

Technologies ▼

References & Guides ▼ Feedback ~



Sign in 🚺

Next »

# Hit regions and accessibility

Languages

Edit

Web technology for « Previous developers >

> The <canvas> element on its own is just a bitmap and does not provide information about any drawn objects. Canvas content is not exposed to accessibility tools like semantic HTML is. In general, you should avoid using canvas in an accessible website or app. The following guidelines can help to make it more accessible.

Related Topics

Canvas tutorial >

Web APIs > Canvas API >

Hit regions and accessibility

### **Canvas API**

**Canvas tutorial** 

Basic usage Drawing shapes

Applying styles and colors

Drawing text

Using images

**Transformations** 

Compositing and clipping

Basic animations Advanced animations

Pixel manipulation

Hit regions and accessibility

Optimizing the canvas Finale

**Examples** 

A basic raycaster

Canvas code snippets Drawing DOM objects into

a canvas

Manipulating video using canvas

**Interfaces HTMLCanvasElement** 

CanvasRendering

CanvasGradient CanvasPattern

Context2D

ImageBitmap

ImageData TextMetrics

A Path2D

**Documentation:** 

Useful lists

Contribute

#### The content inside the <canvas> ... </canvas> tags can be used as a fallback for browsers which don't support canvas rendering. It's also very useful for assistive

Fallback content

technology users (like screen readers) which can read and interpret the sub DOM in it. A good example at <a href="html5accessibility.com">html5accessibility.com</a> demonstrates how this can be done: <canvas> <h2>Shapes</h2>

```
A rectangle with a black border.
    3
          In the background is a pink circle.
    4
           Partially overlaying the <a href="http://en.wikipedia.org/wiki/"
    5
           Partially overlaying the circle is a green
    6
           <a href="http://en.wikipedia.org/wiki/Square" onfocus="drawSqua
           and a purple <a href="http://en.wikipedia.org/wiki/Triangle" or
    8
           both of which are semi-opaque, so the full circle can be seen u
   9
       </canvas>
  10
See the video how NVDA reads this example by Steve Faulkner.
```

#### Web applications more accessible to people with disabilities. You can use ARIA attributes to describe the behavior and purpose of the canvas element. See ARIA and ARIA techniques for more information.

ARIA rules

<canvas id="button" tabindex="0" role="button" aria-pressed="false</pre>

Accessible Rich Internet Applications (ARIA) defines ways to make Web content and

Whether the mouse coordinates are within a particular area on the canvas, is a

common problem to solve. The hit region API allows you to define an area of your

#### canvas and provides another possibility to expose interactive content on a canvas to accessibility tools. It allows you to make hit detection easier and lets you route events

Hit regions

to DOM elements. The API has the following three methods (which are still experimental in current web browsers; check the browser compatibility tables). CanvasRenderingContext2D.addHitRegion() \( \brace{L} \) Adds a hit region to the canvas.

Removes all hit regions from the canvas.

to test if your mouse is hitting your region, for example.

var ctx = canvas.getContext('2d');

CanvasRenderingContext2D.clearHitRegions()

CanvasRenderingContext2D.removeHitRegion() A

Removes the hit region with the specified id from the canvas.

<canvas id="canvas"></canvas> <script>

var canvas = document.getElementById('canvas');

You can add a hit region to your path and check for the MouseEvent.region property

```
5
       ctx.beginPath();
        ctx.arc(70, 80, 10, 0, 2 * Math.PI, false);
        ctx.fill();
        ctx.addHitRegion({id: 'circle'});
   10
        canvas.addEventListener('mousemove', function(event) {
   11
          if (event.region) {
   12
            alert('hit region: ' + event.region);
       });
        </script>
The addHitRegion() method also takes a control option to route events to an
element (that is a descendant of the canvas):
       ctx.addHitRegion({control: element});
```

This can be useful for routing to <input> elements, for example. See also this codepen demo.

on a page. To draw focus rings on a canvas drawing, the drawFocusIfNeeded

When working with the keyboard, focus rings are a handy indicator to help navigating

#### If a given element is focused, this method draws a focus ring around the current path.

property can be used.

Focus rings

CanvasRenderingContext2D.scrollPathIntoView() Scrolls the current path or a given path into the view.

Additionally, the scrollPathIntoView() method can be used to make an element

See also

CanvasRenderingContext2D.drawFocusIfNeeded() \( \Brightarrow{L} \)

visible on the screen if focused, for example.

## Best practices for interactive canvas elements

• # HTML5 Canvas Accessibility in Firefox 13 – by Steve Faulkner

Contributors to this page: peterj35, stephaniehobson, nmve, onurtemizkan, baker1797,

Sign up now

Canvas accessibility use cases

« Previous

Canvas element accessibility issues

- Tutorial Tags: Canvas Graphics
- fscholz Last updated by: peterj35, Jan 16, 2018, 8:13:05 AM

MDN web docs moz://a

Learn the best of web development

Get the latest and greatest from MDN delivered straight to your inbox.

you@example.com

moz://a

About

**About MDN** 

Feedback

Web Technologies

Learn Web Development

Firefox

Contact Us

Other languages: English (US) (en-US) ▼

Cookies Privacy Terms © 2005-2018 Mozilla and individual contributors. Content is available under these licenses.

Next »