

Move It!

CSS3 Transitions, Transforms & Animations

Stephanie (Sullivan) Rewis

- Front-end developer
- Principal w3conversions.com
- WaSP corporate liaison to Adobe
- Corporate Trainer & Consultant
- Author & Speaker
- Twitter addict (@stefsull)

What is movement used for?

- Mobile Web Apps
- Banner Ads
- Enhancing User Interfaces
- HTML5 Immersive Sites

Choices for movement

- Flash/Silverlight or plugin
- jQuery/Javascript
- CSS3
- A Combination framework

Why use CSS for movement?

- Less reliance on plugins
- Less Javascript
 - Can mean less complexity for designers
 - Faster performance when the browser natively renders transitions & animations instead of manipulating the DOM with Javascript

The Importance of Speed

- Google has shown that seemingly minor increases in latency directly impact user engagement and that major increases in latency (> 500ms) can reduce traffic and revenue significantly.

—Alex Russell (Chrome Developer)

CSS3 Transforms - Working Draft

Global user stats*:
Support: 53.47%

Method of transforming an element including rotating, scaling, etc.

Resources: [Live editor](#) [MDN article](#) [Workaround script for IE](#)

	IE	Firefox	Safari	Chrome	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
Two versions back	7.0	3.5 <small>-maz-</small>	3.2 <small>-webkit-</small>	9.0 <small>-webkit-</small>	10.6 <small>-a-</small>	3.2 <small>-webkit-</small>			2.1 <small>-webkit-</small>
Previous version	8.0	3.6 <small>-maz-</small>	4.0 <small>-webkit-</small>	10.0 <small>-webkit-</small>	11.0 <small>-a-</small>	4.0-4.1 <small>-webkit-</small>		10.0	2.2 <small>-webkit-</small>
Current	9.0 <small>-ms-</small>	4.0 <small>-maz-</small>	5.0 <small>-webkit-</small>	11.0 <small>-webkit-</small>	11.1 <small>-a-</small>	4.2-4.3 <small>-webkit-</small>	5.0-6.0	11.0	2.3 <small>-webkit-</small> 3.0 <small>-webkit-</small>
Near future		5.0 <small>-maz-</small>		12.0 <small>-webkit-</small>	11.5 <small>-a-</small>				
Farther future	10.0 <small>-ms-</small>	6.0 <small>-maz-</small>	6.0 <small>-webkit-</small>	13.0 <small>-webkit-</small>	12.0 <small>-a-</small>				

Note: Current implementations may change significantly from final specification, since W3C adoption is quite recent. The scale transform can be emulated in IE < 9 using Microsoft's "zoom" extension, others are (not easily) possible using the MS Matrix filter

[Feedback](#)

Transforms



CSS Transforms

- “The 2D Transformations Module defines a property that applies rotations, translations and other **affine** transformations to a box.”
- affine |ə'fīn; 'afīn|
adjective Mathematics
allowing for or preserving parallel relationships.

CSS Transforms

- In the 2D space, think of the transform controls in your favorite graphics program
 - Scale
 - Rotate
 - Skew
 - Translate

If only it were that simple...

- As individual browsers add support for transforms (and transitions), they use **vendor-specific prefixes**.
- You should list the vendor specific implementations first. The actual CSS property (without a browser prefix) should **always be last**.
- CSSPrefixer

Ready When You Are

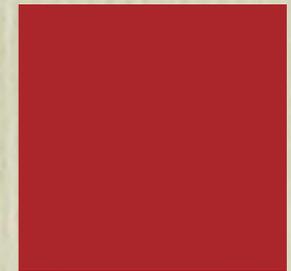
- CSS3 Transforms can be applied to any element —which now includes generated content
- Transforms are applied immediately at the time the page is rendered or triggered via a pseudo class or Javascript

transform: scale() *

```
.square {  
  height: 250px;  
  width: 250px;  
  background-color: rgb(171, 38, 41);  
}
```

```
.square:hover {  
  transform: scale(2);  
}
```

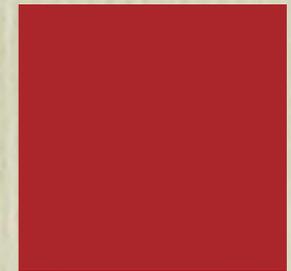
*scaleX and scaleY are also possible



transform: scale() in reality

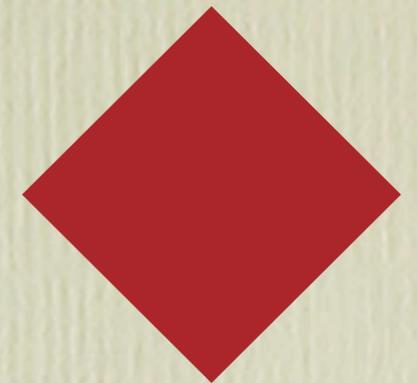
```
.square {  
  height: 250px;  
  width: 250px;  
  background-color: rgb(171, 38, 41);  
}
```

```
.square:hover {  
  -webkit-transform: scale(2);  
  -moz-transform: scale(2);  
  -ms-transform: scale(2);  
  -o-transform: scale(2);  
  transform: scale(2);  
}
```



transform: rotate() & transform: skew()*

```
.square {  
  height: 250px;  
  width: 250px;  
  background-color: rgb(171, 38, 41);  
  transform: rotate(45deg);  
}
```



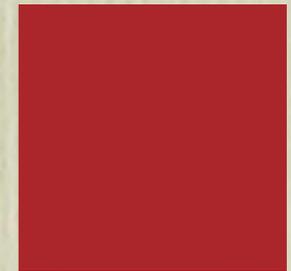
```
.square {  
  height: 250px;  
  width: 250px;  
  background-color: rgb(171, 38, 41);  
  transform: skew(45deg);  
}
```



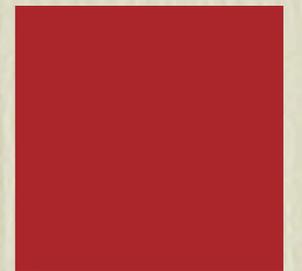
***skewX** and **skewY** are also possible
[browser specific variations intentionally omitted]

transform: translate()

```
.square {  
  height: 250px;  
  width: 250px;  
  background-color: rgb(171, 38, 41);  
}
```



```
.square:hover {  
  transform: translate(50px, 50px);  
}
```



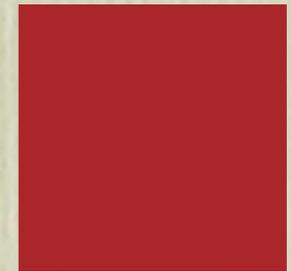
***translateX and translateY** are also possible
[browser specific variations intentionally omitted]

multiple transforms*

```
.square {  
  height: 250px;  
  width: 250px;  
  background-color: rgb(171, 38, 41);  
}
```

```
.square:hover {  
  transform: scale(2) rotate(45deg);  
}
```

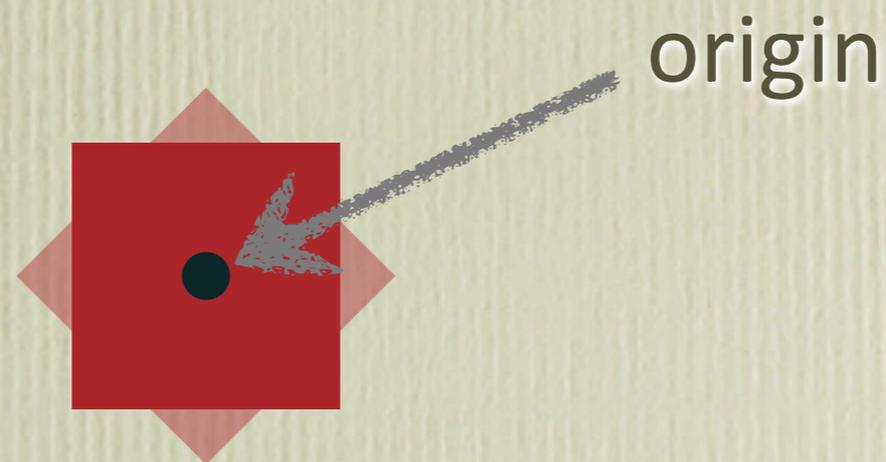
*all properties are applied **simultaneously**
[browser specific variations intentionally omitted]



Transform-origin

- By default, all transforms occur “around” the **center of the object** ($x=50%$, $y=50%$)

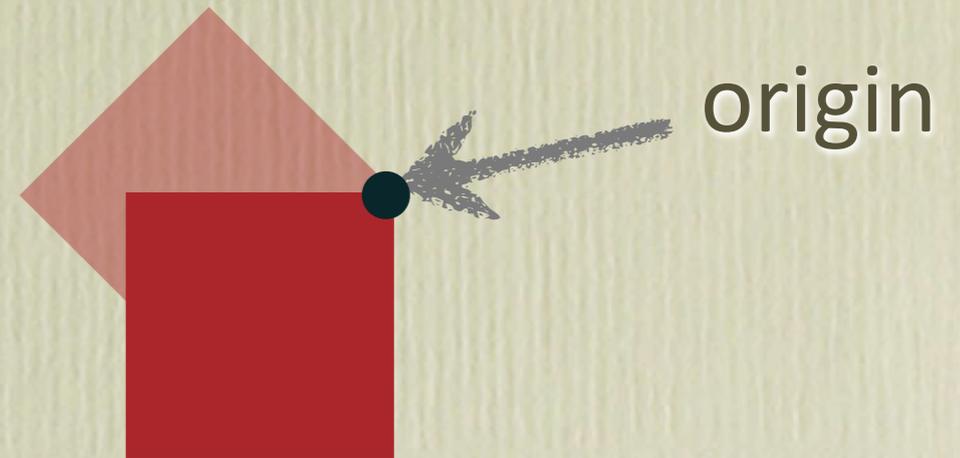
```
.square:hover {  
  transform: rotate(45deg);  
}
```



Transform-origin

```
.square:hover {  
  transform: rotate(45deg);  
  transform-origin: right top;  
}
```

- left, center, right and top, center, bottom



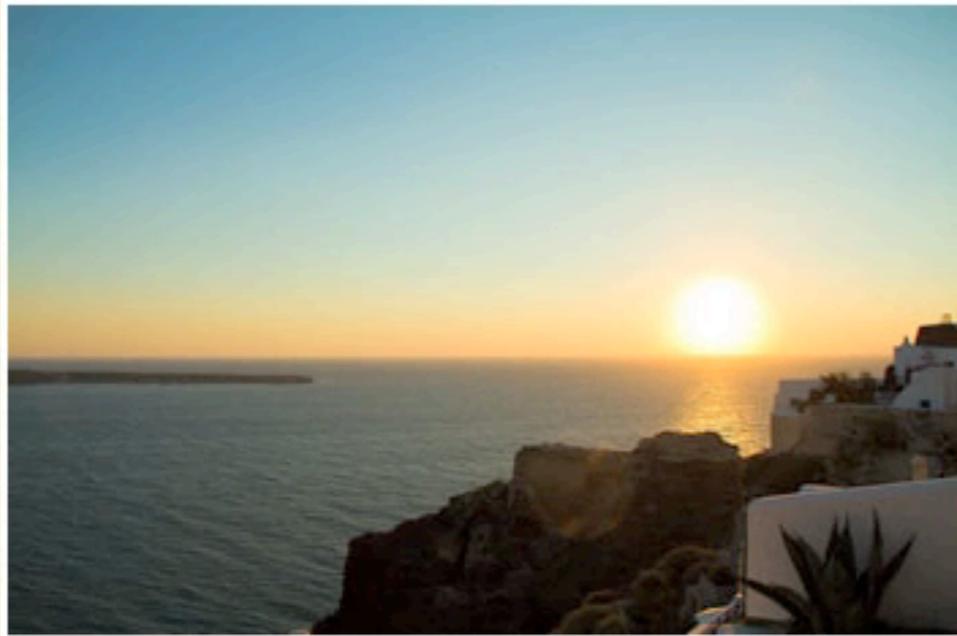
```
.square:hover {  
  transform: rotate(45deg);  
  transform-origin: 100% 100%;  
}
```

- the point of origin can even be **outside** of the object itself



[browser specific variations intentionally omitted]

demo :: transform



A cube created with CSS3 Transforms and cssSandpaper. Go back to the User Agent Man CSS3 article.

This design is based on one by Paul Hayes and uses the Graublau Web and Droid Sans fonts.

User Agent Man

Discoveries of a Client Side Web Developer

What is This?

This is the blog of Zoltan "Du Lac" Hawryluk. It contains mostly random (and not-so-random) thoughts on client side web technology. Feel free to use any of the ideas presented for your own purposes, and if you have anything to add, please share. Please forgive him if he rambles on a bit. Or if he refers to himself in the third person.

Categories

- General Announcements
- Technologies
 - CSS
 - Fonts
 - HTML
 - HTML5
- Images
- JavaScript
- XML
 - SVG

Recent Articles:

Advanced Cross-Browser CSS3 — Even in IE!
The last month and a bit has been spent working on working on a method to help developers use advanced CSS3 effects without using multiple CSS properties. [Read More...](#)

Cross Browser HTML5 Drag



Yes, this is a scrollbar, and yes, it works. If you are using an older version of Chrome or Safari, you may not see the scrollbars, but they do work. Unfortunately, it doesn't appear in Opera :-)

Feature Testing

[Home](#)[News](#)[Docs](#)[Sites](#)

Latest release:

1.7

FEBRUARY 20: Version 1.7 nerfs the WebGL/Chrome9 issue, Opera color input fix, faster (no release notes yet).

What is Modernizr?

Modernizr adds classes to the `<html>` element which allow you to target specific browser functionality in your stylesheet. You don't actually need to write any Javascript to use it.

Have you ever wanted to do if-statements in your CSS for the availability of cool features like `border-radius`? Well, with **Modernizr** you can accomplish just that! The syntax is very intuitive, too:

```
.multiplebgs div p {  
  /* properties for browsers that  
  support multiple backgrounds */  
}  
.no-multiplebgs div p {  
  /* optional fallback properties  
  for browsers that don't */  
}
```

Modernizr detects support for:

- @font-face ✓
- Canvas ✓
- Canvas Text ✓
- HTML5 Audio ✓
- HTML5 Video ✓
- rgba() ✓
- hsla() ✓
- border-image: ✓
- border-radius: ✓
- box-shadow: ✓
- text-shadow: ✓
- opacity: ✓
- Multiple backgrounds ✓
- Flexible Box Model ✓
- CSS Animations ✓
- CSS Columns ✓
- CSS Gradients ✓
- CSS Reflections ✓
- CSS 2D Transforms ✓
- CSS 3D Transforms
- CSS Transitions ✓
- Geolocation API ✓
- localStorage ✓
- sessionStorage ✓
- SVG ✓
- SMIL ✓
- SVG Clipping ✓
- Inline SVG ✓
- Drag and Drop ✓
- hashchange ✓
- X-window Messaging ✓
- History Management ✓
- applicationCache ✓
- Touch events
- Web Sockets ✓
- Web Workers ✓
- Web SQL Database ✓
- WebGL ✓
- IndexedDB ✓
- Input Types†
- Input Attributes‡

versions

Modernizr - Feature Detection

- Link to the script & add `<html class="no-js">`
- Modernizr changes the no-js class to a long string of classes that you can use to serve different CSS or JS to a browser depending on its lack of capabilities. ie:

```
<html class=" js canvas canvastext geolocation  
crosswindowmessaging no-websqldatabase no-indexeddb  
hashchange no-historymanagement draganddrop no-websockets  
rgba hsla multiplebgs backgroundsize borderimage borderradius  
boxshadow opacity no-cssanimations csscolumns cssgradients no-  
cssreflections csstransforms no-csstransforms3d no-csstransitions  
video audio localstorage sessionstorage webworkers  
applicationcache svg no-smilsvgclippaths fontface">
```

CSS based on abilities

```
.no-csstransforms #foo #soo {  
  display:none;  
}
```

```
.no-csstransforms #foo:hover #soo {  
  display:block;  
}
```

Or write CSS so it's only shown to capable browsers:

```
.csstransforms #foo {  
  transform: rotate(10deg);
```

CSS3 Transitions - Working Draft

Global user stats* :
Support: **31.62%**

Simple method of animating certain properties of an element

Resources: [Article on usage](#) [Webkit blog post](#) [Examples on timing functions](#)

	IE	Firefox	Safari	Chrome	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
Two versions back	7.0	3.5	3.2 <small>-webkit-</small>	9.0 <small>-webkit-</small>	10.6 <small>-a-</small>	3.2 <small>-webkit-</small>			2.1 <small>-webkit-</small>
Previous version	8.0	3.6	4.0 <small>-webkit-</small>	10.0 <small>-webkit-</small>	11.0 <small>-a-</small>	4.0-4.1 <small>-webkit-</small>		10.0 <small>-a-</small>	2.2 <small>-webkit-</small>
Current	9.0	4.0 <small>-maz-</small>	5.0 <small>-webkit-</small>	11.0 <small>-webkit-</small>	11.1 <small>-a-</small>	4.2-4.3 <small>-webkit-</small>	5.0-6.0	11.0 <small>-a-</small>	2.3 <small>-webkit-</small> 3.0 <small>-webkit-</small>
Near future		5.0 <small>-maz-</small>		12.0 <small>-webkit-</small>	11.5 <small>-a-</small>				
Farther future	10.0 <small>-ms-</small>	6.0 <small>-maz-</small>	6.0 <small>-webkit-</small>	13.0 <small>-webkit-</small>	12.0 <small>-a-</small>				

Feedback

Transitions



CSS Transitions

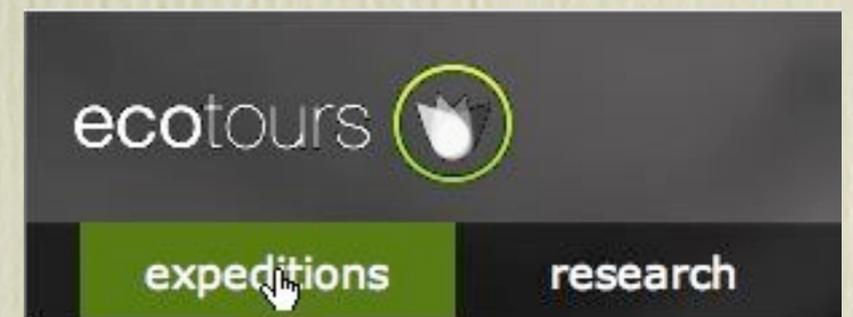
- CSS Transitions allows property changes in CSS values to occur smoothly over a specified duration.
- Transitions can be thought of as animations in their simplest form.
- Only animatable CSS properties can be transitioned.

:hover - the simplest transition

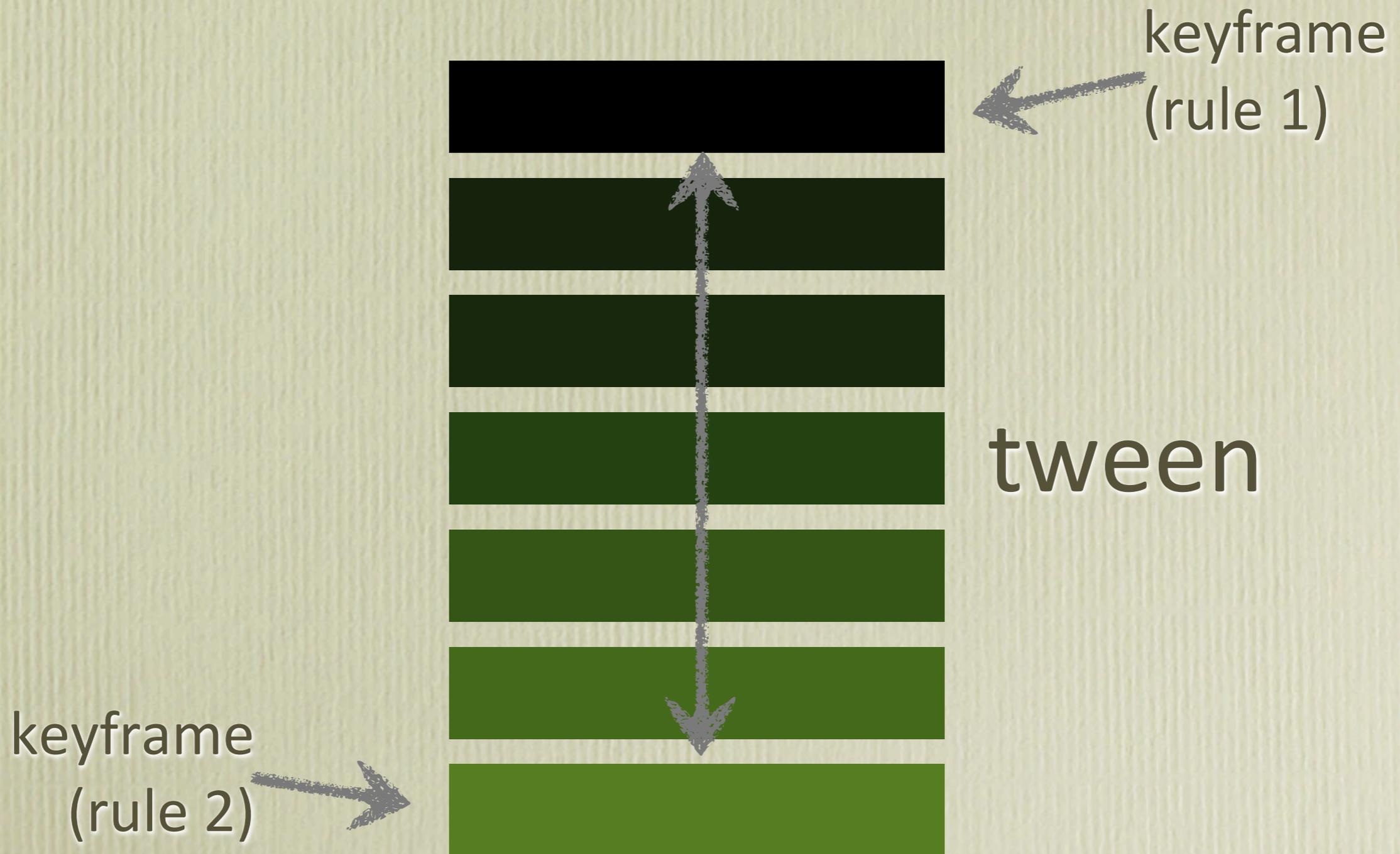
```
nav li a {  
  background-color: rgb(0,0,0);  
}
```



```
nav li a:hover {  
  background-color: rgb(86, 125,33);  
}
```



Anatomy of a Transition



Anatomy of a Transition

- Basic styling of the element in rule 1
 - It may contain properties that won't be modified
 - The properties to be modified are identified with the **transition property**
- Rule 2 contains the modifications to the element & uses a pseudo-class (:hover, :focus, :target)
- **OR** Javascript can also be used to trigger a transition

:hover - the simplest transition

```
nav li a {  
  background-color: rgb(0,0,0);  
  transition-property: background-color;  
  transition-duration: 1s;  
}
```

```
nav li a:hover {  
  background-color: rgb(86, 125,33);  
}
```



Property Listings

- In the previous example, only a single property was transitioned:
`transition-property: background-color;`
- Multiple properties can be defined at one time:
`transition-property: background-color, padding, font-size, color;`
`transition-property: all;`

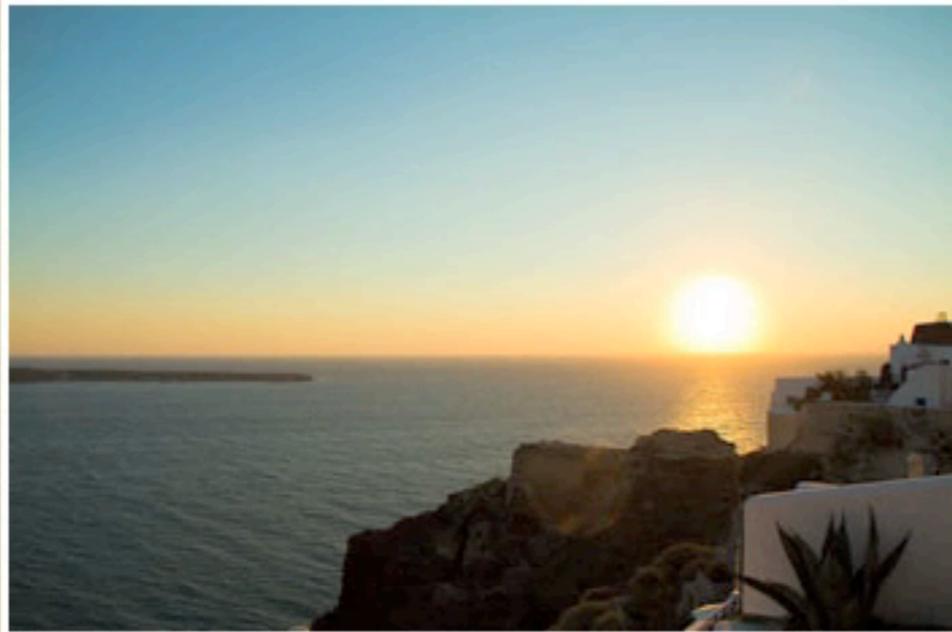
Animatable Properties

- background-color
- background-image (gradients)
- background-position
- background-size
- border (-color, -width)
- border-radius
- border-spacing
- box-flex
- box-shadow
- color
- column (-count, -gap, -width)
- clip
- fill
- font-size
- font-weight
- height
- letter-spacing
- line-height
- margin (-top, -right, -bottom, -left)
- max- / min-height
- max- / min-width
- opacity
- outline (color, width)
- outline-offset
- padding (-top, -right, -bottom, -left)
- position (top, right, bottom, left)
- stroke (-opacity, -width)
- text-indent
- text-shadow
- transform
- transform-origin
- vertical-align
- visibility
- width
- word-spacing
- z-index

Time it Right!

- The duration of the transition is determined by the **transition-duration** property.
- Time can be designated in seconds or milliseconds
 - transition-duration: 2s;
 - transition-duration: 500ms;
- Time is not precise and can vary slightly between browsers

demo :: transition



A Very Good Day

Nullam sed condimentum nisi. In hendrerit pharetra lectus, eget viverra ante tempor quis. Phasellus massa odio, vehicula et pretium ac, ultrices ac leo. Aenean gravida tristique eleifend. Maecenas blandit enim eu tellus dictum vitae consetetur neque aliquam. Duis non ligula nibh. Fusce interdum feugiat urna a venenatis. Suspendisse lacus libero, aliquet nec ultrices vel, mattis ut eros. Aenean feugiat ultrices tellus, eget varius risus iaculis vitae. Nunc eleifend luctus convallis. Ut convallis, mauris eget fermentum tincidunt, felis nisl pretium libero, vel eleifend nisl nunc eget turpis. Vestibulum sagittis vehicula condimentum.

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Praesent eros arcu, lacinia eu dictum vel, pharetra vel odio. Mauris pellentesque laoreet feugiat. Nulla facilisi. Pellentesque fringilla metus diam, at vulputate risus. In ligula turpis, rutrum in venenatis eget, mattis et nunc. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Vestibulum consequat, lorem quis ultricies

suscipit tortor nunc in lectus. Donec id tortor eget velit vulputate porta. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nulla facilisi. Nam rutrum, massa sed cursus interdum, orci orci sagittis est, id vulputate nibh enim in leo.

Fusce arcu velit, iaculis eget porttitor nec, volutpat nec purus. Nam molestie auctor nisl quis tempor. Nulla rhoncus egestas egestas. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Duis a mauris magna, vel eleifend velit. Donec rutrum aliquam gravida. Aliquam rhoncus purus turpis, at tempus leo. Integer ante urna, laoreet at vestibulum id, ullamcorper eget dui. Phasellus diam enim, tincidunt lobortis pharetra et, suscipit ut augue. Pellentesque ullamcorper, augue nec tristique mattis, elit leo volutpat quam, in venenatis metus tellus at lorem. Maecenas mi quam, ultrices quis consequat sed, malesuada at diam. Maecenas pellentesque lobortis erat sed vehicula. Pellentesque non mauris sit amet risus euismod vestibulum a nec ante. Curabitur non nisl nisi, ac rutrum orci. Curabitur id nisl lectus, in tincidunt est. Maecenas eget sem rutrum urna cursus luctus viverra quis nulla.

Transition Speed

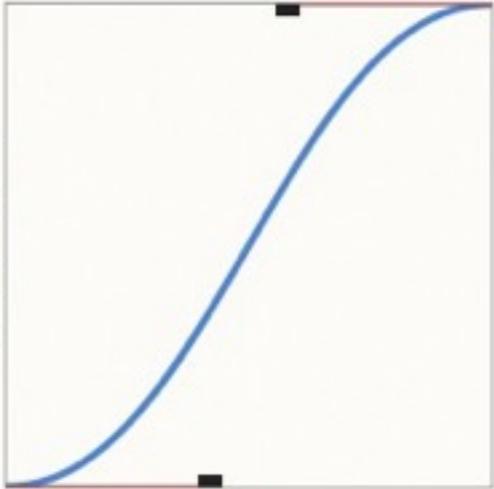
- By default, transitions play at a **constant rate** (called linear).
- The **transition-timing-function** allows you to specify a variable rate of speed for the transition
 - linear
 - ease
 - ease-in
 - ease-out
 - ease-in-out
 - cubic-bezier(0,.33,1,.75)
- **transition-timing-function: ease;**

Transition-timing

Demo - Different timing functions



Ceasar - Easing tool



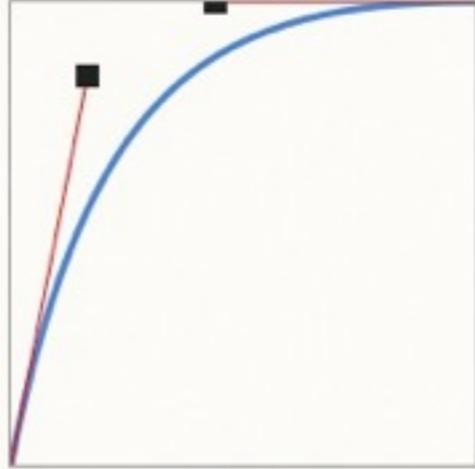
Animation (%)

Time

Easing:

Duration:

Effect:



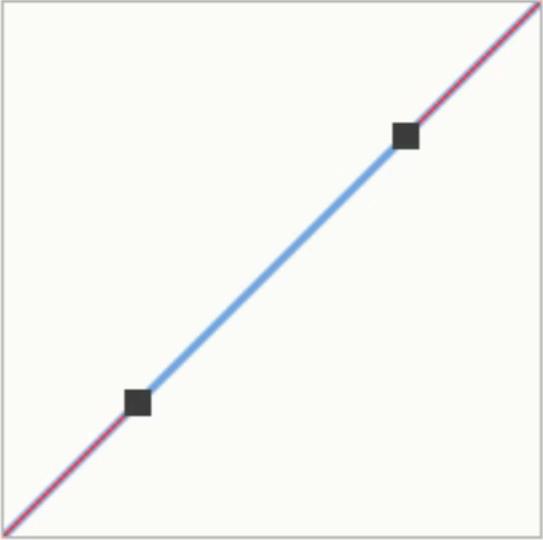
Animation (%)

Time

Easing:

Duration:

Effect:



Animation (%)

Time

Easing:

Duration:

Effect:



Wait a Second!

- By default, a transition will **play immediately** when triggered
- The **transition-delay** property allows you to specify a period of time to wait before performing the transition
- As with the transition-duration property, time can be specified in either seconds or milliseconds

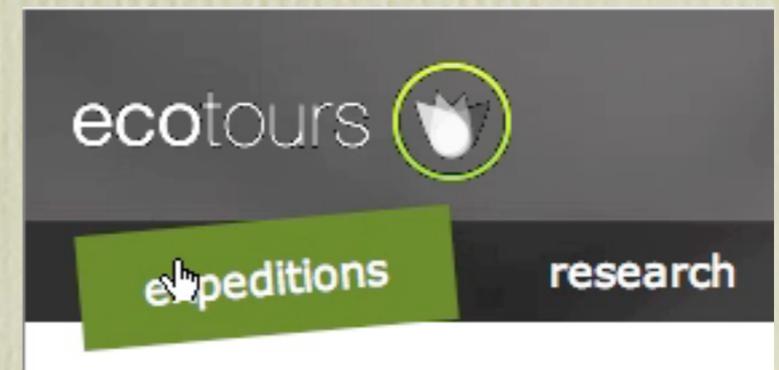
transition-delay: 3s;

transition-delay: 250ms;

Putting It All Together

```
nav li a {  
  background-color: rgb(0,0,0);  
  padding: 9px 5px;  
  transition-property: background-color, padding, transform;  
  transition-duration: 1s;  
  transition-delay: 100ms;  
  transition-timing-function: ease;  
}
```

```
nav li a:hover {  
  background-color: rgb(86, 125,33);  
  padding: 12px 5px;  
  transform: rotate(-5deg);  
}
```



Well, Actually...

```
nav li a {
  background-color: rgb(0,0,0);
  padding: 9px 5px;
  -webkit-transition-property: background-color, padding, transform;
  -moz-transition-property: background-color, padding, transform;
  -o-transition-property: background-color, padding, transform;
  transition-property: background-color, padding, transform;
  -webkit-transition-duration: 1s;
  -moz-transition-duration: 1s;
  -o-transition-duration: 1s;
  transition-duration: 1s;
  -webkit-transition-delay: 100ms;
  -moz-transition-delay: 100ms;
  -o-transition-delay: 100ms;
  transition-delay: 100ms;
  -webkit-transition-timing-function: ease;
  -moz-transition-timing-function: ease;
  -o-transition-timing-function: ease;
  transition-timing-function: ease;
}
nav li a:hover {
  background-color: rgb(86, 125,33);
  padding: 12px 5px;
  -webkit-transform: rotate(-5deg);
  -moz-transform: rotate(-5deg);
  -ms-transform: rotate(-5deg);
  -o-transform: rotate(-5deg);
  transform: rotate(-5deg);
}
```



Let's Mix It Up

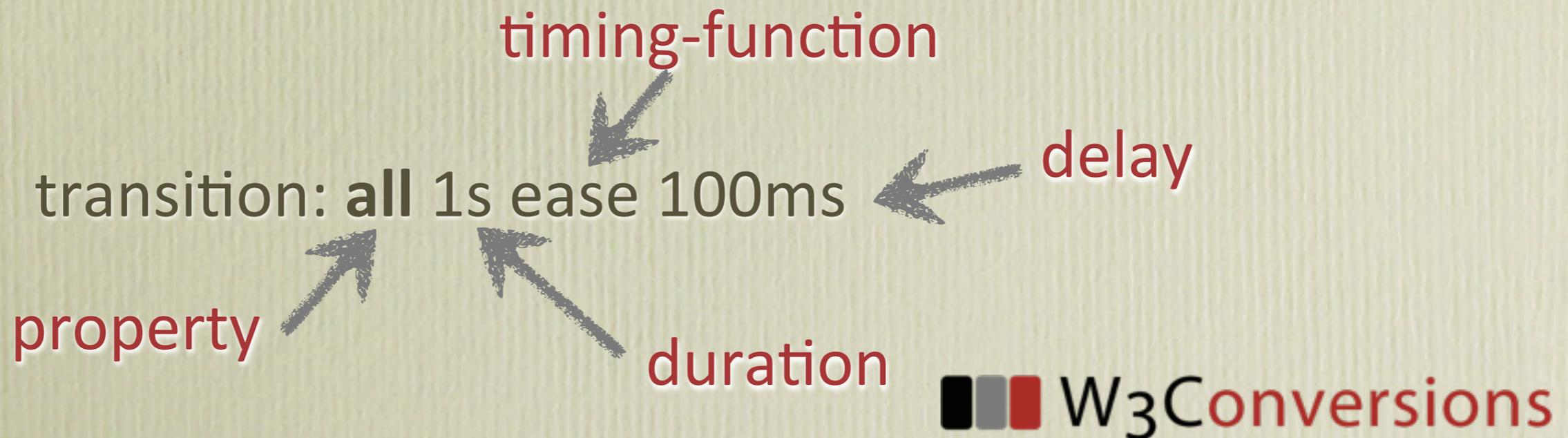
- You can **target multiple properties and assign differing speeds** — all within the same rule using comma-separated values.

```
nav li a {  
  transition-property: background-color, padding, transform;  
  transition-duration: 1s, 2s, 500ms;  
  transition-delay: 100ms, 0, 2s;  
  transition-timing-function: ease, linear, ease-in;  
}
```

Transition Shorthand

- Using shorthand, we can **combine** all four of the transition properties into a single property
- Order is **important** due to the existence of **2 timing properties** — the first *time* value will be assigned to transition-duration and the last to the delay
- Values are separated by a **space** (not comma)

transition: **transition-property transition-duration transition-timing-function transition-delay;**



That's Better

```
nav li a {
  background-color: rgb(0,0,0);
  padding: 9px 5px;
  -webkit-transition: all 1s ease 100ms;
  -moz-transition: all 1s ease 100ms;
  -o-transition: all 1s ease 100ms;
  transition: all 1s ease 100ms;
}
nav li a:hover {
  background-color: rgb(86, 125,33);
  padding: 12px 5px;
  -webkit-transform: rotate(-5deg);
  -moz-transform: rotate(-5deg);
  -ms-transform: rotate(-5deg);
  -o-transform: rotate(-5deg);
  transform: rotate(-5deg);
}
```

Let's Mix It Up

- It's also possible to shorthand the **multiple properties with differing speeds** (use a comma between the groupings)

```
nav li a {  
  transition:  
    background  
    padding 2s  
    transform 5  
}
```

[placed on separate

demo :: multiple properties and speeds

expeditions

research

community

news room

gallery

I

In the Wild

WUFOO

I work for Wufoo, the online form building service. Wufoo takes all the hard, crappy work out of building powerful forms.

LYNDA.COM

I created an online training course for Lynda.com called "WordPress: creating and editing custom themes."

DIGGING INTO WORDPRESS

A book and blog co-authored by Jeff Starr and myself about the World's most popular publishing platform.

QUOTES ON DESIGN

Design, like Art, can be an elusive word to define and an awfully fun thing to have opinions about.

HTML-IPSUM

One-click copy to clipboard access to *Lorem Ipsum* text that comes wrapped in a variety of HTML.

BOOKSHELF

Hey Chris, what books do you recommend? These, young fertile mind, these.

TWITTER

Dog photos, customer service complaints, what I ate for breakfast, how far I just ran, and sometimes web design stuff.

FORRST

My preferred social site for sharing code snippets, links, and screenshots of good design.

* This website may or may not contain any **CSS** or **Tricks**. Member of the Smashing Network.

[Contact](#)

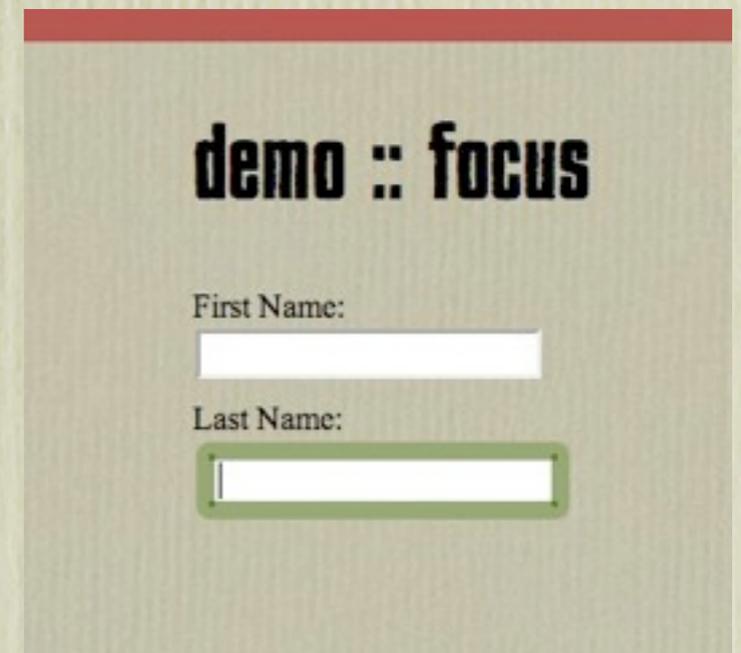
UI Enhancements

- Transitions and transforms can be added to pseudo-elements (::before and ::after).
- The **:focus** pseudo-class can add interesting effects to form elements by highlighting the element with focus.
- :focus is also good to use in combination with :hover — keyboard navigators have a visual cue of their focus on the page.

Stay :focus'ed

- Use the **:focus** pseudo-class to add interesting effects to form elements.

```
input {  
    -webkit-transition: border .5s;  
    -moz-transition: border .5s;  
    -o-transition: border .5s;  
    transition: border .5s;  
}  
input:focus {  
    outline:0;  
    border: rgba(72, 107, 11, .7) 9px solid;  
    -webkit-border-radius: 6px;  
    -moz-border-radius: 6px;  
    border-radius: 6px;  
}
```



Stay :focus'ed

```
textarea {  
  height: 20px;  
  -webkit-transition: height .5s,  
  -moz-transition: height .5s,  
  -o-transition: height .5s,  
  transition: height .5s,  
}  
textarea:focus {  
  outline: 0;  
  height: 250px;  
  border: rgba(72, 107, 11, .7) 1px solid;  
  -webkit-border-radius: 6px;  
  -moz-border-radius: 6px;  
  border-radius: 6px;  
}
```

demo :: focus

First Name:

Last Name:

Comment:



May 12–13 2011, Seattle
 Building *awesome* native & web apps
 with open web technologies



[Home](#) • [Program](#) • [Workshop](#) • [Sponsor](#) • [Expo](#) • [Venue](#) • [Pricing](#) • [Contact](#)

[Register now!](#)

[Keynotes](#)

[Design Track](#)

[Development Track](#)

[Platforms and Solutions Track](#)

[Speakers](#)

[Networking](#)

Program

CONFERENCE SCHEDULE: MAY 12 & 13

Web Directions Unplugged is a **two day conference**, an optional **workshop**, and a whole host of **friendly social events** and **networking opportunities**. Looking to get up to speed with the latest on using **web technologies** to build apps for any device? Need to learn the principles of **great app design**? See you there!

Thursday, May 12 2011

TIME	DESIGN TRACK	DEVELOPMENT TRACK	PLATFORMS & SOLUTIONS TRACK
8:00–9:00	Registration		

SEARCH

STAY IN TOUCH

Twitter: [@webdirections](#)

Follow [the Unplugged 2011 speakers](#)

Keep up with the latest in HTML5, app development, platforms, devices and more at our blog, [the web, unplugged](#)

Track the [latest Web Directions news](#)

DOWNLOAD OUR ONE-PAGER

Need to convince the person who writes the cheques? Grab our [ready to print one-pager](#).

RESOURCES

Check out [presentations from previous Web Directions conferences](#), covering all aspects of design, development, and online business.

Click If You Love Javascript

- Transitions and transformations are simply **a change from one state** of an element to another. CSS can only do so much.
- Using Javascript, we can trigger transformations and transitions for any catchable event, such as **onload, onchange** or **click**.
- jQuery or any other Javascript framework can be used.

Progressive Enhancement-Not

← → ↻ 🏠 🌐 www.thecssninja.com/demo/css_accordion/ ⌨️ ⭐ ⚙️ 🖱️ 🟢 🔑 📧 😊 🛠️

 Example from the [CSS] Ninja

Section 1

Section 2

Section 3

Section 4

Regressive Enhancement

- When you absolutely, positively must work in older browsers
 - Modernizr
 - yepnope.js
 - JS polyfills/shims (for selectors and animations)

Polyfills, Shims & Fallbacks

HTML5 Cross browser Polyfills

Page History

So here we're collecting all the shims, fallbacks, and polyfills in order to implant html5 functionality in browsers that don't natively support them.

The general idea is that: we, as developers, should be able to develop with the HTML5 apis, and scripts can create the methods and objects that should exist. Developing in this future-proof way means as users upgrade, your code doesn't have to change but users will move to the better, native experience cleanly.

Looking for a way to conditionally load these scripts client-side based on feature detects? See [yepnope.js](#)

The All-In-One Entirely-Not-Alphabetical No-Bullshit Guide to HTML5 Fallbacks

svg

- [svgweb](#) by brad neuberg & others
 - fallback via Flash
- [Raphaël](#) by dmitry baranovsky
 - abstracted API. adds features. fallback for IE via VML.
- [Ample SDK](#) by Sergey ilinsky
- [canvg](#) by Gabe Lerner
 - writes SVG to canvas. Good for Android.
- [SVG Boilerplate](#) alpha and still buggy, but handle multiple concurrent SVG shims together

svg smil

- [fakesmil](#)

canvas

- [FlashCanvas](#) by Shinya Muramatsu
 - reported to have [33x better performance](#) than excanvas.
- [excanvas](#) by google, erik arvidsson
- [slicanvas project](#) ([Original Silverlight bridge](#), [demo page](#))
- [canvas-text](#) *only necessary for canvastext in IE, really*
- [fxCanvas](#)

web storage

- [storage polyfill](#) by remy sharp

nversions

yepnope.js

Script loader for your Javascript (and CSS if needed) based on what feature detection finds.

```
yepnope([
  {
    test : Modernizr.indexeddb,
    yep  : ['/js/indexeddb-wrapper.js', '/css/coolbrowser.css'],
    nope : ['/js/polyfills/lawnchair.js', '/js/cookies.js',
            '/css/oldbrowser.css']
  }
]);
```

Polyfilling a transition/transform

- transition: all 3s ease-out;
transform: translateX(600px);
- if (Modernizr.no-csstransforms &&
Modernizr.no-csstransitions) {

// if an older browser, fall back to jQuery animate
jQuery(elem).animate({ 'left': '600px'}, 3000);
}

CSS3 Animation - Working Draft

Global user stats* :

Support: 23.21%

Complex method of animating certain properties of an element

Resources: [Blog post on usage](#) [Webkit blog post](#)

	IE	Firefox	Safari	Chrome	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser	
Two versions back	7.0	3.5	3.2	9.0 <small>-webkit-</small>	10.6	3.2 <small>-webkit-</small>			2.1	<small>-webkit-</small>
Previous version	8.0	3.6	4.0 <small>-webkit-</small>	10.0 <small>-webkit-</small>	11.0	4.0-4.1 <small>-webkit-</small>		10.0	2.2	<small>-webkit-</small>
Current	9.0	4.0	5.0 <small>-webkit-</small>	11.0 <small>-webkit-</small>	11.1	4.2-4.3 <small>-webkit-</small>	5.0-6.0	11.0	2.3 <small>-webkit-</small>	3.0 <small>-webkit-</small>
Near future		5.0 <small>-maz-</small>		12.0 <small>-webkit-</small>	11.5					
Farther future	10.0	6.0 <small>-maz-</small>	6.0 <small>-webkit-</small>	13.0 <small>-webkit-</small>	12.0					

Note: Support in Firefox may not land until version 6.

Feedback

Animations



Like Transitions on Steroids

- A CSS3 Animation, animates between CSS properties exactly like transitions – however, we can specify the number of times it plays and whether it plays in a single direction or both directions
 - webkit-animation-**iteration-count**: **number** or **infinite**;
 - webkit-animation-**direction**: **normal** or **alternate**;
- Animations are named, and can therefore be reused by more than one CSS rule
 - webkit-animation-**name**: **moveIt**;

I Was Keyframed, I Tell You

- An animation defines a set of **start** (from) properties and a set of **end** (to) properties as values of keyframes assigned to the named animation

```
@-webkit-keyframes moveIt {  
  from {  
    properties/values go here  
  }  
  to {  
    properties/values go here  
  }  
}
```

A Simple Animation

```
.animation-rule {  
  -webkit-animation-name: movelt;  
  -webkit-animation-duration: 4s;  
  -webkit-animation-iteration-count: infinite;  
  -webkit-animation-direction: normal;  
  -webkit-animation-timing-function: ease;  
}  
  
@-webkit-keyframes movelt {  
  from {  
    width: 200px;  
    background: #f00;  
    opacity: 0.5;  
    -webkit-transform: scale(0.5) rotate(15deg);  
  }  
  to {  
    width: 400px;  
    background: #fffa2;  
    opacity: 1;  
    -webkit-transform: scale(1) rotate(0deg);  
  }  
}
```

Smoothing out the bumps

- As with transitions, the **animation-timing-function** can use linear, ease, ease-in, ease-out, ease-in-out
- For more control over the look of the animation, **additional keyframes** allow you to specify the values for properties at a certain **percentage** of the animation
- Animations can also be written as shorthand
-webkit-animation: **name duration timing iterations direction;**

A Simple Animation

```
.animation-rule {  
  height: 60px;  
  padding: 10px;  
  -webkit-animation: 'movelt' 4s ease 3 normal;  
}  
@-webkit-keyframes 'movelt' {  
  0% {  
    width: 200px;  
    background: #A74240;  
    opacity: 0.5;  
    -webkit-transform: scale(0.5) rotate(15deg);  
  }  
  40% {  
    width: 300px;  
    background: #A74240;  
    opacity: 1;  
    -webkit-transform: scale(1.3) rotate(15deg);  
  }  
  100% {  
    width: 400px;  
    background: #ffffa2;  
    opacity: 0.5;  
    -webkit-transform: scale(1) rotate(0deg);  
  }  
}
```

demo :: animation

Animation Values

animation-name: 'yourName';

animation-duration: *time*;

animation-timing-function:

linear | ease | ease-in | ease-out |
ease-in-out | cubic-bezier();

animation-iteration-count: *number* or infinite;

animation-direction: normal | alternate;

animation-delay: *time*;

animation-fill-mode:

backwards | forwards | none | both;

-webkit-animation: 'disappear' 0.6s linear 0.5s alternate 5 forwards;

Animation fine-tuning

```
@-webkit-keyframes 'flicker' {  
  0%    { opacity: 1;}  
  10%   { opacity: 0;}  
  20%   { opacity: 1;}  
  40%   { opacity: 0;}  
  60%   { opacity: 1;}  
  100% { opacity: 0;}  
}  
  
.alertMessage {  
  -webkit-animation:  
    'flicker' 0.6s linear 0.5s forwards;  
}
```

tuning the animation

```
@-webkit-keyframes 'flicker' {  
  0%, 20%, 60% { opacity: 1;}  
  10%, 40%, 100% { opacity: 0;}  
}  
  
.alertMessage {  
  border: 1px solid #000000;  
  background-color: #660000;  
  color: #FFFFFF;  
  opacity: 1;  
  -webkit-animation:  
    'flicker' 0.6s linear 0.5s forwards;  
}
```

Accelerate!

-webkit-transform: translateZ(0);

-webkit-transform: translate3d(0,0,0)

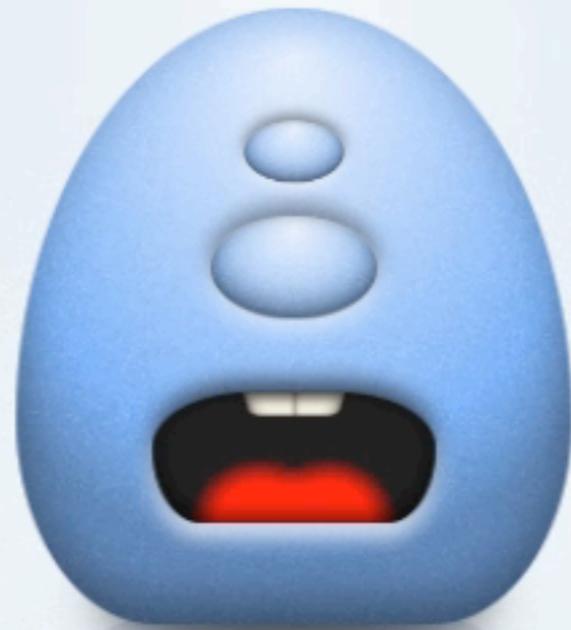
- Both these tricks will make your webkit browser believe it needs to use the GPU, rather than the CPU, which can increase performance.
- As always, test to be sure performance is increased —and that you're not simply decreasing battery life for your user.

Animated Radial Gradients



CSS3: Beyond the Basics

**CAN HAZ UR
CURSOR?**



ons

Tools

- Wallaby - Flash to HTML
- Westciv Transform Tool - sliders for creation
- CSS3 Please (test transforms & transitions)
- Project Edge - Upcoming tool for creating animations & transitions
- Animatable

CSS3 3D Transforms - Working Draft

Global user stats* :
Support: 4.82%

Method of transforming an element in the third dimension

Resources: [Webkit blog post](#) [Demo page](#) [Cubic panorama demo \(Mobile Safari only\)](#)

	IE	Firefox	Safari	Chrome	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
Two versions back	7.0	3.5	3.2	9.0	10.6	3.2			2.1
Previous version	8.0	3.6	4.0	10.0	11.0	4.0-4.1		10.0	2.2
Current	9.0	4.0	5.0 <small>-webkit-</small>	11.0	11.1	4.2-4.3	5.0-6.0	11.0	2.3 3.0
Near future		5.0		12.0	11.5				
Farther future	10.0 <small>-ms-</small>	6.0	6.0 <small>-webkit-</small>	13.0	12.0				

Note: Current implementations may change significantly from final specification, since W3C adoption is quite recent.

[Feedback](#)

3D Transforms



 W3C conversions

Putting it in Perspective

- “The 3D Transformations Module extends the 2D transformations with a **perspective** transformation.”
- Actually, it’s really more about 2.5D — think postcards in space
 - X, Y & Z axis rotation
 - Perspective
- Applied via the same transform property as 2D transforms (our examples will use -webkit-)

Translate in 3d

- Several new transform functions. Remember:

```
transform: translate(50px, 50px);  
transform: translateX(50px);
```

- We now add:

```
-webkit-transform: translate3d(x, y, z);  
-webkit-transform: translateZ(z);
```

- For the z axis, a positive number represents a move toward the viewer.

* The z value cannot be a percentage.

3d Scale & Rotation

- In addition to movement (translate), objects can be scaled and rotated

`scale3d(sx, sy, sz)`
`scaleZ(sz)`

`rotateX(angle), rotateY(angle),`
`rotate3d(x, y, z, angle)`

When rotating, the first two examples rotate an element around a horizontal or vertical axis defined using degrees (deg), radians (rad) or gradians (grad). The last example rotates an element around a defined vector in 3D space.

Animations

- You can use these in animations as well:

```
@-webkit-keyframes x-spin {  
  0% { -webkit-transform: rotateX(0deg); }  
  50% { -webkit-transform: rotateX(180deg); }  
  100% { -webkit-transform: rotateX(360deg); }  
}
```

I See Where You're Coming From

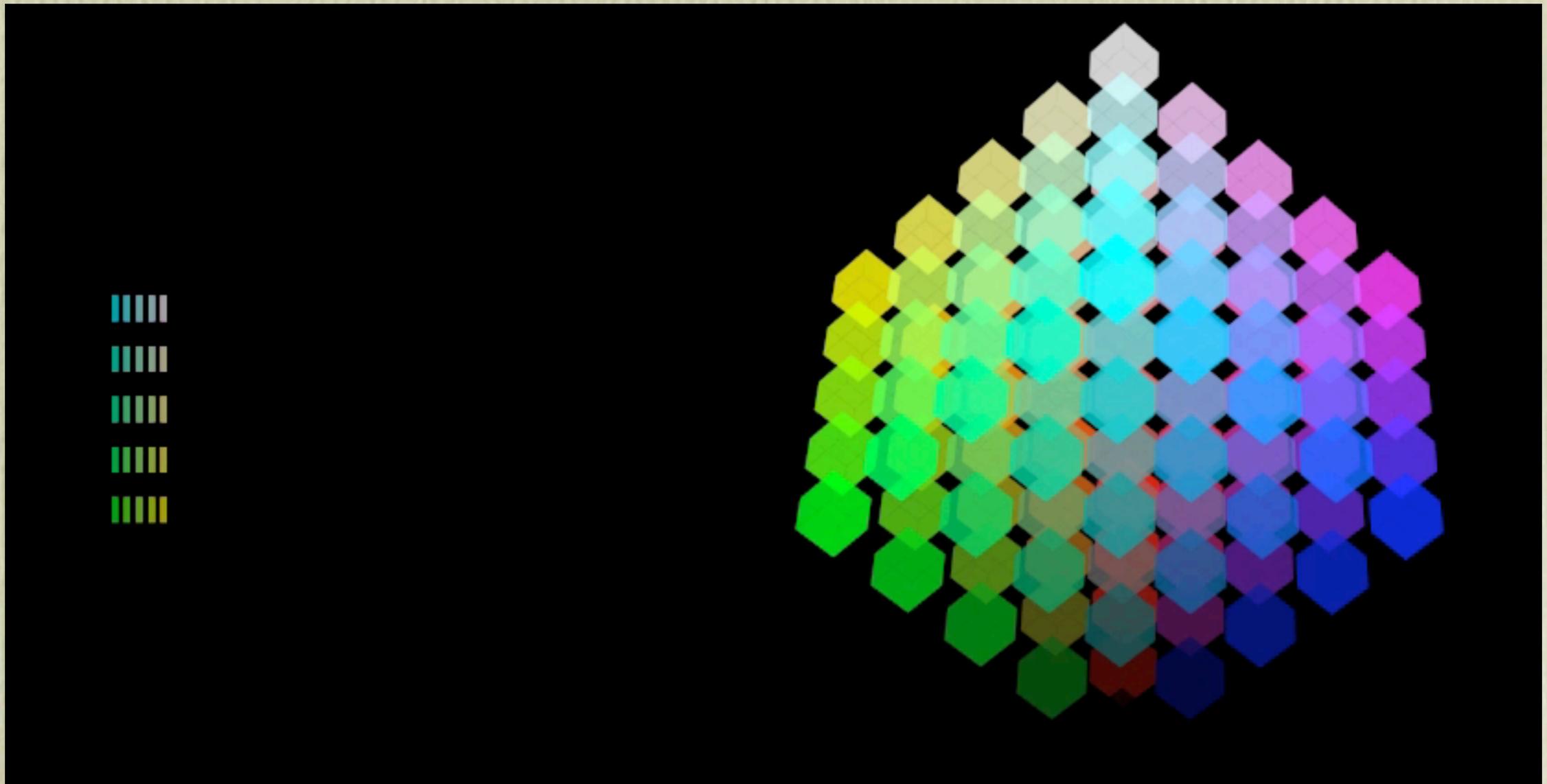
- For 3D Transforms, a 3rd value (Z) is added to the `-webkit-transform-origin` property
- Perspective can be applied to a single element using the `perspective(p)` function or applied to all the children of an element using the `-webkit-perspective` property
 - webkit-transform: rotateY(45deg) perspective(400px);
 - webkit-perspective: 800px;
- Smaller numbers indicate that the perspective begins closer to the viewer

Stay Away From Me

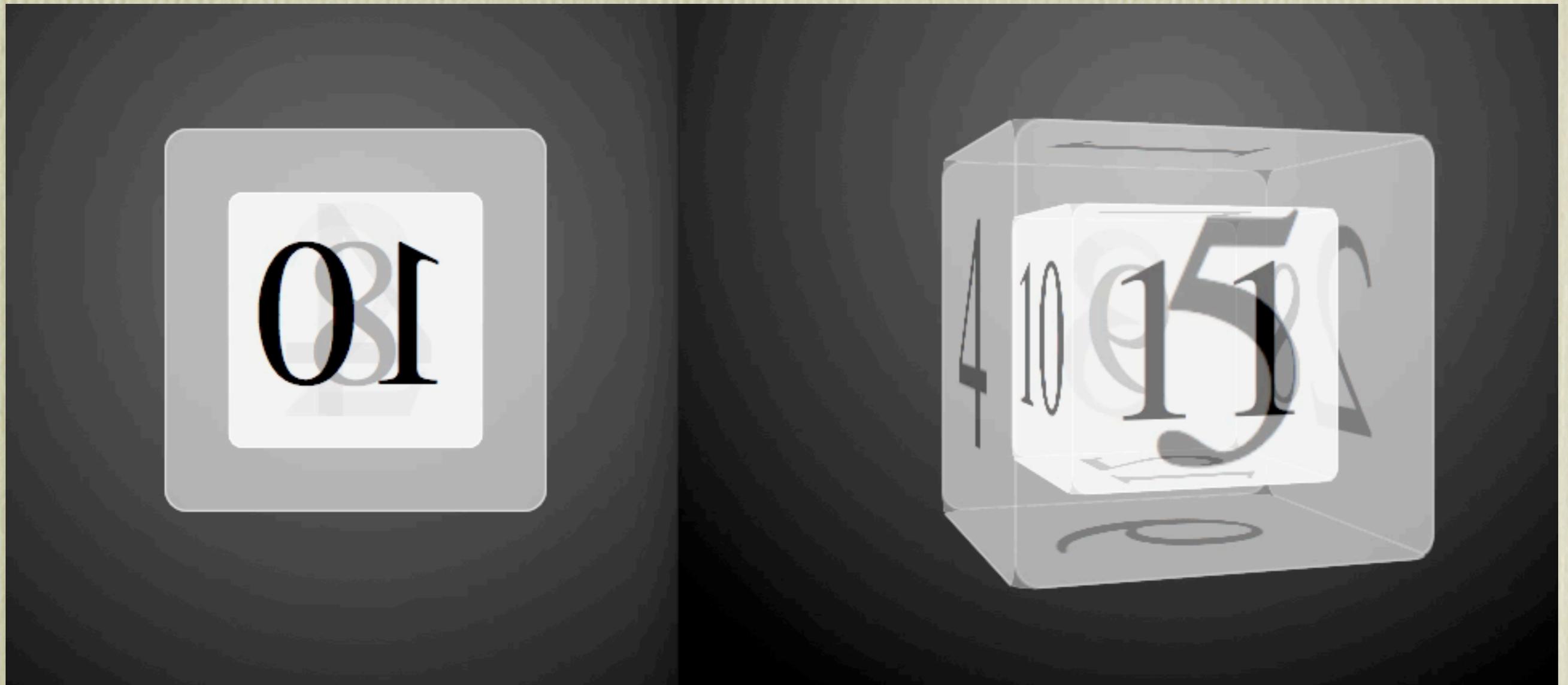
- By default the children of a transformed element are flattened into the parent element, taking on the transformation of the parent
- Using `-webkit-transform-style` allows you to specify whether the elements should follow their default behavior (flat) or “live” in the 3D world with their own transformations (preserves-3d)

<http://www.webkit.org/blog-files/3d-transforms/transform-style.html>

3D Cube (Chrome/Safari)



backface-visibility



Helper apps and tutorials

- [Trigger CSS3 Animations with jQuery](#)
- [cssSandpaper](#)
- [Transformie](#)
- [Selectivizr](#)

Thank you!

- www.w3conversions.com/resources.html
- @stefsull
- info@w3conversions.com