目录

[1：操作DOM 1](#_Toc14144)

[2：事件 3](#_Toc30087)

[应用： 4](#_Toc11807)

[3：ajax 4](#_Toc3653)

[4：核心 8](#_Toc25464)

[5：工具函数 9](#_Toc2204)

[应用： 10](#_Toc23012)

[综合应用： 11](#_Toc22306)

## 1：操作DOM

创建 var $div = $(‘<div></div>’)

**内部插入操作：**

append(content|fn)：向每个匹配的元素内部追加内容。

prepend(content)：向每个匹配的元素内部前置内容。

**外部插入操作：**

after(content|fn)：在每个匹配的元素之后插入内容。

before(content|fn)：在每个匹配的元素之前插入内容。

**包裹操作：**

wrap(html|element|fn)：把所有匹配的元素用其他元素的结构化标记包裹起来。

unwrap()：这个方法将移出元素的父元素。

wrapAll(html|ele)：将所有匹配的元素用单个元素包裹起来。

wrapInner(htm|element|fnl)：将每一个匹配的元素的子内容(包括文本节点)用一个HTML结构包裹起来。

**替换操作：**

replaceWith(content|fn)：将所有匹配的元素替换成指定的HTML或DOM元素。

replaceAll(selector)：用匹配的元素替换掉所有 selector匹配到的元素。

**删除操作：**

empty()：删除匹配的元素集合中所有的子节点。

remove([expr])：从DOM中删除所有匹配的元素。这个方法不会把匹配的元素从jQuery对象中删除，因而可以在将来再使用这些匹配的元素。但除了这个元素本身得以保留之外，其他的比如绑定的事件，附加的数据等都会被移除。

detach([expr])：从DOM中删除所有匹配的元素。这个方法不会把匹配的元素从jQuery对象中删除，因而可以在将来再使用这些匹配的元素。与remove()不同的是，所有绑定的事件、附加的数据等都会保留下来。

**复制操作：**

clone(Even])：克隆匹配的DOM元素并且选中这些克隆的副本。

even：一个布尔值（true 或者 false）指示事件处理函数是否会被复制。

## 2：事件

**页面载入事件：**

ready(fn)：当DOM载入就绪可以查询及操纵时绑定一个要执行的函数。这是事件模块中最重要的一个函数，因为它可以极大地提高web应用程序的响应速度。简单地说，这个方法纯粹是对向window.load事件注册事件的替代方法。

**绑定事件：**

on(events,[selector],[data],fn)：在选择元素上绑定一个或多个事件的事件处理函数。

off(events,[selector],[fn])：在选择元素上移除一个或多个事件的事件处理函数。

bind(type,[data],fn)：为每个匹配元素的特定事件绑定事件处理函数。

unbind(type,[data|fn]])：bind()的反向操作，从每一个匹配的元素中删除绑定的事件。如果没有参数，则删除所有绑定的事件。

one(type,[data],fn)：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。

hover([over,]out)：当鼠标移动到一个匹配的元素上面时，会触发指定的第一个函数。当鼠标移出这个元素时，会触发指定的第二个函数。

click([[data],fn])：触发每一个匹配元素的click事件。这个函数会调用执行绑定到click事件的所有函数。

注：其他事件方法使用方式一样。例如：mouseover、mouseout、dblclick、change、blur、focus、keydown、keyup、keypress、mousedown、mouseup、mousemove、mouseenter、mouseleave、resize、scroll、select、submit、unload等。

## 应用：

1：选项卡。



## 3：ajax

**load**(url, [data], [callback])：载入服务器HTML文件代码并插入至DOM中。默认为GET请求。

注：

1：url不一定是html文件地址，css、js、txt、php等可以。

2：筛选加载进来的HTML文档：Load(url #content p)。

3：data可以是key/value数据，也可以是字符串类型。如果是键值对对象类型，则为POST请求，如果是字符串，则为GET请求。

4：回调函数三个参数：1：获取的HTML代码，2：请求状态的描述，3：XHR对象。

**jQuery.ajax**(url, [settings])：通过HTTP请求加载远程数据，返回其创建的XHR对象。

url：请求地址，settings也有url配置选项，如果同时设置，则以外面的url为基准。如果都不设置，则默认请求的是当前发起请求的页面。

settings配置选项：

dataType：支持的数据类型：xml，html、json、jsonp、script、text。

text和xml类型返回的数据不会经过处理。数据仅仅将简单的XHR的responseText和responseXML属性传递给success回调函数。

script类型：jquery会先执行服务器端生成的JS代码，然后将脚本作为一个文本数据返回。

jsonp类型：会创建一个查询字符串参数callback=?，这个参数会加在请求的URL后面。

xml类型：返回XML文档，可用jquery处理。

html类型：返回纯文本HTML信息，包含的script标签会在插入DOM时执行。

json类型：返回JSON数据。

text类型：返回纯文本字符串。

async：Boolean类型。默认为true（异步）。

contentType：string类型。默认为application/x-www-form-urlencoded，发送信息至服务器时内容编码类型。

context：object类型。这个对象用于设置AJAX相关回调函数的上下文，也就是说，让回调函数内部的this指向这个对象。

timeout：设置请求超时时间（毫秒）。

type：请求方式（GET或POST），默认为GET。

url：默认当前页地址，发送请求的地址。

data：object/string类型。发送到服务器的数据。

jsonp：string类型。在一个jsonp请求中重写回调函数的名字。这个值用来替代”callback=?”中的callback部分。

jsonpCallback：string类型。为jsonp请求指定一个回调函数名。这个值用来取代jQuery自动生成的随机函数名。

statusCode：map类型。一组数值的HTTP代码和函数对象，当响应时调用相应的代码。

回调函数：

beforeSend：在发送请求之前调用，并且传入XHR作为参数。

error：在请求出错时调用。传入XHR对象，描述错误类型的字符串，以及异常对象（如果存在的话）。

dataFilter：在请求成功后调用，传入返回的数据以及dataType参数的值，**并且必须返回新的数据给success回调函数**。

success：当请求成功之后调用，传入返回后的数据，以及包含成功代码的字符串。

complete：当请求完成后调用这个函数，无论成功或失败。传入XHR对象，以及包含成功或错误代码的字符串。

beforeSend：function(xhr, ajax){}

dataFilter：function (data, dataType){}

success：function(data, sInfo, xhr){}

complete：function(xhr, sInfo){}

error：function(xhr, sInfo[, exception]){}

**成功执行顺序：beforeSend、dataFilter、success、complete**

**失败执行顺序：beforeSend、error、complete**

**$.get**(url, [data], [callback], [type])：通过HTTP GET请求获取数据。这是一个简单的GET请求取代复杂的$.ajax。请求成功时可调用回调函数。如果需要在出错时执行函数，则使用$.ajax。

type值：\_default。HTML或者XML取决于返回值。

**$.post**(url, [data], [callback], [type])：通过HTTP POST请求获取数据。这是一个简单的POST请求取代复杂的$.ajax。请求成功时可调用回调函数。如果需要在出错时执行函数，则使用$.ajax。

type值：\_default。HTML或者XML取决于返回值。

**$.getJSON**(url, [data], [callback])：通过HTTP GET请求获取JSON数据。可以通过使用JSONP形式的回调函数来加载其他网站的数据。如：”myulr?cb=?”，jQuery会自动替换 ? 为正确的函数名，以执行回调函数。

**$.getScript**(url, [callback])：通过HTTP GET请求载入并执行一个JS文件。

**$(DOM).serialize()**：序列**表单内容**为字符串，用于AJAX请求。

**$(DOM).serializeArray()**：序列化表单元素，返回JSON数据结构数据。

## 4：核心）

$(“.box”).each()

**对象访问：**

**each(callback)**：以每一个匹配的元素作为上下文来执行一个函数。回调函数有两个参数，第一个参数代表索引，第二个参数代表当前遍历到的DOM对象。return true则跳至下一个循环(就像在普通循环中使用continue)，return false则终止循环（就像在普通循环中使用break）。

**size()**： jQuery 对象中元素的个数。

**get([index])**：取得其中一个匹配的元素。 index表示取得第几个匹配的元素。与$(this)[0]等价。

**index([selector|element])**：搜索匹配的元素，并返回相应元素的索引值，从0开始计数。如果参数是一组DOM元素或者jQuery对象，那么返回值就是传递的元素相对于原先集合的位置。如果不传递参数，那么返回值就是这个jQuery对象集合中第一个元素相对于其同辈元素的位置。

**数据缓存：**

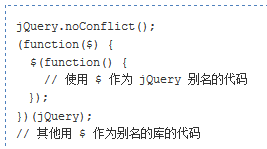
**data([key],[value])** ：在元素上存放数据,返回jQuery对象。

**removeData([name|list])**：在元素上移除存放的数据。

**多库共存：**

**jQuery.noConflict()**：运行这个函数将变量$的控制权让渡给第一个实现它的那个库。

jQuery(“.box”)



## 5：工具函数

$.each()

**jQuery.each(object, [callback])：**通用例遍方法，可用于例遍对象和数组。此方法可用于例遍任何对象。回调函数拥有两个参数：第一个为对象的成员或数组的索引，第二个为对应变量或内容。

**jQuery.grep(array, callback, [invert])**：使用过滤函数过滤数组元素。此函数至少传递两个参数：待过滤数组和过滤函数。过滤函数必须返回 true 以保留元素或 false 以删除元素。如果 "invert" 为 false 或为设置，则函数返回数组中由过滤函数返回 true 的元素，当"invert" 为 true，则返回过滤函数中返回 false 的元素集。

**jQuery.map(arr|obj,callback)**：将一个数组中的元素转换到另一个数组中。

**jQuery.inArray(value,array,[fromIndex])：**确定第一个参数在数组中的位置，从0开始计数(如果没有找到则返回 -1 )。

**jQuery.toArray()**：把jQuery集合中所有DOM元素恢复成一个数组。

**jQuery.merge(first,second)**：合并两个数组。

**jQuery.unique(array)**：删除数组中重复元素。只处理删除DOM元素数组，而不能处理字符串或者数字数组。

**jQuery.parseJSON(json)**：接受一个JSON字符串，返回解析后的对象。传入一个畸形的JSON字符串会抛出一个异常。

**jQuery.trim(str)：**去掉字符串起始和结尾的空格。

**jQuery.contains(container,contained)：**一个DOM节点是否包含另一个DOM节点。

**jQuery.type(obj)**：检测obj的数据类型。

**jQuery.isArray(obj)**：测试对象是否为数组。

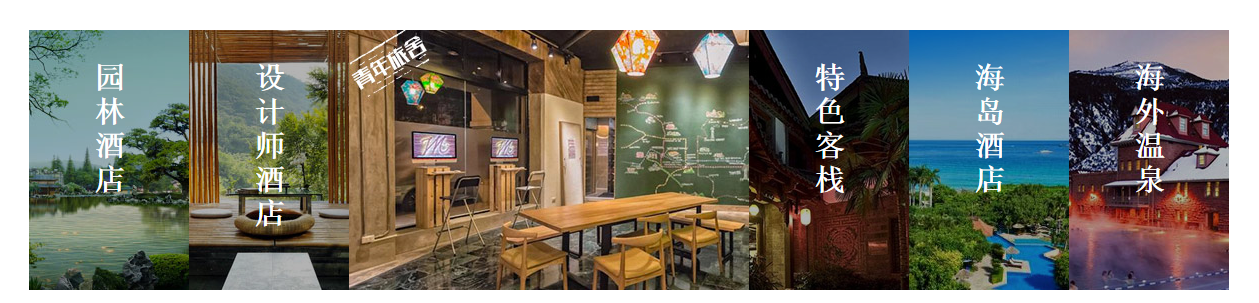
**jQuery.isFunction(obj)**：测试对象是否为函数。

## 应用：

1：京东首页图片轮播。



2：手风琴。



## 综合应用：

1：掌握jquery的常见API。