



CSS Effects, Part 2

Galleries and 3D effects

Simon Fraser
Safari and WebKit Engineer

HTML5

CSS3

JavaScript

Apple - iPad - View photos and images of iPad


http://www.apple.com/ipad/gallery/ Google

Store Mac iPod iPhone iPad iTunes Support Search


iPad Features Design Apps for iPad Gallery Guided Tours Tech Specs Buy Now

The best way to experience the web, email, photos, and video. Hands down.

All of the built-in apps on iPad were designed from the ground up to take advantage of the large Multi-Touch screen and advanced capabilities of iPad. And they work in any orientation. So you can do things with these apps that you can't do on any other device.



[Watch the iPad video >](#)



Safari

iPad is the best way to experience the web. View whole pages in portrait or landscape on the large Multi-Touch screen. And let your fingers do the surfing. [Learn more >](#)

Get to know iPad.
Watch the Guided Tours.

A whole new world of apps.
Discover apps made just for iPad.

Outfit your iPad.
Find the accessories that make iPad even better.

Buy iPad. From \$499
Apple Online Store
[Buy your iPad and get free shipping >](#)

Panic - Transmit - The ultimate Mac OS X FTP + SFTP + S3 app

http://panic.com/transmit/

How do you make Transmit 4 the best ever? 45+ new features, 25x speed, and one more thing...

You need to transfer files. Maybe to an FTP or SFTP server, or the cloud via Amazon S3, or using WebDAV. You maintain a website, do backups, or upload photos.

You need Transmit, the #1 Mac OS X FTP client.¹ Now, in Transmit 4, we took everything good about Transmit, and added a dollop of **unbelievably great**.

Features

We've gone all out.

See All New Features

Apple - HTML5 Showcase - Transitions

http://www.apple.com/html5/showcase/transitions/ Google

Store Mac iPod iPhone iPad iTunes Support Search


HTML5 Showcase

[← Back to overview](#)

The demos below show how Apple's Safari web browser supports the capabilities of web standards such as HTML5, CSS3, and JavaScript.

Transitions

- Cube
- Rotate In
- Horizontal Flip
- Multi-Flip
- Dissolve
- Unfold
- Toss
- Slide In
- Iris
- Fade Through



Try it out: Click any of the transitions above to see them in action.

About this demo: Add Keynote-style transitions to objects on your web pages by using CSS 2D and 3D transforms like the ones displayed here.

[Visit the Safari Dev Center to learn more and download sample code ▶](#)

Technology Framework

- It is just web development
 - CSS properties
 - DOM APIs, events, etc.
 - JavaScript
- Works in
 - Safari on iOS, Mac, and Windows
 - UIWebView and WebView
 - iAds



Open Standards

- Enhancing web technology hand-in-hand with
 - WHATWG, HTML Working Group
 - CSS Working Group
- CSS draft properties
 - `-webkit` prefix



What You'll Learn

- Lightbox
 - Simplicity of CSS Transitions
 - Consistent behavior between iOS and desktop
- Photo gallery
 - Dynamic content
 - Rich, cinematic effects
- Tips and tricks

Demo

Enrica Casucci
Safari and WebKit Engineer


Building a Better Lightbox

Lightbox Demo

http://www.example.com/pictures

Trip to Tuscany

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.



Poppy field in Tuscany, Italy

Foggy Tuscany landscape

Val D'Orcia landscape in Tuscany, Italy

Ponte Vecchio at sunset

Florence Duomo after the sunset

Val D'Orcia landscape in Tuscany, Italy

Wine grapes on the vine one Early Fall Morning

Seaside Villas near Portofino, Italy

Farmhouse and cypress trees in Crete Senesi area, Tuscany Italy

A Better Lightbox

- Smooth transitions
- Loading indicator with CSS
- Works consistently on desktop, iPad, and iPhone

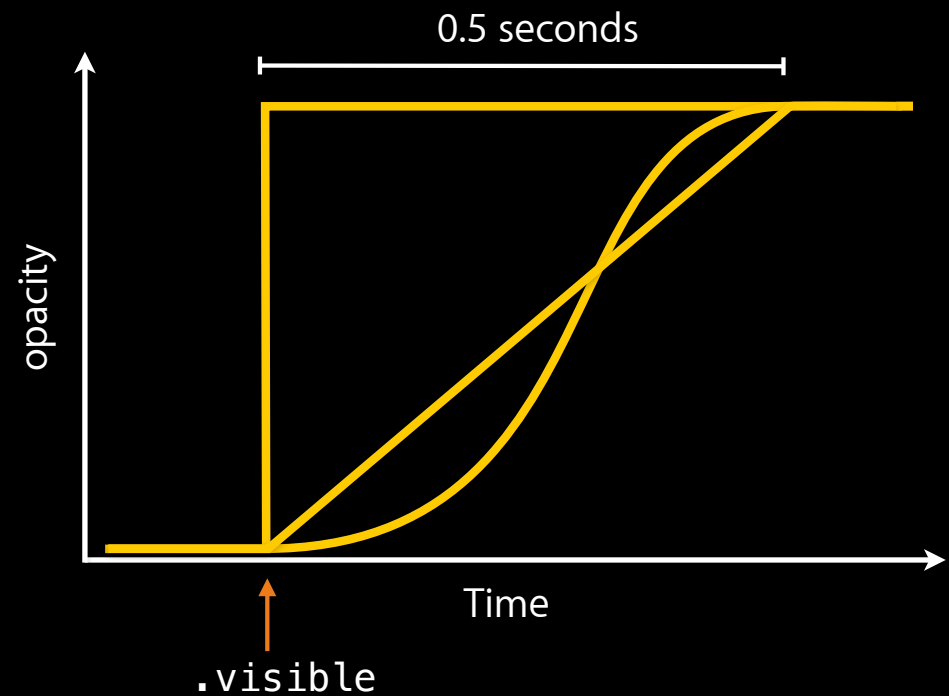
Step 1: Start Loading the Image

- When the user clicks on an image
 - Make image in JavaScript
 - Register load listener
- Don't make the user wait!

```
// Start loading the large image
var largeImage = new Image();
largeImage.onload = imageLoaded();
largeImage.src = highResImageURL;
```

Step 2: Fade In the Overlay

```
#overlay {  
  opacity: 0;  
  -webkit-transition: opacity 0.5s  
    linear;  
}  
  
#overlay.visible {  
  opacity: 1;  
}
```



CSS Transitions

- Which properties transition
 - webkit-transition-property
- Duration
 - webkit-transition-duration
- Timing function
 - webkit-transition-timing-function
- Delay before start
 - webkit-transition-delay

Step 2 Continued: Fade In the Overlay

```
#overlay {  
  opacity: 0;  
  -webkit-transition: opacity 0.5s  
}
```

```
#overlay.visible {  
  opacity: 1;  
}
```

```
var overlay = document.createElement('div');  
overlay.id = 'overlay';  
document.body.appendChild(overlay);  
overlay.className = 'visible';
```



Aside: How Browsers Apply CSS Styles

- Browsers optimize away redundant style changes
- This matters with transitions, because they are temporal

```
var box = document.getElementById('box');  
box.style.backgroundColor = 'red';  
box.style.backgroundColor = 'blue';
```

Aside: How Browsers Apply CSS Styles

- Browsers optimize away redundant style changes
- This matters with transitions, because they are temporal

```
var box = document.getElementById('box');  
box.style.backgroundColor = 'red';  
box.style.webkitTransition = 'background-color 2s';  
box.style.backgroundColor = 'blue';
```



Aside: How Browsers Apply CSS Styles

- Browsers optimize away redundant style changes
- This matters with transitions, because they are temporal

```
var box = document.getElementById('box');  
box.style.backgroundColor = 'red';  
box.style.webkitTransition = 'background-color 2s';  
window.setTimeout(function() {  
    box.style.backgroundColor = 'blue';  
}, 0);
```



Step 2 Continued: Fade In the Overlay

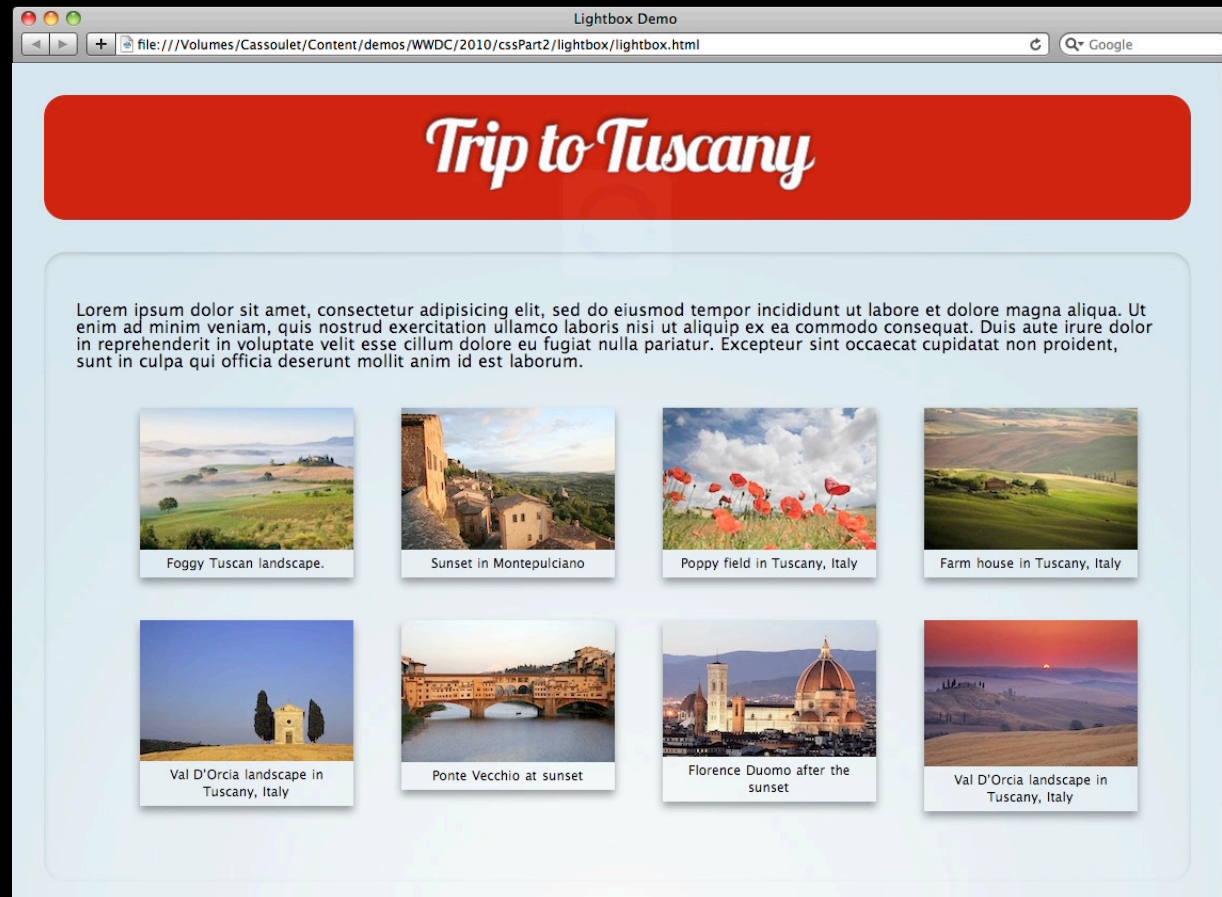
```
#overlay {  
  opacity: 0;  
  -webkit-transition: opacity 0.5s  
}
```

```
#overlay.visible {  
  opacity: 1;  
}
```

```
var overlay = document.createElement('div');  
overlay.id = 'overlay';  
document.body.appendChild(overlay);  
window.setTimeout(function() {  
  overlay.className = 'visible';  
} 0);
```



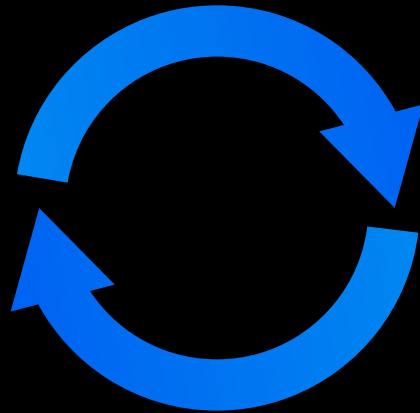
Step 2: Fade In the Overlay



Step 3: Animate the Loading Indicator

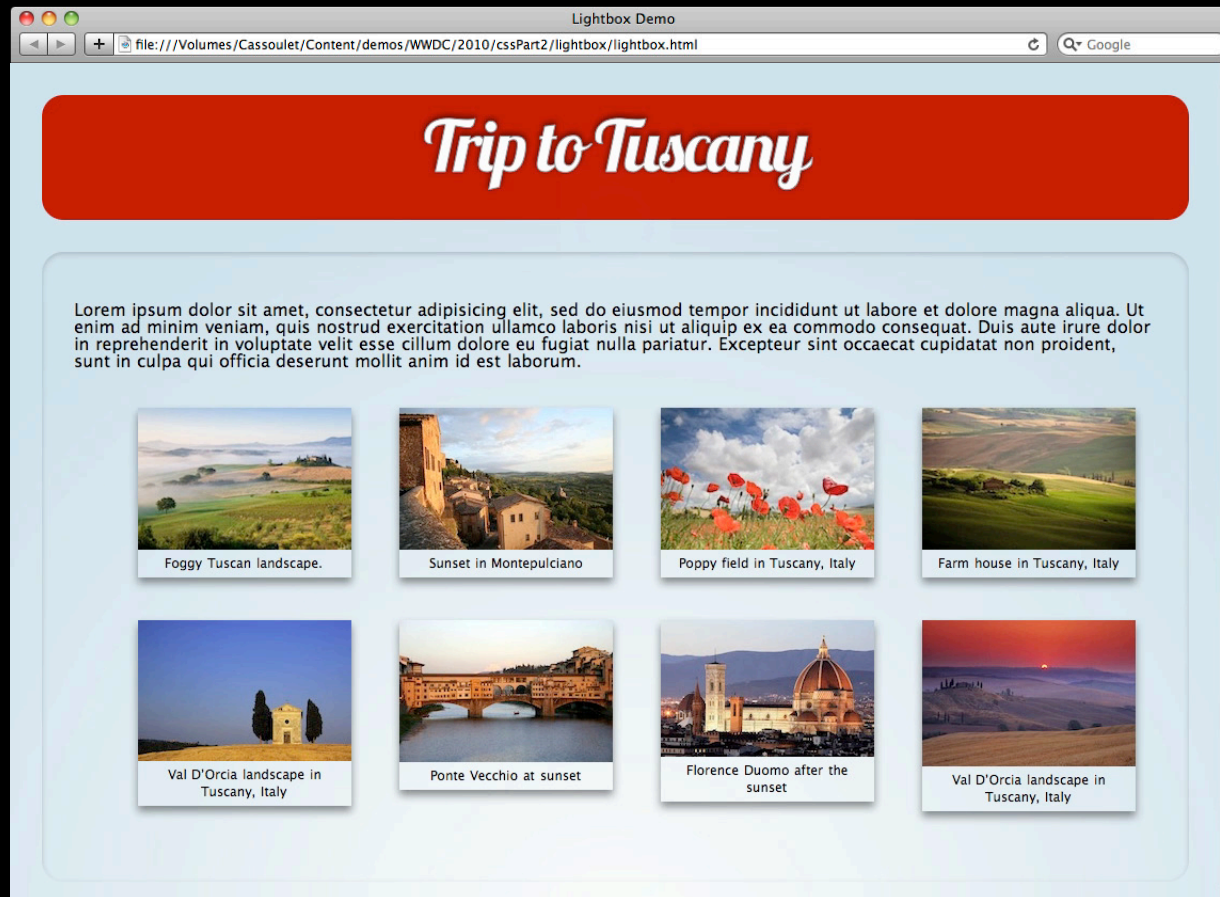
- Rotate with CSS Transforms
- Animate with CSS Keyframe Animations
- Use a high-quality image

Loading Indicator Using CSS Animations

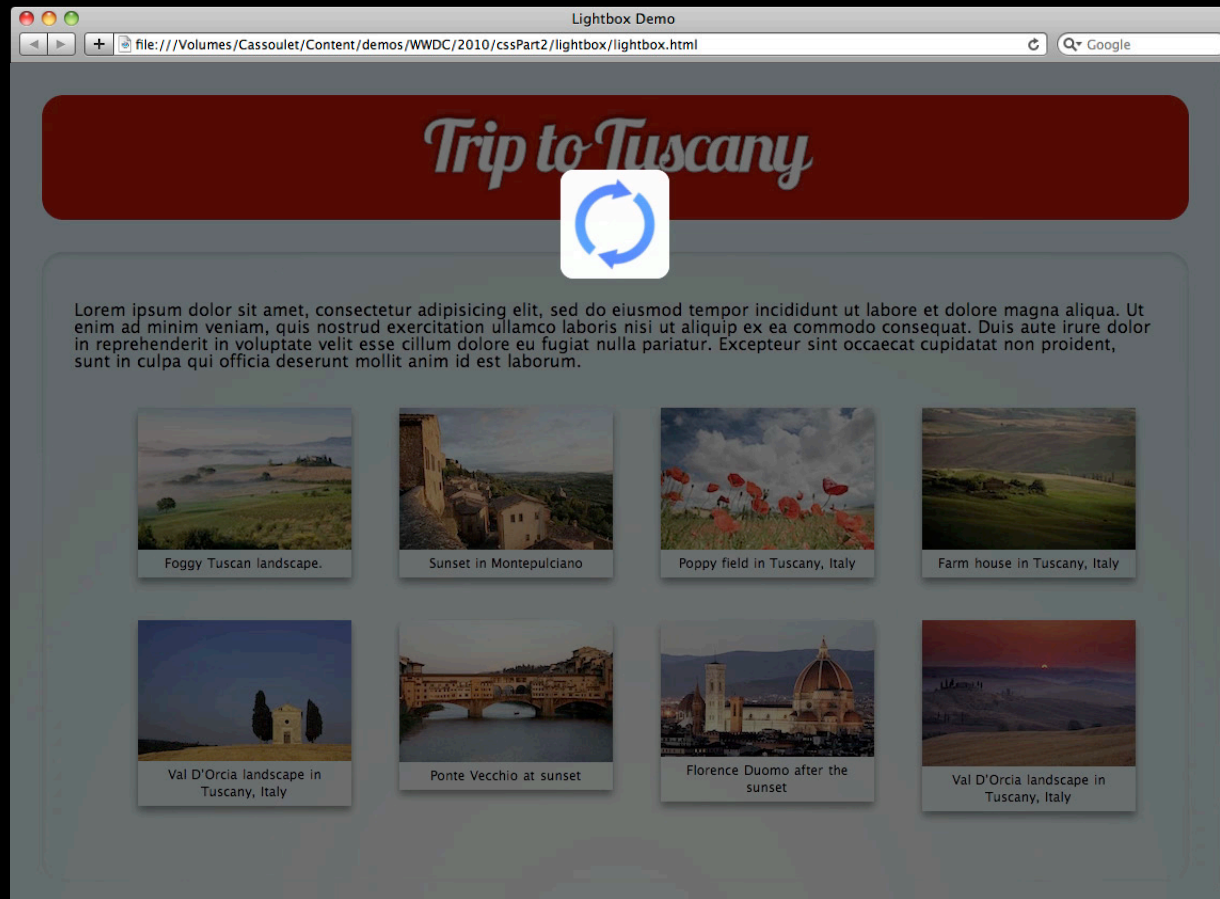


```
@-webkit-keyframes spin {  
  from { -webkit-transform: rotate(0); }  
  to   { -webkit-transform: rotate(360deg); }  
}  
  
.running {  
  -webkit-animation: spin 1s infinite linear;  
}
```

Step 3: Animate the Loading Indicator



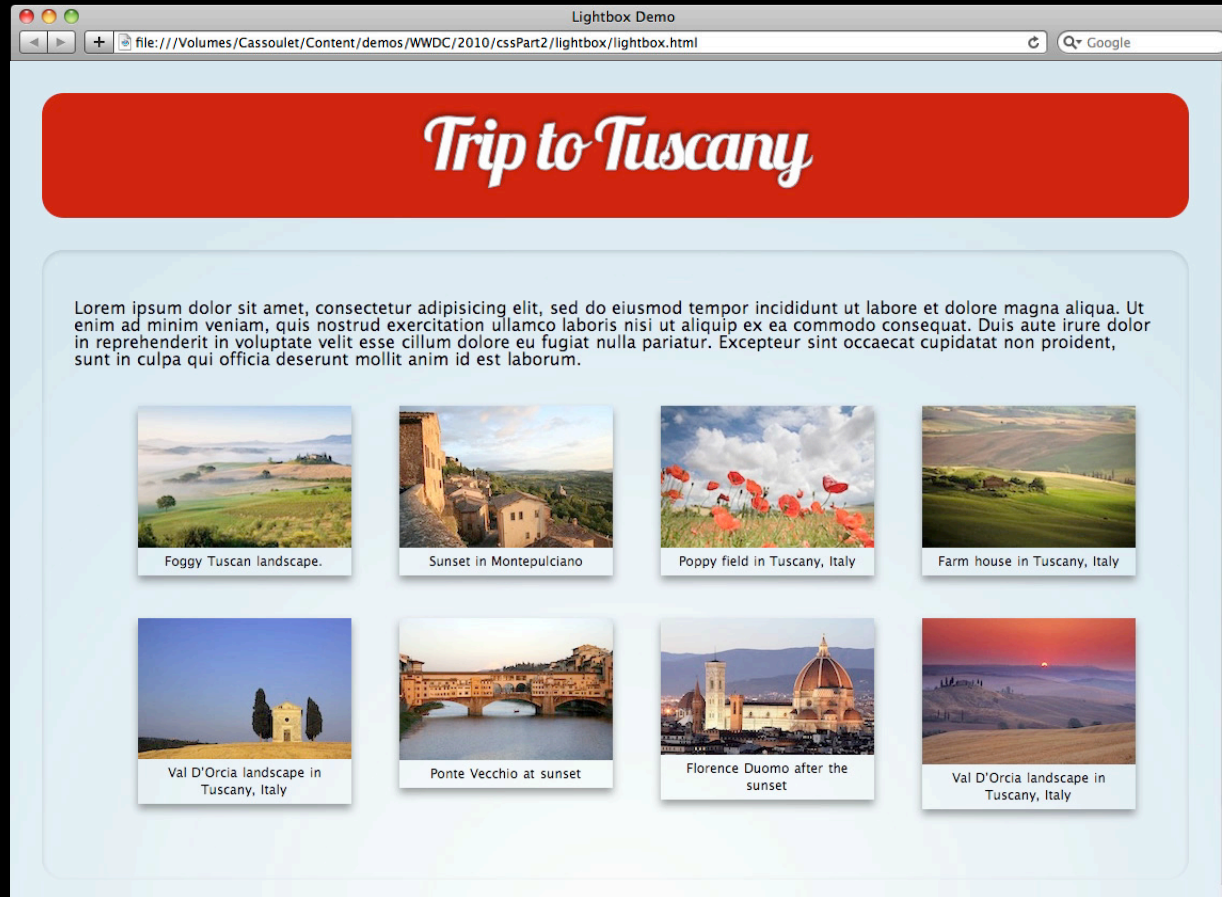
Step 4: Reveal the Image



Step 4: Reveal the Image

- Wait for both the large image to load, and the transition to finish
- CSS Transition for height/width
- Reveal caption
- Reveal close button

The Final Result

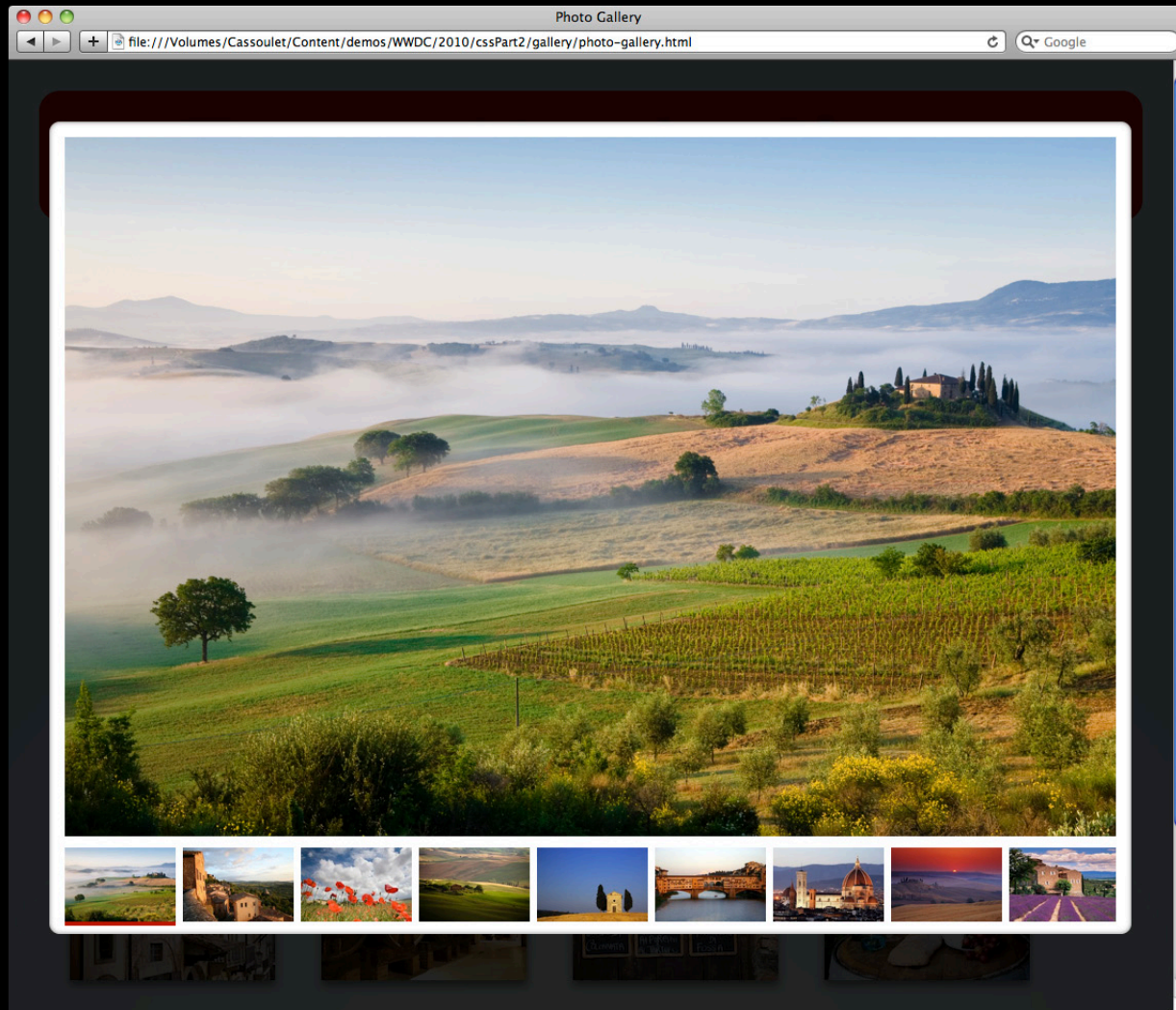


Optimizing for iPad and iPhone

- For best performance, use transitions and animations on
 - -webkit-transform
 - opacity

A Better Lightbox

- Use CSS Transitions for
 - Cinematic experience
 - Ease of authoring
 - Good performance on all platforms
- Script-free loading indicator using
 - CSS Transforms
 - CSS Animations



Awesome Photo Galleries





Gallery Best Practices

- Stay within the same page
- Load each image on demand
- Describe the states using CSS classes
 - “incoming” and “outgoing” images
 - “initial” and “final” states
- Minimal JavaScript to load images and toggle classnames

Gallery Scaffolding

``

`<div id="photo-container">`

``



Building a Push Down Transition



Push Down Transition

```
#photo-container.initial.incoming {  
  -webkit-transform: translateY(-100%);  
}
```

```
#photo-container.initial.outgoing {  
  -webkit-transform: translateY(0);  
}
```

```
#photo-container.final.incoming {  
  -webkit-transform: translateY(0);  
}
```

```
#photo-container.final.outgoing {  
  -webkit-transform: translateY(100%);  
}
```

```
#photo-container > img {  
  -webkit-transition: -webkit-transform 0.5s;  
}
```



```
<div id="photo-container" class="final">>
```



Running the Transition

```
// Get the container
var photoContainer = document.getElementById('photo-container');

// Set initial state
var outgoingImage = photoContainer.firstChild;
outgoingImage.className('outgoing');

incomingImage.className('incoming');
photoContainer.appendChild(incomingImage);

photoContainer.className('initial');

// Kick off the transition
window.setTimeout(function() {
  photoContainer.removeClassName('initial');
  photoContainer.addClassName('final');
}, 0);
```

Cleaning Up

```
incomingImage.addEventListener('webkitTransitionEnd', transitionDone, false);
```

```
function transitionDone(event)
{
  photoContainer.removeClassName('final');

  // Remove old image
  photoContainer.removeChild(photoContainer.firstChild);

  // Remove temporary class name
  incomingImage.removeClassName('incoming');
}
```

Some Transition Examples

Scale Transition



```
#photo-container.initial .incoming {  
  -webkit-transform: scale(0);  
}  
  
#photo-container.final .incoming {  
  -webkit-transform: scale(1);  
}
```

Spin Transition



```
#photo-container.initial .incoming {  
  -webkit-transform: rotate(-180deg);  
}  
  
#photo-container.final .incoming {  
  -webkit-transform: rotate(0);  
}
```

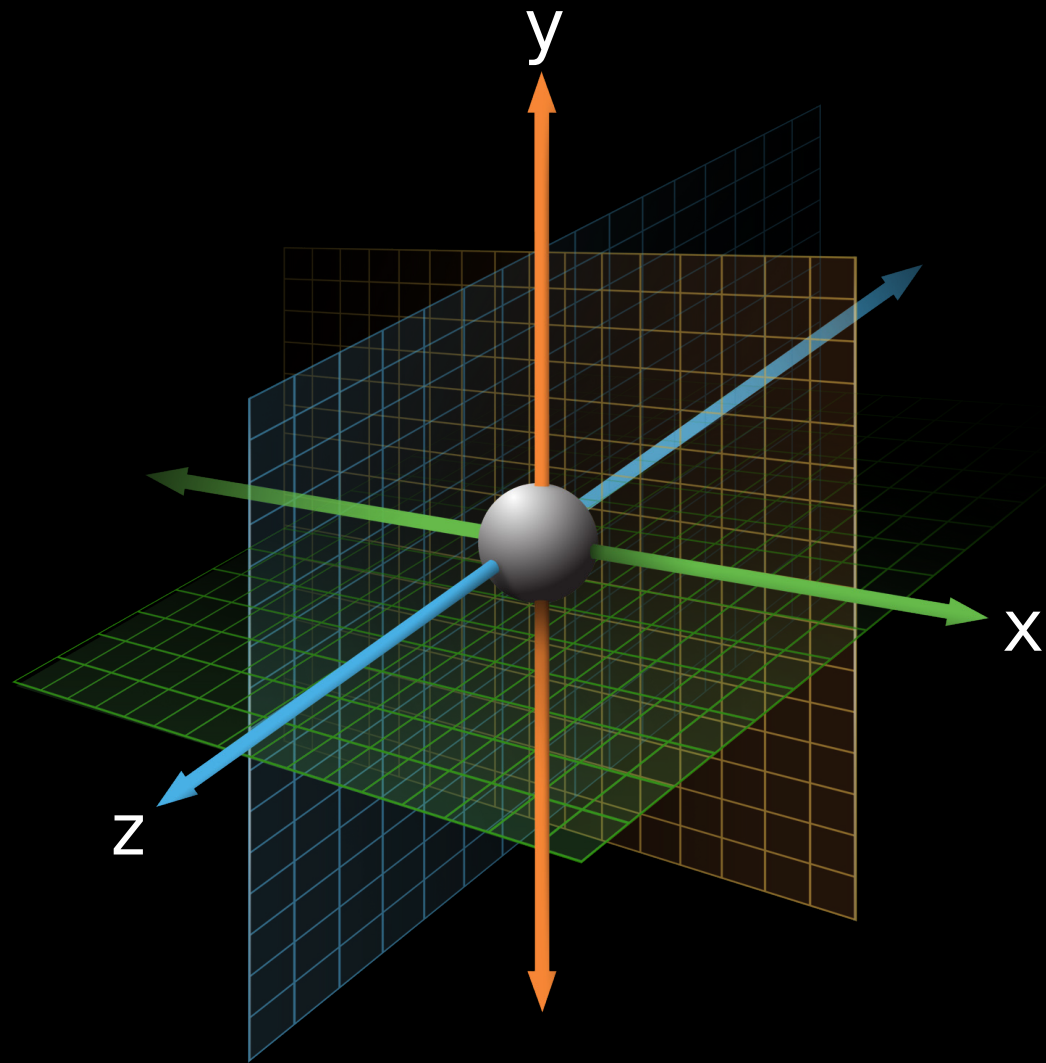
What You've Learned

- Dynamically load each image in JavaScript
- Use classes to describe states
- Minimal JavaScript to toggle states
- Use the "transitionEnd" event to detect when the transition is complete
- Rich set of photo transitions using CSS Transforms



3D

3D Axes



CSS 3D Transforms

- Positioning planes in 3D space
 - New values for `-webkit-transform`
 - `translate3d()`, `rotate3d()`, `scale3d()` etc.
- Make it look 3D
 - `-webkit-perspective`
- Build 3D hierarchies
 - `-webkit-transform-style`
- Single-sided elements
 - `-webkit-backface-visibility`

Revolving Door Effect Using 3D Transforms



Setting the Origin of Rotation



Setting the Origin of Rotation

Transform Origin



```
#photo-container > img {  
  -webkit-transition: -webkit-transform 0.5s;  
  -webkit-transform-origin: right 50%;  
}
```

Revolving Door Transition



```
#photo-container.initial .incoming {  
  -webkit-transform: rotateY(90deg);  
}  
  
#photo-container.final .incoming {  
  -webkit-transform: rotateY(0);  
}
```

Revolving Door Transition



```
#photo-container.initial .outgoing {  
  -webkit-transform: rotateY(0);  
}  
  
#photo-container.final .outgoing {  
  -webkit-transform: rotateY(-90deg);  
}
```



It looks flat!

Add Depth with Perspective

```
-webkit-perspective:
```

Add Depth with Perspective



```
-webkit-perspective: 700px;
```

Add Depth with Perspective



```
#photo-container {  
  -webkit-perspective: 800px;  
}
```


More Complex Animations



CSS Keyframe Animations

- Keyframe animations allow
 - Repeating
 - Richer behavior with more keyframes
- DOM events

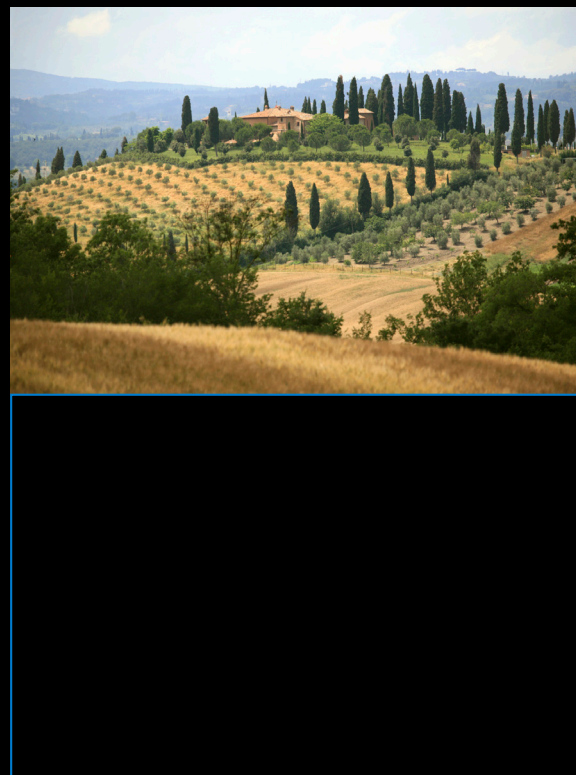
CSS Keyframe Animation Properties

- Apply a set of keyframes
 - `@-webkit-keyframes name { ... }`
 - `-webkit-animation-name`
- Length of one cycle
 - `-webkit-animation-duration`
- Number of cycles
 - `-webkit-animation-iteration-count`
- Reversing
 - `-webkit-animation-direction`
- Delay before start
 - `-webkit-animation-delay`

Drop Animation

```
@-webkit-keyframes drop {  
  0%   { -webkit-transform: translateY(-100%); }  
  70%  { -webkit-transform: translateY(0);    }  
  83%  { -webkit-transform: translateY(-10%); }  
  100% { -webkit-transform: translateY(0);    }  
}
```

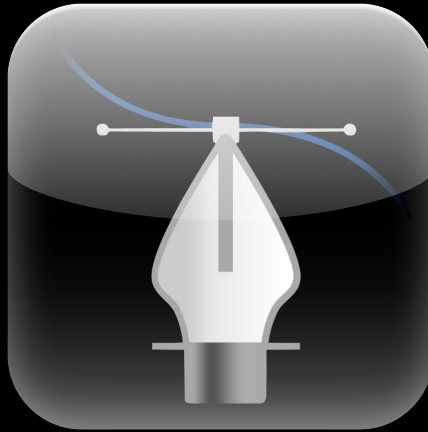
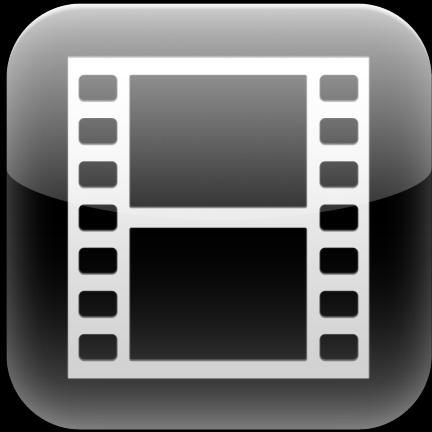
```
#photo-container.final .incoming {  
  -webkit-animation: drop 1s;  
}
```



Transitions vs. Keyframe Animations

	Transitions	Keyframe Animations
Triggered	Automatically	Explicitly
Timing Function	✓	✓
Keyframes		✓
Repeating		✓
Alternating		✓

Apply CSS Effects To...



Tile Effects



Tiling With CSS Background Image



Tiling with CSS Background Image

```
<div class="tile-container">
```



Tiling with CSS Background Image

```
<div class="tile-container">
```



Tiling with CSS Background Image

```
<div class="tile-container">
```

background-image:
background-position:



Tiling with CSS Background Image

```
for (var row = 0; row < rowCount; ++row) {
  for (var col = 0; col < columnCount; ++col) {
    var curTile = document.createElement('div');

    var tileX = col * tileWidth;
    var tileY = row * tileHeight;

    curTile.style.left = tileX + 'px';
    curTile.style.top = tileY + 'px';

    curTile.style.backgroundImage = 'url(landscape.jpg)';
    curTile.style.backgroundPosition = (-tileX) + 'px ' + (-tileY) + 'px';

    tileContainer.appendChild(curTile);
  }
}
```

Tile Drop Animation

```
@-webkit-keyframes incoming-tile-drop {  
  0%    { -webkit-transform: translateY(-100%);  
          opacity: 0;  
        }  
  70%   { -webkit-transform: translateY(0);  
          opacity: 1;  
        }  
  83%   { -webkit-transform: translateY(-10%); }  
  100%  { -webkit-transform: translateY(0);    }  
}
```

```
#photo-container.final .tile-container.incoming > div {  
  -webkit-animation: incoming-tile-drop 1.5s ease-in;  
}
```

Building In the Tiles



```
for (var row = 0; row < rowCount; ++row) {  
  for (var col = 0; col < columnCount; ++col) {  
    ...  
  }  
}
```

Animation Delay



```
for (var row = 0; row < rowCount; ++row) {  
  for (var col = 0; col < columnCount; ++col) {  
    ...  
    curTile.style.webkitAnimationDelay = Math.random() + 's';  
  }  
}
```

Tile Effects Summary

- Use a grid of `<div>` elements with `background-image` style
- Delays are useful for creating interesting effects
 - `-webkit-transition-delay`
 - `-webkit-animation-delay`

Demo

Enrica Casucci
Safari and WebKit Engineer

Support for CSS Effects



2D Transforms	✓	✓	✓	✓
3D Transforms	✓	✓		
Transitions	✓	✓	✓	✓
Animations	✓	✓	✓	

Summary

- CSS Transitions and Animations
 - Compelling animations with little effort
 - Good performance
- CSS Transforms
 - Position elements in 2D and 3D
 - Smoothly animate
- Works well with all HTML elements

More Information

Vicki Murley

Safari Technologies Evangelist
vicki@apple.com

Documentation

Safari Dev Center
<http://developer.apple.com/safari>

CSS Working Group

<http://www.w3.org/Style/CSS/current-work>

Apple Developer Forums

<http://devforums.apple.com>

Related Sessions

CSS Effects, Part 1: UI Elements and Navigation

Marina
Tuesday 3:15PM

Adding Touch and Gesture Detection to Web Pages on iPhone OS

Nob Hill
Wednesday 2:00PM

Creating Info Graphics with Standard Web Technologies

Marina
Wednesday 4:30PM

Labs

CSS Effects Lab	Internet & Web Lab A Wednesday 9:00AM–1:45PM
WebKit Lab	Internet & Web Lab B Wednesday 2:00PM–6:30PM
Safari on iPhone OS Lab	Internet & Web Lab B Thursday 9:00AM–11:15AM
Safari on iPhone OS Lab	Internet & Web Lab B Thursday 2:00PM–4:00PM

Q&A



