本文以实例的形式总结了jQuery的常见实用函数。分享给大家供大家参考之用。具体示例如下：

**1.修剪字符串**

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1 | $('#id').val($.trim($('#someid').val())) |

**2.遍历集合**

可能这样写：

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4 | var anArray = ['one','two'];  for(var n = 0; n < anArray.length; n++){    } |

还有可能这样写：

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4 | var anObject = {one: 1, two: 2};  for(var p in anObject){    } |

但有了$.each函数后，就可以这样写了：

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | var anArray = ['one','two'];  $.each(anArray, funtion(n, value){    })  var anObject = {one: 1, two: 2};  $.each(anObjct, function(name,value){    }) |

**3.筛选数组**

使用$.grep()方法能筛选数组。先来看grep方法的定义：

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | grep: function(elems, callback, inv){    var ret = [], retVal;    inv = !!inv;    for(var i = 0; length = elems.length; i < length; i++){      retVal = !!callback(elems[i],i)      if(inv !== retVal){        ret.push(elems[i]);      }    }    return ret;  } |

以上示例中：  
① grep方法的第二个参数是回调函数，该回调函数接收2个参数，一个是数组的元素，一个是数组的索引。  
② grep方法的第三个参数inv，默认情况下是undefined，所以!!inv为false，即inv的默认值是false

举例1：int类型数组

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5 | var arr = [1, 2, 3, 4, 5, 6];  arr = $.grep(arr, function(val, index){    return val > 3;  })  console.log(arr);//结果是：4 5 6 |

如果把grep的第三个参数显式地设置为true，结果怎样呢？

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5 | var arr = [1, 2, 3, 4, 5, 6];  arr = $.grep(arr, function(val, index){    return val > 3;  }, true)  console.log(arr);//结果是：1 2 3 |

可见，当把grep方法的第三个参数设置为true后，会把不符合回调函数的数组元素过滤出来。

举例2：object类型数组

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | var arr = [    {      first: "Jeffrey",      last: 'Way'    },{      first: 'Allison',      last: 'Way'    },{      first: 'John',      last: 'Doe'    },{      first: 'Thomas',      last: 'Way'    };    arr = $.grep(arr, function(obj, index){      return obj.last === 'Way';    });    console.log(arr);  ]; |

**4.转换数组**

使用$.map(arr, callback)为数组的每一个元素调用回调函数，并返回一个新的数组

给数组的每一个元素加1:

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1 | var oneBased = $.map([0, 1, 2, 3, 4], function(value){return value +1;}) |

把字符串数组转换成整型数字数组，判断数组元素是否是数字：

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5 | var strings = ['1', '2', '3','4','S','6'];  var values = $.map(strings, function(value){    var result = new Number(value);    return isNaN(result) ? null : result;  }) |

把转换后的数组合并到原先的数组中：

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1 | var chars = $.map(['this','that'], function(value){return value.split(' ')}); |

**5.返回数组元素的索引**

使用$.inArray(value, array)返回传入的值第一次出现的下标，即索引。

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1 | var index = $.inArray(2, [1, 2, 3]); |

**6.将对象转换成数组**

$.makeArray(object)将传入类似数组的对象转换成Javascript数组。

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | <div>First</div>  <div>Second</div>  <div>Third</div>  <div>Fourth</div>  <script>    var elems = document.getElementsByTagName("div");    var arr = jQuery.makeArray(elems);    arr.reverse();    $(arr).appendTo(document.body);  </script> |

**7.得到不含重复元素的数组**

使用$.unique(array)返回由原始数组中不重复的元素组成的数组。

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | <div>There are 6 divs in this document.</div>  <div></div>  <div class="dup"></div>  <div class="dup"></div>  <div class="dup"></div>  <div></div>  //把到所有div，get方法转换成javascript数组，总共6个div  var divs = $("div").get();  //再把3个class名为dup的div合并到前面的6个div  divs = divs.concat($(".dup").get());  alert(divs.length); //9个div  //过滤去掉重复  divs = jQuery.unqiue(divs);  alert(divs.length);//6个div |

**8.合并2个数组**

$.merge(array1, array2)把第二个数组合并到第一个数组中，并返回第一个数组。

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4 | var a1 = [1, 2];  var a2 = [2, 3];  $.merge(a1, a2);  console.log(a1);//[1, 2, 2, 3] |

**9.把对象序列化成查询字符串**

$.param(params)把传入的jquery对象或javascript对象转换成字符串形式。

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | $(document).ready(function(){   personObj=new Object();   personObj.firstname="John";   personObj.lastname="Doe";   personObj.age=50;   personObj.eyecolor="blue";   $("button").click(function(){    $("div").text($.param(personObj));   });  }); |

结果：firstname=John&lastname=Doe&age=50&eyecolor=blue

**10.一些判断函数**

$.isArray(o) 如果o是javascript数组，则返回true,如果是类似数组的jquery对象数组，返回false  
$.isEmptyObject(o) 如果o是不包含属性的javascript对象，则返回true  
$.isFunction(o) 如果o是javascript函数就返回true  
$.isPlainObject(o) 如果o是通过{}或new Object()创建的对象，则返回true  
$.isXMLDoc(node) 如果node是XML文档或者是XML文档中的节点，则返回true

**11.判断一个元素是否包含在另外一个元素中**

$.contains(container, containee)第二个参数是被包含

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2 | $.contains( document.documentElement, document.body ); // true  $.contains( document.body, document.documentElement ); // false |

**12.把值存储到某元素上**

$.data(element, key, value)第一个是javascript对象，第二、第三个是键值。

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | //得到一个div的javascript对象  var div = $("div")[0];  //把键值存储到div上  jQuery.data(div, "test",{    first: 16,    last: 'pizza'  })  //根据键读出值  jQuery.data(div, "test").first  jQuey.data(div, "test").last |

**13.移除存储到某元素上的值**

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | <div>value1 before creation: <span></span></div>  <div>value1 after creation: <span></span></div>  <div>value1 after removal: <span></span></div>  <div>value2 after removal: <span></span></div>  var div = $( "div" )[ 0 ];  //存储值  jQuery.data( div, "test1", "VALUE-1" );  //移除值  jQuery.removeData( div, "test1" ); |

**14.绑定函数的上下文**

jQuery.proxy( function, context)返回一个新的function函数，上下文是context。

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | $(document).ready(function(){   var objPerson = {    name: "John Doe",    age: 32,    test: function(){     $("p").after("Name: " + this.name + "<br> Age: " + this.age);    }   };   $("button").click($.proxy(objPerson,"test"));  }); |

以上，点击按钮，执行的是test方法，不过test方法的上下文做了设置。

**15.解析JSON**

jQuery.parseJSON( json )第一个参数json的类型是字符串。

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2 | var obj = jQuery.parseJSON( '{ "name": "John" }' );  alert( obj.name === "John" ); |

**16.表达式求值**

有时候，希望一段代码在全局上下文中执行，可以使用jQuery.globalEval( code )。code的类型是字符串。

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4 | function test() {   jQuery.globalEval( "var newVar = true;" )  }  test(); |

**17.动态加载脚本**

$(selector).getScript(url,success(response,status))用来动态加载js文件，第一个参数是js的文件路径，第二个参数可选，表示获取js文件成功的回调。

[?](http://www.jb51.net/article/54487.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | $.getScript( "ajax/test.js", function( data, textStatus, jqxhr ) {   console.log( data ); // Data returned   console.log( textStatus ); // Success   console.log( jqxhr.status ); // 200   console.log( "Load was performed." );  }); |