

jQuery中的100个技巧

1.当document文档就绪时执行JavaScript代码。

我们为什么使用jQuery库呢？原因之一就在于我们可以使jQuery代码在各种不同的浏览器和存在bug的浏览器上完美运行。



```
<script src="http://code.jquery.com/jquery-1.10.2.min.js"></script>
```

```
<script>
```

```
    // Different ways to achieve the Document Ready event
```

```
    // With jQuery
```

```
    $(document).ready(function() { /* ... */});
```

```
    // Short jQuery
```

```
    $(function() { /* ... */});
```

```
    // Without jQuery (doesn't work in older IE versions)
```

```
    document.addEventListener('DOMContentLoaded', function() {
```

```
        // Your code goes here
```

```
    });
```

```
    // The Trickshot (works everywhere):
```

```
    r(function() {
```

```
        alert('DOM Ready!');
```

```
    })
```

搜索

找找看

随笔分类

Ajax(1)
Bootstrap(1)
chrome(6)
Computer English(5)
CSS3(4)
ES6(2)
HTML&CSS(16)
HTML5(2)
JavaScript(23)
jQuery(3)
JSON&&XML(2)
photoshop与切图(2)
PHP(1)
Regular Expression(4)
sass/less(2)
产品(1)
常用软件和插件(5)
关于前端(3)
计算机(3)
前端性能优化(1)
实例(3)
网站记录(3)
问题记录(3)

```
function r(f){in/.test(document.readyState)?setTimeout('r('+f+')',9):f() }
```

```
</script>
```



2.使用route.



```
<script src="http://code.jquery.com/jquery-1.10.2.min.js"></script>
```

```
<script>
```

```
var route = {  
  _routes : {},    // The routes will be stored here  
  
  add    : function(url, action){  
    this._routes[url] = action;  
  },  
  
  run : function(){  
    jQuery.each(this._routes, function(pattern){  
      if(location.href.match(pattern)){  
        // "this" points to the function to be executed  
        this();  
      }  
    });  
  }  
}  
  
// Will execute only on this page:  
route.add('002.html', function(){  
  alert('Hello there!')  
});  
  
route.add('products.html', function(){  
  alert("this won't be executed :(")  
});
```

移动端(2)
杂文(2)

随笔档案

2016年12月 (40)
2016年11月 (37)
2016年10月 (14)

积分与排名

积分 - 15136
排名 - 15288

最新评论

1. Re:火车票购买技巧
电话购票你试过吗？我感觉花的时间多还没效果，还收电话费。

至于买区间票，上一站等的票，感觉逻辑是这样没错，但不舍的啊。你说多买一站吧，成功率不高，多买几站又太贵了。

--Jay54520

2. Re:jQuery中的100个技巧
48 \$.fn.placeholder 这个挺有意思的，但是我有一个问题就是像ie7这种浏览器 它默认就是不支持placeholder的，那不可能把页面上所有的input 都像这样\$('#te.....

--静逸

3. Re:不骄不躁，努力前行

```
// You can even use regex-es:
route.add('*.html', function(){
    alert('This is using a regex!')
});

route.run();

</script>
```



3.使用JavaScript中的AND技巧。

使用&&操作符的特点是如果操作符左边的表达式是false，那么它就不会再判断操作符右边的表达式了。所以：



```
// Instead of writing this:
if($('#elem').length){
    // do something
}

// You can write this:

$('#elem').length && log("doing something");
```



4. is()方法比你想象的更为强大。

下面举几个例子，我们先写一个id为elem的div。js代码如下：



```
// First, cache the element into a variable:
var elem = $('#elem');

// Is this a div?
elem.is('div') && log("it's a div");

// Does it have the bigbox class?
elem.is('.bigbox') && log("it has the bigbox class!");
```

楼主，每个人都会遇到什么当中的一些坎，时间会抹平一切困难。愿你未来的道路一切顺顺利利。加油
--静逸

4. Re:jQuery中的100个技巧
4 之前很小用 is方法确实很实用呀5
length是属性而非方法7 写的非常好 之前我都没有这样用过 学到了
原来可以这样简写8 外部链接 this.
hostname 是html上一个.....
--静逸

5. Re:jQuery中的100个技巧
\$("*").length () length是属性
不是方法

文章写的很赞 必须夸一个
--静逸

阅读排行榜

1. jQuery中的100个技巧(1662)
2. JavaScript函数之美~(839)
3. 深入理解JavaScript中的属性和特性(699)
4. 如何写出优雅的css代码 ? (679)
5. JavaScript Ajax之美~(576)

评论排行榜

1. jQuery中的100个技巧(21)
2. 深入理解css中的margin属性(13)
3. 如何写出优雅的css代码 ? (8)
4. 用三种方式实现导航菜单中的二级下拉菜单(5)

推荐排行榜

1. jQuery中的100个技巧(27)
2. 正则表达式之基本概念(11)
3. 如何写出优雅的css代码 ? (10)
4. JavaScript函数之美~(5)
5. 如何用iframe标签以及JavaScript制作时钟 ? (5)

```
// Is it visible? (we are hiding it in this example)
elem.is(':not(:visible)') && log("it is hidden!");
```

```
// Animating
elem.animate({'width':200},1);
```

```
// is it animated?
elem.is(':animated') && log("it is animated!");
```



其中判断是否为动画我觉得非常不错。

5.判断你的网页一共有多少元素。

通过使用\$("*").length属性可以判断网页的元素数量。

```
// How many elements does your page have?
log('This page has ' + $('*').length + ' elements!');
```

6.使用length()属性很笨重，下面我们使用exists()方法。



```
/ Old way
log($('#elem').length == 1 ? "exists!" : "doesn't exist!");
```

```
// Trickshot:
```

```
jQuery.fn.exists = function(){ return this.length > 0; }
```

```
log($('#elem').exists() ? "exists!" : "doesn't exist!");
```



7.jQuery方法\$()实际上是拥有两个参数的，你知道第二个参数的作用吗？



```

// Select an element. The second argument is context to limit the search
// You can use a selector, jQuery object or dom element

$('li', '#firstList').each(function() {
    log($(this).html());
});

log('-----');

// Create an element. The second argument is an
// object with jQuery methods to be called

var div = $('<div>', {
    "class": "bigBlue",
    "css": {
        "background-color": "purple"
    },
    "width" : 20,
    "height": 20,
    "animate" : { // You can use any jQuery method as a property!
        "width": 200,
        "height": 50
    }
});

div.appendTo('#result');

```



8.使用jQuery我们可以判断一个链接是否是外部的，并添加一个icon在非外部链接中，且确定打开方式。

这里用到了hostname属性。



```

<ul id="links">
    <li><a href="007.html">The previous tip</a></li>
    <li><a href="./009.html">The next tip</a></li>
    <li><a href="http://www.google.com/">Google</a></li>
</ul>

// Loop through all the links
$('#links a').each(function() {

```

```

if(this.hostname != location.hostname){
    // The link is external
    $(this).append('')
        .attr('target','_blank');
}

});

```



9.jQuery中的end()方法可以使你的jQuery链更加高效。

```

<ul id="meals"> <li> <ul class="breakfast"> <li class="eggs">No</li> <li class="toast">No</li> <li
class="juice">No</li> </ul> </li> </ul>
// Here is how it is used:

var breakfast = $('#meals .breakfast');

breakfast.find('.eggs').text('Yes')
    .end() // back to breakfast
    .find('.toast').text('Yes')
    .end()
    .find('.juice').toggleClass('juice coffee').text('Yes');

breakfast.find('li').each(function(){
    log(this.className + ': ' + this.textContent)
});

```



10.也许你希望你的web 应用感觉更像原生的，那么你可以阻止contextmenu默认事件。

```

<script>
    // Prevent right clicking on this page
    $(function(){
        $(document).on("contextmenu",function(e){
            e.preventDefault();

```



```
    });  
  });  
</script>
```



11.一些站点可能会使你的网页在一个bar下面，即我们所看到在下面的网页是iframe标签中的，我们可以这样解决。



```
// Here is how it is used:  
  
if(window != window.top){  
    window.top.location = window.location;  
}  
else{  
    alert('This page is not displayed in a frame. Open 011.html to see it in action.');}
```



12.你的内联样式表并不是被设置为不可改变的，如下：

```
// Make the stylesheet visible and editable  
$('#regular-style-block').css({'display':'block', 'white-space':'pre'})  
    .attr('contentEditable',true);
```

这样即可改变内联样式了。

13.有时候我们不希望网页的某一部分内容被选择比如复制粘贴这种事情，我们可以这么做：



```
<p class="descr">In certain situations you might want to prevent text on the page from being selectable.  
Try selecting this text and hit view source to see how it is done.</p>  
  
<script>  
    // Prevent text from being selected
```

```
$(function(){  
    $('p.descr').attr('unselectable', 'on')  
        .css('user-select', 'none')  
        .on('selectstart', false);  
});  
</script>
```



这样，内容就不能被选择啦。

14.从CDN中引入jQuery，这样的方法可以提高我们网站的性能，并且引入最新的版本也是一个不错的主意。

下面会介绍四种不同的方法。

```
<!-- Case 1 - requesting jQuery from the official CDN -->  
<script src="http://code.jquery.com/jquery-1.10.2.min.js"></script>  
  
<!-- Case 2 - requesting jQuery from Google's CDN (notice the protocol) -->  
<!-- <script src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script> -->  
  
<!-- Case 3 - requesting the latest minor 1.8.x version (only cached for an hour) -->  
<!-- <script src="//ajax.googleapis.com/ajax/libs/jquery/1.10/jquery.min.js"></script> -->  
  
<!-- Case 4 - requesting the absolute latest jQuery version (use with caution) -->  
<!-- <script src="http://code.jquery.com/jquery.min.js"></script> -->
```



15.保证最小的DOM操作。

我们知道js操作DOM是非常浪费资源的，我们可以看看下面的例子。



```
CODE  
// Bad  
//var elem = $('#elem');  
//for(var i = 0; i < 100; i++){
```



```
//     elem.append('<li>element '+i+'</li>');
//}

// Good
var elem = $('#elem'),
    arr = [];

for(var i = 0; i < 100; i++){
    arr.push('<li>element '+i+'</li>');
}

elem.append(arr.join(''));
```



16.更方便的分解URL。

也许你会使用正则表达式来解析URL，但这绝对不是一种好的方法，我们可以借用a标签来实现它。



```
// You want to parse this address into parts:
var url = 'http://tutorialzine.com/books/jquery-trickshots?trick=12#comments';

// The trickshot:
var a = $('<a>',{ href: url });

log('Host name: ' + a.prop('hostname'));
log('Path: ' + a.prop('pathname'));
log('Query: ' + a.prop('search'));
log('Protocol: ' + a.prop('protocol'));
log('Hash: ' + a.prop('hash'));
```



17.不要害怕使用vanilla.js。

jQuery背负的太多，这便是原因，你可以用一般的js。



```
// Print the IDs of all LI items
$('#colors li').each(function(){

    // Access the ID directly, instead
    // of using jQuery's $(this).attr('id')

    log(this.id);

});
```



18.最优化你的选择器



```
// Let's try some benchmarks!

var iterations = 10000, i;

timer('Fancy');

for(i=0; i < iterations; i++){
    // This falls back to a SLOW JavaScript dom traversal
    $('#peanutButter div:first');
}

timer_result('Fancy');

timer('Parent-child');

for(i=0; i < iterations; i++){
    // Better, but still slow
    $('#peanutButter div');
}

timer_result('Parent-child');

timer('Parent-child by class');
```

```
for(i=0; i < iterations; i++){  
    // Some browsers are a bit faster on this one  
    $('#peanutButter .jellyTime')
```



19.缓存你的selector。



```
// Bad:  
// $('#pancakes li').eq(0).remove();  
// $('#pancakes li').eq(1).remove();  
// $('#pancakes li').eq(2).remove();  
  
// Good:  
var pancakes = $('#pancakes li');  
  
pancakes.eq(0).remove();  
pancakes.eq(1).remove();  
pancakes.eq(2).remove();  
  
// Alternatively:  
// pancakes.eq(0).remove().end()  
//           .eq(1).remove().end()  
//           .eq(2).remove().end();
```



20.对于重复的函数只定义一次

如果你追求代码的更高性能，那么当你设置事件监听程序时必须小心，只定义一次函数然后把它的名字作为事件处理程序传递是不错的方法。



```
$(document).ready(function(){  
    function showMenu(){  
        alert('Showing menu!');
```

```
// Doing something complex here
}

$('#menuButton').click(showMenu);
$('#menuLink').click(showMenu);

});
```



21.像对待数组一样地对待jQuery对象

由于jQuery对象有index值和长度，所以这意味着我们可以把对象当作普通的数组对待。这样也会有更好地性能。



```
var arr = $('li'),
    iterations = 100000;

timer('Native Loop');

for(var z=0;z<iterations;z++){

    var length = arr.length;
    for(var i=0; i < length; i++){
        arr[i];
    }
}

timer_result('Native Loop');

timer('jQuery Each');

for(z=0;z<iterations;z++){

    arr.each(function(i, val) {
        this;
    });
}

timer_result('jQuery Each');
```



22. 当做复杂的修改时要分离元素。

修改一个dom元素要求网页重绘，这个代价是高昂的，所以如果你想要再提高性能，就可以尝试着当对一个元素进行大量修改时**先从页面中分离这个元素**，修改完之后再添加到页面。



```
// Modifying in place
var elem = $('#elem');

timer('In place');

for(i=0; i < iterations; i++){

    elem.width(Math.round(100*Math.random()));
    elem.height(Math.round(100*Math.random()));

}

timer_result('In place');

var parent = elem.parent();

// Detaching first
timer('Detached');

elem.detach();

for(i=0; i < iterations; i++){

    elem.width(Math.round(100*Math.random()));
    elem.height(Math.round(100*Math.random()));

}

elem.appendTo(parent);

timer_result('Detached');
```



23.不要一直等待load事件。

我们已经习惯了把我们所有的代码都放在ready的事件处理程序中，但是，如果你的html页面很庞大，document ready恐怕会被延迟了，所以对于一些我们不希望ready后才可以触发的事件可以放在html的head元素中。



```
<script>

// jQuery is loaded at this point. We can use
// event delegation right away to bind events
// even before $(document).ready:

$(document).on('click', '#clickMe', function(){
    alert('Hit view source and see how this is made');
});

$(document).ready(function(){

    // This is where you would usually bind event handlers,
    // but as we are using delegation, there is no need to.

    // $('#clickMe').click(function(){ alert('Hey!'); });
});

// Note: You should place your script tags at the bottom of the page.
// I have included them in the head only to demonstrate that we can bind
// events before document ready and before the elements are created.

</script>
```



24.当使用js给多个元素添加样式时更好的做法是创建一个style元素。

我们之前提到过，操作dom是非常慢的，所以当添加多个元素的样式时创建一个style元素并添加到document中是更好的做法。



```
<ul id="testList">
  <li>Item</li> <li>Item</li> <li>Item</li> <li>Item</li> <li>Item</li> <li>Item</li> <li>Item</li>
<li>Item</li> <li>Item</li> <li>Item</li> <li>Item</li> <li>Item</li> <li>Item</li> <li>Item</li>
<li>Item</li> <li>Item</li> <li>Item</li>
</ul>
```

```
var style = $('<style>');
```

```
// Try commenting out this line, or change the color:
```

```
style.text('#testList li{ color:red;}');
```

```
// Placing it before the result section so it affects the elements
```

```
style.prependTo('#result');
```



25.给html元素分配一个名为JS的class。

现代的web apps非常的依赖js，这里的一个技巧就是只有当js可用时才能显示特定的元素。看下面的代码。



```
$(document).ready(function() {
  $('#html').addClass('JS');
});
```

```
html.JS #message { display:block; }
#message {display:none;}
```



这样，只有js可用的时候id为message的元素才会显示；如果不支持js，则该元素不会显示。

26.监听不存在的元素上的事件。

jQuery拥有一个先进的事件处理机制，通过on()方法可以监听还不存在的事件。这是因为on方法可以传递一个元素的子元素选择器作为参数。看下面的例子：



```
<ul id="testList"> <li>Old</li> <li>Old</li> <li>Old</li> <li>Old</li> </ul>
```

```
var list = $('#testList');
```

```
// Binding an event on the list, but listening for events on the li items:
```

```
list.on('click','li',function(){
```

```
    $(this).remove();
```

```
});
```

```
// This allows us to create li elements at a later time,
```

```
// while keeping the functionality in the event listener
```

```
list.append('<li>New item (click me!)</li>');
```



这样，即使li是后创建的，也可以通过on()方法来监听。

27.只使用一次事件监听。

有时，我们只需要绑定只运行一次的事件处理程序。那么one()方法是一个不错的选择，通过它你就可以高枕无忧了。



```
<button id="press">Press me!</ul>
```

```
var press = $('#press');
```

```
// There is a method that does exactly that, the one():
```

```
press.one('click',function(){
```

```
    alert('This alert will pop up only once');
```

```
});
```

```
// What this method does, is call on() behind the scenes,
```

```
// with a 1 as the last argument:
```

```
// press.on('click',null,null,function(){alert('I am the one and only!');}, 1);
```



28.模拟触发事件。

我们可以通过使用trigger模拟触发一个click事件。



```
<button id="press">Press me!</ul>
var press = $('#press');

// Just a regular event listener:
press.on('click',function(e, how){
    how = how || '';
    alert('The buton was clicked ' + how + '!');
});

// Trigger the click event
press.trigger('click');

// Trigger it with an argument
press.trigger('click',['fast']);
```



29.使用触摸事件。

使用触摸事件和相关的鼠标事件并没有太多不同，但是你得有一个方便的移动设备来测试会更好。看下面这个例子。



```
// Define some variables
var ball = $('<div id="ball"></div>').appendTo('body'),
startPosition = {}, elementPosition = {};

// Listen for mouse and touch events
ball.on('mousedown touchstart',function(e){
    e.preventDefault();

    // Normalizing the touch event object
    e = (e.originalEvent.touches) ? e.originalEvent.touches[0] : e;
```

```

// Recording current positions
startPosition = {x: e.pageX, y: e.pageY};
elementPosition = {x: ball.offset().left, y: ball.offset().top};

// These event listeners will be removed later
ball.on('mousemove.rem touchmove.rem',function(e){
    e = (e.originalEvent.touches) ? e.originalEvent.touches[0] : e;

    ball.css({
        top:elementPosition.y + (e.pageY - startPosition.y),
        left: elementPosition.x + (e.pageX - startPosition.x),
    });

});

});

ball.on('mouseup touchend',function(){
    // Removing the heavy *move listeners
    ball.off('.rem');
});

```



30.更好地使用on()/off()方法。

在jQuery1.7版本时对事件处理进行了简化，看看下面的例子吧。



```

<div id="holder"> <button id="button1">1</button> <button id="button2">2</button> <button
id="button3">3</button> <button id="button4">4</button> <button id="clear" style="float:
right;">Clear</button> </div>

```

```

// Lets cache some selectors

```

```

var button1 = $('#button1'),
    button2 = $('#button2'),
    button3 = $('#button3'),
    button4 = $('#button4'),
    clear = $('#clear'),
    holder = $('#holder');

```

```
// Case 1: Direct event handling
button1.on('click',function(){
    log('Click');
});

// Case 2: Direct event handling of multiple events
button2.on('mouseenter mouseleave',function(){
    log('In/Out');
});

// Case 3: Data passing
button3.on('click', Math.round(Math.random()*20), function(e){

    // This will print the same number over and over again,
    // as the random number above is generated only once:
    log('Random number: ' + e.data);

});

// Case 4: Events with a namespace
button4.on('click.temp', function(e){
    log('Temp event!');
});

button2.on('click.temp', function(e){
    log('Temp event!');
});

// Case 5: Using event delegation
$('#holder').on('click', '#clear', function(){
    log.clear();
});

// Case 6: Passing an event map
var t; // timer

clear.on({

    'mousedown':function(){

        t = new Date();

    },
```

```

        'mouseup':function(){

            if(new Date() - t > 1000){

                // The button has been held pressed
                // for more than a second. Turn off
                // the temp events

                $('button').off('.temp');
                alert('The .temp events were cleared!');
            }

        }
    });

```



31.更快地阻止默认事件行为。

我们知道js中可以使用preventDefault()方法来阻止默认行为，但是jQuery对此提供了更简单的方法。如下：

```

<a href="http://google.com/" id="goToGoogle">Go To Google</a>

$('#goToGoogle').click(false);

```

32.使用event.result链接多个事件处理程序。

对一个元素绑定多个事件处理程序并不常见，而使用event.result更可以将多个事件处理程序联系起来。看下面的例子。



```

<button id="press">点击</button>
<script src="http://code.jquery.com/jquery-1.10.2.min.js"></script>
<script>

var press = $('#press');
press.on('click',function(){
    return 'Hip';
});

```

```
// The second event listener has access  
// to what was returned from the first
```

```
press.on('click',function(e){  
    console.log(e.result + ' Hop!');  
});  
</script>
```



这样，控制台会输出Hip Hop!

33.创建你自己习惯的事件。

你可以使用on()方法创建自己喜欢的事件名称，然后通过trigger来触发。举例如下：



```
<button id="button1">Jump</button> <button id="button2">Punch</button> <button id="button3">Click</button>  
<button id="clear" style="float: right;">Clear</button> <div id="eventDiv"></div>  
<script src="http://code.jquery.com/jquery-1.10.2.min.js"></script>
```

```
<script>
```

```
var button1 = $('#button1'),  
    button2 = $('#button2'),  
    button3 = $('#button3'),  
    clear = $('#clear'),  
    div = $('#eventDiv');  
  
div.on({  
    jump : function(){  
        alert('Jumped!');  
    },  
  
    punch : function(e,data){  
        alert('Punched '+data+'!');  
    }  
});
```

```

    },

    click : function(){
        alert('Simulated click!');
    }

});

button1.click(function(){
    div.trigger('jump');
});

button2.click(function(){
    // Pass data along with the event
    div.trigger('punch',['hard']);
});

button3.click(function(){
    div.trigger('click');
});

clear.click(function(){
    //some clear code
});

</script>

```



34.在下载文件旁显示文件大小。

你知道如何在下载一个文件的情况下通过发送一个ajax请求头得到一个文件的大小吗？使用jQuery就很容易。



```

<a href="001.html" class="fetchSize">First Trickshot</a> <br />
<a href="034.html" class="fetchSize">This Trickshot</a> <br />
<a href="assets/img/ball.png" class="fetchSize">Ball.png</a> <br />

// Loop all .fetchSize links

```

```

$('a.fetchSize').each(function(){

    // Issue an AJAX HEAD request for each one
    var link = this;

    $.ajax({
        type      : 'HEAD',
        url       : link.href,
        complete  : function(xhr){

            // Append the filesize to each
            $(link).append(' (' + humanize(xhr.getResponseHeader('Content-Length')) + ') ');

        }
    });

});

function humanize(size){
    var units = ['bytes', 'KB', 'MB', 'GB', 'TB', 'PB'];

    var ord = Math.floor( Math.log(size) / Math.log(1024) );
    ord = Math.min( Math.max(0, ord), units.length-1);

    var s = Math.round((size / Math.pow(1024, ord))*100)/100;
    return s + ' ' + units[ord];
}

```



注意：这个例子如何我们直接使用浏览器是没法得到的，必须使用本地的web服务器打开运行才可以。

35.使用延迟简化你的Ajax请求

延迟(deferreds)是一个强大的工具。jQuery对于每一个Ajax请求都会返回一个deferred对象。deferred.done()方法接受一个或多个参数，所有这些都参数可以是一个单一的函数或一个函数数组。当Deferred（延迟）解决时，doneCallbacks被调用。回调是依照他们添加的顺序执行。一旦deferred.done()返回Deferred（延迟）对象，Deferred（延迟）可以链接其它的延迟对象，包括增加额外的.done()方法。下面这样就会使你的代码更易读：



```
// This is equivalent to passing a callback as the
// second argument (executed on success):

$.get('assets/misc/1.json').done(function(r) {
    log(r.message);
});

// Requesting a file that does not exist. This will trigger
// the failure response. To handle it, you would normally have to
// use the full $.ajax method and pass it as a failure callback,
// but with deferreds you can simply use the fail method:

$.get('assets/misc/non-existing.json').fail(function(r) {
    log('Oops! The second ajax request was "' + r.statusText + '" (error ' + r.status + ')!');
});
```



36.平行的运行多个Ajax请求。

当我们需要发送多个Ajax请求是，相反于等待一个发送结束再发送下一个，我们可以平行地发送来加速Ajax请求发送。

```
// The trick is in the $.when() function:

$.when($.get('assets/misc/1.json'), $.get('assets/misc/2.json')).then(function(r1, r2) {
    log(r1[0].message + " " + r2[0].message);
});
```

37.通过jQuery获得ip

我们不仅可以在电脑上ping到一个网站的ip，也可以通过jQuery得到。

```
$.get('http://jsonip.com/', function(r) { log(r.ip); });

// For older browsers, which don't support CORS
// $.getJSON('http://jsonip.com/?callback=?', function(r) { log(r.ip); });
```


38.使用最简单的ajax请求

jQuery (使用ajax) 提供了一个速记的方法来快速下载内容并添加在一个元素中。



```
<p class="content"></p> <p class="content"></p>
```

```
var contentDivs = $('.content');
```

```
// Fetch the contents of a text file:
```

```
contentDivs.eq(0).load('1.txt');
```

```
// Fetch the contents of a HTML file, and display a specific element:
```

```
contentDivs.eq(1).load('1.html #header');
```



39.序列化对象

jQuery提供了一个方法序列化表单值和一般的对象成为URL编码文本字符串。这样，我们就可以把序列化的值传给ajax()作为url的参数，轻松使用ajax()提交表单了。



```
<form action="">
```

```
First name: <input type="text" name="FirstName" value="Bill" /><br />
```

```
Last name: <input type="text" name="LastName" value="Gates" /><br />
```

```
</form>
```

```
// Turn all form fields into a URL friendly key/value string.
```

```
// This can be passed as argument of AJAX requests, or URLs.
```

```
$(document).ready(function() {
```

```
    console.log($("#form").serialize()); // FirstName=Bill&LastName=Gates
```

```
});
```

```
// You can also encode your own objects with the $.param method:  
log($.param({'pet':'cat', 'name':'snowbell'}));
```



40.使用jQuery上传二进制文件

现在的浏览器都支持FormData API，这可以是我们很轻松的通过ajax来发送数据。 并将之结合HTML5中的File API，我们就可以上传二进制文件了。



```
// The file input field  
  
var fileInput = $('input[type=file]'),  
    button = $('#upload');  
  
button.on('click', function(){  
  
    // Access the files property, which holds  
    // an array with the selected files  
  
    var files = fileInput.prop('files');  
  
    // No file was chosen!  
    if(files.length == 0) {  
        alert('Please choose a file to upload!');  
        return false;  
    }  
  
    // Create a new FormData object  
  
    var fd = new FormData();  
  
    fd.append('file', files[0]);  
  
    // Upload the file to assets/php/upload.php. Open that file in a text  
    // editor to get a better idea of how it works.  
  
    $.ajax({  
        url: './assets/php/upload.php',  
        data: fd,  
        contentType:false,           // This will make the browser use the multipart/formdata encoding, which
```

```
is required for transferring binary data.
    processData:false,          // jQuery shouldn't do any processsing on the data - the browser will
                                handle this when it sees we are passing a formdata object.
    type:'POST',
    success: function(m) {
        log(m);
    }
});
});
```



41.使用Facebook的图表

我们可以引入facebook中的一个很强大的API来是我们的app更加社交化。下面是一个简单的例子：



CODE

```
// Fetch the publicly accessible data on Tutorialzine's Page
var api = 'http://graph.facebook.com/Tutorialzine/?callback=?',
    holder = $('#fbdata');

$.getJSON(api, function(r) {

    // This will always give the current picture
    holder.append('');

    holder.append('<p>'+ r.about + '</p>')
    holder.append('<a href="'+ r.website +' ">'+ r.website + '</a>');

});
```



42. 获取天气信息

Open Weather Map提供了免费的天气信息，我们可以通过使用它们的JSON API来获取数据。简单的例子如下：



```
// Request weather data:
var api = 'http://openweathermap.org/data/2.1/find/name?q=paris,france&callback=?';

$.getJSON(api, function(r){
  // This will always give the current picture
  log(r.list[0].name + ', ' + r.list[0].sys.country);
  log(r.list[0].main);

  // Temperatures are in kelvin, subtract 273.15 to convert to Celsius,
});
```



43. 获取你的最近的汤博乐(Tumblr)内容

现在非常流行的汤博乐博客服务提供了简单的方法使用JSON api，这样我们可以使用它来获取任何博客内容，下面是使用的方法。



```
<div id="post"></div>
// Define variables
var blog = 'minimaldesks.tumblr.com',
    api = 'http://' + blog + '/api/read/json?callback=?',
    post = $('#post');

$.getJSON(api, function(r){

  log('Blog title: ' + r.tumblelog.title);
  log('Description: ' + r.tumblelog.description);

  // If this post has a photo, show it
```

```

    if(r.posts[0]['photo-url-250']){
        post.append('');
    }
    else{
        log('Latest post: ' + r.posts[0]['regular-title']);
    }
}

});

```



44.通过IP地址获得地理位置

有很多在线服务可以告诉我们IP地址所在的城市和国家，下面我们先ping到百度的IP地址，然后获取其地理位置：



```

// Define variables
var ip = '119.75.218.70', // you can optionally put an ip address here
    api = 'http://freegeoip.net/json/' + ip + '?callback=?';

$.getJSON(api, function(r){

    console.log('How is the weather in ' + r.city + ', ' + r.country_name + '?');

}); //How is the weather in Beijing, China?

```



45.使用YQL来爬网站

YQL对JavaScript开发者来说有无限的API，下面的例子是我们如何使用它来获取并解析其他站点的HTML。



CODE

```

// Define variables

```

```

var query = 'select * from data.html.cssselect where url="http://www.chucknorrisfacts.com/chuck-norris-top-50-facts" and css=".field-content a"';
var yqlAPI = 'http://query.yahooapis.com/v1/public/yql?q=' + encodeURIComponent(query) + '&format=json&env=store%3A%2F%2Fdatatables.org%2Falltableswithkeys&callback=?';

$.getJSON(yqlAPI, function(r){

    log('Chuck Norris Facts:');

    $.each(r.query.results.results.a, function(){
        log('-----');
        log(this.content);
    });

});

```



46.使用全局的Ajax方法

我们可以通过ajax的全局方法来简化web app中处理的ajax请求。



CODE

```

// Create an indicator that would be shown whenever an AJAX request occurs:

var preloader = $('<div>', { 'class': 'preloader' }).appendTo('body');
var doc = $(document);

// Show the preloader whenever you are making an AJAX request:

doc.ajaxStart(function(){
    preloader.fadeIn();
});

// Hide it when the ajax request finishes

```

```
doc.ajaxComplete(function(){
    // Keep it visible for 0.8 seconds after the request completes
    preloader.delay(800).fadeOut();
});

// It will show automatically every time you do an AJAX request:

$.get('assets/misc/1.json');
```



47. 学会爱上console吧。

我们的浏览器给了我们一系列有用的方法使用来调试代码，找出bug，下面就是一个例子，打开console看看吧。



```
// The simple case. Use this instead of alert():
console.log('This is a console message!');

// It supports embedding of variables as well:
var a = 'morning', b = 'Miss';
console.log('Good %s %s! How are you feeling today?', a, b);

// Interactively browse the properties of a method (similar to console.log):
console.dir(window);

// Information message (in webkit it looks like console.log)
console.info('Everything is OK');

// Warning message
console.warn('Something may be wrong');

// Error message (will print a stack trace)
console.error('Oops. That was bad.');
```

```
// Counting things
for(var i = 0; i<20; i++){
    console.count('Counter Name');
```

```
}

// Starts a collapsable group of log messages
console.group("Preflight check");
console.info('Fuel is OK');
console.info('Temperature is normal');
console.error('Wings are missing');
console.groupEnd()

// Timing things
console.time('The million-dollar loop')

var dollars = 0;

for(var i=0;i<100000; i++){
    dollars+=10;
}

console.timeEnd('The million-dollar loop');

// Profiling code (it will show up in your console's Profile tab)
console.profile('My app performance');

var arr = [];
$.each([0,1,2,3,4,5,6,7,8,9],function(){
    arr.push(this+1);
});

console.profileEnd('My app performance');
```



48.把代码转化为插件以提高重用率。

如果有一些代码你总是在不同的项目之间复制粘贴，你就可以考虑着把它转化成一个插件了。下面的例子就是这样。




```
<input id="testInput" placeholder="YourName"/>
<script src="http://code.jquery.com/jquery-1.10.2.min.js"></script>

<script>
// Define the placeholder plugin

$.fn.placeholder = function(){

    if ('placeholder' in document.createElement('input')){

        // This browser already supports placeholders.
        // Nothing to do here.
        return this;
    }

    this.each(function(){
        var input = $(this);

        input.on('focus', function(){

            if(input.val() == input.attr('placeholder')){

                input.val('');
            }

        }).on('blur', function(){

            if(input.val() == ''){
                input.val(input.attr('placeholder'));
            }

        });

        // Show the placeholder on load
        input.trigger('blur');
    });

    return this;
};

// And here is how to use it:
$('#testInput').placeholder();
```



49.使用匿名函数来产生一个独立的代码块

定义全局变量和函数是一种代码很粗糙的行为，更好的方式是通过使用匿名函数使你的代码独立于块之中。看下面的例子：



```
// Isolating a block of code:

(function($) {

    // Declare a variable. It will only be visible in this block.
    var c = 1;

    // Define a simple plugin

    $.fn.count = function() {

        // Increment and log the counter
        log(c++);

        return this;
    };

})(jQuery);

// The c variable is only visible for the plugin and will keep
// its value between invocations:

$(document).count();

$('body').count().count();
```



50. 用extend融合对象

当提到从多个项目到一个项目结合属性时，你最好的猜测就是扩展方法。



```
// Combine properties (useful in plugins).
// The defaults are passed as the first argument.

var supplied = { height: 400 };

var options = $.extend({
  color    : 'blue',
  width    : 200,
  height   : 150
}, supplied);

log('New options:', options);

// You can also pass more than one object

log('Three parents:', $.extend({a:2}, {b:3}, {c:4}) );

log('-----');

// Cloning objects.
// To clone an object, simply pass an empty one
// as the first argument

var original = {a:123, b:'#fff'};
var clone = $.extend({}, original);

log('Clone:', clone);

log('-----');

// Extending jQuery.
// You can define plugins with extend

$.extend($.fn, {
  plugin1: function(){
    log('Plugin 1');
  }
});
```

```
        return this;
    },
    plugin2: function(){
        log('Plugin 2');
        return this;
    }
});

$('body').plugin1().plugin2();

log('-----');

// If you pass only one arguments to $.extend,
// it will add the properties to the jQuery object

$.extend({ dontDoThis : 123});

log($.dontDoThis);

log('-----');

// Deep cloning.
// If you have nested objects, you will have to
// pass one additional argument to extend:

var obj1 = { a: 1, b: 2, c: {d: 3} };
var obj2 = { c: {e: 4}, f:5};

// This won't work
// $.extend(obj1, obj2);

// This will
$.extend(true, obj1,obj2);

log('Deep clone:', obj1);
```



持续更新中...

注：原创文章，如需转载，请注明出处。博客地址：<http://www.cnblogs.com/zhuzhenwei918/p/6181760.html>

分类：[jQuery](#)

[好文要顶](#)[关注我](#)[收藏该文](#)



[918之初](#)
[关注 - 20](#)
[粉丝 - 56](#)
[+加关注](#)

27

 推荐

1

 反对

« 上一篇：[不骄不躁，努力前行](#)
» 下一篇：[真机调试之android手机+chrome](#)

posted @ 2016-12-15 01:13 918之初 阅读(1662) 评论(21) 编辑 收藏

评论列表

#1楼 2016-12-15 08:41 幻天芒 

实践出精华~

支持(0) 反对(0)

#2楼 2016-12-15 09:18 Sam Xiao 

用的熟练啊！一个类库都能总结出这么多经验，不得了！！

支持(1) 反对(0)

#3楼 2016-12-15 13:54 迪克牛逼 

厉害了，哥

支持(0) 反对(0)

#4楼 2016-12-15 13:55 聿若婉青 

```
clear.click(function(){  
log.clear();  
});
```


这个是什么意思啊
这块儿没看懂
大神帮解释一下呗

支持(1) 反对(0)

#5楼 2016-12-15 14:14 卷柏的花期 

厉害了，我的哥，JQ现在已经是烂大街的，用的人多，但是真正去细致使用的确没几个...

支持(0) 反对(0)

#6楼[楼主 

@ 聿若婉青

这里有些问题，function里可以写clear()来做清理工作，然后在外部声明这个函数。我稍后会改，谢谢提醒。

支持(0) 反对(0)

#7楼 2016-12-15 15:42 聿若婉青 

@ 918之初
好的 ok

支持(0) 反对(0)

#8楼 2016-12-15 18:32 hambert 

都是干货啊！大家快来

支持(1) 反对(0)

#9楼 2016-12-16 09:36 LGGGGG 

不错	支持(0) 反对(0)
#10楼 2016-12-16 09:51 反骨仔 (二五仔) 	
good	支持(0) 反对(0)
#11楼 2016-12-16 14:34 2604529 	
不错~	支持(0) 反对(0)
#12楼 2016-12-16 15:15 long 	
博主牛逼！恍然大悟！	支持(0) 反对(0)
#13楼 2016-12-16 16:26 潇十一郎 	
@ Sam Xiao 我只服你这个评论	支持(0) 反对(0)
#14楼 2016-12-16 21:11 小i屁孩学编程 	
写这么多，还是看不懂	支持(0) 反对(0)
#15楼 2016-12-19 13:48 奔走的光 	
看懂了一半，，看来还要不断积累啊。	支持(0) 反对(0)
#16楼 2016-12-19 17:45 反骨仔 (二五仔) 	

好多不懂，可能我只是个搞.NET

支持(0) 反对(0)

#17楼[楼主👤] 2016-12-19 17:49 918之初 📧

@ 反骨仔 (二五仔)

有一部分是不太容易懂... 这个很正常的 加油吧!

支持(0) 反对(0)

#18楼 2016-12-21 09:07 安心晨 📧

真的是你自己写的吗？表示新手基本看不懂，
听老师更容易让人一看就懂的才是好代码。
你这纯英文注释加上注释太少。
新手汗颜，我还得看看别的教程进阶了才能来看这个了。
西安收藏了。

支持(0) 反对(0)

#19楼 2016-12-24 21:55 静逸 📧

\$(".*").length () length是属性不是方法

文章写的很赞 必须夸一个


支持(0) 反对(0)

#20楼 2016-12-24 22:28 静逸 📧

4 之前很小用 is方法确实很实用呀
5 length是属性而非方法
7 写的非常好 之前我都没有这样用过 学到了 原来可以这样简写
8 外部链接 this.hostname 是html上一个原生的属性 跟jQuery无关
9 end方法可让链式更加方法 把this指向原先对象
12 contentEditable 可让任何可编辑 包括css js
13 也很实用
16 解析url的思路特别好 通过创建a标签 挺秒的
18 不知道你要表述的是什么意思？
23 思路也很清晰，但我想不到应用场景
28 之前用过trigger 但确实不知道原来trigger还可以传递参数进去的

30 没有看懂这个想表述的意思
31 这个挺妙的 估计没几个会这样使用的吧
32 e.result还可以这样


支持(0) 反对(0)

#21楼 2016-12-25 10:47 静逸 

48 \$.fn.placeholder 这个挺有意思的，但是我有一个问题就是像ie7这种浏览器 它默认就是不支持placeholder的，那不可能把页面上所有的input 都像这样\$('#testInput').placeholder(); 使用一番吧 也没有考虑textArea的情况？

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【福利】微软Azure给博客园的你专属双重好礼

【推荐】融云发布 App 社交化白皮书 IM 提升活跃超 8 倍

【推荐】BPM免费下载



最新IT新闻:

- 同是靠广告“掘金”，微信和Facebook相比差在哪？
- 程序员不想让你知道的5件事，你知道吗？
- 花式推销手表：苹果将要在1月推出健身挑战

- Crytek上海工作室再创业，想要打造基于CryEngine的引擎生态
- 90后爱开网店影响经济？董明珠的困惑也是经济的挑战
- » 更多新闻...



最新知识库文章：

- 写给未来的程序媛
- 高质量的工程代码为什么难写
- 循序渐进地代码重构
- 技术的正宗与野路子
- 陈皓：什么是工程师文化？
- » 更多知识库文章...



Copyright ©2016 918之初