

Basic HTML

HTML5

ABOUT WEB ELEMENTS

Weaving your threads on the net

CSS

JavaScript

CSS

Cascading Style Sheets

Mr. K's Technology



Welcome to HTML.net
Free tutorials on HTML and CSS
<http://www.html.net/>



ABOUT
WEB
ELEMENTS

CSS
{ } , ;



<http://www.w3schools.com/css/>

Information for this Unit
was obtained from



<http://www.tizag.com>

CSS

Cascading Style Sheets

Mr. K's Technology

Mr. K's
Technology



The WebKit Open Source Project

CSS (Cascading Style Sheets)

Welcome to the website for the WebKit Open Source Project!

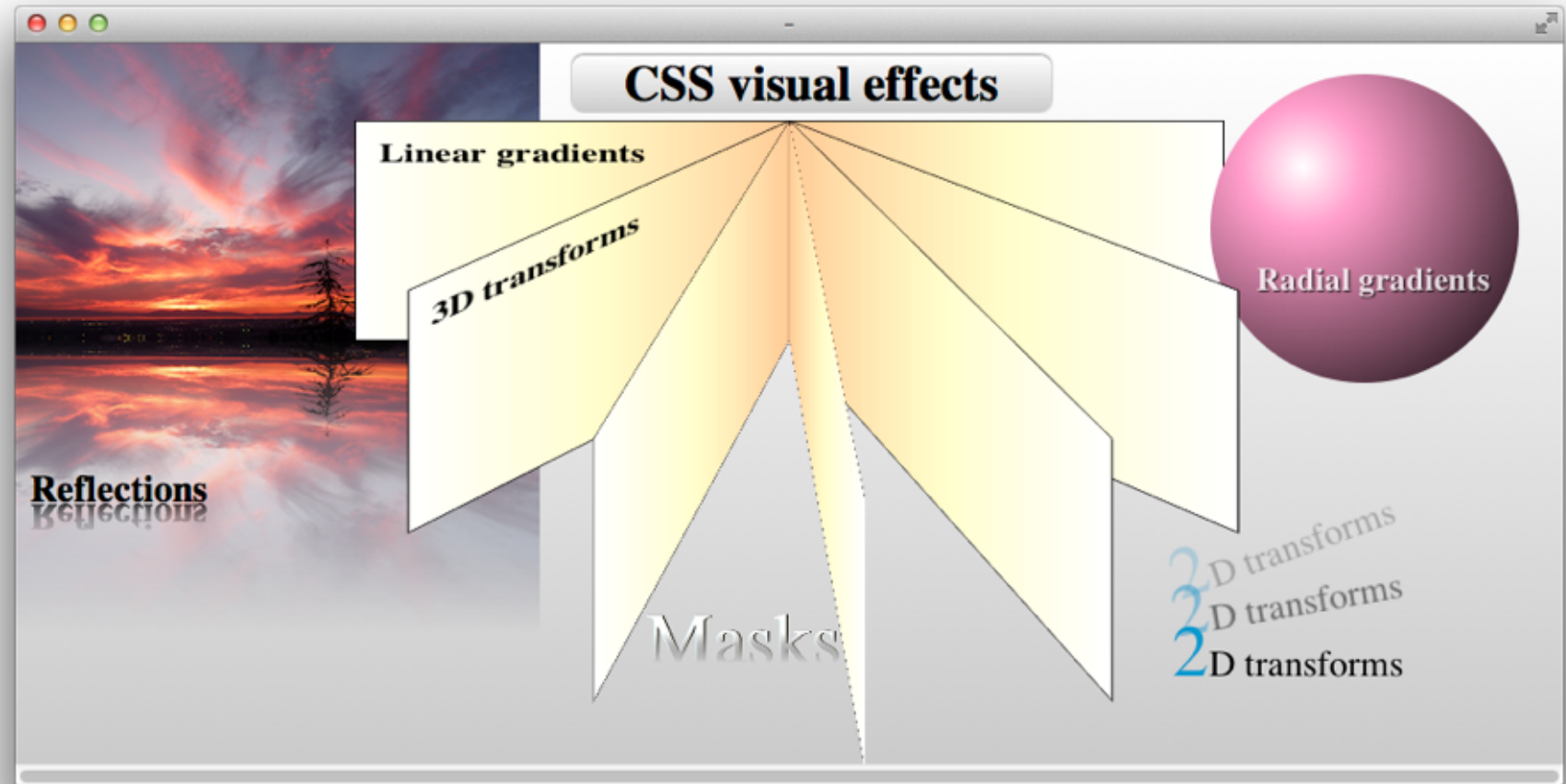
WebKit is an open source web browser engine. WebKit is also the name of the Mac OS X system framework version of the engine that's used by [Safari](#), Dashboard, Mail, and many other OS X applications. WebKit's HTML and JavaScript code began as a branch of the [KHTML](#) and KJS libraries from [KDE](#).



About Web Elements is an independent publication and has not been authorized, sponsored, or otherwise approved by Apple Inc.



Use CSS to create stunning visual effects—masks, gradients, reflections, lighting effects, animations, transitions, 3D rotations, and more. Apply any or all of these effects interactively, triggered by mouse events or touch events; make HTML elements visibly respond to the user—without requiring plug-ins, graphics libraries, or elaborate JavaScript programs.





There are advantages to using CSS instead of graphic images to create visual effects:

- Because they are resolution-independent, CSS effects scale up smoothly when zoomed.
- Text formatted with CSS is searchable; images are not.
- CSS is compact and compresses well compared with graphic images.
- CSS is just text; it can be quickly modified using a text editor or the output of a script.

Safari supports CSS visual effects on Mac OS X and iOS.



Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

Here's what we are planning to demonstrate

Using Gradients (17),
Using a Gradient as a Mask (18),
Using Reflections (18),
Using CSS Filters (18),
Animating CSS Transitions,
Animating With Keyframes,
Using 2D and 3D Transforms, and
Adding Interactive Control to Visual Effects.





Using Masks

A mask is a two-dimensional shape that acts as an overlay on an element and clips the element visually or renders portions of the element transparent or translucent. A mask consists of an image or a gradient that contains an alpha channel; where the alpha value of the mask is 0, the underlying element is transparent, or masked out; where the alpha value is 1, the underlying element displays normally; where the alpha value is between 0 and 1, the underlying element is proportionally transparent. Only the alpha values of the mask are used; any other color values are ignored.

An element's mask applies to its children and their descendants. The children and descendants may also have masks, in which case the masks are stacked. The transparent areas of a masked element reveal the underlying webpage contents, which may also contain masked elements; these masks too are stacked. The final image is rendered after concatenating all the masks in the stack, tiling and stretching them as specified.

When a mask is applied to an element, both the contents and background of the element are masked. A mask may be applied to the entire element or it may exclude either the border or the combined border and padding.

Using an Image as a Mask

To apply a mask using an image, first create a .png or .svg image that contains an alpha channel. Pass the URL of this image into either the `-webkit-mask-image` property or the `-webkit-mask-box-image` property. The `-webkit-mask-image` property uses the native size of the mask image. The `-webkit-mask-box-image` property scales the mask image to meet the edges of the element.

If the image used as a mask consists of an opaque shape on a transparent background, the mask acts as a “cookie cutter” to clip an element in the shape of the mask. For example, the following snippet applies a heart-shaped mask to an image. Figure 2-1 shows the result.

```

```

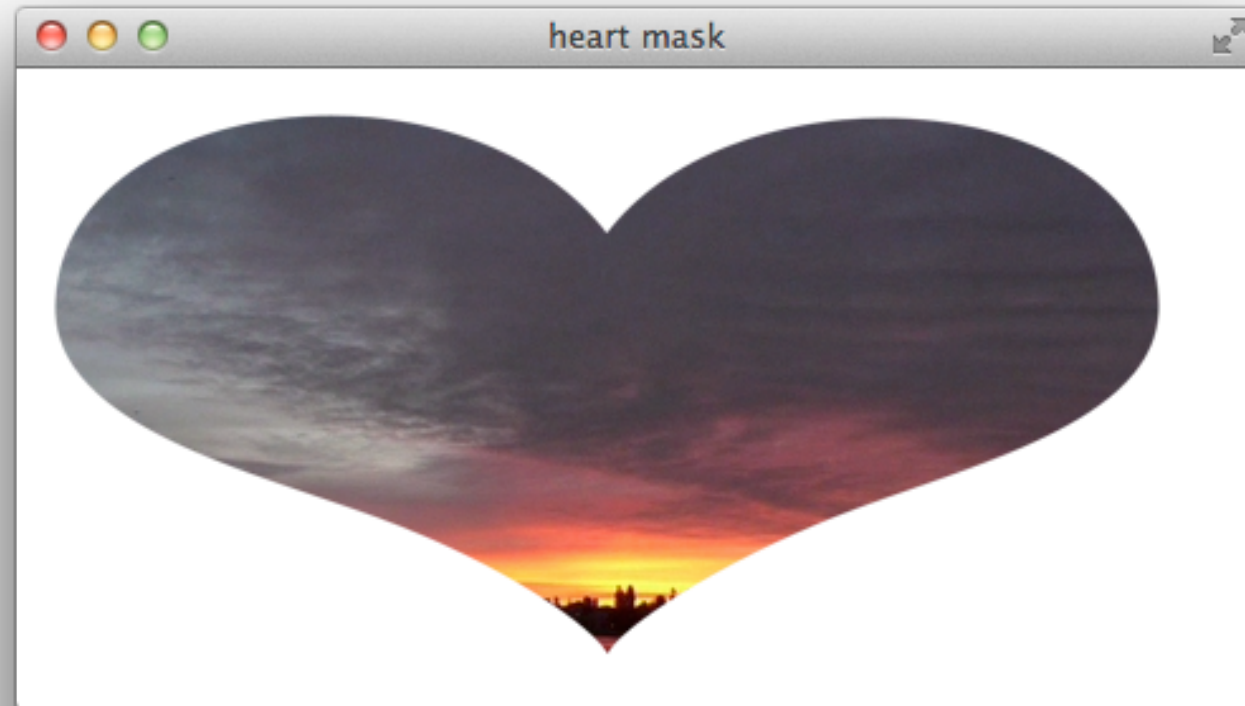


Figure 2-1 Heart-shaped “cookie-cutter”

The `-webkit-mask-box-image` property takes optional arguments that work like the `-webkit-border-image` parameters. You can specify offsets into the mask image to slice it into top, right, bottom, and left images, applying the mask slices, stretched or repeated, to the edges of an element. For example, the following snippet uses a 50-pixel heart-shaped mask and adds a 50-pixel border to the image. The snippet also adds parameters to the `-webkit-mask-box-image` property, setting the slice size to the whole mask image and specifying repeat both horizontally and vertically. Figure 2-2 shows the result.

```

```

`<img src=` *is one HTML line of code*

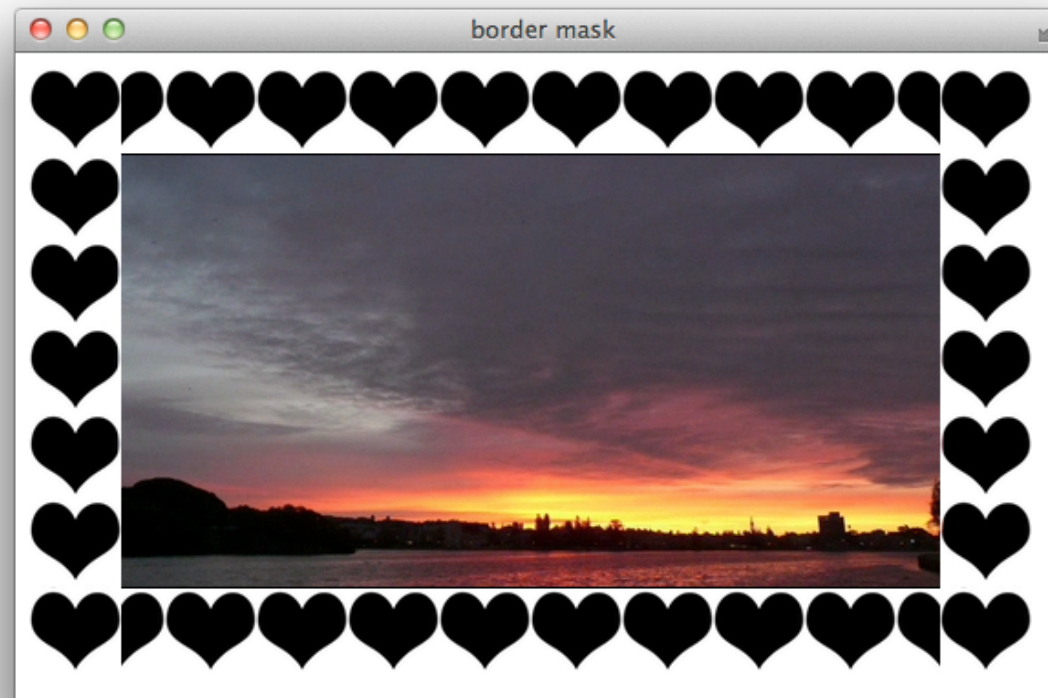


Figure 2-2 Masking a border



All **webkit** elements can be used within an internal style sheet



As mentioned previously, you can mask any HTML element, and you can concatenate masks by inheritance and by stacking masked elements. For example, Listing 2-1 creates a pair of masked `div` elements, each of which contains a paragraph element. The example positions the second `div` element so that it partly overlaps the first `div` element. The paragraph elements are masked by their parent `div` elements, and the two masks are stacked. The result is shown in Figure 2-3.

Listing 2-1 Stacking masked elements

```
<html>
<head> <title>stacked masks</title>
</head>
<body style="background:beige;">

<div style="position:absolute; left:20px; top:20px; background:#e9e9e9; width:240; -webkit-mask-box-image: url(heartmask.png);">

    <p>This div element has a heart-shaped mask.This div element has a heart-shaped mask.This div element has a heart-shaped
mask.This div element has a heart-shaped mask.This div element has a heart-shaped mask.This div element has a heart-shaped
mask.This div element has a heart-shaped mask.This div element has a heart-shaped mask.
    </p>
</div>

<div style="position:absolute; left:80px; top:80px; background:white; width:240; -webkit-mask-box-image: url(heartmask.png);">

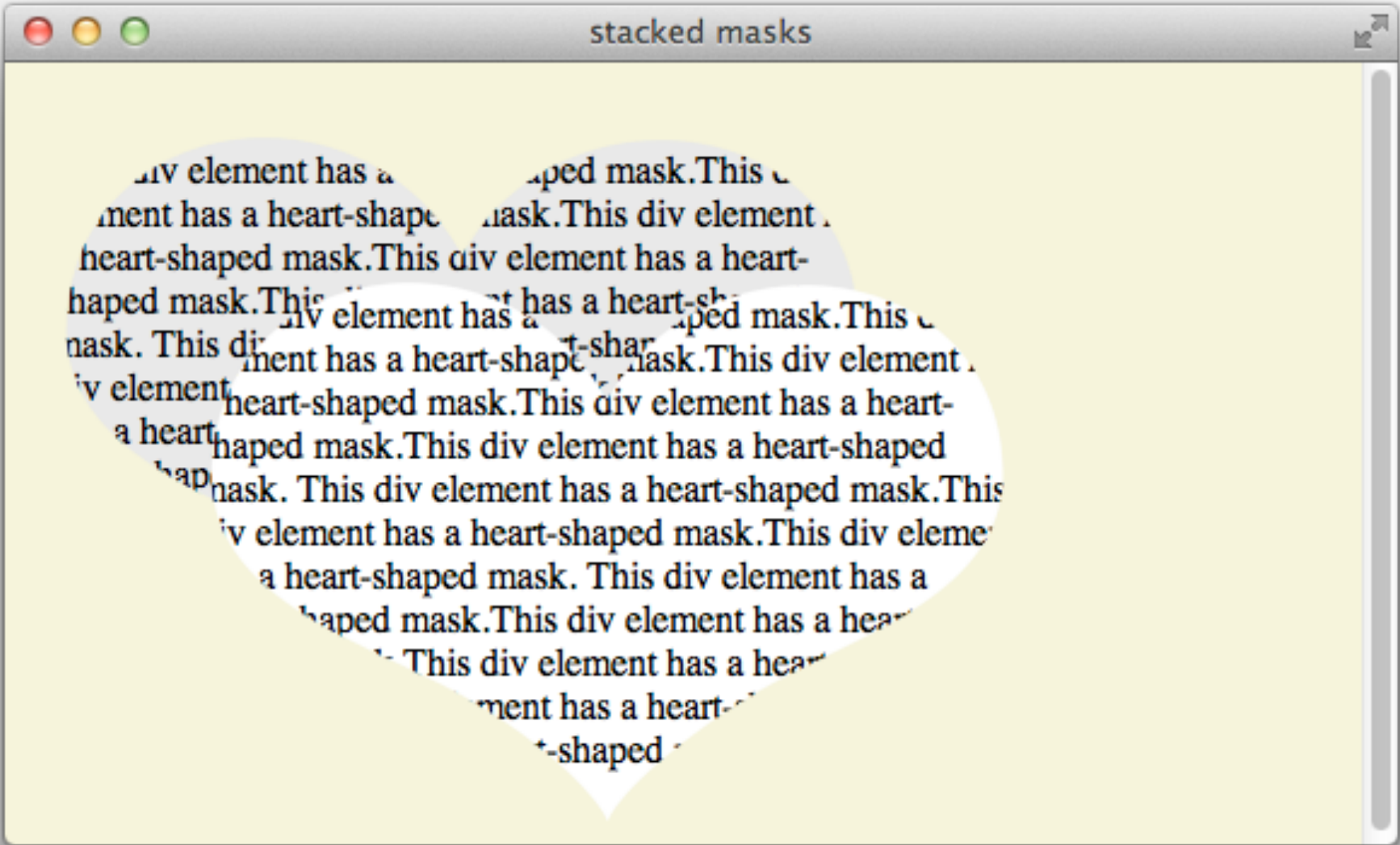
    <p>This div element has a heart-shaped mask.This div element has a heart-shaped mask.This div element has a heart-shaped
mask.This div element has a heart-shaped mask.This div element has a heart-shaped mask.This div element has a heart-shaped
mask.This div element has a heart-shaped mask.This div element has a heart-shaped mask.
    </p>
</div>

</body>
</html>
```

revealed on next slide

All **webkit** elements can be used
inline within the HTML style tag

Figure 2-3 Stacking masks



You can use a mask to fade an element gradually into the background by including semitransparent pixels in the mask. For example, the following snippet applies a mask image with a solid center and increasingly transparent edges to the image of a field. The `-webkit-mask-box-image` property is used, so the mask is scaled to fit the `img` element. The result is shown in Figure 2-4.

```

```

Figure 2-4 Applying a mask with a fuzzy border



Image



Mask



Result

Using a Gradient as a Mask

Because gradients can be used in place of any image, a gradient can be passed as a mask. The following snippet defines a gradient that goes from opaque to transparent as a mask for an `img` element. The result is shown in Figure 2-5.

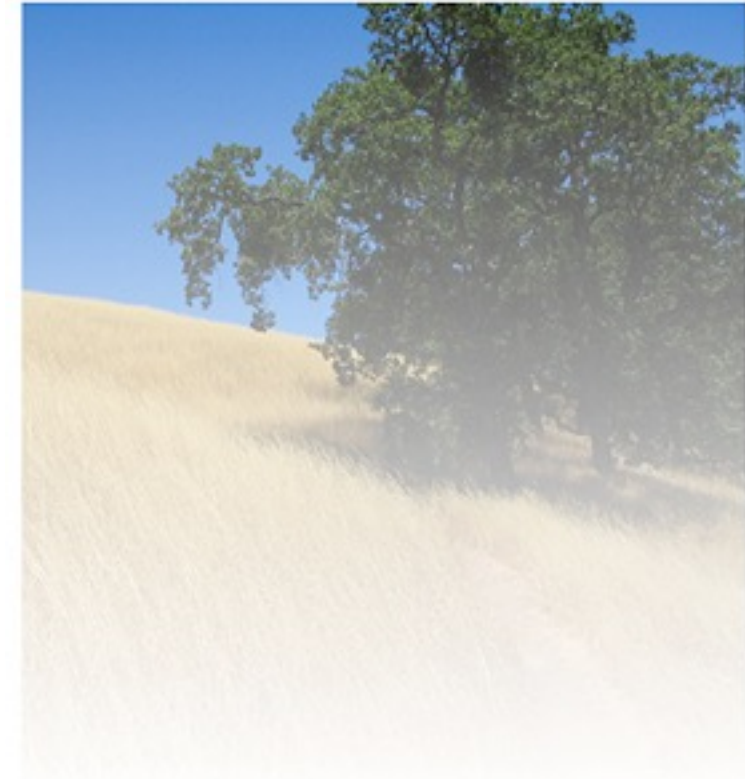
```

```

Figure 2-5 Result of applying a gradient mask

Note: The gradient in the example uses black as the opaque color, but any completely opaque color would be equivalent.

The example just given goes from opaque to transparent over the height of the image, but you can use different gradient parameters to achieve different effects. The following snippet creates a gradient from left to right and uses color stops to fade an image sharply at the edges, as shown in Figure 2-6. Notice that the image's border radius works in concert with the mask.



All **webkit** elements can be used from an external style sheet

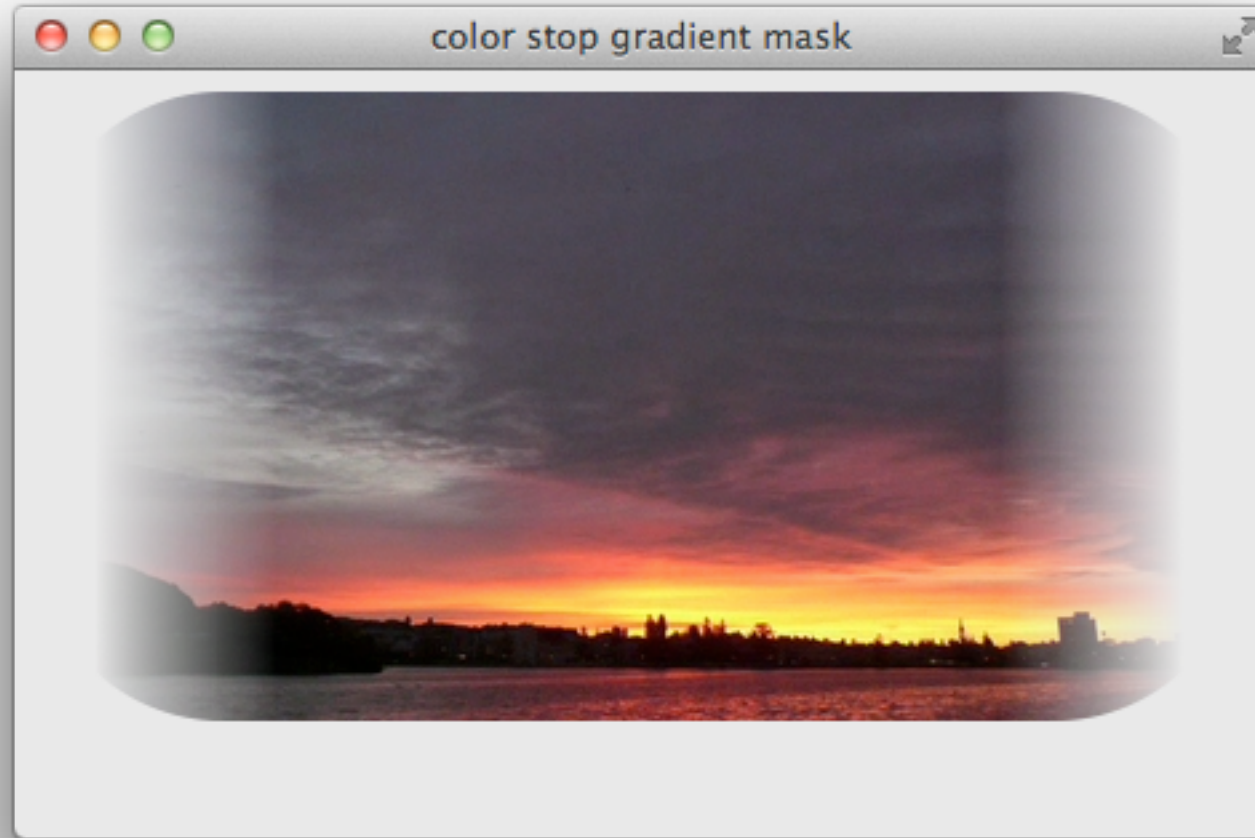


```

```

Figure 2-6 Horizontal gradient mask with color stops

`<img src=` *is one HTML line of code*



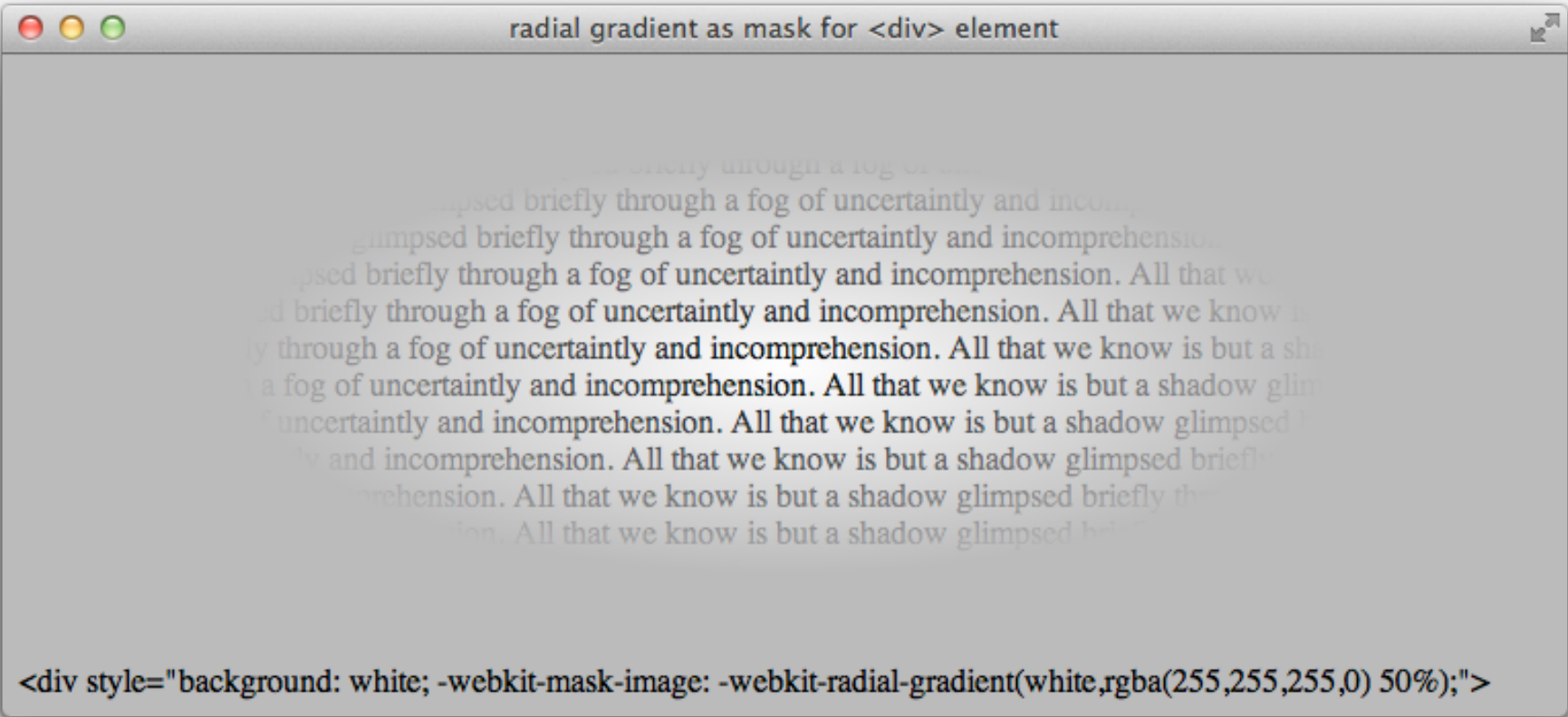
You can apply a gradient mask to any HTML element, not just images. For example, the following snippet creates a radial gradient as a mask for a `div` element containing several paragraphs of text. The result is shown in Figure 2-7.

`<div style="background: white; padding: 20px; -webkit-mask-box-image: -webkit-radial-gradient(white,transparent 50%);">`

`<div style=` *is one HTML line of code*

Figure 2-7 Masking text with a radial gradient

That's the complete inline HTML style but remember to close the `<div>` tag with `</div>`



Using Reflections

A reflection is a mirror image, optionally with its own mask. You can add a reflection to any visible HTML element. Reflections update automatically as the element's appearance changes. If you hover over a link with a reflection, for example, you see the hover effect in the reflection. If you add a reflection to a [video](#) element, the video plays in the reflection as well. Resize the element and the reflection is resized. Reflections are not interactive elements, however—clicking or touching a reflection does not generate any events.

Reflections have no effect on layout other than being part of a container's overflow, and are similar to box shadows in this respect. In other words, you must position elements explicitly to allow space for reflections; if you add a reflection below an image, and follow the image with a paragraph of text, the reflection may cover the text (depending on the size of the image, padding, margins, and so on).



Adding a Reflection

Use the `-webkit-box-reflect` property to add a reflection to an element. The first, mandatory, parameter is the direction in which the reflection projects, which can be `above`, `below`, `left`, or `right`. For example, the following snippet adds a reflection below an `h1` element, as Figure 3-1 illustrates:

```
<h1 style="-webkit-box-reflect: below;">
```

Reflection in Blue

```
</h1>
```

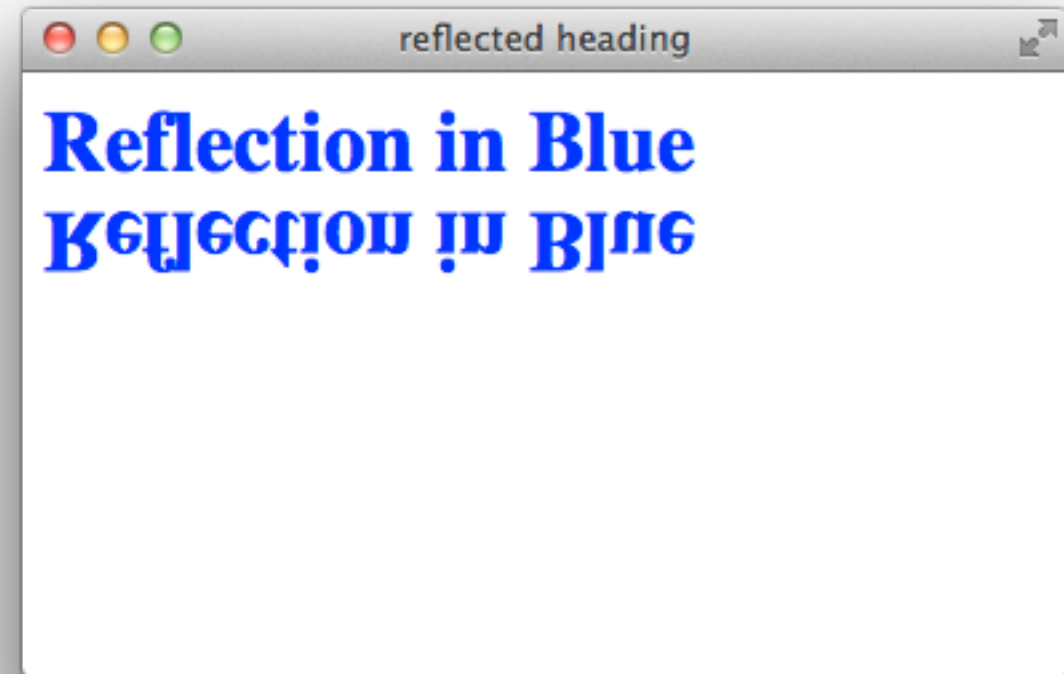


Figure 3-1 Reflection below a heading

The reflection is the same visible size as the element. The element border and padding are included in the reflection.



Adjusting the Reflection's Position

The first argument of the `-webkit-box-reflect` property specifies the direction of the reflection. An optional second argument specifies the offset or space between the original image and the reflection.

By default, the reflection begins with no separation from the element's bounding box. To change this behavior, add an offset parameter. A positive offset causes the reflection to be separated from the element by additional space; a negative offset causes the reflection to overlap the element. For example, the bounding box of an `h1` element includes substantial white space; to tuck the reflection in closer to the text, add a negative offset. The following example adds an offset of `-15px` to the reflection. The result is shown in Figure 3-1:

```
<h1 style = "-webkit-box-reflect: below -15px;">
```

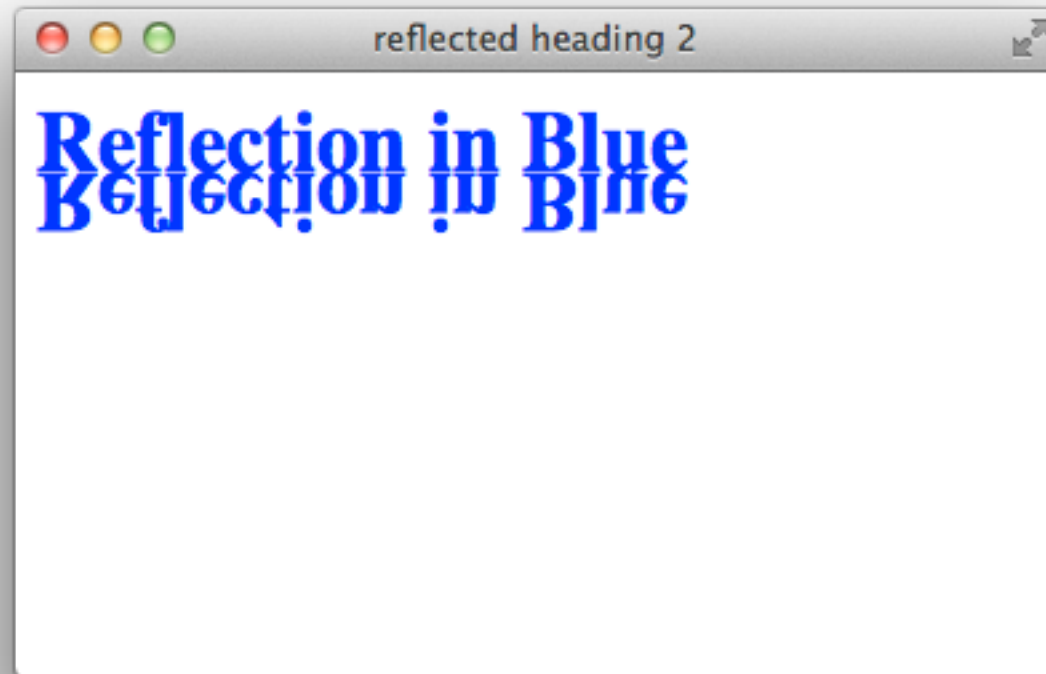


Figure 3-2 Reflection with negative offset

Masking a Reflection

An optional third argument to `-webkit-box-reflect` specifies a mask to apply to the reflection.

A mask renders part of the reflection transparent or translucent. The mask is either an image file, specified as a URL, or a gradient. A mask image file can be a `.png`, an `.svg`, or a `.gif` with transparent areas. The alpha value of the mask is applied to the reflection on a pixel-by-pixel basis. Only the alpha value of the mask's color values has any effect on the reflection.

When creating a mask, shape it to mask the target element itself, not to mask the reflection. The reflection shows a mirror image of the masked element, even though the element does not appear masked. The following snippet adds a mask image to a reflection. Figure 3-3 shows the resulting masked reflection side-by-side with the mask image.

`style="-webkit-box-reflect: below 0px url(mask.png);"`

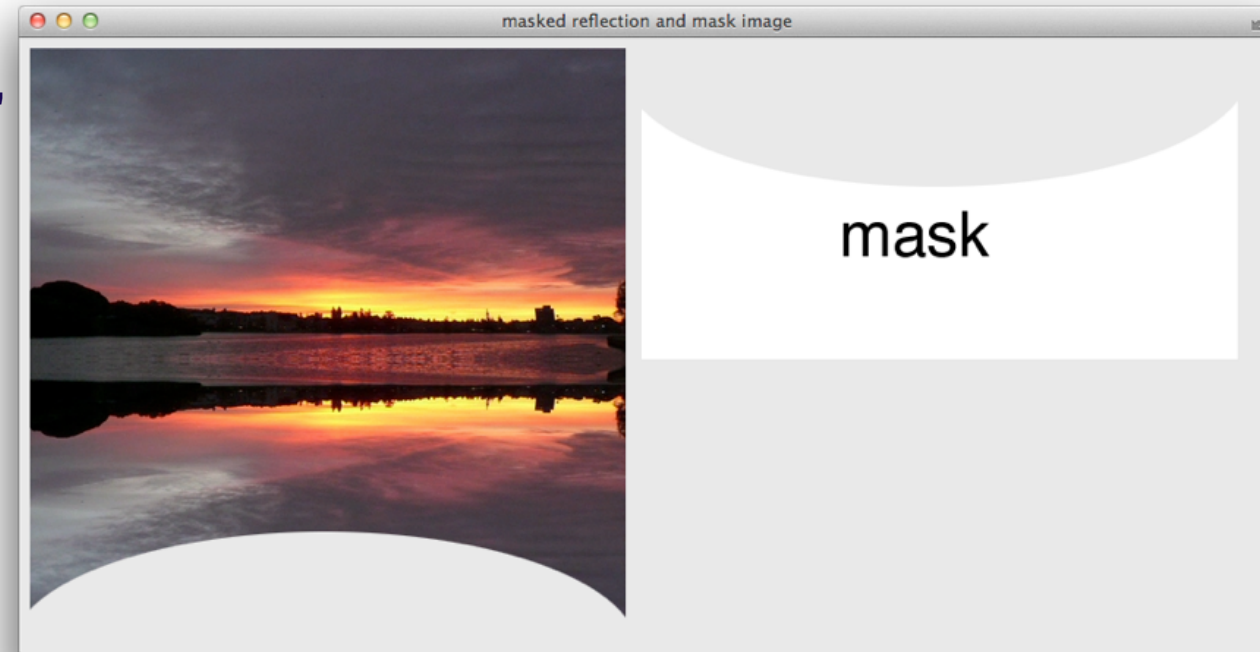
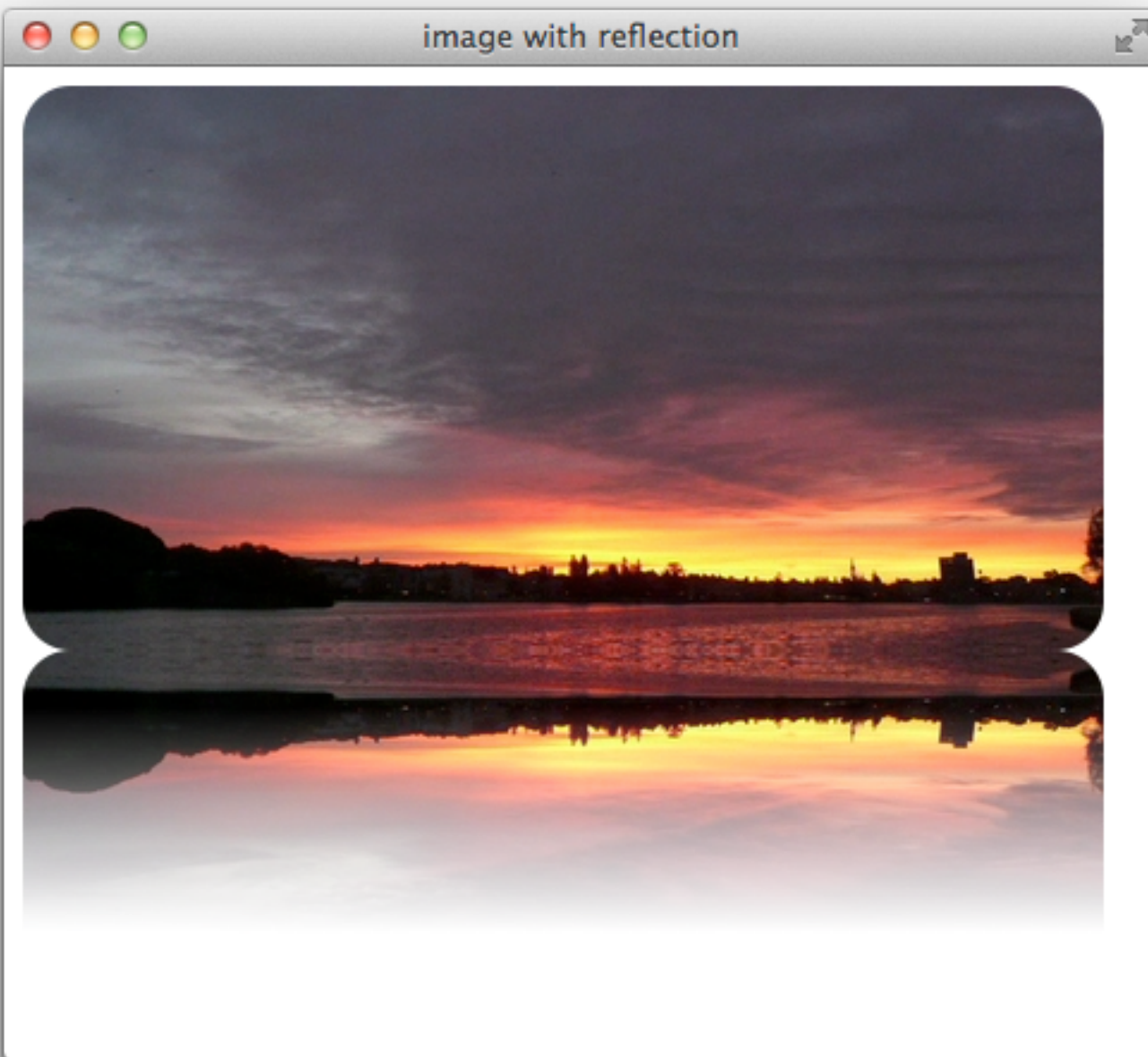


Figure 3-3 Reflection with image as mask



The following snippet adds a reflection below an image, using a gradient as a mask. Figure 3-4 shows the result.

```
<img src = "sunrise.png" style = " border-radius: 20px; -webkit-box-reflect: below 0px; -webkit-linear-gradient(transparent, transparent 50%, white 90%);">
```

<img src= *is one HTML line of code*

Figure 3-4 Image with reflection and gradient mask

Note that the reflection includes the rounded borders of the img element. Notice also that the gradient goes from transparent at the top to opaque at the bottom—the opposite of its appearance in the reflection; this is because the reflection shows a mirror image of the masked element.

Inner Reflections

When you reflect an element, all of the element's descendants are displayed in the reflection. If you add a reflection to a child element, the child's reflection is rendered first; this inner reflection is included in the parent element's reflection. The following snippet adds a reflection to a `div` element containing an `img` element that has its own reflection. Figure 3-5 shows the result:

```
<div style="-webkit-box-reflect: below;">
<img src = "sunrise.png" height="177" width="339" style = " border-radius:
20px; -webkit-box-reflect: below 0px; -webkit-linear-
gradient(transparent,transparent 80%,white);">
  <P>
  <P>Text...
  <P>Text...
</div>
```

`<img src=` *is one HTML line of code*

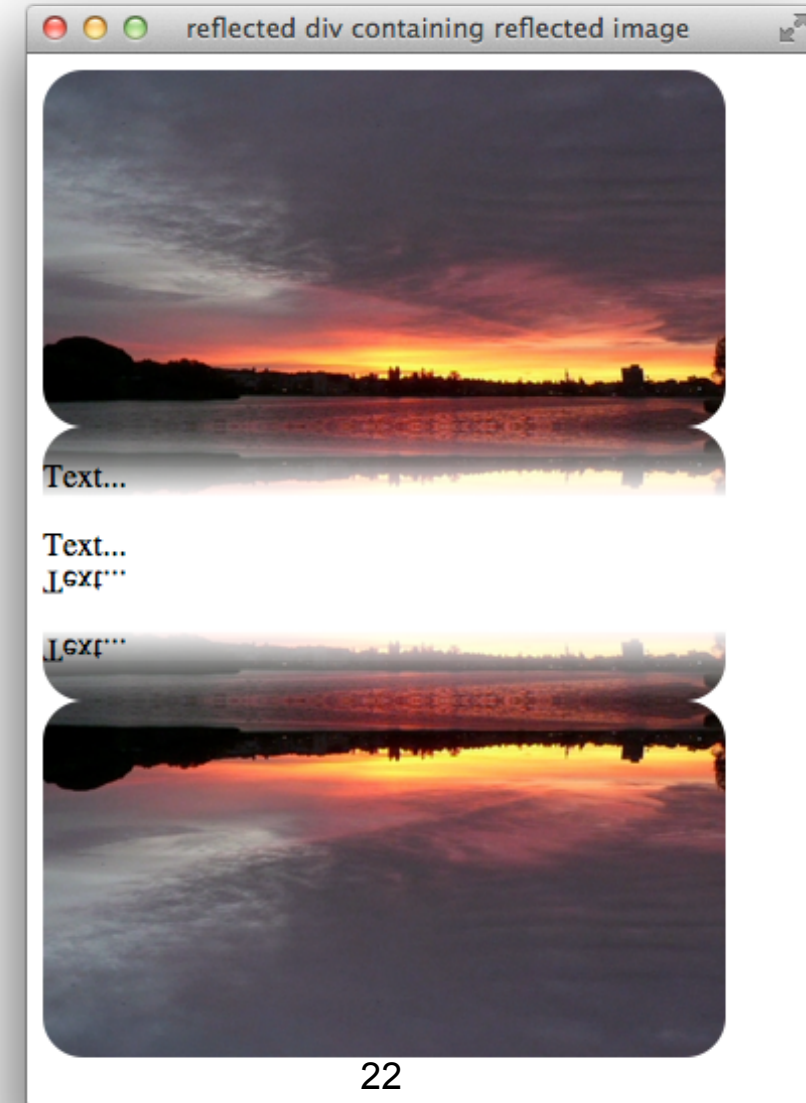


Figure 3-5 Inner reflection, reflected

Using CSS Filters

Safari 6 and later supports CSS filters, or special visual effects, that you can apply to many elements, including videos (see [Figure 4-1](#)). These hardware-accelerated filters (such as brightness, contrast, saturation, and blur) can be stacked on top of and animated against one another.

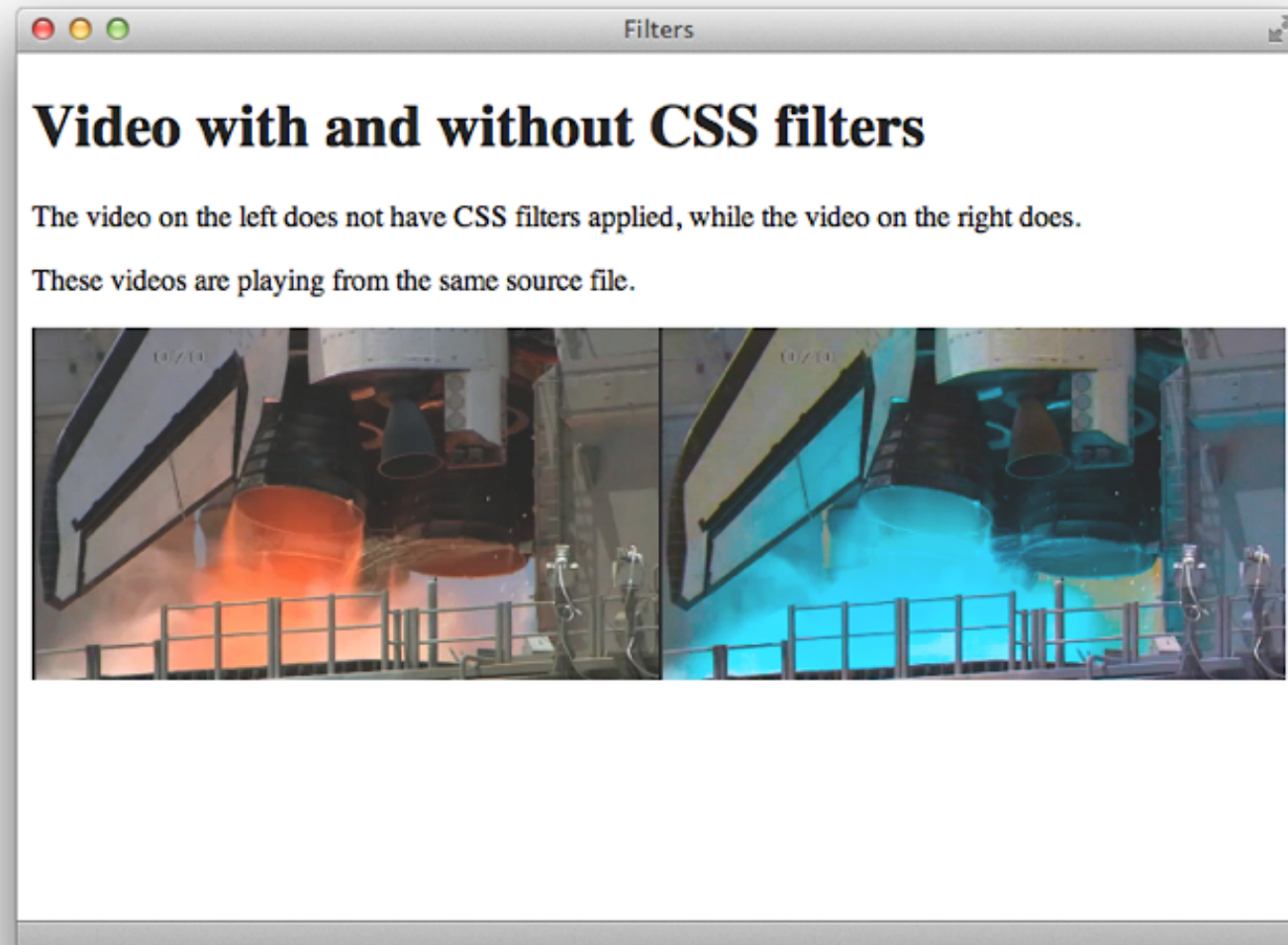


Figure 4-1 CSS filters on a video

Using CSS Filters

To add a CSS filter to an HTML element, include the `-webkit-filter` property in the element's CSS declaration, as shown in [Listing 4-1](#). You can list as many of the functions as you'd like.

Listing 4-1 Applying CSS filters to HTML elements

```
<!doctype html>
<html>
<head>
  <title>Filters</title>
  <style>
    video {
      float: left;
      width: 50%;
    }
    .filtered {
      -webkit-filter: hue-rotate(180deg) saturate(200%);
    }
  </style>
</head>
<body>
  <h1>Video with and without CSS filters</h1>
  <p>The video on the left does not have CSS filters applied, while the video on the right does.</p>
  <p>These videos are playing from the same source file.</p>
  <video src="shuttle.m4v" autoplay></video>
  <video src="shuttle.m4v" autoplay class="filtered"></video>
</body>
</html>
```

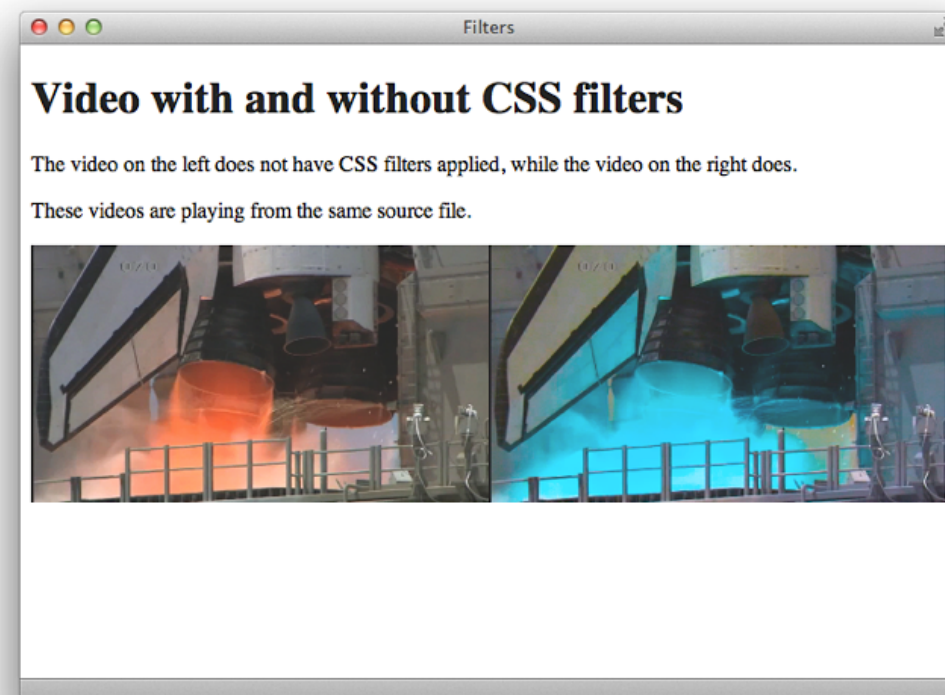


Figure 4-1 CSS filters on a video



About Web Elements



CSS

Cascading Style Sheets

Mr. K's Technology

Using the information you learned about masks and reflections, create a page similar to the one on the left

ABOUT WEB ELEMENTS

Webkits

About Web Elements is an independent publication and has not been authorized, sponsored, or otherwise approved by Apple Inc.

SAVE YOUR WORK

CLOSE ALL OPEN PROGRAMS

ABOUT
WEB
ELEMENTS

LOG OFF

CSS
Cascading Style Sheets
Mr. K's Technology



Safari Developer Library