1. Zepto 对象 不能自定义事件

例如执行： $({}).bind('cust', function(){});

结果： TypeError: Object has no method 'addEventListener'

解决办法是创建一个脱离文档流的节点作为事件对象：

例如： $('').bind('cust', function(){});

2. Zepto 的选择器表达式: [name=value] 中value 必须用 双引号 " or 单引号 ' 括起来

例如执行：$('[data-userid=123123123]')

结果：Error: SyntaxError: DOM Exception 12

解决办法： $('[data-userid="123123123]"') or $("[data-userid='123123123']")

2-1.zepto的选择器没有办法选出 $("div[name!='abc']") 的元素

2-2.zepto获取select元素的选中option不能用类似jq的方法$('option[selected]'),因为selected属性不是css的标准属性

应该使用$('option').not(function(){ return !this.selected })

比如：jq:$this.find('option[selected]').attr('data-v') \* 1

zepto:$this.find('option').not(function() {return !this.selected}).attr('data-v') \* 1

但是获取有select中含有disabled属性的元素可以用 $this.find("option:not(:disabled)") 因为disabled是标准属性

2-3、zepto在操作dom的selected和checked属性时尽量使用prop方法，以下是官方说明：

prop v1.0+

prop(name) ⇒ value

prop(name, value) ⇒ self

prop(name, function(index, oldValue){ ... }) ⇒ self

读取或设置dom元素的属性值。它在读取属性值的情况下优先于 attr，因为这些属性值会因为用户的交互发生改变，如checked 和 selected。

3.Zepto 是根据标准浏览器写的，所以对于节点尺寸的方法只提供 width() 和 height()，省去了 innerWidth(), innerHeight(),outerWidth(),outerHeight()

Zepto.js: 由盒模型（ box-sizing ）决定

jQuery: 忽略盒模型，始终返回内容区域的宽/高（不包含 padding 、 border ）解决方式就是使用 .css('width') 而不是 .width() 。

3-1.边框三角形宽高的获取

假设用下面的 HTML 和 CSS 画了一个小三角形：

<div class="caret"></div>

.caret {

width: 0;

height: 0;

border-width: 0 20px 20px;

border-color: transparent transparent blue;

border-style: none dotted solid;

}

jQuery 使用 .width() 和 .css('width') 都返回 ，高度也一样；

Zepto 使用 .width() 返回 ，使用 .css('width') 返回 0px 。

所以，这种场景，jQuery 使用 .outerWidth() / .outerHeight() ；Zepto 使用 .width() / .height() 。

3-2.offset()

Zepto.js: 返回 top 、 left 、 width 、 height

jQuery: 返回 width 、 height

3-3.隐藏元素

Zepto.js: 无法获取宽高；

jQuery: 可以获取。

4.Zepto 的each 方法只能遍历数组，不能遍历JSON对象

5.Zepto 的animate 方法参数说明 ，即zepto中animate的用法

$("#some\_element").animate(

{

opacity: 0.25,

left: '50px',

color:'#abcdef',

rotateZ:'45deg',

translate3d: '0,10px,0'

}, //一个动画对象的各个参数

500, //时间

'ease-out') //动画类型

6.zepto的jsonp callback函数名无法自定义

7.DOM 操作区别

jq代码：

(function($) {

$(function() {

var $list = $('<ul><li>jQuery 插入</li></ul>', {

id: 'insert-by-jquery'

});

$list.appendTo($('body'));

});

})(window.jQuery);

jQuery 操作 ul 上的 id 不会被添加。

zepto代码：

Zepto(function($) {

var $list = $('<ul><li>Zepto 插入</li></ul>', {

id: 'insert-by-zepto'

});

$list.appendTo($('body'));

});

Zepto 可以在 ul 上添加 id 。

8.事件触发区别

jq代码：

(function($) {

$(function() {

$script = $('<script />', {

src: 'htt p:/ /cdn.amaze ui.o rg/amazeui/1.0.1/js/amazeui.min.js',

id: 'ui-jquery'

});

$script.appendTo($('body'));

$script.on('load', function() {

console.log('jQ script loaded');

});

});

})(window.jQuery);

使用 jQuery 时 load 事件的处理函数不会执行

zepto代码：

Zepto(function($) {

$script = $('<script />', {

src: 'http:/ /cdn.amazeui.org/amazeui/1.0.1/js/amazeui.js',

id: 'ui-zepto'

});

$script.appendTo($('body'));

$script.on('load', function() {

console.log('zepto script loaded');

});

});

使用 Zepto 时 load 事件的处理函数会执行。