

Make it, shake it, break it

Funka Accessibility Days 2016

Vibration API

[W3C Recommendation \(2015\)](#)

What?

Enables simple haptic feedback from webapps

Why?

- New email notifications
- Game effects
- Incoming call notifications

How?

- Patterns of on/off pulses
- Each pulse is described in milliseconds
- Patterns are a combination of pulses/pauses

Where?

- Chrome on Android
- Chrome, Firefox and Opera on desktop

Basic HTML

```
<button id="v1">Vibrate once</button>
```

```
<button id="v2">Vibrate twice</button>
```

```
<button id="v3">Vibrate more</button>
```

Check for API support

```
if ("vibrate" in navigator) {  
    //Do something.  
} else {  
    alert("Browser does not support the Vibration API");  
}
```


Vibrate once

```
var vibrateOnce = function(e) {  
    window.navigator.vibrate(500);  
};
```

Vibrate twice

```
var vibrateTwice = function(e) {  
    window.navigator.vibrate([500, 500, 500]);  
};
```

Vibrate more

```
var vibrateMore = function(e) {  
    window.navigator.vibrate([500, 500, 500, 500, 500]);  
};
```

Event listeners

```
document.getElementById("v1")  
    .addEventListener("click", vibrateOnce);  
document.getElementById("v2")  
    .addEventListener("click", vibrateTwice);  
document.getElementById("v3")  
    .addEventListener("click", vibrateMore);
```

Vibration API demo

Vibration API demo

Vibrate once

Vibrate twice

Vibrate more



CSS Flexible Box Layout module

[W3C Candidate Recommendation \(2016\)](#)

What?

Enables flexible layout control and visual ordering

Why?

- Easier responsive layouts
- Drag and drop functionality
- No more float hacks

How?

- display: flex;
- order: int;

Where?

Pretty much everywhere except IE before 11

Basic HTML

```
<div>  
  <button>1</button>  
  <button>2</button>  
  <button>3</button>  
</div>
```

Unflexed buttons



Keyboard experience



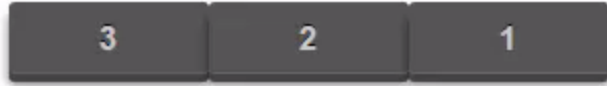
display and order properties

```
<div style="display: flex;">  
  <button style="order: 3;">1</button>  
  <button style="order: 2;">2</button>  
  <button style="order: 1;">3</button>  
</div>
```

Flexed buttons



Keyboard experience



tabindex attribute

```
<div style="display: flex;">  
  <button style="order: 3;" tabindex="3">1</button>  
  <button style="order: 2;" tabindex="2">2</button>  
  <button style="order: 1;" tabindex="1">3</button>  
</div>
```

Spec advice?

“ *Authors must use 'order' only for visual, not logical, reordering of content.* ”

The Firefox bug



Web Speech API

[W3C Community Group specification \(2012\)](#)

What?

Enables speech input and output for webapps

Why?

- Ask calendar for today's appointments
- Get help with complex interfaces
- Hands-free recipe books

How?

- SpeechSynthesis and SpeechRecognition interfaces
- Methods for controlling and manipulating speech output/input

Where?

SpeechSynthesis interface:

- Chrome, Opera and Safari on desktop
- Chrome and Safari on mobile

SpeechRecognition interface:

- Chrome and Opera on the desktop
- Chrome on Android

Basic HTML

```
<button data-hint="Use space or enter to activate">  
  Foo  
</button>
```

Check for API support

```
if (window.SpeechSynthesisUtterance === undefined) {  
    alert("Browser does not support the Web Speech API");  
} else {  
    // Do something  
}
```

Get controls with hints

```
hints = document.querySelectorAll('[data-hint]'),  
      hoverTimeout;
```

Create speech callback

```
speechHint = function(e) {  
    var msg =  
        new SpeechSynthesisUtterance(e.target.dataset.hint);  
    window.speechSynthesis.speak(msg);  
};
```

Keyboard functionality

```
function focusEventListener(e) {  
    hoverTimeout = window.setTimeout(speakHint, 500, e);  
}
```

```
function blurEventListener(e) {  
    window.clearTimeout(hoverTimeout);  
}
```

Mouse functionality

```
function mouseoverEventListener(e) {  
    hoverTimeout = window.setTimeout(speakHint, 500, e);  
}
```

```
function mouseoutEventListener(e) {  
    window.clearTimeout(hoverTimeout);  
}
```

Event listeners

```
hints.forEach(function(hint) {  
    hint.addEventListener('focus', focusEventListener);  
    hint.addEventListener('blur', blurEventListener);  
    hint.addEventListener('mouseover', mouseoverEventListener);  
    hint.addEventListener('mouseout', mouseoutEventListener);  
});
```

Web Speech API demo

Web Speech API SpeechSynthesis interface: hints demo

Enable spoken hints

Foo



Scalable Vector Graphics (SVG)

- [SVG W3C Recommendation 1.0 \(2001\)](#)
- [SVG 1.1 2nd edition W3C Recommendation \(2011\)](#)
- [SVG 2.0 W3C Working Draft \(2015\)](#)

What?

Create graphics that scale without loss of quality, and which can have semantic meaning

Why?

- Responsive images
- Icons
- Reduced bandwidth

How?

Draw on-screen using elements and animations

Where?

Pretty much everywhere

Rectangles



rect element

```
<svg width="250" height="100">  
  <rect x="0" y="0" width="100%" height="100%"  
    style="fill: #b55fff;" />  
</svg>
```

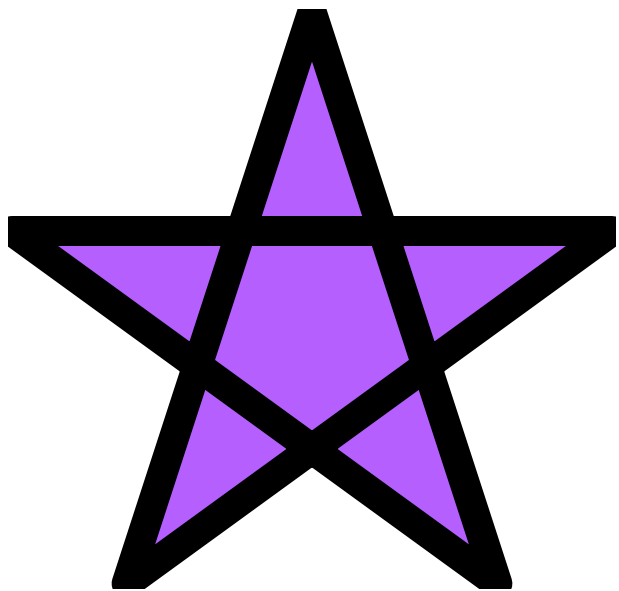
Circles



circle element

```
<svg width="250" height="100">  
  <circle tabindex="0" cx="125" cy="50" r="32.5"  
    style="fill: #820bbb;" />  
</svg>
```

Pentagrams

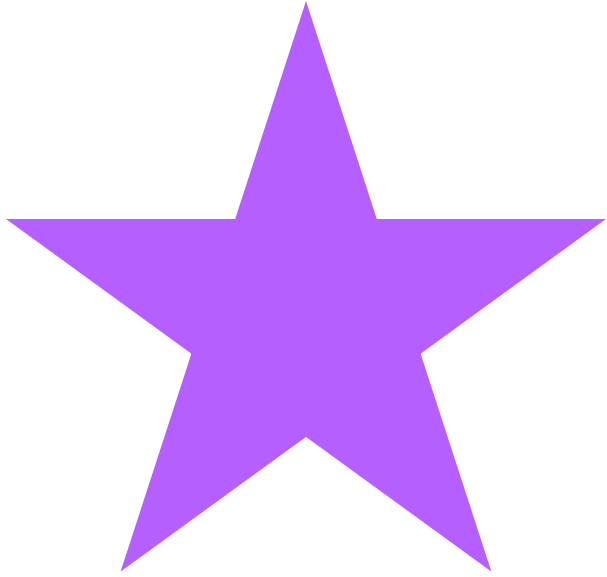


path element

```
<svg width="304" height="290">  
<path d="M2,111 h300 l-242.7,176.3 92.7,-285.3 92.7,285.3 z"  
      style="fill:#b55fff;stroke:#000000;stroke-width:15;  
           stroke-linejoin:round"/>  
</svg>
```

Scalability

Screen reader experience



img role

```
<svg role="img" width="304" height="290">  
<path d="M2,111 h300 l-242.7,176.3 92.7,-285.3 92.7,285.3 z"  
      style="fill:#b55fff;" />  
</svg>
```

title element

```
<svg role="img" width="304" height="290">  
  <title>Violet pentagram</title>  
  <path d="M2,111 h300 l-242.7,176.3 92.7,-285.3 92.7,285.3 z"  
    style="fill:#b55fff;>  
</svg>
```

desc element

```
<svg role="img" width="304" height="290">  
  <title>Violet pentagram</title>  
  <desc>Five pointed star</desc>  
  <path d="M2,111 h300 l-242.7,176.3 92.7,-285.3 92.7,285.3 z"  
    style="fill:#b55fff;>  
</svg>
```

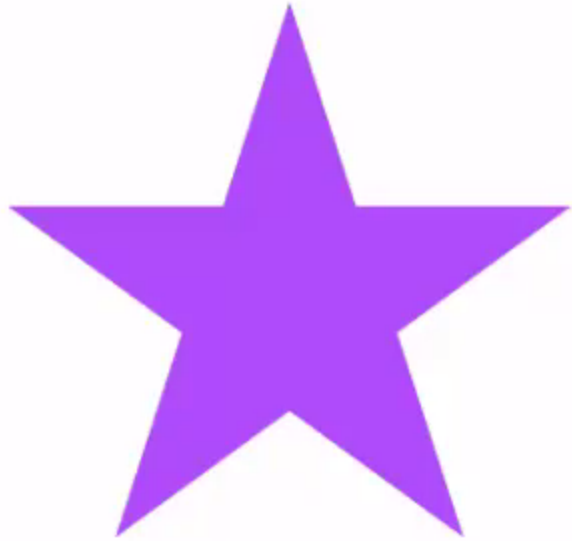

aria-labelledby attribute

```
<svg role="img" aria-labelledby="title"  
    width="304" height="290">  
  <title id="title">Violet pentagram</title>  
  <desc>Five pointed star</desc>  
  <path d="M2,111 h300 l-242.7,176.3 92.7,-285.3 92.7,285.3 z"  
    style="fill:#b55fff;>  
</svg>
```

aria-describedby attribute

```
<svg role="img" aria-labelledby="title"
      aria-describedby="desc" width="304" height="290">
  <title id="title">Violet pentagram</title>
  <desc id="desc">Five pointed star</desc>
  <path d="M2,111 h300 l-242.7,176.3 92.7,-285.3 92.7,285.3 z"
        style="fill:#b55fff;>
</svg>
```

Screen reader experience



Text

Tequila

text element

```
<svg width="250" height="100">  
  <text x="50" y="50" style="fill: #000000; font-size: 2em;">  
    Tequila  
  </text>  
</svg>
```

Rotated text

Tequila

transform attribute

```
<svg width="250" height="150">  
  <text x="50" y="25" style="fill: #000000; font-size: 2em;"  
    transform="rotate(30 20,40)">Tequila</text>  
</svg>
```

SVG Accessibility API Mappings (AAM)

[W3C Working draft \(2015\)](#)

Playground

ljwatson.github.io/playground/

Thank you

Questions?