transform($transforms) {  
2     -moz-transform: $transforms;  
3     -o-transform: $transforms;  
4     -ms-transform: $transforms;  
5     -webkit-transform: $transforms;  
6     transform: $transforms;  
7 }  
8 .box {  
9     @include transform(rotate(1deg));  
10    background-color: blue;  
11 }
```

cSS

```
1 .box {  
2     -moz-transform: rotate(1deg);  
3     -o-transform: rotate(1deg);  
4     -ms-transform: rotate(1deg);  
5     -webkit-transform: rotate(1deg);  
6     transform: rotate(1deg);  
7     background-color: blue;  
8 }
```

styles.scss

styles.css

<http://codepen.io/sawmac/pen/yBdvK>

# Mixins

```
@mixin mixin-name($arg) {  
    -o-blah: $arg;  
    -moz-blah: $arg;  
    -webkit-blah: $arg;  
    blah: $arg;  
}
```

```
.style {  
    @include mixin-name(stuff);  
}
```

create the mixin

use the mixin

```
.style {  
    -o-blah: stuff;  
    -moz-blah: stuff;  
    -webkit-blah: stuff;  
    blah: stuff;  
}
```

the results

# Transform Mixins

create the mixin

```
@mixin transform($transforms) {  
    -moz-transform: $transforms;  
    -o-transform: $transforms;  
    -ms-transform: $transforms;  
    -webkit-transform: $transforms;  
    transform: $transforms;  
}
```

```
.box {  
    @include transform(rotate(1deg));  
    background-color: blue;  
}
```

use the mixin

# Clearfloat Mixins

ScSS

```
1 @mixin clearfloat() {  
2   zoom: 1; // for ie6-7  
3   &:before, &:after {  
4     content: " ";  
5     display: table;  
6   }  
7   &:after {  
8     clear: both;  
9   }  
10 }  
11 .box {  
12   @include clearfloat();  
13 }
```

cSS

```
1 .box {  
2   zoom: 1;  
3 }  
4 .box:before, .box:after {  
5   content: " ";  
6   display: table;  
7 }  
8 .box:after {  
9   clear: both;  
10 }
```

<http://nicolasgallagher.com/micro-clearfix-hack/>



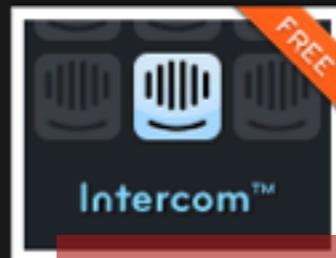
**Compass** is an open-source *CSS Authoring Framework.*

### ❤ Why designers love Compass.

1. Experience cleaner markup without presentational classes.
2. It's chock full of the web's best reusable patterns.
3. It makes creating sprites a breeze.
4. Compass mixins make CSS3 easy.
5. Create beautiful typographic rhythms.
6. Download and create extensions with ease.

### ✓ Compass uses Sass.

Sass is an extension of CSS3 which adds nested rules, variables, mixins, selector inheritance, and more. Sass generates well formatted CSS and makes your stylesheets easier to organize and maintain.



Thousands of web businesses use Intercom to convert signups to paying customers.

[Powered by Fusion](#)

Brilliant people use Compass, including these *wildly talented* folks:

<http://compass-style.org>



# A simple and lightweight mixin library for Sass.

```
ol {
  li {
    background-color: #F4F5F7;
    @include background-image( linear-gradient(#FFF, #F4F5F7) );
    float: left;
    font-family: $lucida-grande;
    position: relative;
    @include transition( all 0.8s );

    &::after {
      content: " ";
      height: 23px;
      @include position( absolute, 10px -13px 0 0 );
      @include transform( rotate(45deg) skew(20deg, 30deg) );
      width: 23px;
    }
  }
}
```

CSS Output

```
background-image: -webkit-linear-gradient(top, #FFF, #F4F5F7);
background-image: -moz-linear-gradient(top, #FFF, #F4F5F7);
background-image: -ms-linear-gradient(top, #FFF, #F4F5F7);
background-image: -o-linear-gradient(top, #FFF, #F4F5F7);
background-image: linear-gradient(top, #FFF, #F4F5F7);
```

Forget about vendor prefixes.

The mixins contain up-to-date vendor prefixes for support in modern browsers.

<http://bourbon.io>

# Extend

ScSS

```
1 .error {  
2   background-color: red;  
3 }  
4 .seriousError {  
5   @extend .error;  
6   border: 3px solid green;  
7 }
```

cSS

```
1 .error, .seriousError {  
2   background-color: red;  
3 }  
4 .seriousError {  
5   border: 3px solid green;  
6 }
```

ui elements with shared visual properties

buttons, alert types, form fields

<http://codepen.io/sawmac/pen/rucne>

# Extend

ScSS

```
1 .error {  
2   background-color: red;  
3 }  
4 .seriousError {  
5   @extend .error;  
6   border: 3px solid green;  
7 }  
8 .totalFailure {  
9   @extend .seriousError;  
10  border-width: 10px;  
11  text-decoration: blink;  
12 }
```

cSS

```
1 .error, .seriousError, .totalFailure {  
2   background-color: red;  
3 }  
4 .seriousError, .totalFailure {  
5   border: 3px solid green;  
6 }  
7 .totalFailure {  
8   border-width: 10px;  
9   text-decoration: blink;  
10 }
```

<http://codepen.io/sawmac/pen/bEagq>

# CSS Transitions

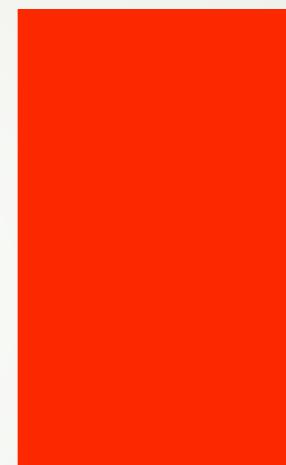
Animating between CSS properties



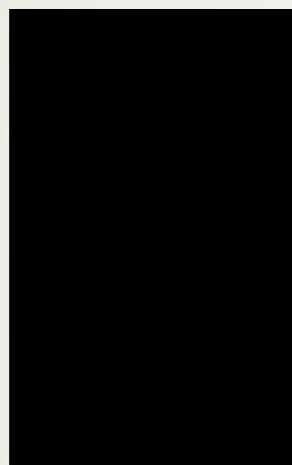
```
nav a {  
    background-color: black  
}
```



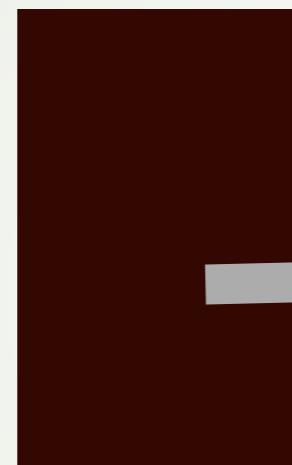
```
nav a:hover {  
    background-color: red;  
}
```



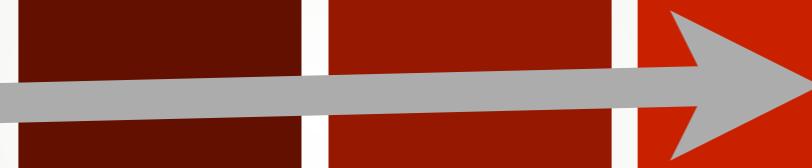
keyframe



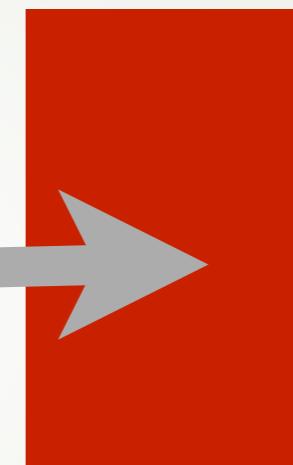
nav a



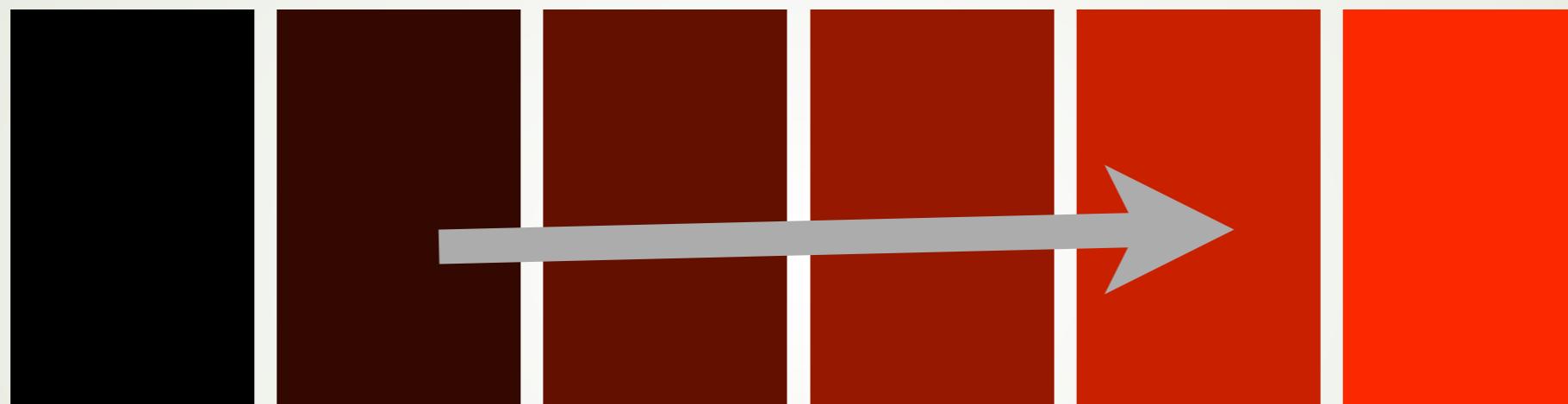
tween



keyframe



nav a:hover



nav a

you

↑  
browser

nav a:hover

you

# Step 1: Create First Keyframe

```
nav a {  
    background-color: black;  
}
```

## Step 2: Add Transition

```
nav a {  
  background-color: black;  
  transition-property: background-color;  
  transition-duration: .75s;  
}
```

add transition to the first keyframe

## Step 3: Add Last Keyframe

```
nav a:hover {  
    background-color: red;  
}
```

# transition-property

transition-property: background-color;

property to animate



transition-property: background-color, color, height;

multiple properties  
(comma separated)



transition-property: all;

animate all properties



# transition-property

```
-moz-transition-property: background-color;  
-o-transition-property: background-color;  
-webkit-transition-property: background-color;  
transition-property: background-color;
```

requires vendor prefixes

don't need a -ms prefix, since IE10 supports transitions without

# animatable properties

background-color

background-position

border-color

border-width

border-spacing

bottom

clip

color

font-size

font-weight

height

left

letter-spacing

line-height

margin-top, -right,  
-bottom, -left

max-height

max-width

min-height

min-width

opacity

outline-color

outline-width

padding-top, -right,  
-bottom, -left

right

text-indent

text-shadow

top

vertical-align

visibility

width

word-spacing

z-index

<http://www.w3.org/TR/css3-transitions/#animatable-properties->

# Animatable Properties



**CSS Property**

background-color ▾

**Value**

**Animate it!**

[basic-transitions/animatable-properties.html](#)

# transition-duration

transition-duration: `1s;`

in seconds



transition-duration: `1000ms;`

or milliseconds



requires vendor prefixes

# transition-timing-function

how the transition animates

transition-timing-function: **linear**;

boring 

**ease**

linear

ease-in

default 

ease-out

ease-in-out

step-start

step-end

steps(4, start)

cubic-bezier(.42,0,.58,1)

requires vendor prefixes

# transition-timing-function

5s duration

ease

linear

ease-in

ease-in-out

cubic-

step(5,end)

step(5,start)

step-start

step-end

basic-transitions/rollover-timing.html



## Examples

transition-timing-function:  
ease



transition-timing-function:  
ease-in



transition-timing-function:  
ease-out



transition-timing-function:  
ease-in-out



transition-timing-function:  
linear



transition-timing-function:  
step-start



transition-timing-function:  
step-end



transition-timing-function:  
steps(4, end)



## Specifications

Specification	Status	Comment
<a href="#">CSS Transitions</a>	Working Draft	

## Browser compatibility

Desktop	Mobile				
Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari
Basic support	(Yes) <code>-webkit-</code>	4.0 (2.0) <code>-moz-</code> 16.0 (16.0)	10	11.6 <code>-o-</code> 12.10 #	(Yes) <code>-webkit-</code>

## See also

<https://developer.mozilla.org/en-US/docs/CSS/transition-timing-function>

- [Using CSS transitions](#)
- [TransitionEvent](#)
- [Transitions-related properties: transition, transition-delay, transition-duration, transition-property, transition-timing-function](#)

1. Choose an easing type and test it out with a few effects.
2. If you don't quite like the easing, grab a handle and fix it.
3. When you're happy, snag your code and off you go.

Now that we can use CSS transitions in all the modern browsers, let's make them pretty. I love the classic Penner equations with Flash and jQuery, so I included most of those. If you're anything like me\*, you probably thought this about the default easing options: "ease-in, ease-out...yawn." The mysterious cubic-bezier has a lot of potential, but was cumbersome to use. Until now. Also, touch-device friendly!

\*If you are anything like me, we should be friends [@matthewlein](#)

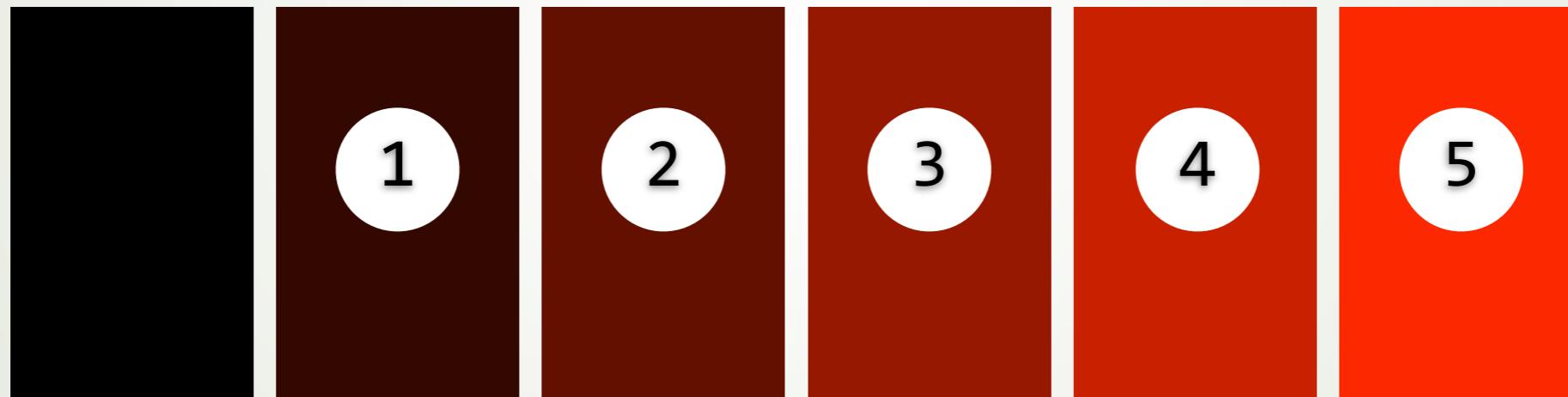
**Note:** Bugfixes have landed, so the newest Webkit now supports values above 1 and below 0. For the time being, I am including fallback code for older Webkit that is clamped between 0 and 1 when needed.



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

# steps

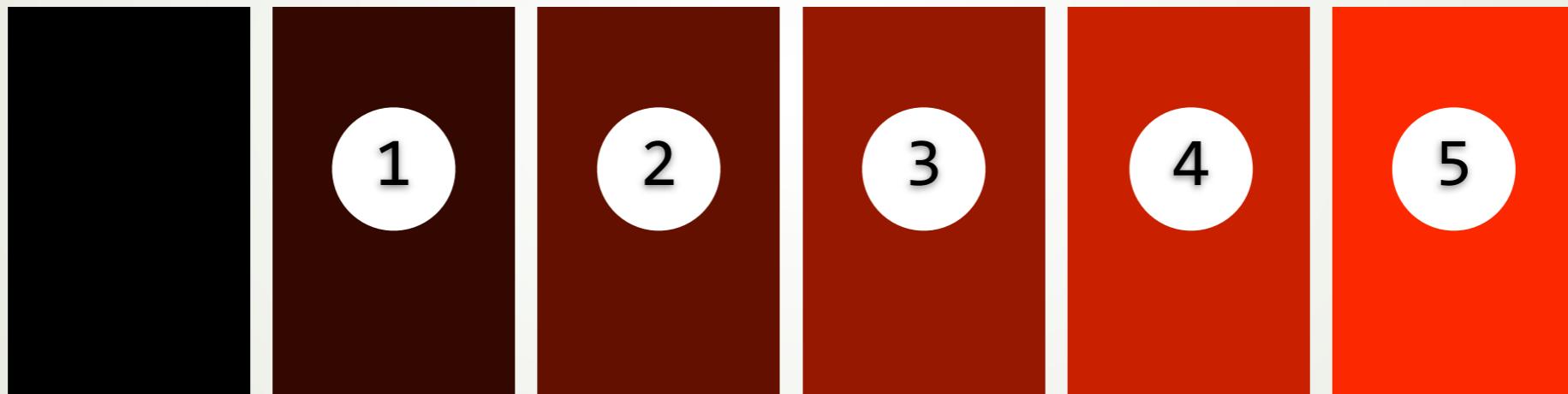
```
transition-property: background-color;  
transition-duration: 5s;  
transition-timing-function: steps(5,end);
```



1 second delay

# steps

```
transition-property: background-color;  
transition-duration: 5s;  
transition-timing-function: steps(5,start);
```



no delay

entire animation is actually only 4 seconds long!

# steps

```
transition-property: background-color;  
transition-duration: 5s;  
transition-timing-function: step-start;
```

useless? 



# steps

```
transition-property: background-color;  
transition-duration: 5s;  
transition-timing-function: step-end;
```

wait 5 seconds 



# transition-delay

wait before you start the animation

transition-delay: `1s`;

wait 1 second

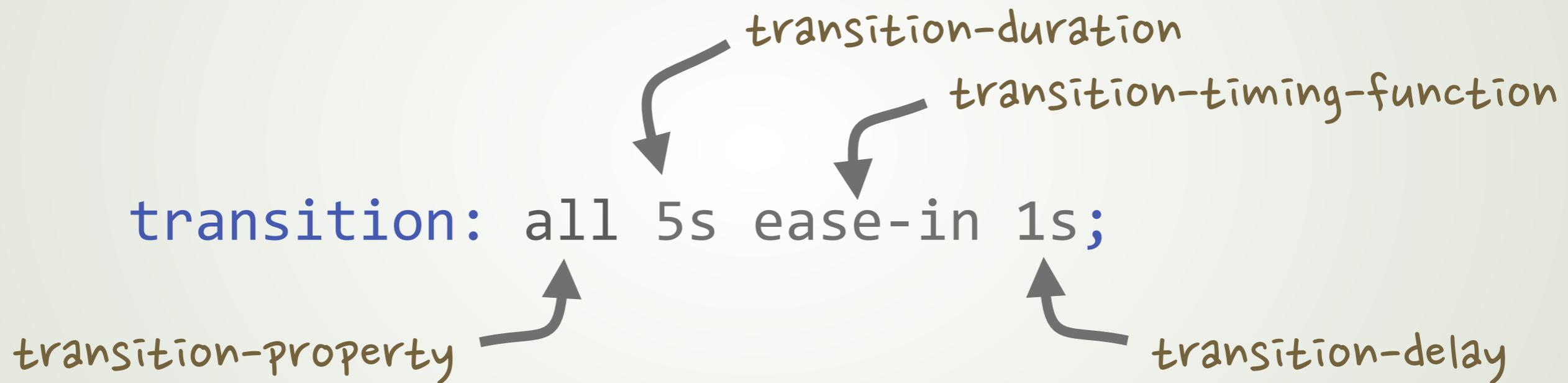
transition-delay: `1000ms`;

...in milliseconds

requires vendor prefixes

# transition shorthand

way easier to combine all properties into one!



# transition shorthand

requires vendor prefixes

```
-moz-transition: all 5s ease-in 1s;  
-o-transition: all 5s ease-in 1s;  
-webkit-transition: all 5s ease-in 1s;  
transition: all 5s ease-in 1s;
```

# Transition Mixin

ScSS

```
1 @mixin transition($transition...) {  
2   -moz-transition: $transition;  
3   -o-transition: $transition;  
4   -webkit-transition: $transition;  
5   transition: $transition;  
6 }  
7 .box {  
8   @include transition(all 1s);  
9   background-color: blue;  
10 }
```

cSS

```
1 .box {  
2   -moz-transition: all 1s;  
3   -o-transition: all 1s;  
4   -webkit-transition: all 1s;  
5   transition: all 1s;  
6   background-color: blue;  
7 }
```

styles.scss

styles.css

<http://codepen.io/sawmac/pen/cayhK>

# multiple transitions

can have separate timings per property

```
transition: background-color 5s ease-in 1s,  
           color 1s ease-in-out,  
           font-size .5s;
```

or

```
transition-property: background-color, color, font-size;  
transition-duration: 5s, 1s, .5s;  
transition-timing-function: ease-in, ease-in-out, ease;  
transition-delay: 1s, 0, 0;
```

# transition triggers

- ▶ :hover
- ▶ :focus
- ▶ :active
- ▶ :target
- ▶ JavaScript

# transition triggers

:hover

hover over this div



:active

click on this div and hold mouse button down

:target

[click on this link](#)

:focus

click or tab into the text field below

JavaScript

[click on me](#)

[basic-transitions/transition-triggers.html](#)

# :hover

```
nav a {  
    background-color: black;  
    transition: all 1s;  
}  
  
nav a:hover {  
    background-color: red;  
}
```

# :active

only while you are clicking on element

```
div {  
    background-color: black;  
    transition: all 1s;  
}
```

```
div:active {  
    background-color: red;  
}
```

not limited to links!

# :focus

```
input[type='text'] {  
  background-color: black;  
  transition: all 1s;  
}
```

```
input[type='text']:focus {  
  background-color: red;  
}
```

when tabbing or clicking into a form field or link

# :target

tricky way to leverage a click

```
div {  
  background-color: black;  
  transition: all 1s;  
}  
  
div:target {  
  background-color: red;  
}
```

# What is a target?

`<a href="#an-id">Click to go to target</a>`

`<div id="an-id">This is the target when you click on the link above</div>`

browser location bar before clicking the link

`http://.../somepage.html`

and after clicking the link

`http://.../somepage.html#an-id`

Sign up today

# :target trigger

basic-transitions/signup-target.html

# Problems with :target

- ▶ Requires a link to activate
- ▶ changes the URL
- ▶ may scroll the page when you don't want to

# Triggering Transitions with JavaScript

- ▶ any change to the specified transition properties animate
- ▶ use JavaScript to apply new CSS properties dynamically via click, mouseover, double click -- any javascript event
- ▶ simplest way: apply a class style to the element with new CSS properties

# Using jQuery

```
<style>
div {
    background-color: #FF0;
    transition: all 1s;
}
.change {
    background-color: #00F;
}
</style>
<script src="jquery.js"></script>
<script>
$(document).ready(function() {
    $('div').click(function() {
        $(this).addClass('.change');
    }); // end click
}); // end ready
</script>
```

# Using jQuery

```
<style>
div {
    background-color: #FF0;
    transition: all 1s;
}
.change {
    background-color: #00F;
}
</style>
<script src="jquery.js"></script>
<script>
$(document).ready(function() {
    $('div').click(function() {
        $(this).toggleClass('.change');
    }); // end click
}); // end ready
</script>
```

# Revealing Content

Fun ways to reveal hidden content



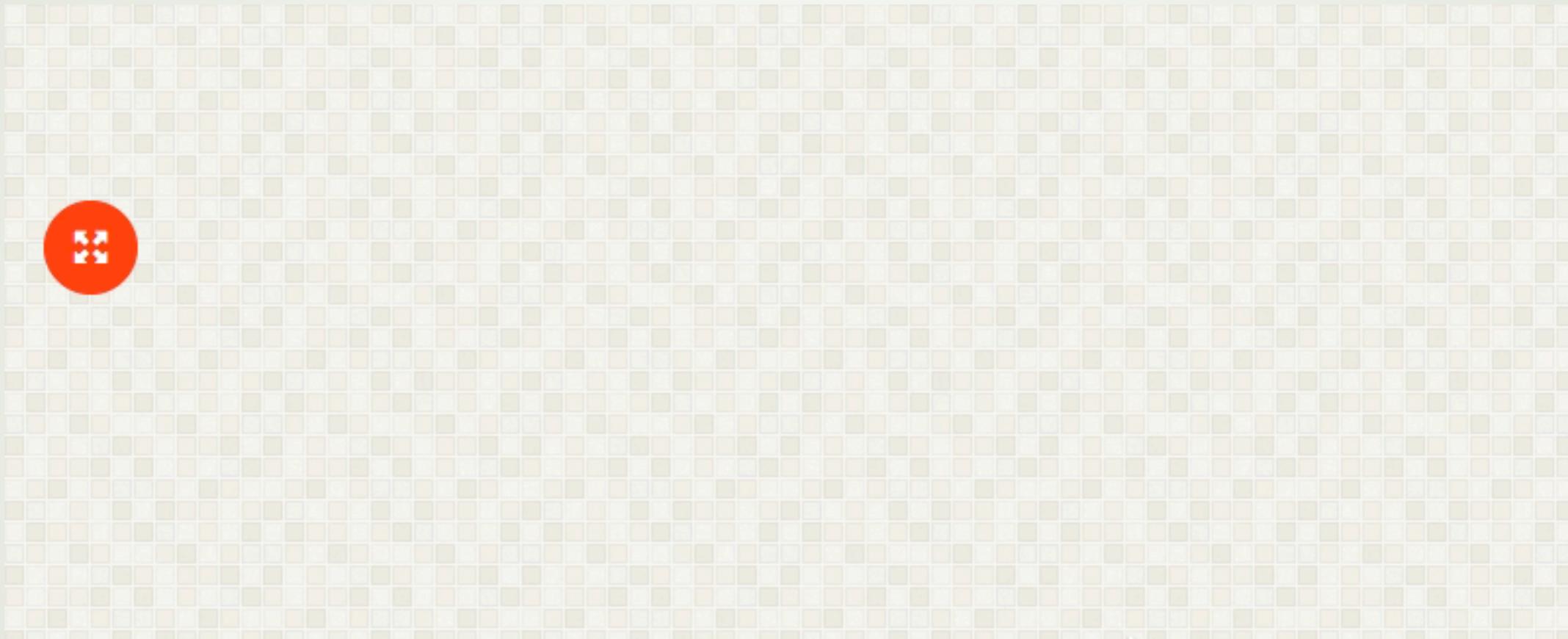
reveals/01-reveal.html

# **Nav Bars**

**Creating drop down menus with CSS3**

# Control Panel

Reveal control panel links on hover



# **CSS3 Modal Dialog**

# CSS Animations

when transitions aren't enough



# CSS Animations

- ▶ Transitions can only animate between two CSS states (2 keyframes)
- ▶ Transitions: blue to red
- ▶ Animations can animate between multiple CSS states (multiple keyframes).
- ▶ Animations: blue to red to green to orange
- ▶ can apply the same animation to multiple elements
- ▶ animation can run when page loads!

# Step 1: Define the animation

keyframes “at-rule”



```
@keyframes myAnimation {  
    from {  
        background-color: black;  
    }  
    to {  
        background-color: red;  
    }  
}
```

whatever you want to call it



requires vendor prefixes

# Step 1: Define the animation

```
@keyframes myAnimation {  
    from {  
        background-color: black;  
    }  
    to {  
        background-color: red;  
    }  
}
```

first  
keyframe

# Step 1: Define the animation

```
@keyframes myAnimation {  
  from {  
    background-color: black;  
  }  
  to {  
    background-color: red;  
  }  
}
```

second  
keyframe

## Step 2: Apply the animation

```
div:hover {  
    animation-name: myAnimation;  
    animation-duration: 1s;  
}
```

requires vendor prefixes

# @keyframes

any number of keyframes

```
@keyframes cycleColor {  
    from {  
        background-color: white;  
    }  
    33% {  
        background-color: green;  
    }  
    66% {  
        background-color: blue;  
    }  
    to {  
        background-color: red;  
    }  
}
```

# @keyframes

requires vendor prefixes

@-moz-keyframes

@-o-keyframes

@-webkit-keyframes

@keyframes

don't worry, I've got a mixin for that!

# properties like transitions

- ▶ animation-duration  
(works just like transition-duration)
- ▶ animation-timing-function  
(works just like transition-timing-function)
- ▶ animation-delay  
(works just like transition-delay)

add these to the element you are animating

# properties like transitions

requires vendor prefixes

```
-moz-animation-duration: .5s;  
-o-animation-duration: .5s;  
-webkit-animation-duration: .5s;  
animation-duration: .5s;
```

```
-moz-animation-timing-function: ease-in-out;  
-o-animation-timing-function: ease-in-out;  
-webkit-animation-timing-function: ease-in-out;  
animation-timing-function: ease-in-out;
```

```
-moz-animation-delay: 1s;  
-o-animation-delay: 1s;  
-webkit-animation-delay: 1s;  
animation-delay: 1s;
```

don't worry, I've got a mixin for that!

# animation-iteration-count

play the animation more than once

animation-iteration-count: 5;

number of times to repeat



animation-iteration-count: infinite;

forevah



requires vendor prefixes

# animation-direction

use with animation-iteration-count

```
animation-iteration-count: 5;  
animation-direction: alternate;
```

plays animation forwards  
and backwards 

requires vendor prefixes

# animation-fill-mode

`animation-fill-mode: none;`

😊 `animation-fill-mode: forwards;`  
stay at the last keyframe after animation

`animation-fill-mode: backwards;`  
first keyframe loads even with delay

`animation-fill-mode: both;`

requires vendor prefixes

# no delay

animation-fill-mode **animation-duration: 5s**

[Start animation \(no delay\)](#)

[Start animation \(5s delay\)](#)

animation-fill-mode: none

animation-fill-mode: forwards

animation-fill-mode: backwards

animation-fill-mode: both

# 5 second delay

animation-fill-mode **animation-duration: 5s**

[Start animation \(no delay\)](#)

[Start animation \(5s delay\)](#)

animation-fill-mode: none

animation-fill-mode: forwards

animation-fill-mode: backwards

animation-fill-mode: both

# animation-play-state

pause the animation

```
div:hover {  
    animation-name: grow;  
    animation-duration: 10s;  
}
```

```
div:hover {  
    animation-play-state: paused;  
}
```

requires vendor prefixes

# animation shorthand

seven properties in 1 !

`animation: <name> <dur> <timing> <count> <dir> <delay> <fill-mode>`

```
@keyframes colorChange {  
  from {  
    color: rgba(10,255,30,.5);  
  }  
  to {  
    color: rgba(255,255,30,1);  
  }  
}  
a:hover {  
  animation: colorChange 5s ease 4 alternate 5s forwards;  
}
```

requires vendor prefixes

# multiple animations

run more than one animation at once!

```
@keyframes colorChange {  
  from { color: rgba(10,255,30,.5); }  
  to { color: rgba(255,255,30,1); }  
}  
@keyframes grow {  
  from { transform: scale(1); }  
  33% { transform: scale(1.5); }  
  to { transform: scale(3); }  
}  
a:hover {  
  animation: colorChange 5s ease 4 alternate 5s forwards,  
            grow 5s ease-in-out 2;  
}
```

requires vendor prefixes

**Let's Build an  
Animation**