**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）**

　　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）**

　　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：

http://www.diybloghome.com/zb_users/upload/2015/4/2015040357064177.jpg

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：

$('#ButtonToClick').click(function(){

    $('#ImageToHide').hide();

});

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)**

　　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。

**6. JavaScript window.onload 事件和 jQuery ready 函数有何不同？（answer）**

　　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：

$('[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：

$('[name=NameOfSelectedTag] :selected').each(function(selected) {

    alert($(selected).text());

});

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：

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

   alert($(this).attr('href'));

});

**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 CDN 加载 jQuery 库的主要优势是什么 ? (答案)**

　　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。

**jQuery 对象**

* jQuery 对象就是通过 jQuery 包装 DOM 对象后产生的对象。
* jQuery 对象是 jQuery 独有的。
* 只有 jQuery 对象才能使用 jQuery 的方法，在 jQuery 对象中无法使用 DOM 对象的任何方法，反之 DOM 对象也无法使用任何 jQuery 的方法。
* 约定：如果获取的是 jQuery 对象，那么要在变量前面加上 $
* jQuery 对象中封装了多个 DOM 对象，同时 jQuery 对象是类数组对象
* 数组与类数组对象的区别

1. 数组的类型是Array
2. 类数组对象的类型是 Object

**DOM 对象转 jQuery 对象**

* 使用 $（）将 DOM 对象包装起来，就可以转换成 jQuery 对象

var item = document.getElementsByTagName('ul')[0], // DOM对象

$item = $(item); // jQuery对象

**jQuery 对象转换为 DOM 对象**

jQuery 对象通过 jQuery 提供的 get（index）方法，得到对应的 DOM 对象

var $ul = $('ul'),

ul = $ul.get(0);

jQuery 对象是一个类数组对象，可以通过 [] 方式，得到对应的 DOM 对象。

**类数组对象**

类数组对象本质就是一个对象，只不过存储方式类似于数组的结构

* arguments 对象 ---- 接受函数实参的个数
* jQuery 对象 ---- 底层就是 dom 对象

**属性**

* length 属性（数组的长度 | 元素的个数）

**方法**

* get（index）：根据 index 放回对应的 dom 对象
* eq（index）：根据 index 返回对应的 jQuery 对象
* index（）：查找元素的索引值

**ready 和 onlaod 的区别**

**ready**

1. 具有简写方式
2. 在一个 HTML 页面中允许出现多个
3. 加载完 DOM 结构就执行
4. 执行速度快

**onload**

1. 没有简写方式
2. 在一个 HTML 页面中只能使用一个
3. 需要等页面所有资源加载完才执行
4. 执行速度比 ready 慢

**jQuery 动画**

基本隐藏、显示效果

* show（）/ hide（）

$('div').show(1000).hide(1000);

若是对同一个 jQuery 对象使用，可以采用链式操作。

滑动式动画效果

* slideDown（）/ slideUp（）

$('div').slideUp(1000).slideDown(1000);

淡入淡出

* fadeIn（）淡入
* fadeOut（）淡出

$('div').fadeIn(1000).fadeOut(1000);

并发和排队效果

* 并发效果：设置多个动画同时执行
* 排队效果：设置多个动画，按照先后顺序依次执行

**jQuery 插件**

**jQuery 插件的作用**

* 扩展 jQuery 的功能
* 呈现组件化特点

**日期插件 - layDate插件**

* layDate初步使用

1. 引入 laydate.js
2. laydate（options）

**开发插件**

**全局函数**

全局函数，实际上就是 jQuery 本身的方法。

jQuery 内置的一些功能是通过全局函数实现的。

* 比如$.ajax（）就是典型的全局函数

向 jQuery 命名空间添加一个函数，只需要将这个新函数指定为 jQuery 本身的一个属性

$.globalFunction = function(){

// todo...

};

可以通过 jQuery.globalFunction（）或者 $.globalFunction（）来调用

当需要添加多个函数可以使用$.extend（）函数

$.extend({

functionOne: function(){

// todo...

},

functionTwo: function(){

// todo...

}

});

通过上述代码可以添加全局函数，但是代码存在有关命名空间的风险

我们可以把属于一个插件的所有全局函数封装到一个对象

$.plugins = function(){

functionOne: function(){

// todo...

},

functionTwo: function(){

// todo...

}

};

通过上述代码，其实是为全局函数创建了另一个方法 --- plugins

functionOne 和 functionTwo 已经成为 plugins 对象的属性。

$.plugins.functionOne();

$.plugins.functionTwo();

**添加 jQuery 实例对象的方法**

$.fn.method = function(){};

**对象方法的环境**

在任何插件方法内部，关键字 this 引用的都是当前调用方法的 jQuery对象，因此可以在 this 上面调用任何内置的 jQuery 方法。

**方法连缀**  
通过 return this 来实现链式操作