Device Orientation events

Internet Explorer 11 adds support for DOM events that provide information about the physical orientation and motion of a device, as defined in the emerging W3C DeviceOrientation Event Specification. Using device orientation and motion events in IE11, you can use modern device sensors to explore new input mechanisms for games, new gestures for apps (such as "shake to clear the screen" or "tilt to zoom") or even augmented reality experiences.

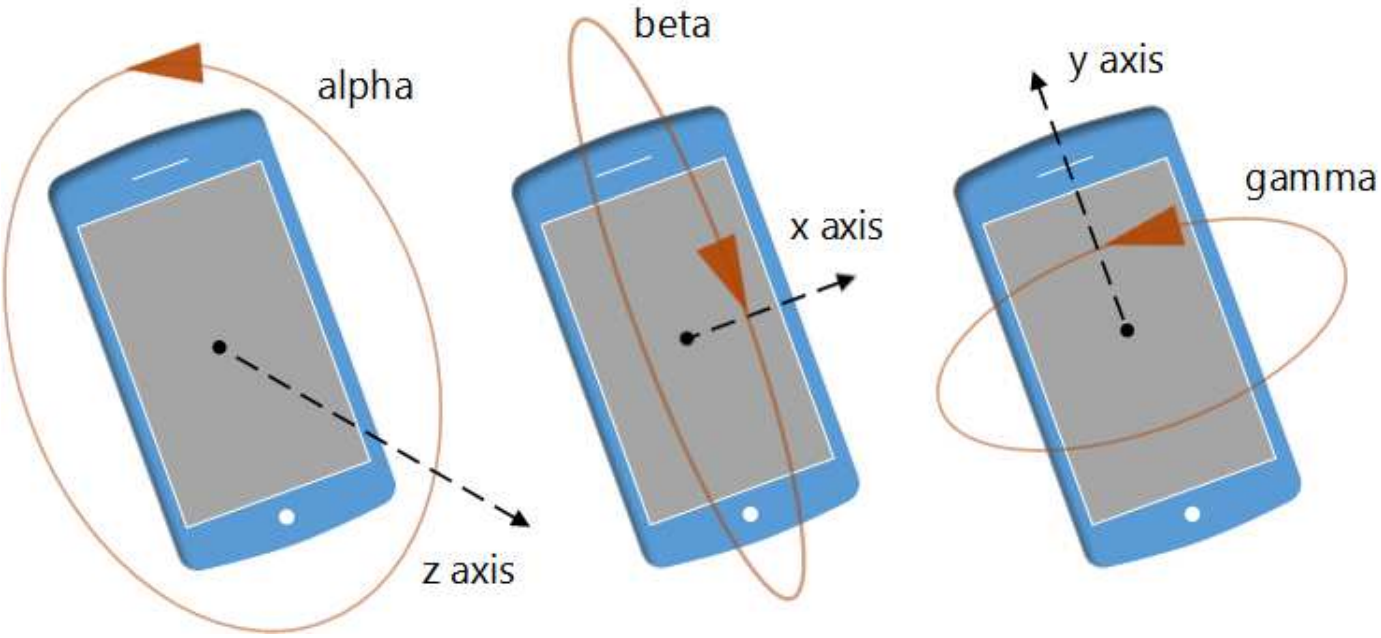
Important This feature is not supported in IE11 on Windows 7.
The W3C DeviceOrientation Events specification defines two different types of sensor data: orientation and motion.

Device orientation events

When the physical orientation of the device has changed (when the user tilts or rotates it) by 0.01 degrees or more, Internet Explorer fires the **DeviceOrientationEvent** object at the **window**. The data provided by the **DeviceOrientationEvent** object specifies the orientation of the host device in relation to a coordinate frame fixed on the Earth. Specifically, this Earth coordinate frame has the following three axes:

- East (*X*) is in the ground plane, perpendicular to the north axis and positive towards the East.
- North (*Y*) is in the ground plane and positive towards true north (towards the North Pole).
- Up (*Z*) is perpendicular to the ground plane and positive upwards.

These *X*, *Y*, and *Z* axes correspond to the **beta**, **gamma**, and **alpha** properties of the **DeviceOrientationEvent**, respectively.



The following code shows how to use the **deviceorientation** event to guide the user to point their device northward.

JavaScript

```
<div id="directions"></div>
<script>
    window.addEventListener("deviceorientation", findNorth);
```

```
    function findNorth(evt) {
        var directions = document.getElementById("directions");
        if (evt.alpha < 5 || evt.alpha > 355) {
            directions.innerHTML = "North!";
        } else if (evt.alpha < 180) {
            directions.innerHTML = "Turn Left";
        } else {
            directions.innerHTML = "Turn Right";
        }
    }
</script>
```

Device motion events

When a device is being moved or rotated (or more accurately, *accelerated*), the **DeviceMotionEvent** object is fired at the window and provides **acceleration data** (both **with** and **without** the effects of gravitational acceleration on the device, expressed in m/s^2) in the **x**, **y**, and **z** axes as well as **rotational rate of change data** in the **alpha**, **beta**, and **gamma** rotation angles (expressed in deg/s). Rotations use the *right-hand rule*, such that positive rotation around an axis is clockwise when viewed looking towards the positive direction of the axis.

The following sample demonstrates how to use the **ondevicemotion** event to detect and report any movement of the device above a specified threshold.

JavaScript

```
<div id="status"></div>
<script>
    window.addEventListener("devicemotion", detectShake);
    function detectShake(evt) {
        var status = document.getElementById("status");
        var accl = evt.acceleration;
        if (accl.x > 1.5 || accl.y > 1.5 || accl.z > 1.5) {
            status.innerHTML = "EARTHQUAKE!!!";
        } else {
            status.innerHTML = "All systems go!";
        }
    }
</script>
```

Calibrating the compass

The **compassneedscalibration** event fires when the host device compass requires calibration by the user in order to provide more accurate data from the **DeviceOrientationEvent**.

The IE implementation of this event fires whenever the host device magnetometer changes to a state of unreliable or approximate accuracy, as defined by the **MagnetometerAccuracy** enumeration of the **Windows.Devices.Sensors** namespace.

API reference

Detecting device orientation

by 22 contributors: Show all...

This is an experimental technology
Because this technology's specification has not stabilized, check the compatibility table for the proper prefixes to use in various browsers. Also note that the syntax and behavior of an experimental technology is subject to change in future versions of browsers as the spec changes.

Increasingly, web-enabled devices are capable of determining their **orientation**; that is, they can report data indicating changes to their orientation with relation to the pull of gravity. In particular, hand-held devices such as mobile phones can use this information to automatically rotate the display to remain upright, presenting a wide-screen view of the web content when the device is rotated so that its width is greater than its height.

There are two JavaScript events that handle orientation information. The first one is the `DeviceOrientationEvent`, which is sent when the accelerometer detects a change to the orientation of the device. By receiving and processing the data reported by these orientation events, it's possible to interactively respond to rotation and elevation changes caused by the user moving the device.

The second event is the `DeviceMotionEvent`, which is sent when a change in acceleration was added. It is different from the `DeviceOrientationEvent` because it is listening for changes in acceleration as opposed to orientation. Sensors that are commonly capable of detecting `DeviceMotionEvent` include sensors in laptops to protect moving storage devices. `DeviceOrientationEvent` are more commonly found in mobile devices.

Processing orientation events

All you need to do in order to begin receiving orientation change is to listen to the `deviceorientation` event:

Note: *gyronorm.js is a polyfill for normalizing the accelerometer and gyroscope data on mobile devices. This is useful for overcoming some of the differences in device support for device orientation.*

```
window.addEventListener("deviceorientation", handleOrientation, true);
```

After registering your event listener (in this case, a JavaScript function called handleOrientation()), your listener function periodically gets called with updated orientation data.

The orientation event contains four values:

- `DeviceOrientationEvent.absolute`
- `DeviceOrientationEvent.alpha`
- `DeviceOrientationEvent.beta`
- `DeviceOrientationEvent.gamma`

The event handler function can look something like this:

```
function handleOrientation(event) {
  var absolute = event.absolute;
  var alpha    = event.alpha;
  var beta     = event.beta;
  var gamma    = event.gamma;

  // Do stuff with the new orientation data
}
```

Orientation values explained

The value reported for each axis indicates the amount of rotation around a given axis in reference to a standard coordinate frame. These are described in greater detail in the Orientation and motion data explained article which is summarized below.

- The `DeviceOrientationEvent.alpha` value represents the motion of the device around the z axis, represented in degrees with values ranging from 0 to 360.
- The `DeviceOrientationEvent.beta` value represents the motion of the device around the x axis, represented in degrees with values ranging from -180 to 180. This represents a front to back motion of the device.
- The `DeviceOrientationEvent.gamma` value represents the motion of the device around the y axis, represented in degrees with values ranging from -90 to 90. This represents a left to right motion of the device.

Orientation example

This example will work on any browser supporting the `deviceorientation` event and running on a device able to detect its orientation.

So let's imagine a ball in a garden:

```
<div class="garden">
  <div class="ball"></div>
</div>

<pre class="output"></pre>
```

This garden is 200 pixel wide (Yes, it's a tiny one), and the ball is in the center:

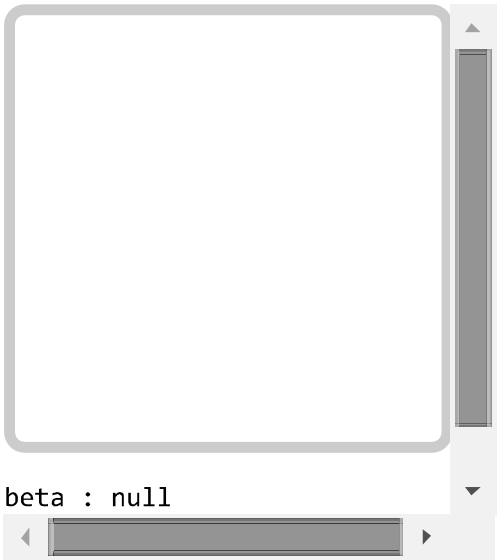
```
.garden {
  position: relative;
  width : 200px;
  height: 200px;
  border: 5px solid #CCC;
  border-radius: 10px;
}

.ball {
  position: absolute;
  top    : 90px;
  left   : 90px;
  width : 20px;
  height: 20px;
  background: green;
  border-radius: 100%;
}
```

Now, if we move our device, the ball will move accordingly:

```
var ball   = document.querySelector('.ball');
var garden = document.querySelector('.garden');
var output = document.querySelector('.output');

var maxX = garden.clientWidth  - ball.clientWidth;
var maxY = garden.clientHeight - ball.clientHeight;

function handleOrientation(event) {
  var x = event.beta;  // In degree in the range [-180,180]
  var y = event.gamma; // In degree in the range [-90,90]

  output.innerHTML  = "beta : " + x + "\n";
  output.innerHTML += "gamma: " + y + "\n";

  // Because we don't want to have the device upside down
  // We constrain the x value to the range [-90,90]
  if (x >  90) { x =  90};
  if (x < -90) { x = -90};
```

```
  // To make computation easier we shift the range of
  // x and y to [0,180]
  x += 90;
  y += 90;

  // 10 is half the size of the ball
  // It center the positioning point to the center of the ball
  ball.style.top  = (maxX*x/180 - 10) + "px";
  ball.style.left = (maxY*y/180 - 10) + "px";
}

window.addEventListener('deviceorientation', handleOrientation);
```

Here's the live result:



> ***Warning:*** *Chrome and Firefox do not handle the angles the same way, so on some axes the direction are reversed.*

Processing motion events

Motion events are handled the same way as the orientation events except that they have their own event's name: `devicemotion`

```
window.addEventListener("devicemotion", handleMotion, true);
```

What's really changed are the information provided within the `DeviceMotionEvent` object passed as a parameter of the *HandleMotion* function.

The motion event contains four properties:

- `DeviceMotionEvent.acceleration`

- `DeviceMotionEvent.accelerationIncludingGravity`
- `DeviceMotionEvent.rotationRate`
- `DeviceMotionEvent.interval`

Motion values explained

The `DeviceMotionEvent` objects provide web developers with information about the speed of changes for the device's position and orientation. The changes are provided along three axis (see Orientation and motion data explained for details).

For `acceleration` and `accelerationIncludingGravity`, those axes correspond to the following:

- `x`: Represents the axis from West to East
- `y`: Represents the axis from South to North
- `z`: Represents the axis perpendicular to the ground

For `rotationRate`, the situation is a bit different; the information corresponds to the following in each case:

- `alpha`: Represents a rotation rate along the axis perpendicular to the screen (or keyboard for desktop).
- `beta`: Represents a rotation rate along the axis going from left to right of the plane of the screen (or keyboard for desktop).
- `gamma`: Represents a rotation rate along the axis going from bottom to top of the plane of the screen (or keyboard for desktop).

Finally, `interval` represents the interval of time, in milliseconds, at which data are obtain from the device.

Specifications

Specification	Status	Comment
Device Orientation Events	WD Working Draft	Initial specification.

Browser compatibility

Desktop Mobile

Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari (WebKit)
`DeviceOrientationEvent`	7.0	3.6[1] 6	?	?	?
`DeviceMotionEvent`	(Yes)	6	?	?	?

[1] Firefox 3.6 to 5 supported mozOrientation versus the standard `DeviceOrientationEvent` event.

See also

- `DeviceOrientationEvent`
- `DeviceMotionEvent`
- The legacy `MozOrientation` event.
- Orientation and motion data explained
- Using deviceorientation in 3D Transforms
- Cyber Orb: 2D maze game with device orientation

Screen.orientation

> **This is an experimental technology**
> Because this technology's specification has not stabilized, check the compatibility table for usage in various browsers. Also note that the syntax and behavior of an experimental technology is subject to change in future versions of browsers as the specification changes.

The `Screen.orientation` property give the current orientation of the screen.

Syntax

```
var orientation = window.screen.orientation;
```

Return value

The return value is a string representing the orientation of the screen. It can be portrait-primary, portrait-secondary, landscape-primary, landscape-secondary (See `lockOrientation` for more info about those values).

Example

```
var orientation = screen.orientation || screen.mozOrientation || screen.msOrientation;

if (orientation === "landscape-primary") {
  console.log("That looks good.");
} else if (orientation === "landscape-secondary") {
  console.log("Mmmh... the screen is upside down!");
} else if (orientation === "portrait-secondary" || orientation === "portrait-primary")
  console.log("Mmmh... you should rotate your device to landscape");
}
```

Specifications

Specification	Status	Comment
Screen Orientation API The definition of 'Screen Orientation' in that specification.	WD Working Draft	Initial definition

Browser compatibility

Desktop Mobile

Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari
Basic support	38	(Yes) moz [1]	11 ms [2]	25	No support

[1] This API is only implemented as a prefixed method (`mozOrientation`) in B2G and Firefox for Android.

[2] This API is implemented using a prefix (`msOrientation`) in Internet Explorer for Windows 8.1 and Windows RT 8.1. It is not supported on Windows 7.

See also

- `Screen.orientation`
- `Screen.unlockOrientation()`
- `Screen.onorientationchange`
- Managing screen orientation