# 什么是zepto.js

**概念： 移动端开发框架,是jquery的轻量级代替品；API及语句同jquery相似，但文件更小(可压缩至8KB)。**

**是目前功能完备的库中，最小的一个。**

**\* zepto.js特点**

**1、针对的是移动端**

**2、轻量级，压缩版本只有8KB**

**3、语法大部分同jquery一样，学习成本低，上手快。**

**4、响应，执行快。**

**5、同jquery一样以$作为核心函数和核心对象。**

# Zepto和jQuery的异同

## 相同

**\* 共同点：**

**jquery:**

**\* 核心函数$**

**1、作为函数使用（参数）**

**1.function**

**2.html字符串**

**3.DOM code**

**4.选择器字符串**

**2、作为对象调用(方法)**

**1.$.ajax() $.get() $.post()**

**2.$.isArray()**

**$.each()**

**$.isFunction()**

**$.trim()**

**......**

**\* jquery对象**

**概念：jquery核心函数$()调用返回的对象就是jquery对象的数组（可能有只有一个）；**

**使用：**

**1.addClass()**

**2.removeClass()**

**3.show()**

**4.find()**

**zepto:**

**以上jquery的特性zepto同样适用**

## 区别

### attr与prop的区别

**jQuery:**

**\* 1、prop多用在标签的固有属性，布尔值属性。比如：a标签的href，class，selected等。**

**\* 2、attr多用在自定义属性上。**

**\* 3、在jquery中如果用attr去获取布尔值属性且该布尔值属性在标签体内没有定义的时候，会返回undefined**

**\*/**

**$('option').each(function (index, item) {**

**//console.log(index, item.innerHTML);**

**//console.log($(this).attr('selected'));//没有定义获取到的是undefined**

**console.log($(this).prop('selected'));//4false 1true**

**});**

**//在zepto中用attr也可以获取布尔值属性.**

**//prop在读取属性的时候优先级高于attr，布尔值属性的读取还是建议用prop**

**//坑！------zepto中removeProp()的方法。在1.2及以上才支持。**

**$(function () {**

**$('option').each(function (index, item) {**

**//console.log($(this).attr('selected'));**

**console.log($(this).prop('selected'));**

**});**

**$('#btn').click(function () {**

**$(this).prop('disabled', 'true');**

**setTimeout(function () {**

**$('#btn').removeAttr('disabled');**

**}, 2000);**

**});**

**});**

### DOM操作

**<!--**

**jquery中插入DOM元素的时候添加配置对象（属性选择器：id，class。。。)的时候不会显示；**

**-->**

**<div id="box"></div>**

**<script src="../../../js/jquery-1.10.1.min.js" type="text/javascript" charset="utf-8"></script>**

**<script type="text/javascript">**

**$(function () {**

**var $insert = $('<p>我是新添加的p标签</p>', {**

**id:'insert'**

**});**

**$('#box').append($insert)**

**});**

**Zepto：**

**<!--**

**插入DOM元素的时候添加配置对象的时候可以添加进去，并且添加的配置对象的内容会直接反应在标签属性内，且可以操作，影响对应的DOM元素。**

**-->**

**<div id="box"></div>**

**<script src="../../../js/zepto.js" type="text/javascript" charset="utf-8"></script>**

**<script type="text/javascript">**

**$(function () {**

**var $insert = $('<p>我是新添加的p标签</p>', {**

**id:'insert',**

**class:'insert'**

**});**

**$('#box').append($insert)**

**});**

### Each方法

**jQuery：**

**//$.each(collection, function(index, item){ ... })**

**//可以遍历数组,以index，item的形式，**

**//可以遍历对象，以key-value的形式**

**//不可以遍历字符串。**

**//不可以遍历json字符串**

**var arr = [1,'a',3,'b',5];**

**var obj = {name:'tom',age:13};**

**//遍历数组，**

**$.each(arr, function(index,item) {**

**console.log(index + '-' + item);**

**});**

**$.each(obj, function(key,value) {**

**console.log(key + '-' + value);**

**});**

**//不可以遍历字符串。**

**var str = 'adfdfdf';**

**$.each(str,function(index){**

**console.log(index);**

**})**

**Zepto：**

**//$.each(collection, function(index, item){ ... })**

**//可以遍历数组,以index，tiem的形式，**

**//可以遍历对象，以key-value的形式，**

**//可以遍历字符串。**

**var arr = [1,'a',3,'b',5];**

**var obj = {name:'tom',age:13};**

**$.each(arr, function(index,item) {**

**console.log(index + '-' + item);**

**});**

**$.each(obj, function(key,value) {**

**console.log(key + '-' + value);**

**});**

**var str = 'abcdef';**

**//可以遍历字符串，同对数组的遍历方法一样以index-item的形式。**

**$.each(str, function(index,item) {**

**console.log(index + '-' + item);**

**});**

### Offset方法

**jQuery：**

**<!--**

**1、获取匹配元素在当前视口的相对偏移。**

**2、返回的对象包含两个整型属性：top 和 left。此方法只对可见元素有效。**

**-->**

**<div id="box">新年快乐，大吉大利</div>**

**<script src="../../../js/jquery-1.10.1.min.js" type="text/javascript" charset="utf-8"></script>**

**<script type="text/javascript">**

**$(function(){**

**var $offset = $('#box').offset();**

**console.log($offset);**

**console.log($offset.top);**

**//获取width，height时为undefined**

**console.log($offset.width);**

**})**

**Zepto**

**<!--**

**\* offset()**

**1、获得当前元素相对于视口的位置。**

**2、返回一个对象含有： top, left, width和height(获取的宽高是盒模型可见区域的宽高)**

**-->**

**<div id="box">新年快乐，大吉大利</div>**

**<script src="../../../js/zepto.js" type="text/javascript" charset="utf-8"></script>**

**<script type="text/javascript">**

**$(function(){**

**var $offset = $('#box').offset();**

**console.log($offset);**

**console.log($offset.left);**

**console.log($offset.width);**

**})**

**</script>**

### Width() height()方法

**/\***

**\* jquery获取宽高的方法**

**1、width(),height()---content内容区的宽高，没有单位px;**

**2、.css('width')----可以获取content内容区的宽高，padding，border的值，有单位px;**

**3、innerHeight(),innerWidth()---outerHeight(),outerWidth()来获取**

**\* \*/**

**$(function(){**

**//jquery中width(),height(),.css()返回的始终是content区域的宽高**

**console.log($('#box').width());**

**console.log($('#box').height());**

**console.log($('#box').css('width'));**

**//可以通过css（）获取padding，border的值**

**console.log($('#box').css('padding'));**

**console.log($('#box').css('border-width'));**

**//也可以利用innerHeight(),outerHeight(),innerWidth(),outerWidth()等来获取padding和border的值**

**console.log($('#box').innerHeight());**

**console.log($('#box').outerHeight()+'outerHeight');**

**})**

**$(function(){**

**//zepto用width(),height()是根据盒模型决定的，并且不包含单位PX**

**//包含padding的值,border的值**

**//zepto中没有innerHeight(),innerWidth()---outerHeight(),outerWidth()**

**console.log($('#box').width());**

**console.log($('#box').height());**

**//用.css('width')获取的是content的宽高。**

**console.log($('#box').css('width'));**

**//\* 单独获取padding，border的值**

**console.log($('#box').css('padding'));**

**console.log($('#box').css('border-width')+'border');**

**})**

### 事件委托

**jQuery：**

**$(function(){**

**console.log($('.a').length);**

**//1.7以后已经不 支持live了。**

**// $('#a').live('click',function(){**

**// alert('a');**

**// })**

**// $('#box').delegate('.a','click',function(){**

**// alert('delegate');**

**// })**

**//在jquery中事件委托只是找相应的event.target触发元素进行的回调函数执行,其他的并不关心。**

**$('#box').on("click",'.a',function(){**

**alert('a事件');**

**$(this).removeClass('a').addClass('b');**

**});**

**$('#box').on("click",'.b',function(){**

**alert('b事件');**

**$(this).removeClass('b').addClass('a');**

**});**

**$('#box').append('<div class="a" style="font-size: 30px;">我是新添加的div</div>')**

**})**

**Zepto**

**$(function(){**

**//在zepto的官网表示已经废除了live，delegate等。**

**// $('.a').live('click',function(){**

**// alert('a')**

**// })**

**//\* 坑！！！！**

**/\***

**\* ---在zepto中事件委托**

**委托的事件先被依次放入数组队列里，然后由自身开始往后找直到找到最后，期间符合条件的元素委托的事件都会执行。**

**1、委托在同一个父元素,或者触发的元素的事件范围小于同类型事件(冒泡能冒到自身范围)**

**2、同一个事件**

**3、委托关联 操作的类要进行关联**

**4、绑定顺序---从当前的位置往后看**

**\*/**

**$('#box').on("click",'.a',function(){**

**alert('a事件');**

**$(this).removeClass('a').addClass('b');**

**});**

**$('#box').on("click",'.b',function(){**

**alert('b事件');**

**$(this).removeClass('b').addClass('a');**

**});**

**$('#box').append($('<p>我是新添加的p标签</p>',{class:'a'}));**

**})**

### 获取隐藏元素的宽高

**<!--**

**jquery可以获取隐藏元素的宽高**

**-->**

**<div id="box"></div>**

**<script src="../../../js/jquery-1.10.1.min.js" type="text/javascript" charset="utf-8"></script>**

**<script type="text/javascript">**

**$(function () {**

**console.log($('#box').width());**

**console.log($('#box').height());**

**});**

**</script>**

**<!--**

**zepto无法获取隐藏元素宽高**

**-->**

**<div id="box">hello</div>**

**<script src="../../../js/zepto.js" type="text/javascript" charset="utf-8"></script>**

**<script type="text/javascript">**

**$(function () {**

**console.log($('#box').width());**

**console.log($('#box').height());**

**});**

**</script>**

# Touch event

## 同jQuery类似事件

**<!--**

**on() 绑定事件处理程序**

**off() 方法移除用目标oon绑定的事件处理程序。**

**bind() 为每个匹配元素的特定事件绑定事件处理函数，可同时绑定多个事件，也可以自定义事件。**

**one() 为每一个匹配元素的特定事件（像click）绑定一个一次性的事件处理函数。只执行一次。**

**trigger() 触发有bind定义的事件（通常是自定义事件）**

**unbind() bind的反向操作，删除匹配元素所绑定的bind事件。**

**-->**

## Zepto

**<!--**

**\* zepto touch方法**

**\* tap()点击事件 利用在document上绑定touch事件来模拟tap事件的，并且tap事件会冒泡到document上**

**\* singleTap() 点击事件**

**\* doubleTap() 双击事件**

**\* longTap() 当一个元素被按住超过750ms触发。**

**\* swipe, swipeLeft, swipeRight, swipeUp, swipeDown — 当元素被划过(同一个方向滑动距离大于30px)时触发。(可选择给定的方向)**

**在一个方向滑动大于30px即为滑动。否则算点击。**

**要给一个元素设置swipe事件，必须在这个元素上的css设置touch-action:none;来消除默认的拖动滚动条，否则无效**

**-->**

**<div id="box" class="box">按钮1</div>**

**<br />**

**<div id="box2" class="box">按钮2</div>**

**<br />**

**<div id="box3" class="box">按钮3</div>**

**<button id="btn" class="btn">button</button>**

**<script src="../js/zepto.js" type="text/javascript" charset="utf-8"></script>**

**<script src="../js/touch.js" type="text/javascript" charset="utf-8"></script>**

**<script type="text/javascript">**

**$(function () {**

**//tap()点击事件---真机测没问题**

**// $('#box').tap(function () {**

**// alert('tap事件');**

**// });**

**$('#box').on('tap',function () {**

**alert('tap事件');**

**});**

**//singleTap()单击事件**

**$('#box2').singleTap(function () {**

**alert('我单击了一下');**

**});**

**//doubleTap() 双击事件**

**$('#box2').doubleTap(function () {**

**alert('我双击了一下');**

**});**

**//longTap()长按事件----按住屏幕时间超过750ms触发**

**$('#box3').longTap(function () {**

**alert('我长按了一下box3');**

**});**

**//swipe()滑动事件---在同一个方向连续滑动超过30px**

**// $('#btn').swipe(function () {**

**// alert('我滑动了超过30px');**

**// });**

**$('#btn').swipeLeft(function () {**

**alert('我向左滑动了');**

**});**

**});**

**</script>**

**\* zepto有自己的一套事件机制，并且对不同的浏览器做了兼容的内部封装处理。**

**\* 1、像新版本的zepto中已经舍弃了bind，delegate，die，同样jquery中舍弃了live，die等。**

**\* 2、现在统一使用on，off标准事件来绑定事件。**

**\*/**

**$(function(){**

**//简单绑定**

**$('#box').on('touchstart',function(){**

**alert('ddd');**

**});**

**//事件委托**

**$('#box1').on('touchstart','p',function(){**

**alert($(this).text());**

**});**

**//解除绑定事件**

**//$('#box').off('touchstart');**

**//一次函数**

**$('#box2').one('touchstart',function(){**

**alert('我就执行一次');**

**});**

**})**