本文会给你们展示50个jquery代码片段，这些代码能够给你的javascript项目提供帮助。其中的一些代码段是从jQuery1.4.2才开始支持的做法，另一些则是真正有用的函数或方法，他们能够帮助你又快又好地把事情完成。如果你发现你任何可以做得更好的地方的话，欢迎把你的版本粘贴在评论中!

1. 如何修改jQuery默认编码（例如默认UTF-8改成改GB2312）：

$.ajaxSetup({

    ajaxSettings:{ contentType:"application/x-www-form-urlencoded;chartset=GB2312"}

});

2. 解决jQuery, prototype共存，$全局变量冲突问题：

view source

print?

|  |  |
| --- | --- |
| 1 | <script src="http://upload.server110.com/image/20131221/0036233454-0.jpg"></script> |
| 2 | <script src="http://upload.server110.com/image/20131221/00362313b-1.jpg"></script> | |
| 3 | <script type="text/javascript"> jQuery.noConflict();</script> |  |

3. jQuery 判断元素上是否绑定了事件

//jQuery event封装支持判断元素上是否绑定了事件，此方法只适用于jQuery绑定的事件

var $events = $("#foo").data("events");if( $events && $events["click"] ){

//your code

}

4. 如何使用jQuery来切换样式表

//找出你希望切换的媒体类型（media-type），然后把href设置成新的样式表。

view source

print?

|  |  |
| --- | --- |
| 1 | $('link[media='screen']').attr('href','alternative.css'); |

5. 如何限制选择范围（基于优化目的）：

//尽可能使用标签名来作为类名的前缀，

//这样jQuery就不需要花费更多的时间来搜索

//你想要的元素。还要记住的一点是，

//针对于你的页面上的元素的操作越具体化，

//就越能降低执行和搜索的时间。

view source

print?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | varin\_stock = $('#shopping\_cart\_items input.is\_in\_stock'); | | | |
| 2 |  |  |  |  |
| 3 | <ul id="shopping\_cart\_items"> | | |  |
| 4 |  |  |  |  |
| 5 | <li><input type="radio"value="Item-X"name="item"/> Item X</li> 　　<li><input type="radio"value="Item-Y"name="item"class="3-5\_days"/> Item Y</li> 　　<li><input type="radio"value="Item-Z"name="item"/> Item Z</li> | | | | |
| 6 |  |  |  |  |  |
| 7 | </ul> | |  |  |  |

6. 如何正确地使用toggleClass：

//切换（toggle）类允许你根据某个类的 //是否存在来添加或是删除该类。 //这种情况下有些开发者使用： a.hasClass('blueButton') ? a.removeClass('blueButton') : a.addClass('blueButton');//toggleClass允许你使用下面的语句来很容易地做到这一点 a.toggleClass('blueButton');

7. 如何设置IE特有的功能：

if ($.browser.msie) {// Internet Explorer就是个虐待狂 }

8. 如何使用jQuery来代替一个元素：

$('#thatdiv').replaceWith('fnuh');

9. 如何验证某个元素是否为空：

// 方法一 if (! $('#keks').html()) {//什么都没有找到; }// 方法二 if ($('#keks').is(":empty")) {//什么都没有找到; }

10. 如何从一个未排序的集合中找出某个元素的索引号

$("ul > li").click(function () {　　var index = $(this).prevAll().length; //prevAll([expr]): 查找当前元素之前所有的同辈元素 });

11. 如何把函数绑定到事件上：

//方法一 $('#foo').click(function(event) {

　　alert('User clicked on "foo."');

}); //方法二, 支持动态传参 $('#foo').bind('click', {test1:"abc", test2:"123"}, function(event) {

　　alert('User clicked on "foo."' + event.data.test1 + event.data.test2 );

});

12. 如何追加或是添加html到元素中：

$('#lal').append('sometext');

13. 在创建元素时，如何使用对象字面量（literal）来定义属性

var e = $("", { href: "#", class: "a-class another-class", title: "..." });

14. 如何使用多个属性来进行过滤

//在使用许多相类似的有着不同类型的input元素时，  //这种基于精确度的方法很有用  var elements = $('#someid input[type=sometype][value=somevalue]').get();

15. 如何使用jQuery来预加载图像：

jQuery.preloadImages = function() { 　　for(var i = ; i < arguments.length; i++) {

　　　　$("<img />").attr('src', arguments[i]);

　　}

}; //用法  $.preloadImages('image1.gif', '/path/to/image2.png', 'some/image3.jpg');

16. 如何为任何与选择器相匹配的元素设置事件处理程序：

$('button.someClass').live('click', someFunction); //注意，在jQuery 1.4.2中，delegate和undelegate选项  //被引入代替live，因为它们提供了更好的上下文支持  //例如，就table来说，以前你会用 $("table").each(function(){

　　$("td", this).live("hover", function(){

　　　　$(this).toggleClass("hover");

　　});

}); //现在用  $("table").delegate("td", "hover", function(){

　　$(this).toggleClass("hover");

});

17. 如何找到一个已经被选中的option元素：

$('#someElement').find('option:selected');

18. 如何隐藏一个包含了某个值文本的元素：

$("p.value:contains('thetextvalue')").hide();

19. 如何创建嵌套的过滤器：

//允许你减少集合中的匹配元素的过滤器， //只剩下那些与给定的选择器匹配的部分。在这种情况下， //查询删除了任何没（:not）有（:has） //包含class为“selected”（.selected）的子节点。 .filter(":not(:has(.selected))")

20. 如何检测各种浏览器：

检测Safari (if( $.browser.safari)),    
检测IE6及之后版本 (if ($.browser.msie && $.browser.version > 6 )),    
检测IE6及之前版本 (if ($.browser.msie && $.browser.version <= 6 )),    
检测FireFox 2及之后版本 (if ($.browser.mozilla && $.browser.version >= '1.8' ))

21. 任何使用has()来检查某个元素是否包含某个类或是元素：

//jQuery 1.4.\*包含了对这一has方法的支持。 //该方法找出某个元素是否包含了其他另一个元素类或是其他任何的你正在查找并要在其之上进行操作的东东。 $("input").has(".email").addClass("email\_icon");

22. 如何禁用右键单击上下文菜单：

$(document).bind('contextmenu',function(e){

　　return false;

});

23. 如何定义一个定制的选择器

$.expr[':'].mycustomselector = function(element, index, meta, stack){ // element- 一个DOM元素  // index ? 栈中的当前循环索引  // meta ? 有关选择器的元数据  // stack ? 要循环的所有元素的栈  // 如果包含了当前元素就返回true  // 如果不包含当前元素就返回false };  // 定制选择器的用法：  $('.someClasses:test').doSomething();

24. 如何检查某个元素是否存在

if ($('#someDiv').length) { //万岁！！！它存在……  }

25. 如何使用jQuery来检测右键和左键的鼠标单击两种情况：

$("#someelement").live('click', function(e) {

    if( (!$.browser.msie && e.button == ) || ($.browser.msie && e.button == 1) ) {

        alert("Left Mouse Button Clicked");

    } else if(e.button == 2) {

        alert("Right Mouse Button Clicked");

    }

});

26. 如何替换串中的词

var el = $('#id');

el.html(el.html().replace(/word/ig, ''));

27. 如何在一段时间之后自动隐藏或关闭元素（支持1.4版本）：

//这是1.3.2中我们使用setTimeout来实现的方式  setTimeout(function() {

$('.mydiv').hide('blind', {}, 500)

}, 5000); //而这是在1.4中可以使用delay()这一功能来实现的方式（这很像是休眠）  $(".mydiv").delay(5000).hide('blind', {}, 500);

28. 如何把已创建的元素动态地添加到DOM中：

var newDiv = $('<div></div>');

newDiv.attr('id','myNewDiv').appendTo('body');

29. 如何限制“Text-Area”域中的字符的个数：

jQuery.fn.maxLength = function(max){

    return this.each(function(){

        var type = this.tagName.toLowerCase();

        var inputType = this.type? this.type.toLowerCase() : null;

        if(type == "input" && inputType == "text" || inputType == "password"){

            //Apply the standard maxLength              this.maxLength = max;

        } else if(type == "textarea"){

            this.onkeypress = function(e){

                var ob = e || event;

                var keyCode = ob.keyCode;

                var hasSelection = document.selection? document.selection.createRange().text.length >  : this.selectionStart != this.selectionEnd;

                return !(this.value.length >= max && (keyCode > 50 || keyCode == 32 || keyCode ==  || keyCode == 13) && !ob.ctrlKey && !ob.altKey && !hasSelection);

            };

            this.onkeyup = function(){

                if(this.value.length > max){

                    this.value = this.value.substring(,max);

                }

            };

        }

    });

};//用法  $('#mytextarea').maxLength(500);

30. 如何jQuery注册和禁用jQuery全局事件

//jQuery注册ajax全局事件ajaxStart，ajaxStop： $(document).ajaxStart(function(){

    $("#background,#progressBar").show();

}).ajaxStop(function(){

    $("#background,#progressBar").hide();

});//ajax请求禁用全局事件：$.ajax() 有个参数global (默认: true) 是否触发全局 AJAX 事件.设置为 false 将不会触发全局 AJAX 事件，如 ajaxStart 或 ajaxStop 可用于控制不同的 Ajax 事件。

31. 如何在jQuery中克隆一个元素：

var cloned = $('#somediv').clone();

32. 在jQuery中如何测试某个元素是否可见

if($(element).is(':visible')) { 　　//该元素是可见的  }

33. 如何把一个元素放在屏幕的中心位置：

jQuery.fn.center = function () {

　　return this.each(function(){

　　　　$(this).css({

　　　　　　position:'absolute',

　　　　　　top, ( $(window).height() - this.height() ) / 2 + $(window).scrollTop() + 'px',

　　　　　　left, ( $(window).width() - this.width() ) / 2 + $(window).scrollLeft() + 'px' 　　　　});

　　});

}//这样来使用上面的函数：   $(element).center();

34. 如何把有着某个特定名称的所有元素的值都放到一个数组中：

var arrInputValues = new Array();

$("input[name='xxx']").each(function(){

　　arrInputValues.push($(this).val());

});

35. 如何从元素中除去HTML

(function($) {

$.fn.stripHtml = function() {

　　var regexp = /<("[^"]\*"|'[^']\*'|[^'">])\*>/gi;

　　this.each(function() {

　　　　$(this).html( $(this).html().replace(regexp,'') );

　　});

　　return $(this);

}

})(jQuery); //用法：  $('p').stripHtml();

36. 如何使用closest来取得父元素：

$('#searchBox').closest('div');

37. 如何使用Firebug和Firefox来记录jQuery事件日志：

// 允许链式日志记录 jQuery.log = jQuery.fn.log = function (msg) { 　　if (console){

　　　　console.log("%s: %o", msg, this);

　　}　　return this;

};// 用法：  $('#someDiv').hide().log('div hidden').addClass('someClass');

38. 如何强制在弹出窗口中打开链接：

$('a.popup').live('click', function(){

　　var newwindow = window.open($(this).attr('href'),'','height=200,width=150');

　　if (window.focus) {

　　　　newwindow.focus();

　　}

　　return false;

});

39. 如何强制在新的选项卡中打开链接：

$('a.newTab').live('click', function(){

　　var newwindow=window.open(this.href);

　　$(this).target = "\_blank";

　　return false;

});

40. 在jQuery中如何使用.siblings()来选择同辈元素

// 不这样做  $('#nav li').click(function(){

　　$('#nav li').removeClass('active');

　　$(this).addClass('active');

});//替代做法是  $('#nav li').click(function(){

　　$(this).addClass('active').siblings().removeClass('active');

});

41. 如何切换页面上的所有复选框：

var tog = false; // 或者为true，如果它们在加载时为被选中状态的话  $('a').click(function() {

　　$("input[type=checkbox]").attr("checked",!tog);

　　tog = !tog;

});

42. 如何基于一些输入文本来过滤一个元素列表：

//如果元素的值和输入的文本相匹配的话,该元素将被返回  $('.someClass').filter(function() {

　　return $(this).attr('value') == $('input#someId').val();

})

43. 如何获得鼠标垫光标位置x和y

$(document).ready(function() {

　　$(document).mousemove(function(e){

　　　　$(’#XY’).html(”X Axis : ” + e.pageX + ” | Y Axis ” + e.pageY);

　　});

});

44. 如何扩展String对象的方法

$.extend(String.prototype, {

        isPositiveInteger:function(){

            return (new RegExp(/^[1-9]/d\*$/).test(this));

        },

        isInteger:function(){

            return (new RegExp(/^/d+$/).test(this));

        },

        isNumber: function(value, element) {

            return (new RegExp(/^-?(?:/d+|/d{1,3}(?:,/d{3})+)(?:/./d+)?$/).test(this));

        },

        trim:function(){

            return this.replace(/(^/s\*)|(/s\*$)|/r|/n/g, "");

        },

        trans:function() {

            return this.replace(/</g, '<').replace(/>/g,'>').replace(/"/g, '"');

        },

        replaceAll:function(os, ns) {

            return this.replace(new RegExp(os,"gm"),ns);

        },         skipChar:function(ch) {

            if (!this || this.length===) {return '';}

            if (this.charAt()===ch) {return this.substring(1).skipChar(ch);}

            return this;

        },

        isValidPwd:function() {

            return (new RegExp(/^([\_]|[a-zA-Z0-9]){6,32}$/).test(this));

        },

        isValidMail:function(){

            return(new RegExp(/^/w+((-/w+)|(/./w+))\*/@[A-Za-z0-9]+((/.|-)[A-Za-z0-9]+)\*/.[A-Za-z0-9]+$/).test(this.trim()));

        },

        isSpaces:function() {

            for(var i=; i<this.length; i+=1) {

                var ch = this.charAt(i);

                if (ch!=' '&& ch!="/n" && ch!="/t" && ch!="/r") {return false;}

            }

            return true;

        },

        isPhone:function() {

            return (new RegExp(/(^([0-9]{3,4}[-])?/d{3,8}(-/d{1,6})?$)|(^/([0-9]{3,4}/)/d{3,8}(/(/d{1,6}/))?$)|(^/d{3,8}$)/).test(this));

        },

        isUrl:function(){

            return (new RegExp(/^[a-zA-z]+:////([a-zA-Z0-9/-/.]+)([-/w .//?%&=:]\*)$/).test(this));

        },

        isExternalUrl:function(){

            return this.isUrl() && this.indexOf("://"+document.domain) == -1;

        }

    });

45. 如何规范化写jQuery插件：

(function($){

    $.fn.extend({

        pluginOne: function(){

            return this.each(function(){

                // my code             });

        },

        pluginTwo: function(){

            return this.each(function(){

                // my code             });

        }

    });

})(jQuery);

46. 如何检查图像是否已经被完全加载进来

$('#theImage').attr('src', 'image.jpg').load(function() {

　　alert('This Image Has Been Loaded');

});

47. 如何使用jQuery来为事件指定命名空间：

//事件可以这样绑定命名空间  $('input').bind('blur.validation', function(e){ 　　// ...  }); //data方法也接受命名空间  $('input').data('validation.isValid', true);

48. 如何检查cookie是否启用

var dt = new Date();

dt.setSeconds(dt.getSeconds() + 60);

document.cookie = "cookietest=1; expires=" + dt.toGMTString(); var cookiesEnabled = document.cookie.indexOf("cookietest=") != -1; if(!cookiesEnabled) {

　　//没有启用cookie  }

49. 如何让cookie过期：

var date = new Date();

date.setTime(date.getTime() + (x \* 60 \* 1000));

$.cookie('example', 'foo', { expires: date });

50. 如何使用一个可点击的链接来替换页面中任何的URL

$.fn.replaceUrl = function() { 　　var regexp = /((ftp|http|https):////(/w+:{0,1}/w\*@)?(/S+)(:[0-9]+)?(//|//([/w#!:.?+=&%@!/-//]))?)/gi; 　　return this.each(function() {

　　　　$(this).html(

　　　　　　$(this).html().replace(regexp,'<a href=#>

　　　　);

　　});

}

//用法

$('p').replaceUrl();

http://www.open-open.com/doc/view/09dbe7161d104be99fde38d1a5ac4b4f

http://www.open-open.com/doc/view/64ac60f3febb477cb2d5b8abce0b2f34

Hover(callBack,callback)就是鼠标经过时为：触发了callback函数，如果离开时，触发了callback的函数;

Toggle(fn,fn1)就是触发不同函数，并且循环触发函数;

$.map(array ,fn)数据映射;

$.extend(target,gron1,gron2)继承关系；

$.browser检验浏览器的类型;

$.earch(array, fn)循环array数组中的数据;

$(“p”).bind(even,fn)事件绑定的方法;

$.(“p”).unbind(even,fn);事件删除方法；

Trigger(eventType);在什么元素上触发什么事件；

$(“input”).val(“test”);什么元素设值；

$(“p”).text();获取元素中文本的内容；

$(“#div1,#div2”).val();同时获取元素的值；

setTineout(fn,时间);

Jquery.typeOf()==’object’类型判断的方法;

Has();表示匹配的元素存在；

Not()表示匹配的元素不存在；

hasClass()的作用；

Is检验数据类型是否一致，返回true或false;

juery.inArray(value,array);

Juery,unique();删除重复的元素，返回整理好的元素;

jquery.makeArray(obj)数组返回正真数组；

Reverse()就是数组的数据凡需排序;

jquery.IsFunction(obj)检验返回的参数是否function;

Jquery.browser.verssion获取浏览器的版本;

$(“p”).css(color,’red’);

表单元素过滤选择器：

$(“select option :selected”)

$(“:checked”)

$(“: disabled”);操作不匹配的元素；

$(“:enabled”) 操作匹配的元素；

------------------------------------------------------

$(“:input”),$(“:button”),$(“:submit”);

-------------------------------------------------------------

$(“div span:first-child”)

属性过滤选择器

$(“input[name==letter]”)

内容过滤选择器

 $(“div:hidden”),$(“tr:has(ui)”),$(“div:visable”),$(“div:empty”),

$(“div:parent”),$(“div:contrains(‘jianghu’)”)

获取值得方法为:

$(“tr:eq(1)”),$(“tr:gl(1)”),$(“tr:gt(1)”),$(“tr:even(1)”),$(“tr:odd”),$(“tr:first”)

$(“tr:last”),$(“tr,div”)

Parent(expr)与parents(expr)之间的区别

Parent(expr),获取获取所有匹配中唯一父元素的元素集合；

parents(expr)获取获取所有匹配中唯一祖先元素的元素集合；

Prev(expr)就是获取匹配集合中每一个元素之前相邻同辈元素的元素集合;

prevAll(expr)就是获取匹配集合中每一个元素之前所有同辈元素的元素集合;

Siblings(expr)就是获取匹配集合中每一个元素的所有同辈元素的元素集合；

Next(expr)就是获取匹配集合中每一个元素之后相邻同辈元素的元素集合;

nextAll(expr)就是获取匹配集合中每一个元素之后所有同辈元素的元素集合;

Andself(expr)就是前一个匹配元素集合添加到当前集合中；

Not(expr)就是元素集合中删除与指定表达式匹配的元素；

**1. jQuery 库中的 $() 是什么？（答案如下）**

$() 函数是 jQuery() 函数的别称，乍一看这很怪异，还使 jQuery 代码晦涩难懂。一旦你适应了，你会爱上它的简洁。$() 函数用于将任何对象包裹成 jQuery 对象，接着你就被允许调用定义在 jQuery 对象上的多个不同方法。你甚至可以将一个选择器字符串传入 $() 函数，它会返回一个包含所有匹配的 DOM 元素数组的 jQuery 对象。这个问题我已经见过好几次被提及，尽管它非常基础，它经常被用来区分一个开发人员是否了解 jQuery。

**2. 网页上有 5 个 <div> 元素，如何使用 jQuery来选择它们？（[答案](http://javarevisited.blogspot.sg/2013/07/jquery-selectors-examples-ID-Class-Descendent-Child-Multiple-Pseudo-Selector-find-element-DOM.html" \t "_blank)）**

另一个重要的 jQuery 问题是基于选择器的。jQuery 支持不同类型的选择器，例如 ID 选择器、class 选择器、标签选择器。鉴于这个问题没提到 ID 和 class，你可以用标签选择器来选择所有的 div 元素。jQuery 代码：$("div")，这样会返回一个包含所有 5 个 div 标签的 jQuery 对象。更详细的解答参见上面链接的文章。

**3. jQuery 里的 ID 选择器和 class 选择器有何不同？（[答案](http://javarevisited.blogspot.sg/2014/05/jquery-class-and-id-selector-example.html" \t "_blank)）**

如果你用过 CSS，你也许就知道 ID 选择器和 class 选择器之间的差异，jQuery 也同样如此。ID 选择器使用 ID 来选择元素，比如 #element1，而 class 选择器使用 CSS class 来选择元素。当你只需要选择一个元素时，使用 ID 选择器，而如果你想要选择一组具有相同 CSS class 的元素，就要用 class 选择器。在面试过程中，你有很大几率会被要求使用 ID 选择器和 class 选择器来写代码。下面的 jQuery 代码使用了 ID 选择器和 class 选择器：

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | $('#LoginTextBox')  // Returns element wrapped as jQuery object with id='LoginTextBox' | | |
| 2 | | $('.active') // Returns all elements with CSS class active. |

正如你所见，从语法角度来说，ID 选择器和 class 选择器的另一个不同之处是，前者用字符”#”而后者用字符”.”。更详细的分析和讨论参见上面的答案链接。

4. 如何在点击一个按钮时使用 jQuery 隐藏一个图片？

这是一个事件处理问题。jQuery为按钮点击之类的事件提供了很好的支持。你可以通过以下代码去隐藏一个通过ID或class定位到的图片。你需要知道如何为按钮设置事件并执行hide() 方法，代码如下所示：

|  |  |  |
| --- | --- | --- |
| 1 | $('#ButtonToClick').click(function(){ | |
| 2 | $('#ImageToHide').hide(); |

|  |  |
| --- | --- |
| 3 | }); |

我喜欢这个问题，因为很贴近实际使用，代码也不复杂。

5.  $(document).ready() 是个什么函数？为什么要用它？([**answer**](http://javarevisited.blogspot.sg/2014/11/difference-between-jquery-document-ready-vs-Javascript-window-onload-event.html))

这个问题很重要，并且常常被问到。 ready() 函数用于在文档进入ready状态时执行代码。当DOM 完全加载（例如HTML被完全解析DOM树构建完成时），jQuery允许你执行代码。使用$(document).ready()的最大好处在于它适用于所有浏览器，jQuery帮你解决了跨浏览器的难题。需要进一步了解的用户可以点击 answer链接查看详细讨论。

**6. JavaScript window.onload 事件和 jQuery ready 函数有何不同？（[答案](http://javarevisited.blogspot.sg/2014/11/difference-between-jquery-document-ready-vs-Javascript-window-onload-event.html" \t "_blank)）**

这个问答是紧接着上一个的。JavaScript window.onload 事件和 jQuery ready 函数之间的主要区别是，前者除了要等待 DOM 被创建还要等到包括大型图片、音频、视频在内的所有外部资源都完全加载。如果加载图片和媒体内容花费了大量时间，用户就会感受到定义在 window.onload 事件上的代码在执行时有明显的延迟。

另一方面，jQuery ready() 函数只需对 DOM 树的等待，而无需对图像或外部资源加载的等待，从而执行起来更快。使用 jQuery $(document).ready() 的另一个优势是你可以在网页里多次使用它，浏览器会按它们在 HTML 页面里出现的顺序执行它们，相反对于 onload 技术而言，只能在单一函数里使用。鉴于这个好处，用 jQuery ready() 函数比用 JavaScript window.onload 事件要更好些。

**7. 如何找到所有 HTML select 标签的选中项？（答案如下）**

这是面试里比较棘手的 jQuery 问题之一。这是个基础的问题，但是别期望每个 jQuery 初学者都知道它。你能用下面的 jQuery 选择器获取所有具备 multiple=true 的 <select > 标签的选中项：

|  |  |
| --- | --- |
| 1 | $('[name=NameOfSelectedTag] :selected') |

这段代码结合使用了属性选择器和 :selected 选择器，结果只返回被选中的选项。你可按需修改它，比如用 id 属性而不是 name 属性来获取 <select> 标签。

**8. jQuery 里的 each() 是什么函数？你是如何使用它的？（答案如下）**

each() 函数就像是 Java 里的一个 Iterator，它允许你遍历一个元素集合。你可以传一个函数给 each() 方法，被调用的 jQuery 对象会在其每个元素上执行传入的函数。有时这个问题会紧接着上面一个问题，举个例子，如何在 alert 框里显示所有选中项。我们可以用上面的选择器代码找出所有选中项，然后我们在 alert 框中用 each() 方法来一个个打印它们，代码如下：

|  |  |  |
| --- | --- | --- |
| 1 | $('[name=NameOfSelectedTag] :selected').each(function(selected) { | |
| 2 | alert($(selected).text()); |

|  |  |
| --- | --- |
| 3 | }); |

其中 text() 方法返回选项的文本。

**9. 你是如何将一个 HTML 元素添加到 DOM 树中的？（答案如下）**

你可以用 jQuery 方法 appendTo() 将一个 HTML 元素添加到 DOM 树中。这是 jQuery 提供的众多操控 DOM 的方法中的一个。你可以通过 appendTo() 方法在指定的 DOM 元素末尾添加一个现存的元素或者一个新的 HTML 元素。

**10. 你能用 jQuery 代码选择所有在段落内部的超链接吗？（答案略）**

这是另一个关于选择器的 jQuery 面试题。就像其他问题那样，只需一行 jQuery 代码就能搞定。你可以使用下面这个 jQuery 代码片段来选择所有嵌套在段落（<p>标签）内部的超链接（<a>标签）……

**11. $(this) 和 this 关键字在 jQuery 中有何不同？（答案如下）**

这对于很多 jQuery 初学者来说是一个棘手的问题，其实是个简单的问题。$(this) 返回一个 jQuery 对象，你可以对它调用多个 jQuery 方法，比如用 text() 获取文本，用val() 获取值等等。而 this 代表当前元素，它是 JavaScript 关键词中的一个，表示上下文中的当前 DOM 元素。你不能对它调用 jQuery 方法，直到它被 $() 函数包裹，例如 $(this)。

**12. 你如何使用jQuery来提取一个HTML 标记的属性 例如. 链接的href? (答案)**

attr() 方法被用来提取任意一个HTML元素的一个属性的值. 你首先需要利用jQuery选择及选取到所有的链接或者一个特定的链接，然后你可以应用attr()方法来获得他们的href属性的值。下面的代码会找到页面中所有的链接并返回href值：

|  |  |
| --- | --- |
| 1 | $('a').each(function(){ |
| 2 | alert($(this).attr('href')); | |

|  |  |
| --- | --- |
| 3 | }); |

**13. 你如何使用jQuery设置一个属性值? (答案)**

前面这个问题之后额外的一个后续问题是，attr()方法和jQuery中的其它方法一样，能力不止一样. 如果你在调用attr()的同时带上一个值 例如. attr(name, value), 这里name是属性的名称，value是属性的新值。

**14. jQuery中 detach() 和 remove() 方法的区别是什么? (答案)**

尽管 detach() 和 remove() 方法都被用来移除一个DOM元素, 两者之间的主要不同在于 detach() 会保持对过去被解除元素的跟踪, 因此它可以被取消解除, 而 remove() 方法则会保持过去被移除对象的引用. 你也还可以看看 用来向DOM中添加元素的 appendTo() 方法.

**15. 你如何利用jQuery来向一个元素中添加和移除CSS类? (答案)**

通过利用 addClass() 和 removeClass() 这两个 jQuery 方法。动态的改变元素的class属性可以很简单例如. 使用类“.active"来标记它们的未激活和激活状态，等等.

**16. 使用 CND 加载 jQuery 库的主要优势是什么 ? (**[**答案**](http://javarevisited.blogspot.sg/2013/08/3-cdn-url-to-load-jquery-into-webpage-google-microsoft.html)**)**

这是一个稍微高级点儿的jQuery问题。好吧，除了报错节省服务器带宽以及更快的下载速度这许多的好处之外, 最重要的是，如果浏览器已经从同一个CDN下载类相同的 jQuery 版本, 那么它就不会再去下载它一次. 因此今时今日，许多公共的网站都将jQuery用于用户交互和动画, 如果浏览器已经有了下载好的jQuery库，网站就能有非常好的展示机会。

**17.  jQuery.get() 和 jQuery.ajax() 方法之间的区别是什么?**

ajax() 方法更强大，更具可配置性, 让你可以指定等待多久，以及如何处理错误。get() 方法是一个只获取一些数据的专门化方法。

**18. jQuery 中的方法链是什么？使用方法链有什么好处？**

方法链是对一个方法返回的结果调用另一个方法，这使得代码简洁明了，同时由于只对 DOM 进行了一轮查找，性能方面更加出色。

**19. 你要是在一个 jQuery 事件处理程序里返回了 false 会怎样？**

这通常用于阻止事件向上冒泡。

**20. 哪种方式更高效：document.getElementbyId("myId") 还是 $("#myId")？**

第一种，因为它直接调用了 JavaScript 引擎。