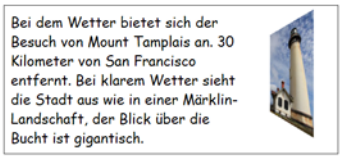

Abb. 11–13 Sowohl Internet Explorer 8 + 9 als auch Opera unterstützen weder `perspective` noch den Wert `rotateY` für `transform`. Die Browser-Präfixe sind hier nur für die Webkit-Browser (`-webkit-`) und Firefox (`-moz-`) angegeben.

```
p {
   padding: 10px;
   background-color: transparent;
   -webkit-perspective: 150px;
   -moz-perspective: 150px;
   perspective: 150px;
}
p img {
   float: right;
   margin: 20px;
   -webkit-transform: rotateY(-50deg);
   -moz-transform: rotateY(-50deg);
   transform: rotateY(-50deg);
}
```

11.3 Video/Audio einbinden

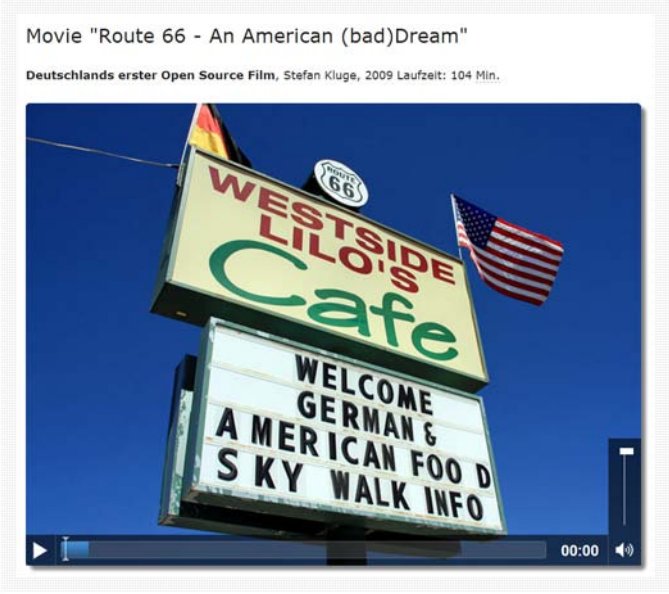

Abb. 11–14 Direkte Videoeinbindung über das `video`-Element

`video`- und `audio`-Elemente sind wahrscheinlich zwei der wichtigsten Neuerungen in HTML5. Gerade in Zeiten von YouTube & Co. sind multimediale Inhalte aus dem Web nicht mehr wegzudenken. Das Dilemma bestand in der Vergangenheit darin, dass die Einbindung von Videos oder Audios nur über eine komplizierte Einbindung per `object`- `param` und `embed`-Tags möglich und dann immer noch zusätzlich ein Flash-Plug-in notwendig war, um Medien abspielen zu können. Langfristig werden diese neuen Tags ein Flash-Plug-in überflüssig machen.

Die Browserunterstützung von `video`- und `audio`-Elementen erweist sich mit Ausnahme des Internet Explorer 8 mittlerweile als gut. Positiv ist zudem, dass auf diese Elemente auch CSS-Eigenschaften, beispielsweise runde Ecken oder Schlagschatten, angewendet werden können, um die Multimediainhalte zusätzlich ansprechender zu präsentieren.

Es wäre allerdings zu schön, wenn sich deren Einbindung auch noch als einfach erweisen würde. Leider haben sich die Browserhersteller bislang auf kein einheitliches Dateiformat geeinigt, sodass die Video- bzw. Audio-Datei noch in mehreren Formaten hinterlegt werden muss, damit alle Browser etwas anzeigen können.

```
<video controls preload poster="img/lilos-cafe-route66.jpg">
   <source src="media/route66.mp4" type="video/mp4">
   <source src="media/route66.ogv" type="video/ogg">
</video>
```

Die Einbindung eines Videos im Beispiellayout erfolgt über das `video`-Element. Hier stehen verschiedene Attribute zur Verfügung. Einige davon sind leere Attribute, die einfach weggelassen werden, wenn die Funktion nicht gewollt ist.

Leere HTML-Attribute: Kapitel 24.3.1

- `controls`: sorgt dafür, dass das Video eine Bedienleiste erhält. Benötigt man keine Bedienelemente, wird das `controls`-Attribut einfach weggelassen (leeres Attribut).
- `autoplay`: startet das Video automatisch beim Laden der Seite (leeres Attribut).
- `loop`: spielt ein Video in einer Endlosschleife ab (leeres Attribut).
- poster: referenziert eine Grafik, die als Startbild verwendet wird. Ohne Grafik bleibt der Player zu Beginn schwarz.
- `width` und `height`: Breiten- und Höhenangaben für das Video. Allerdings fügt sich das Video mit seinen eigenen Proportionen in den vorgegebenen Platz ein, mit diesen Werten wird lediglich die reservierte Fläche für das Video-Element angegeben. Beide Eigenschaften können auch über CSS gesteuert werden.
- `preload`: steuert, ob die Videodatei beim Laden der Seite vorgeladen wird oder nicht beziehungsweise ob nur Metadaten des Videos geladen werden (mögliche Werte: `auto`, `metadata`, `none`).

Innerhalb des `video`-Tags wird über das `source`-Tag die eigentliche Mediendatei angegeben und diese über das `src`-Attribut referenziert. Zusätzlich kann über das Attribut `type` die Codierung der Videodatei angegeben werden; allerdings sind die Browser so intelligent, dass sie das Medienformat automatisch erkennen.

Im Quelltext sehen Sie zwei Angaben von `source`. So wie die Browser das Medienformat automatisch erkennen, so verwenden sie auch automatisch jene Datei, die sie darstellen können. Alle weiteren `source`-Tags werden ignoriert, sobald der Browser eine verwendbare Datei gefunden hat. Im Folgenden die Medienformate und Browserunterstützung im Detail:

- `type="video/mp4"`: gibt an, dass eine MP4-Datei vorliegt (das Videoformat selbst wird als H.264-Codec bezeichnet). Wird von Internet Explorer 9, Safari und Google Chrome angezeigt.
- `type="video/ogg"`: der Typ für eine Ogg-Vorbis-Containerdatei und ein Alternativformat zu MP4. Wird von Firefox, Opera und ebenfalls Google Chrome unterstützt.

```
audio,
video {
   box-shadow: gray 5px 5px 5px;
   border-radius: 5px;
}
```

Für das Beispiellayout wird für `video`- und `audio`-Elemente ein leichter Schlagschatten über `box-shadow` und die Ecken über `border-radius` definiert.

Für Abwärtskompatibilität sorgen

Abb. 11–15
Älteren Browsern kann das Video zum Download angeboten werden.

```
<video controls preload poster="img/lilos-cafe-route66.jpg">
   <source src="media/route66.mp4" type="video/mp4">
   <source src="media/route66.ogv" type="video/ogg">
   <p>Ihr Browser unterstützt die direkte Anzeige von Videos nicht.<br>
      Das Video zum Download: <a href="media/route66.mp4">
         Route 66 - An American (bad)Dream</a>
   </p>
</video>
```

Anwender von älteren Browsern bekommen das HTML zu sehen, welches von den betreffenden Browsern verstanden wird – in diesem Fall das `p`-Element, das innerhalb der `video`-Tags geschachtelt ist.

In zahlreichen Beispielen im Internet zum `video-Element` finden Sie die Fallback-Information »Ihr Browser unterstützt kein HTML5« oder Vergleichbares. Das ist nicht nett! Zum guten Stil gehört, ein »echtes« Fallback anzubieten, wie zum Beispiel einen Link zu der Video-Datei anzugeben, sodass sich der Nutzer das Video zumindest herunterladen kann.

Zusätzlich finden Sie im Quelltext unten zwei wichtige Angaben: das Dateiformat sowie die Dateigröße. Damit wissen Nutzer, was sie beim Klick auf den Link erwartet.

Alternativ besteht die Möglichkeit, einen Flash-Player zu implementieren, womit Videos auch in alten Browsern direkt angesehen werden können. Auch dieser wird dann innerhalb der `video`-Tags notiert. Eine beispielhafte Umsetzung finden Sie im Ergebnispaket.

Abb. 11–16
Das Aussehen der Bedienelemente hängt vom Browser ab. Hier die Bedienleiste eines `audio`-Elements in Opera.

```
1  <audio controls>
2     <source src="media/route_66_soundtrack.mp3">
3     <source src="media/route_66_soundtrack.ogg">
4     <p>Das Audio zum Download: <a href="media/route_66_soundtrack.mp3">
5           Soundtrack Route 66 (MP3, 44MB)</a></p>
6  </audio>
```

Die Einbindung von Audio-Dateien ist mit der von Videos vergleichbar. Auch hier unterstützen die verschiedenen Browser unterschiedliche Formate, die daher parallel angegeben werden müssen. Eine Übersicht, welcher Browser welches Audioformat unterstützt, finden Sie unter nebenstehendem Link.

Unterstützte Medienformate für video- und audio-Elemente: www.h5c3.de/link-11-2

controls-Attribut ist ein Muss!
Egal ob Audios oder Videos: Bieten Sie immer die Bedienleiste an. Nutzer sollten jederzeit die Möglichkeit haben, das Abspielen einer Datei selbst zu steuern! Auch dies ist ein wichtiger Grundgedanke der Barrierefreiheit.

Kapitel 11 Ergebnispaket: www.h5c3.de/listing-11-3