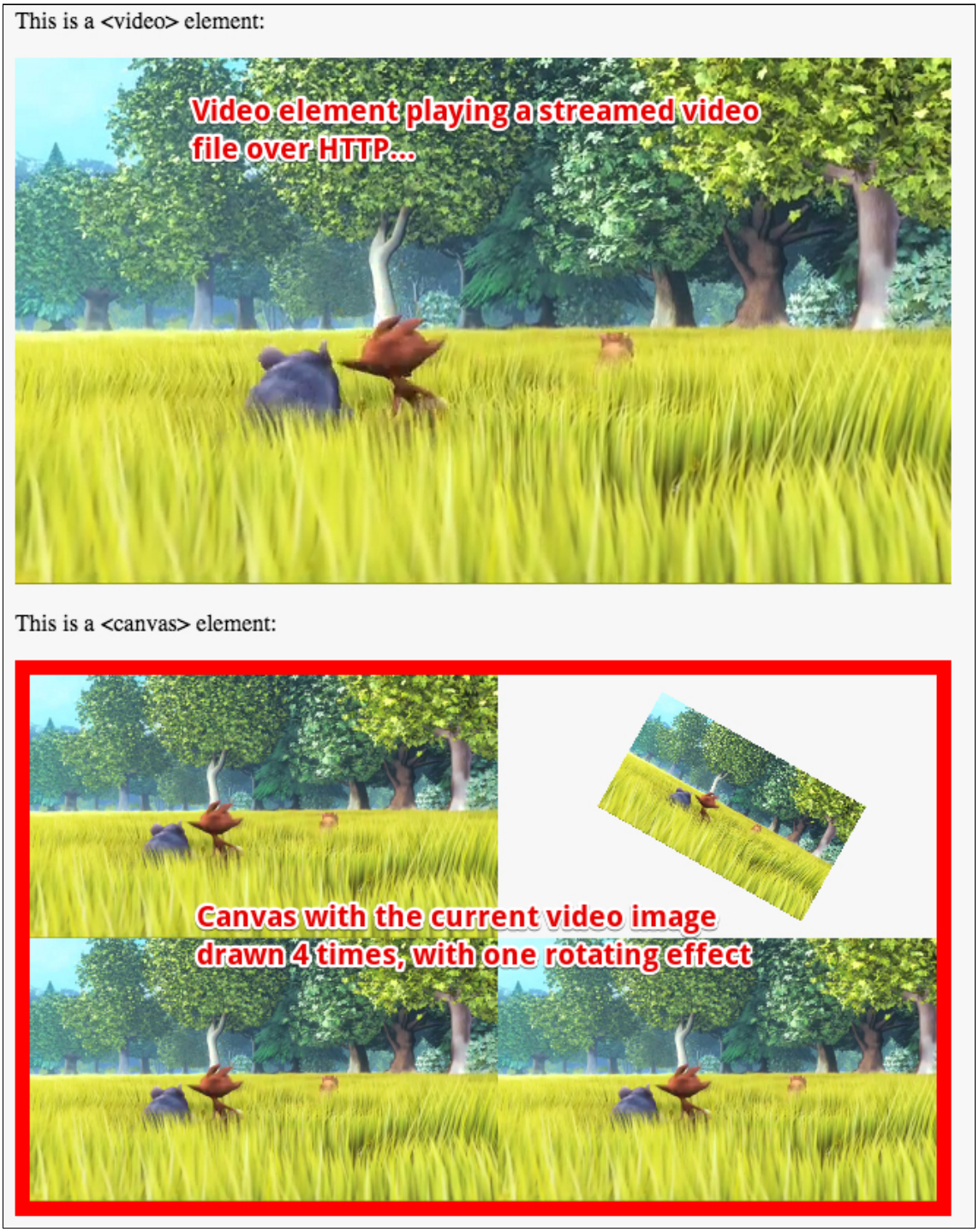
Drawing images from a video stream

The `drawImage(...)` function can take a video element as its first parameter. The image that will be drawn is the one currently played by the video stream. This can be done at video frequency on most modern computers or mobile devices.

Online example at http://jsbin.com/dajena/3/edit

This example shows:

- a `<video>` element on top, and four images drawn in a canvas below.
- The images are drawn every XXX milliseconds using the`setInterval(function, delay)` method.



Source code extract:

```
<script>
  var video;
  var canvas, ctx;
  var angle = 0;

function init() {
  video =document.getElementById('sourcevid');
  canvas =document.getElementById('myCanvas');
```

```
    ctx = canvas.getContext('2d');
10.
    setInterval("processFrame()", 25); // call processFrame each 25ms
  }

  function processFrame() {
    ctx.drawImage(video, 0, 0, 320, 180);

    drawRotatingVideo(480, 90);

    ctx.drawImage(video, 0, 180, 320,180);
    ctx.drawImage(video, 320, 180, 320,180);
  }

  function drawRotatingVideo(x, y) {
      // Clear the zone at the top right quarter of the canvas
23.   ctx.clearRect(320, 0, 320, 180);

    // We are going to change the coordinate system, save the context!
    ctx.save();
    // translate, rotate and recenter the image at its "real" center,
    //not the top left corner
    ctx.translate(x, y);
    ctx.rotate(angle += 0.01); // rotate and increment the current angle
    ctx.translate(-80, -45);

33.   ctx.drawImage(video, 0, 0, 160, 90);

    // restore the context
    ctx.restore();
  }
  </script>
  </head>

  <body onload="init()" >
42. <p>This is a <video> element: </p>
  <video id="sourcevid" autoplay="true"loop="true">
  <sourcesrc="http://www.craftymind.com/factory/html5video/BigBuckBunny_640x360.mp4"
          type="video/mp4" />
  <sourcesrc="http://www.craftymind.com/factory/html5video/BigBuckBunny_640x360.ogv"
          type="video/ogg"/>
  </video>
  <p>This is a <canvas> element: </p>
  <canvas id="myCanvas" width="620"height="360"></canvas>
  </body>
```

- *Line 11*: the call to `setInterval` will make the browser execute the `processFrame` function each 25ms.
- *Lines 15, 17 and 18*: in `processFrame`, `drawImage(...)` is called 3 times with the video element passed as

first parameter.

- *Line 43*: the video element declared at line 43 has`autoplay=true` and `loop=true`, it starts playing the video as soon as possible and will loop it.
- *Line 21*: We implemented a rotating video effect in the`drawRotatingVideo`. The use of context save/restore is primordial as this function changes the coordinate system at each call, translating and rotating it. Notice the extra translation at *line 31* that translates the coordinate system backwards with half of the size of the image that is drawn. We did this in order to make the image rotate around the center of the rectangle, instead of around the top left corner at (0, 0) by default. Try commenting out this line in the running example and you will see what we mean.