

New APIs

- 陈本峰 
4月11日

关于我

- W3C中国区 HTML5布道官



- 前微软美国总部IE浏览器
核心研发团队成员



- 云适配科技创始人CEO



提纲

- New DOM APIs
- Geolocation API
- Orientation API
- Device motion API

New DOM API – innerHTML 属性

- 用途：设置/返回一个元素包含的HTML代码
- 示例代码

```
var v= document.getElementById("id1");  
v.innerHTML = '<h1><i>Some italic title...</i></ h1>';
```

- 浏览器兼容性： 所有主流浏览器

New DOM API - getElementsByClassName()

- 用途：返回匹配class名称的所有元素
- 调用：在document 对象 或者某个元素 对象上调用
- 同类API还有 getElementById(), getElementsByName(), getElementsByTagName()
- 示例代码

```
var list1 = document.getElementsByClassName('important');
for (var i = 0; i < list1.length; i++) {
    // do something with list1.item(i) or list1[i],
    // in our case : put the background in red
    list1[i].style.backgroundColor = 'red';
}
var elem1=
document.getElementById('id1').getElementsByClassName('notImportant')
;
```

New DOM API - `getElementsByClassName()`

- 浏览器兼容性 (<http://caniuse.com/#feat=getelementsbyclassname>)

`getElementsByClassName` - **Working Draft**

*Usage stats:

Global

Support:

83.84%

Method of accessing DOM elements by class name

Show all versions

	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser
								2.1	
								2.2	
						3.2		2.3	
						4.0-4.1		3.0	
						4.2-4.3		4.0	
	8.0	16.0				5.0-5.1		4.1	
	9.0	17.0	23.0	5.1					
Current	10.0	18.0	24.0	6.0	12.1	6.0	5.0-7.0	4.2	7.0
Near future		19.0	25.0		12.5				10.0
Farther future		20.0	26.0						

Notes

Known issues (1)

Resources (2)

Feedback

Edit on GitHub

No notes

New DOM API - querySelector()/querySelectorAll()

- 用途：返回匹配CSS选择器的第一个/全部元素
- 类似jQuery的\$()
- 示例代码

```
// all elements li in ul elements in an element of id=nav
var el = document.querySelector('#nav ul li');

// all li in a ul, but only even elements
var els = document.querySelectorAll('ul li:nth-child(even)');

// all td directly in tr in a form of class test
var els = document.querySelectorAll('form.test > tr > td');

// all paragraphs of class warning or error
querySelectorAll("p.warning, p.error");

// first element of id=foo or id=bar
querySelector("#foo, #bar");

// first p in the div
var div = document.getElementById("bar");
var p = div.querySelector("p");
```

New DOM API - `querySelector()/querySelectorAll()`

- 浏览器兼容性 (<http://caniuse.com/#feat=queryselector>)

querySelector/querySelectorAll - **Candidate Recommendation**

*Usage stats:

Global

Support:

93.62%

Method of accessing DOM elements using CSS selectors

Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser
								2.1	
								2.2	
								2.3	
						3.2		3.0	
						4.0-4.1		4.0	
	8.0	16.0				4.2-4.3		4.1	
	9.0	17.0	23.0	5.1		5.0-5.1		4.2	7.0
Current	10.0	18.0	24.0	6.0	12.1	6.0	5.0-7.0	4.2	10.0
Near future		19.0	25.0		12.5				
Farther future		20.0	26.0						

Notes

Known issues (0)

Resources (4)

Feedback

Edit on GitHub

Only works for the CSS selectors available. Thus the IE8 implementation is limited to the CSS 2.1 selectors

New DOM API - matches()

- 用途：检测一个元素是否匹配输入的CSS选择器
- 示例代码

```
var element1 = document.getElementById('id1');
var element2 = document.getElementsByTagName('div')[0];
if(element1.matches('h1')){
    // element1 is a h1 element
}
if(element2.matches('#myid')){
    // element2 has an id equals to 'myid'
}
```

New DOM API - matches()

- 浏览器兼容性 (<http://caniuse.com/#feat=matchesselector>)

# matches() DOM method - Working Draft									
Method of testing whether or not a DOM element matches a given selector. Formerly known (and largely supported with prefix) as matchesSelector.									
							*Usage stats:		Global
							Support:		0%
							Partial support:		80.9%
							Total:		80.9%
Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser
								2.1	
								2.2 webkit	
						3.2		2.3 webkit	
						4.0-4.1 webkit		3.0 webkit	
	8.0	16.0 moz				4.2-4.3 webkit		4.0 webkit	
	9.0 ms	17.0 moz	23.0 webkit	5.1 webkit		5.0-5.1 webkit		4.1 webkit	
Current	10.0 ms	18.0 moz	24.0 webkit	6.0 webkit	12.1 o	6.0 webkit	5.0-7.0	4.2 webkit	7.0 webkit
Near future		19.0 moz	25.0 webkit		12.5 o				10.0 webkit
Farther future		20.0 moz	26.0 webkit						
Notes Known issues (0) Resources (2) Feedback									
• Specification [w3.org] reference									
• MDN article [developer.mozilla.org] reference info									
							Edit on GitHub		

New DOM API – classList属性

- 用途： 返回元素的class列表对象 classList
- 示例代码

```
var elem= document.getElementById("id1");
var allClasses = elem.classList;

// By default, start without a class in the div: <div class=""/>
// Set "foo" as the class by adding it to the classList
div.classList.add('foo'); // now <div class="foo"/>
// Check that the classList contains the class "foo"
div.classList.contains('foo'); // returns true
// Remove the class "foo" from the list
div.classList.remove('foo'); // now <div class=""/>
// Check if classList contains the class "foo"
div.classList.contains('foo'); // returns false: "foo" is gone
// Check if class contains the class "foo",
// If it does, "foo" is removed, if it doesn't, it's added
div.classList.toggle('foo'); // class set to <div class="foo"/>
div.classList.toggle('foo'); // class set to <div class=""/>
```

New DOM API - classlist属性

- 浏览器兼容性 (<http://caniuse.com/#feat=classList>)

classList (DOMTokenList) - Working Draft

*Usage stats:

Global

Support: 62.02%

Method of easily manipulating classes on elements, using the DOMTokenList object.

Show all versions

	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser
								2.1	
								2.2	
								2.3	
						3.2		2.3	
						4.0-4.1		3.0	
						4.2-4.3		4.0	
	8.0	16.0				5.0-5.1		4.1	
	9.0	17.0	23.0	5.1				4.1	
Current	10.0	18.0	24.0	6.0	12.1	6.0	5.0-7.0	4.2	7.0
Near future		19.0	25.0		12.5				10.0
Farther future		20.0	26.0						

Notes

Known issues (0)

Resources (3)

Feedback

Edit on GitHub

No notes

地理位置 Geolocation API

- 浏览器兼容性 (<http://caniuse.com/#feat=geolocation>)

Geolocation - **Candidate Recommendation**

Method of informing a website of the user's geographical location

*Usage stats:		Global
Support:		81.19%
Partial support:		0.03%
Total:		81.22%

Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser
								2.1	
						3.2		2.2	
	7.0					4.0-4.1		2.3	
	8.0	15.0				4.2-4.3		3.0	
	9.0	16.0	22.0	5.1		5.0-5.1		4.0	
Current	10.0	17.0	23.0	6.0	12.1	6.0	5.0-7.0	4.1	7.0
Near future		18.0	24.0		12.5				10.0
Farther future		19.0	25.0						

[Notes](#)
[Known issues \(1\)](#)
[Resources \(2\)](#)
[Feedback](#)

[Edit on GitHub](#)

- [Specification](#) [w3.org] [reference](#)
- [Simple demo](#) [html5demos.com] [demo](#)

地理位置 Geolocation API – 代码示例

```
<!DOCTYPE html>
<html>
<body>
<p id="msg">Click the button to get your coordinates:</p>
<button onclick="getLocation()">Where am I ?</button>
<script>
var displayCoords=document.getElementById("msg");
function getLocation() {
if (navigator.geolocation) {
    navigator.geolocation.getCurrentPosition(showPosition);
} else {
    displayCoords.innerHTML="Geolocation API not supported by your browser.";
}
}
function showPosition(position) {
    displayCoords.innerHTML="Latitude: " + position.coords.latitude +
    "<br />Longitude: " + position.coords.longitude;
}
</script>
</body>
</html>
```

地理位置 Geolocation API

- 在线演示 <http://jsbin.com/ezivih>

Click the button to get your coordinates:

Where am I ?

地理位置 Geolocation API – coords 对象

- `coords.latitude`: 纬度
- `coords.longitude`: 经度
- `coords.altitude`: 海拔高度
- `coords.accuracy`: 所提供的以米为单位的经度和纬度估计的精确度
- `coords.altitudeAccuracy`: 所提供的以米为单位的高度估计的精确度
- `coords.heading`: 用户设备当前移动的角度方向，相对于正北方向顺时针计算
- `coords.speed`: 以米每秒为单位的设备的当前对地速度

地理位置 Geolocation API – 参数

```
navigator.geolocation.getCurrentPosition(onSuccess, onError,  
{enableHighAccuracy:true});
```

- **enableHighAccuracy**: 是否返回更详细更准确的结构，默认为false不启用，选择true则启用，但是会导致较长的响应时间及增加功耗，这种情况更多的用在移动设备上。
- **timeout**: 设备位置获取操作的超时时间设定（不包括获取用户权限时间），单位为毫秒，如果在设定的timeout时间内未能获取位置定位，则会执行errorCallback()返回code（3）。如果未设定timeout，那么timeout默认为无穷大，如果timeout为负数，则默认timeout为0。
- **maximumAge**: 设定位置缓存时间，以毫秒为单位，如果不设置该值，该值默认为0，如果设定负数，则默认为0。该值为0时，位置定位时会重新获取一个新的位置对象；该值大于0时，即从上一次获取位置时开始，缓存位置对象，如果再次获取位置时间不超过maximumAge，则返回缓存中的位置，如果超出maximumAge，则重新获取一个新的位置。

地理位置 Geolocation API – 错误处理

```
<div id="maposition"></div>
...
// Callback function in case of error
function errorPosition(error) {
var info = "Error during geolocation: ";
switch(error.code) {
  case error.TIMEOUT:
    info += "Timeout !";
    break;
  case error.PERMISSION_DENIED:
    info += "Permission denied, geolocation could not be obtained...";
    break;
  case error.POSITION_UNAVAILABLE:
    info += "Location could not be obtained though the available means...";
    break;
  case error.UNKNOWN_ERROR:
    info += "Unknown error";
    break;
}
document.getElementById("maposition").innerHTML = info;
}
navigator.geolocation.getCurrentPosition(showPosition, errorPosition);
```

地理位置 Geolocation API -地图演示

- <http://browsergeolocation.com/>

Your Browser Geolocation:

Latitude: 39.978899 Longitude: 116.304664

Address: Ancient Mosque, Suzhou Street, Haidian, Beijing, China, 100080

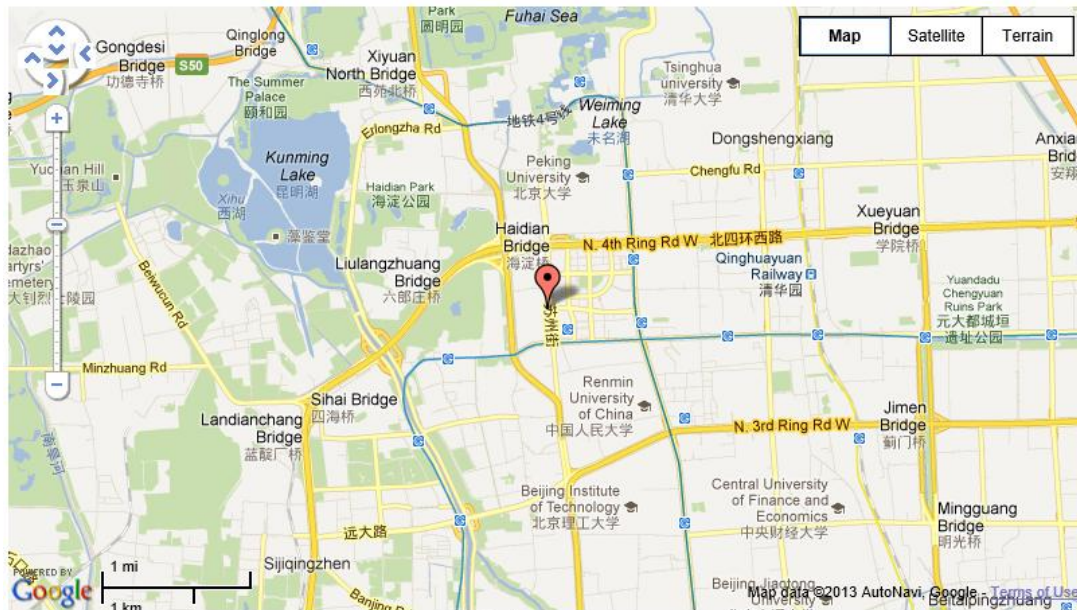
[Related Ads](#)

[Whois IP Address](#)

[Google Map](#)

[Geolocation](#)

[IP Locator](#)



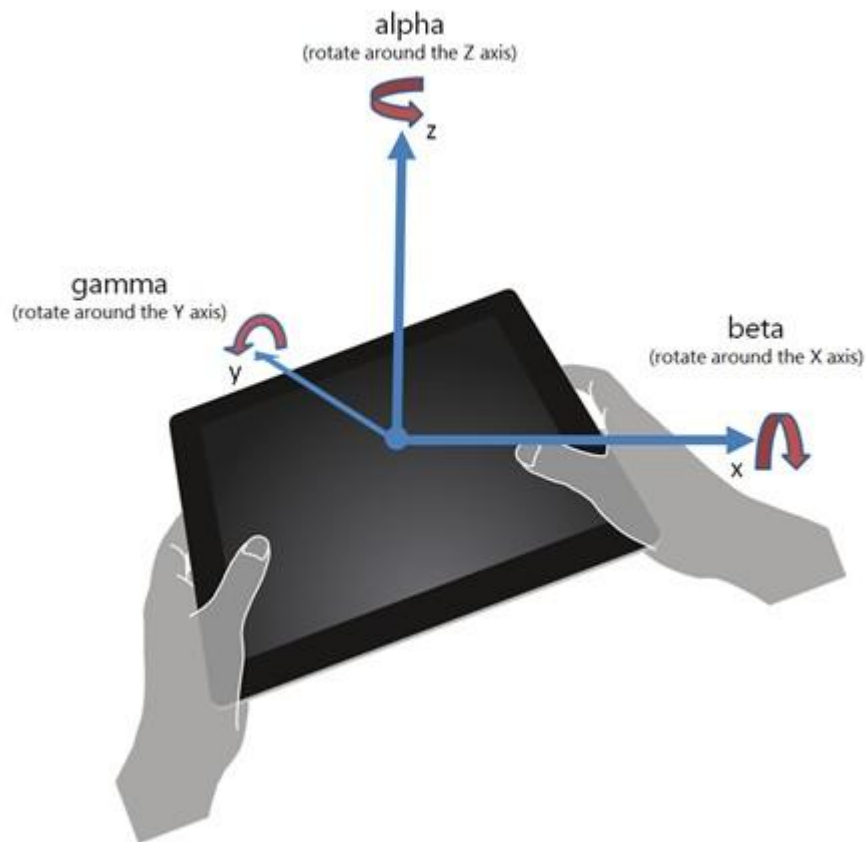
地理位置 Geolocation API - watchPosition

- **用途：** 实时跟踪用户地理位置，每当用户位置改变，回调函数将被调用

```
// get an id of the current tracking, the showPosition callback is like the one we saw in earlier var watchPosId =  
navigator.geolocation.watchPosition(showPosition);  
...  
// stop the tracking  
navigator.geolocation.clearWatch(watchPosId);
```

方向传感器 Orientation API

- 用途：返回手机静止状态下的方向数据
- 返回值
 - gamma: 左右倾斜
 - beta : 前后倾斜
 - alpha: 水平方向

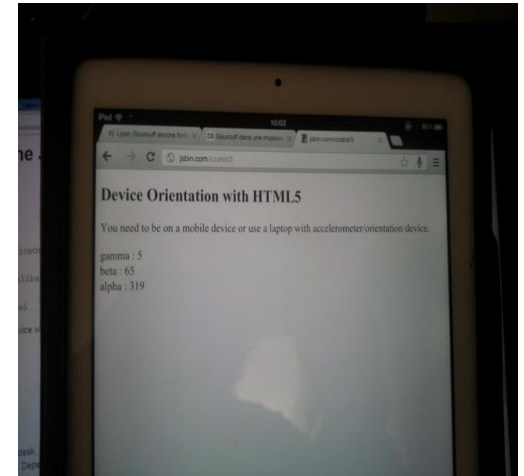
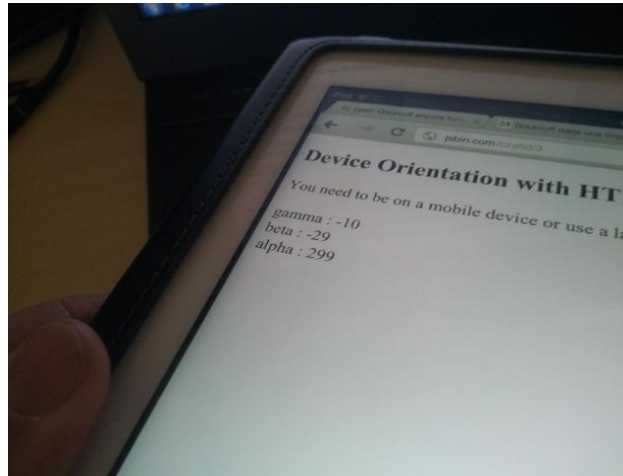
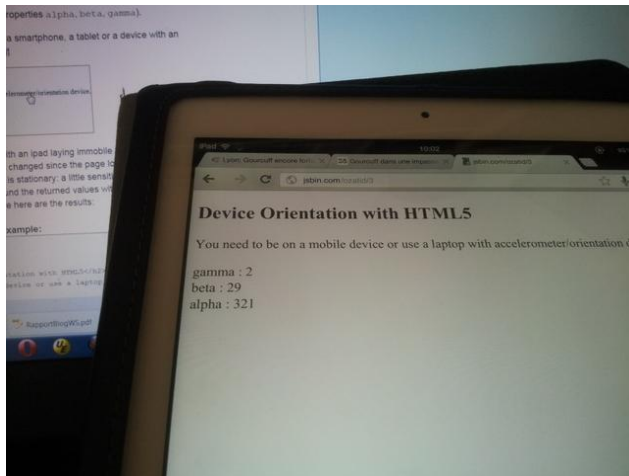


方向传感器 Orientation API - 示例代码

```
<div id="LR"></div>
<div id="FB"></div>
<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 inclinaison
        var LR = eventData.gamma;
        // beta is for front/back inclinaison
        var FB = eventData.beta;
        // alpha is for orientation
        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);
    document.querySelector("#FB").innerHTML = "beta : " + Math.round(FB);
    document.querySelector("#DIR").innerHTML = "alpha : " + Math.round(DIR);
}
</script>
```

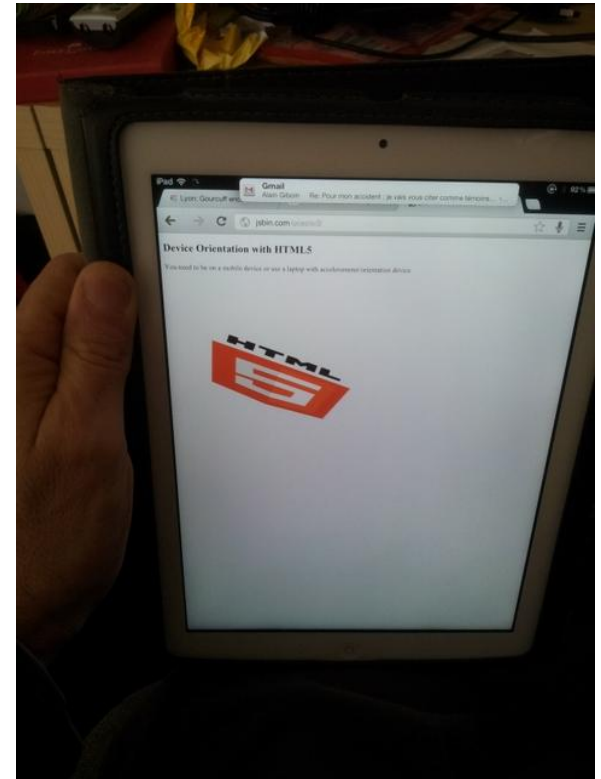
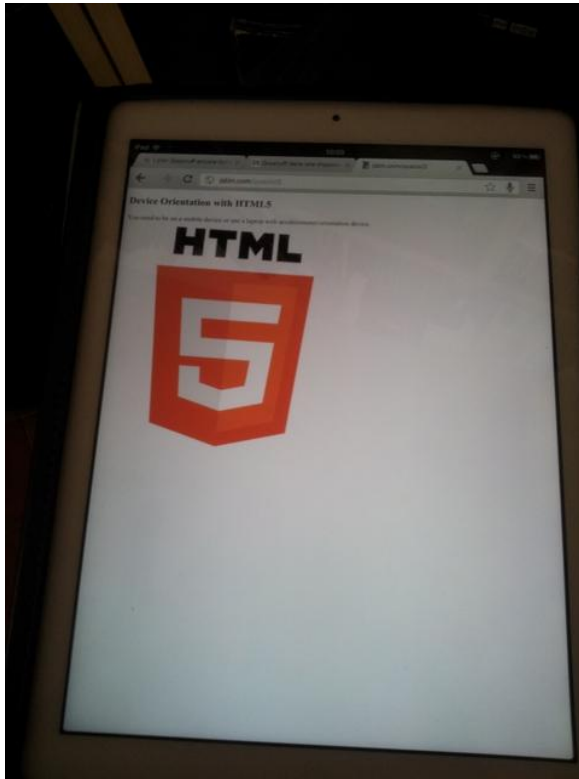
方向传感器 Orientation API – 演示

<http://jsbin.com/ozatid/3>



方向传感器 Orientation API – 结合CSS3演示

<http://jsbin.com/uceciv/2>



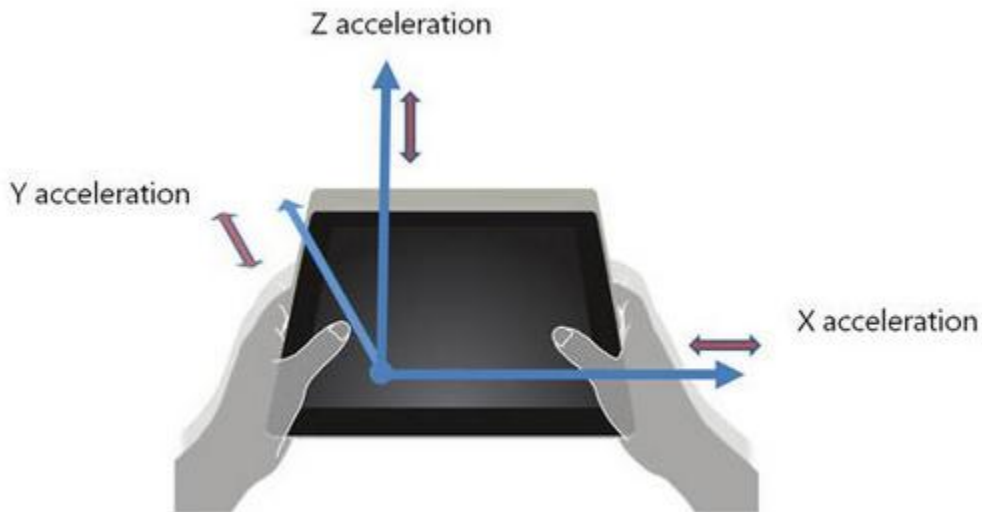
方向传感器 Orientation API

- 浏览器兼容性

# Device Orientation events - Working Draft									
API for detecting orientation and motion events from the device running the browser.									
							*Usage stats:		Global
							Support:		38.03%
							Partial support:		19.12%
							Total:		57.15%
Show all versions	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser
								2.1	
						3.2		2.2	
	7.0					4.0-4.1		2.3	
	8.0	15.0				4.2-4.3		3.0	
	9.0	16.0	22.0	5.1		5.0-5.1		4.0	
Current	10.0	17.0	23.0	6.0	12.1	6.0	5.0-7.0	4.1	7.0
Near future		18.0	24.0		12.5				10.0
Farther future		19.0	25.0						

运动传感器 Device Motion API

- 用途：获取设备运动状态的加速度
- 事件属性
 - `accelerationIncludingGravity`（含重力的加速度）
 - `acceleration`（排除了重力的影响的加速度）



运动传感器 Device Motion API – 代码示例

```
function handleMotionEvent(event) {  
    var x = event.accelerationIncludingGravity.x;  
    var y = event.accelerationIncludingGravity.y;  
    var z = event.accelerationIncludingGravity.z;  
    // Process ...  
}  
window.addEventListener("devicemotion", handleMotionEvent, true);
```

运动传感器 Device Motion API –实现“摇一摇”

```
var SHAKE_THRESHOLD = xxx;
var last_update = 0;
var x, y, z, last_x, last_y, last_z;

function deviceMotionHandler(eventData) {
    var acceleration = eventData.accelerationIncludingGravity;

    var curTime = new Date().getTime();

    if ((curTime - lastUpdate) > 100) {

        var diffTime = curTime - last_update;
        last_update = curTime;

        x = acceleration.x;
        y = acceleration.y;
        z = acceleration.z;

        var delta = Math.abs(x + y + z - last_x - last_y - last_z) / diffTime;

        if ((delta * 10000) > SHAKE_THRESHOLD) {
            alert("shaked!");
        }
        last_x = x;
        last_y = y;
        last_z = z;
    }
}
```

谢谢

- 联系我: Benfeng@gmail.com
- 加入我们



<http://www.chinaw3c.org/hire.html>

- Web应用专家 (Web Application Expert)
- Web应用封装专家 (Packaged Web Applications Specialist)
- 系统管理员 (Systems administrator)
- Web信息无障碍工程师 (Web Accessibility Engineer)



<http://yunshipei.com/jobs.html>

- HTML5 移动前端开发工程师
- Node.js 开发工程师