

Get the different angles using the JavaScript HTML5 orientation API

TYPICAL USE

The use of this API is very straightforward:

1. Test if your browser supports the orientation API (test `if window.DeviceOrientationEvent` is not null),
2. Define a listener for the 'deviceorientation' event as follows: `window.addEventListener('deviceorientation', callback);` with the callback function accepting the event as a single input parameter,
3. Get the angles from the event (use its properties `alpha`, `beta`, `gamma`).

[Let's see this with an example on JsBin](#). Try it with a smartphone, a tablet or a device with an accelerometer:

(If using a mobile device, it's better to open this URL in standalone mode, without the JsBin editor. [Try this URL instead!](#))

Device Orientation with HTML5

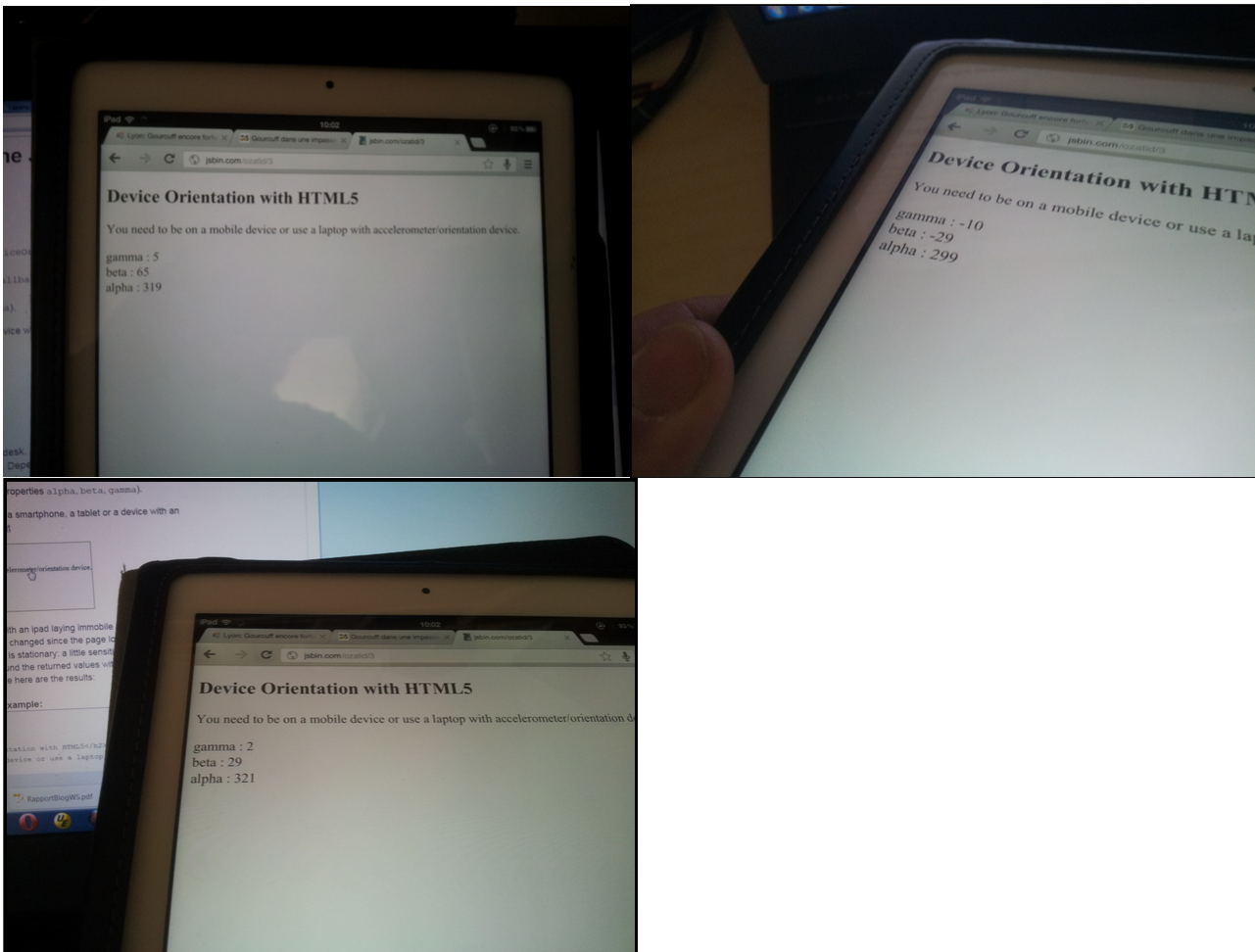
You need to be on a mobile device or use a laptop with accelerometer/orientation device.

```
gamma : 0  
beta : 0  
alpha : 0
```

The above screenshot was taken with an ipad lying immobile on a desk. All the angle values are theoretically 0 when the device is laid flat, and it has not changed since the page loaded.

Depending on the hardware, however, these values may change even if the device is stationary: a little sensitive sensor can report constantly changing values. This is why in the example we round the returned values with `Math.round()` at display time (see code).

If we change the orientation of the device here are the results:



Typical use / code from the above example:

```

...
<h2>Device Orientation with HTML5</h2>
You need to be on a mobile device or use a laptop with
accelerometer/orientation
device.
<p>
<div id="LR"></div>
<div id="FB"></div>
8. <div id="DIR"></div>
<script type="text/javascript">
    if (window.DeviceOrientationEvent) {
        console.log("DeviceOrientation is supported");
        window.addEventListener('deviceorientation', function(eventData) {
            // gamme is for left/right inclination
            var LR = eventData.gamma;
            // beta is for front/back inclination
            var FB = eventData.beta;
            // alpha is for orientation
18. var DIR = eventData.alpha;

```

```

        // display values on screen
        deviceOrientationHandler(LR, FB, DIR);
    }, false);
} else {
    alert("Device orientation not supported on your device or
browser. Sorry.");
}
function deviceOrientationHandler(LR, FB, DIR) {
    document.querySelector("#LR").innerHTML ="gamma :
" + Math.round(LR);
28.    document.querySelector("#FB").innerHTML ="beta :
" + Math.round(FB);
    document.querySelector("#DIR").innerHTML ="alpha :
" + Math.round(DIR);
}
</script>
...

```

ANOTHER EXAMPLE THAT SHOWS HOW TO ORIENT THE HTML5 LOGO USING THE ORIENTATION API + CSS3 3D ROTATIONS

This is just a variation of the previous example, [try it at JsBin](#): if using a mobile device it's better to open it in standalone mode, use [this URL instead](#):

Device Orientation with HTML5

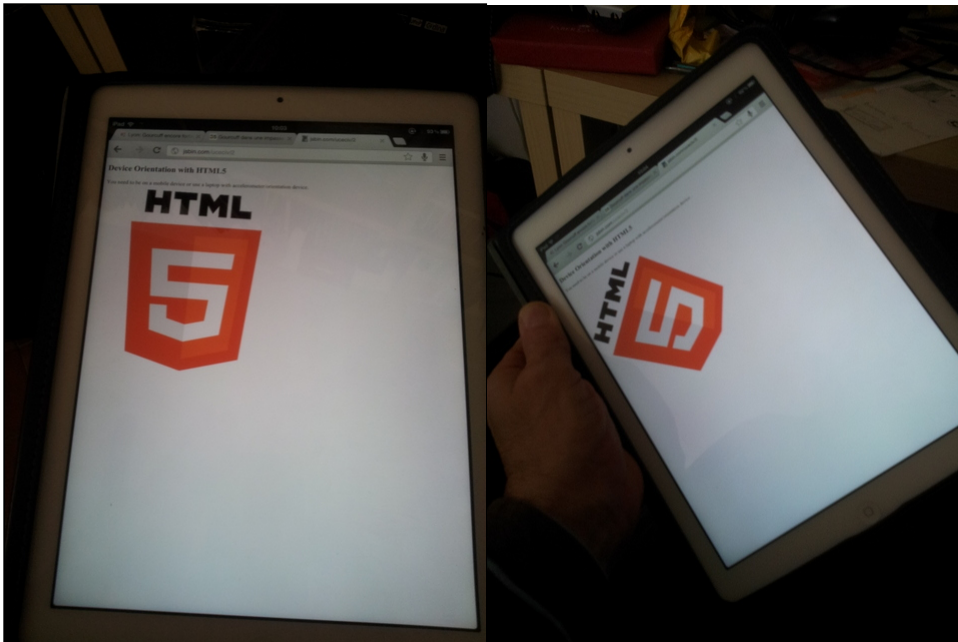
You need to be on a mobile device or use a laptop with accelerometer/orientation device.

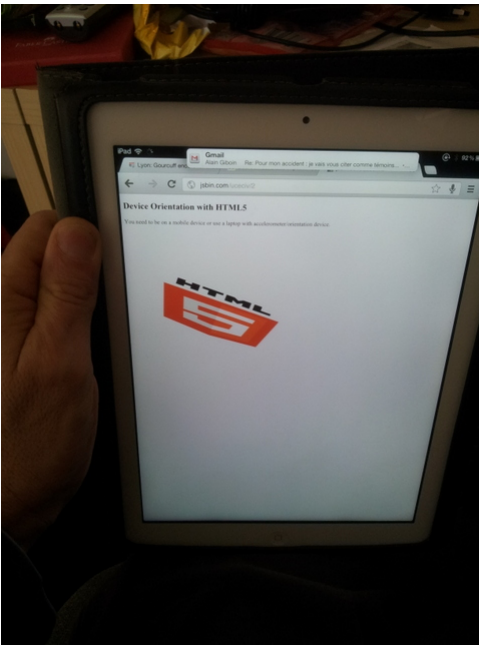
gamma : 0
beta : 0
alpha : 0

HTML



Results on the ipad: the logo rotates when we change the ipad's orientation. This is a good "visual feedback" for an orientation controlled game...





This example as a Youtube video: <http://www.youtube.com/watch?v=OrNLhOAGSdE>

Code from the example:

```
...
<h2>Device Orientation with HTML5</h2>
You need to be on a mobile device or use a laptop with
accelerometer/orientation
device.
5. <p>
    <div id="LR"></div>
    <div id="FB"></div>
    <div id="DIR"></div>
    
    <script type="text/javascript">
        if (window.DeviceOrientationEvent) {
            console.log("DeviceOrientation is supported");
15. window.addEventListener('deviceorientation',function(eventData) {
            var LR = eventData.gamma;
            var FB = eventData.beta;
            var DIR = eventData.alpha;
            deviceOrientationHandler(LR, FB,DIR);
        }, false);
```

```

    } else {
        alert("Not supported on your device or browser. Sorry.");
    }
25.    function deviceOrientationHandler(LR, FB, DIR) {
        // USE CSS3 rotations for rotating the HTML5 logo
        //for webkit browser
        document.getElementById("imgLogo").style.webkitTransform =
            "rotate(" + LR + "deg) rotate3d(1,0,0,
" + (FB * -1) + "deg)";
        //for HTML5 standard-compliance
        document.getElementById("imgLogo").style.transform =
            "rotate(" + LR + "deg) rotate3d(1,0,0,
" + (FB * -1) + "deg)";
35.    document.querySelector("#LR").innerHTML= "gamma :
" + Math.round(LR);
        document.querySelector("#FB").innerHTML= "beta :
" + Math.round(FB);
        document.querySelector("#DIR").innerHTML = "alpha :
" + Math.round(DIR);
    }
</script>
...

```

A SIMPLE LEVEL TOOL USING DEVICE ORIENTATION

This example works in FF, Chrome and IOS Safari. Created by [Derek Anderson](#) @ [Media Upstream](#). Original source code available [GitHub](#).

We adapted the source code so that you can tweak it in [JsBin](#): for testing it using a mobile device, open it in standalone mode at this URL instead.

Level Tool



Created by [Derek Anderson](#) @ [Media Upstream](#).
Fork on [GitHub](#)

For best results use Google Chrome or FireFox
on a device with an Accelerometer!

OTHER INTERESTING USES: MIX ORIENTATION API AND WEBSOCKETS

You can imagine the above example that sends the current orientation of the device to a server using WebSockets. The server in turn updates the logo and position on a PC screen. If multiple devices connect, they can chat together and take control of the 3D Logo.

This video shows one of the above examples slightly modified: the JavaScript code running in the Web page on the iPad sends in real time the device orientation using the Web Sockets API to a server that in turns sends the orientation to a client running on a desktop browser. In this way the tablet "controls" the HTML5 logo that is shown on the desktop browser:

Click on the image to see the YouTube video:

