

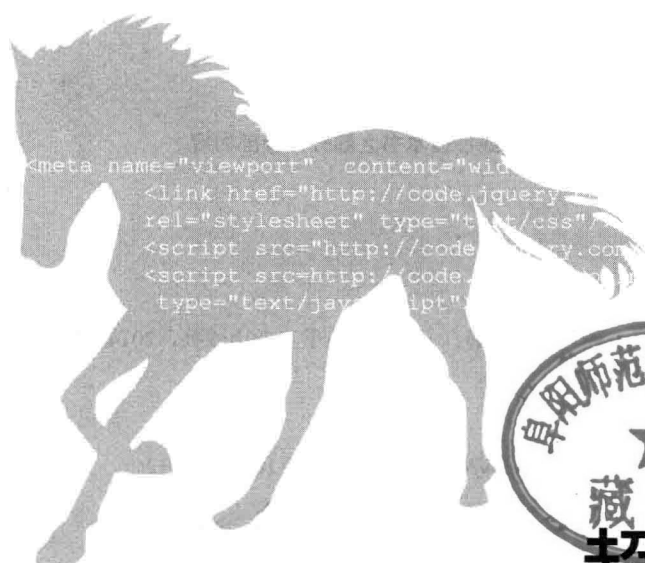
始于功能，终于代码，拿来即用
语法、规范、理念、工具、技巧，尽在代码中



超实用的 jQuery代码段

周敏 编著

将最有用的jQuery代码段汇总在一起，提供各种解决实际问题的方案，跨浏览器兼容、选择器的优化、动态修改CSS样式、流行特效、插件开发、Callback回调机制、jQuery Mobile移动端扩展……



超实用的 jQuery代码段

周敏 编著

电子工业出版社

Publishing House of Electronics Industry

内 容 简 介

本书精选近 350 个 jQuery 代码段, 涵盖页面开发中绝大多数要点、技巧与方法, 堪称史上最实用的 jQuery 代码参考书, 可以视为网页设计与网站建设人员的好帮手。本书的代码跨平台、跨设备、跨浏览器, 充分向读者演示了如何使用 jQuery 的各项技术, 实现令人激动的网页效果。

本书从 jQuery 框架的使用原理与应用场景出发, 对最实用的 jQuery 代码段进行了全方位的介绍和演示。全书分为 11 章, 包含网页效果、DOM 元素与属性、HTML 事件、CSS 样式、用户输入自动完成、拖放、图形图像、AJAX、算法、jQuery Mobile 应用、浏览器、Cookies 等 jQuery 技术内容, 对提高网站开发人员和设计人员的 jQuery 技术水平有指导作用。

本书内容简洁明了、代码精练、重点突出、实例丰富、原理清晰明白、语言通俗易懂, 是广大网页设计者的良好选择, 同时也非常适合大中专院校学生学习阅读。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

超实用的 jQuery 代码段 / 周敏编著. —北京: 电子工业出版社, 2014.9
(代码逆袭)
ISBN 978-7-121-23949-6

I. ①超… II. ①周… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2014) 第 173067 号

责任编辑: 董 英

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱

邮编: 100036

开 本: 787×1092 1/16

印张: 23.25

字数: 566 千字

版 次: 2014 年 9 月第 1 版

印 次: 2014 年 9 月第 1 次印刷

定 价: 59.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件到 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前言

jQuery 是网页设计中一柄锋利的宝剑，使用 jQuery 框架技术，可以快速实现页面上各种强大的效果。所以，要搭建网站和设计网页，就必须学会 jQuery 技术。本书是一本讲述代码实践的书，为读者全面深入地讲解了针对各种页面元素和页面特效的 jQuery 技术。近 350 段代码给读者带来的不仅仅是网页开发速度的提升，还有如何创建高性能网页、如何处理 jQuery 特效、如何应对跨浏览器兼容、如何进行前端代码优化等时时刻刻困扰网站开发人员的问题解决方案。

jQuery 中的那些事儿

有些网站开发人员写了多年 jQuery，依然在面对新问题时束手无策，这些技术难点具有普适性，读者能有多少了解呢？

- 浏览器的兼容
- 选择器的优化
- 事件注册和事件注销
- 动态修改 CSS 样式
- 文字、图片的流行特效
- 基于 jQuery 的插件开发
- Callback 回调机制
- AJAX 无刷新更新页面
- IE 老版本的兼容
- jQuery Mobile 移动端扩展

以上所有内容在本书的代码中都有讲解。除这些常见 jQuery 技术难点外，本书力求将最有用的 jQuery 代码段汇总在一起，提供各种解决实际问题的方案。

jQuery 学习方法

11 个字就能帮助我们更好地学习 jQuery。

- **多看、多练：**观摩成功的网站设计，分析并练习网站开发常用的代码。
- **多想、多问：**思考功能实现的原理，提出自己的问题并通过各种渠道寻找答案。
- **多总结：**记录前人已经探索出来的 jQuery 技巧，总结实战中碰到的问题及解决方案。

只要真正能做到勤思考、勤动手、勤总结，jQuery 学习定能一马平川。

本书的编写特点

1. 独特的 jQuery 切入点

与市面上其他 jQuery 书不同，本书从最常见的网页效果出发，直接应用 jQuery 代码段实现页面操作，全部是最实用的例子，全部是最透彻的分析！

2. 内容丰富，知识全面

本书以网站开发的各个分类和使用场景作为基础，立体式全方位地解释各种场景下的 jQuery 代码段应用，实例丰富，与时俱进，并拥有良好的可扩展性、可复用性。

3. 去中心化，分布式学习

本书的代码实例都是独立的，读者可以从中间开始学，也可以从头开始学。代码跨平台跨设备，可以在平板电脑上学，也可以在 PC 上学，如果可以，手机上写代码也是完全可能的。

4. jQuery 原理与实践相结合

jQuery 是一个强大的 JavaScript 库，无论读者具有什么编程背景，都可以使用它来优化改进自己的网站。本书的代码段基于简单易学的原理提供了丰富多样的操作特性，使得本书成为适用于各类客户端脚本编程的必备工具。

5. 必用、实用

本书的实例从网页元素、HTML 事件、CSS 样式类、输入自动完成、图片加载、AJAX 应用、jQuery Mobile 移动端开发等方方面面内容出发，配合原理解释，呈现给读者一场代码的盛宴。

6. 自发式学习

在学习代码前，先让读者练习实际上最基础却最容易做错的 jQuery 考题和面试题，激发读者学习斗志。本书摒弃传统的说教模式，每段代码都是单独的功能型页面，读者可以从全书的任意一点开始阅读。

本书的设计始于功能、终于代码，是网站开发人员的案头必备。

本书的内容安排

本书共 11 章，各章节实现了不同类别的 jQuery 代码段。

第 1 章主要介绍了使用 jQuery 实现网页操作，包括 jQuery DOM 操作方法与案例、jQuery 动画效果的实现、jQuery 控件下 iframe 控件的使用案例、jQuery 实现各种文本特效、jQuery 控制页面布局的方法与 jQuery 与 Flash 组合案例等内容。

第 2 章主要介绍了使用 jQuery 操作 HTML 中 DOM 元素与属性的方法，包括使用 jQuery 创建、插入、删除 DOM 节点、使用 jQuery 对 DOM 节点进行属性操作与使用 jQuery 对

DOM 节点进行样式操作等内容。

第3章主要介绍了使用 jQuery 操作 HTML 事件的方法,包括 jQuery 事件触发器、jQuery 绑定与反绑定事件监听器、jQuery 事件的交互处理、jQuery 的内置事件类型与 jQuery 页面载入完毕响应事件等内容。

第4章主要介绍了使用 jQuery 操作 CSS 样式类的方法,包括添加样式类、移除样式类、切换样式类、判断样式类、使用 `css()` 方法直接修改样式类等内容。

第5章主要介绍了使用 jQuery 实现用户输入自动完成的方法,包括使用远程数据源的自动完成、滚动条和组合框风格的自动完成、使用 XML 数据源的自动完成和使用多维数据的自动完成等内容。

第6章主要介绍了使用 jQuery 实现拖放功能的方法,包括控制拖放的范围、自动滚动的拖放效果、可拖放的排序列表框、在多个列表框之间拖放、可拖动改变列宽的表格、拖动选择表格的多行数据等内容。

第7章主要介绍了使用 jQuery 操作图形图像的方法,包括处理图片无法显示、解析 JSON 文件、显示 Flickr 与 Google Picasaweb 图片、动态调整图片的大小、自适应背景图像、随机与按顺序的淡入淡出图片等内容。

第8章主要介绍了使用 jQuery AJAX 的方法,包括加载网页并传递参数、无刷新删除网页记录、无刷新验证用户名和密码、无刷新提交表单、无刷新上传图片等内容。

第9章主要介绍了 jQuery 的常用算法,包括 jQuery 的选择器算法、在 jQuery 中对列表排序、查找或移除数组中特定的值、随机选择页面元素、序列化表单到 JSON 数据等内容。

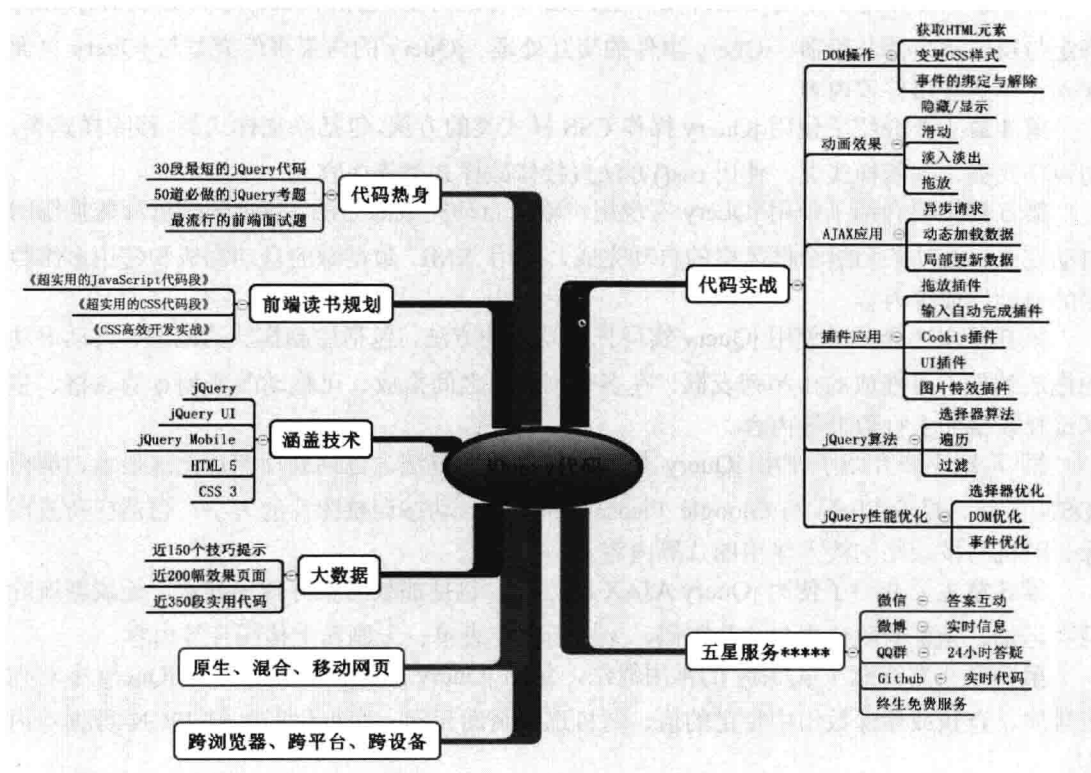
第10章主要介绍了 jQuery Mobile 移动开发技巧,包括 jQuery Mobile 的多页面显示视图、jQuery Mobile 的转场效果、不同导航栏的应用、可折叠列表框的应用、动态加载和切换页面、动态添加列表框的菜单等内容。

第11章主要介绍了一些无法分类的常用 jQuery 代码段,包括检查 jQuery 库是否加载、解决 jQuery 的版本冲突、使用 CDN 加载 jQuery 库、jQuery 的 Cookies 管理、jQuery 选择器和 DOM 操作的性能优化等内容。

本书面对的读者

- 网页设计入门者
- 网站开发入门者
- jQuery 学习爱好者
- 由 JavaScript 向 jQuery 转型的开发人员
- 中小型企业网站开发者
- 大中专院校的学生
- 各种 IT 培训学校的学生
- 网站后台开发人员 (Java、PHP、ASP.NET)
- 网站建设与网页设计的相关威客兼职人员

本书的思维导图



编者推荐

很多人都说 jQuery 很简单，看完文档就会用了！但是“会用”只是学会了 jQuery 的基本语法，你不知道什么情况下用哪段代码最优，你学会的也只限于 jQuery，而网站开发技术是混合型的，一段代码背后有 JavaScript 技巧、浏览器原理、网页数据的交换方法、HTML 的优化布局。本书就是一本最接地气的 jQuery 代码书，你学会的不仅仅是语法，还有 JSON、XML、序列化、AJAX、闭包、Callback、链式操作、插件等贯穿在网站开发中的所有概念和技术。本书近 350 段代码都可以拿来即用，这些代码是最好的、最高效的、最实用的 jQuery 代码段。这是一本市场上绝无仅有的 jQuery 代码书，是一本值得拥有的 jQuery 实战书。

本书的服务

笔者能力有限，如果写作过程中有什么疏漏，或者读者有什么疑问，可通过以下方式与我们沟通。

- QQ 群：296811675，作者在线答疑。
- 扫描封底的微信二维码，时刻参与我们的图书互动和本书的考题答案。
- @博文视点 Broadview 的微博，了解我们发布的信息和各种前端流行技术。

- 博文视点官方网站 <http://www.broadview.com.cn/>，下载本书所有实例源代码。
- Github, <https://github.com/kingwjz/jquery-code-segment>，了解代码的实时更新和迭代过程，可以在每章代码下参与讨论，也可以观看其他读者提出的问题，还可以随时随地下载代码。

很多读者在学习过程中苦于无法交流，小故障无法及时解决，加入我们的服务方阵，我们将为您提供终身免费的服务。

本书主要由周敏编写，王金柱审查并统一修订。参与编写的人员还有席新亮、赵荣娇、任建智、李勇、王铁民、张兴瑜、马新原、薛淑英、殷龙、谢郁、于健、周洋、王金柱、李兰英。

序 1 jQuery 必知必会

jQuery 是一个兼容多浏览器的 JavaScript 框架，核心理念是“write less, do more”（写得更少、做得更多）。jQuery 框架由美国人 John Resig 于 2006 年 1 月在纽约 BarCamp 发布的，一发布就吸引了来自世界各地的众多 JavaScript 高手加入，目前由 Dave Methvin 率领的团队进行开发。如今，jQuery 已经成为最流行的 JavaScript 框架，在世界前 10000 个访问量最多的网站中，有超过 55% 的网站是使用 jQuery 开发的。

本书不是一本普通的 jQuery 书，它涉及脚本编程的技巧、脚本性能的优化、快速开发的原理。本书不仅仅是你的帮助文档，更是你的良师益友，拥有这些代码，你将会在前端工作中战无不胜！

jQuery 如何加载

在项目中加载 jQuery 的方式有很多，废话不多说，先来介绍 4 种加载方式。

（1）页面加载完之前执行，与嵌入的 JS 加载方式一样：

```
(function($){  
    alert('start');  
})(jQuery);
```

（2）页面加载后执行：

```
$(document).ready(function(){  
    alert('jquery ready');  
});
```

（3）页面加载后执行：

```
$(function(){  
    alert('jquery onload');  
});
```

（4）页面加载后执行：

```
jQuery(function($){  
    alert('jQuery ready');  
});
```

以上这几种加载方式相较于传统的 window.onload 方式来讲，只需要加载所有的 DOM 结构，在浏览器把所有的 HTML 元素（加载外部图片和资源）放入 DOM 树之前就可以执行 JS 效果。

最短最实用的 30 段 jQuery 代码

使用 jQuery 框架的最大优势就在于，它可以使用很短的代码实现很强大的功能。下面将会给读者展示 30 个最短的 jQuery 代码段，这些代码都是真正有实用价值的代码，其中有一些代码段是从 jQuery-1.4.2 版本才开始支持的做法，这点需要读者特别注意！

(1) 修改 jQuery 默认编码（例如默认 UTF-8 改为 GB2312）：

```
$.ajaxSetup({
  ajaxSettings:{contentType:"application/x-www-form-urlencoded;charset=GB2312"}
});
```

(2) 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
});
```

(3) 检测各种浏览器：

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

(4) 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();
});
```

(5) 使用 .siblings() 来选择同辈元素：

```
$("#nav li").click(function(){
  $(this).addClass('active').siblings().removeClass('active');
});
```

(6) 选中页面上的所有复选框：

```
var tog=false;
$('a').click(function(){
  $("input[type=checkbox]").attr("checked",!tog);
  tog=!tog;
});
```

(7) 解决 jQuery 与 prototype 共存时，\$全局变量冲突问题：

```
<script src="prototype.js"></script>
<script src="jquery.js"></script>
<script type="text/javascript">
  jQuery.noConflict();
</script>
```

注意：一定要先引入 prototype.js，再引入 jquery.js。

(8) 设置 IE 特有的功能：

```
if($.browser.msie){
    // IE 下要做的工作
}
```

(9) 使用 jQuery 来切换样式表：

```
// 找出希望切换的媒体类型 (media-type)，然后把 href 设置成新的样式表
$('link[media='screen']").attr('href','alternative.css');
```

(10) 在创建元素时，使用对象字面量 (literal) 来定义属性：

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

(11) 查找已经被选中的 option 元素：

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

(12) 判断元素是否绑定了事件：

```
// jQuery event 判断元素上是否绑定了事件，此方法只适用于用 jQuery 绑定的事件
var $events=$('#foo').data("events");
if($events&&$events["click"]){
    //你自己的代码
}
```

(13) 禁用右键单击上下文菜单：

```
$(document).bind('contextmenu',function(e){
    return false;
});
```

(14) 验证某个元素是否为空：

```
if(!$('#keks').html()){
    //什么都没有找到
}
```

(15) 使用 jQuery 来代替一个元素：

```
$('#id-div').replaceWith('id-replace');
```

(16) 隐藏一个包含了某个文本值的元素：

```
$('#p.value:contains(id-text-value)').hide();
```

(17) 检查某个元素是否存在：

```
if($('#someDiv').length){
    // 元素存在……
}
```

(18) 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');
```

(19) 在 jQuery 中克隆一个元素：

```
var id-cloned=$('#somediv').clone();
```

(20) 使用 closest 来取得父元素：

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

(21) 判断某个元素是否为空:

```
if($('#keks').is(":empty")){
    //什么都没有找到，这里用的 empty，比较第 14 段代码。
}
```

(22) 替换字符串中的词:

```
var el=$('#id');
el.html(el.html().replace(/word/ig,""));
```

(23) 添加 html 到元素中:

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

(24) 判断图像是否已经被完全加载:

```
$('#id-Image').attr('src','image.jpg').load(function(){
    alert('This Image Has Been Loaded');
});
```

(25) 基于一些输入文本来过滤一个元素列表:

```
$('.someClass').filter(function(){
    return $(this).attr('value')==$('#input#someId').val();
})
```

(26) 测试某个元素是否可见:

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

(27) 把函数绑定到事件上:

```
$('#foo').click(function(event){
    alert('User clicked on "foo."');
});
```

(28) 让 cookie 过期:

```
var date=new Date();
date.setTime(date.getTime()+(x*60*1000));
$.cookie('example','foo',{expires:date});
```

(29) 使用多个属性来进行过滤:

```
var elements=$('#someid input[type=sometype][value=somevalue]').get();
```

(30) 创建嵌套的过滤器:

```
.filter(":not(:has(.selected))")
```

本书浏览器约定

本书涉及的浏览器测试基准如图 1 所示，可以看到内核和外壳的占有情况，内环为内核，外环为外壳，这也是接下来要提到的浏览器同类项的数据基础。

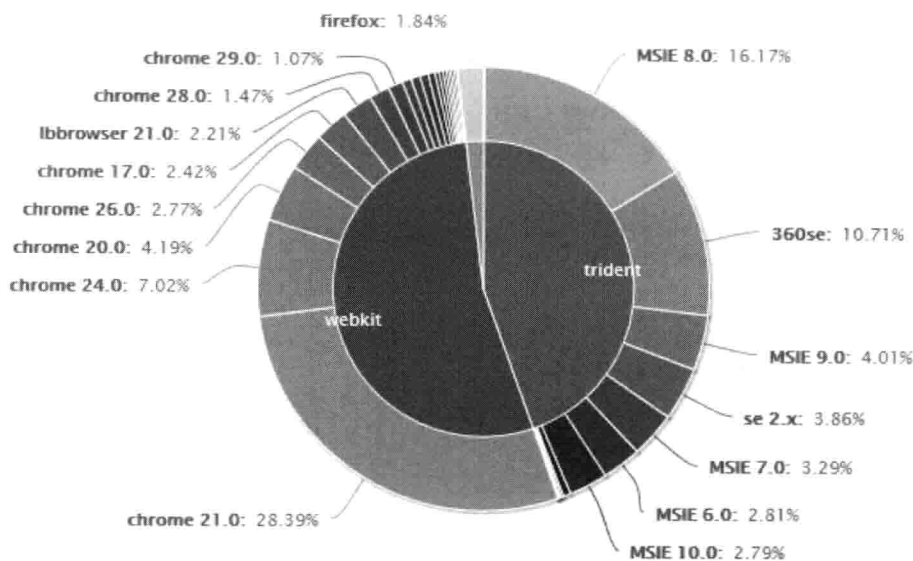


图1 浏览器基准

不同的浏览器可能会采用相同的内核（渲染引擎）。一方面，外壳实现的差异性会影响整个网站功能，这也是为什么内核无法完全代表所有浏览器的原因。另一方面，某些浏览器具有相同的血统，是可以归为同一类的，同一类中，如果其中一个测试通过，那么其他与之同类的浏览器也基本上可以通过测试。图1根据相同的内核或对W3C标准有类似的支持程度将浏览器进行归类。之所以这样做，是因为测试基准除了要基本反映浏览器的市场占有率，还要考虑到开发测试成本。因此，测试基准中的浏览器应当具有典型代表性。

浏览器的不同版本的内核也不一样，因此通常要针对不同版本的浏览器做测试，开发者要了解内核对标准的支持，比如IE的渲染引擎Trident的不同版本差别较大，因此IE需要测试IE 6~10的所有版本，然而由于Firefox和Chrome升级覆盖面广，因此基准中只保留其最新版本。

本书涉及的代码会根据需要，对不同浏览器进行有针对性的测试。

序 2 你绝对不可能全部 做对的 jQuery 题



不用找百度，试试你是否能及格？如果不用百度，你全做对了，相信本书不是你的菜！
如果你的正确率不到 58%，每天坚持看本书，会带来意想不到的技术飞跃！

1. 下面哪种不是 jQuery 的选择器？（单选）
 - A. 基本选择器
 - B. 后代选择器
 - C. 类选择器
 - D. 进一步选择器
2. 绑定一个 DOM 加载完成后要执行的函数，下面哪个是正确的？（单选）
 - A. `jQuery(expression,[context])`
 - B. `jQuery(html,[ownerDocument])`
 - C. `jQuery(callback)`
 - D. `jQuery(elements)`
3. 下面哪一个是追加内容到指定元素的末尾？（单选）
 - A. `insertAfter()`
 - B. `append()`
 - C. `appendTo()`
 - D. `after()`
4. 下面哪一个不是 jQuery 对象可访问的函数？（单选）
 - A. `each(callback)`
 - B. `size()`
 - C. `index(subject)`
 - D. `index()`
5. jQuery 访问对象中的 `size()` 函数的返回值和 jQuery 对象的_____属性一样。
6. jQuery 中 `$(this).get(0)` 的写法和_____是等价的。
7. 想要找到表单中的 hidden 元素，下面哪个是正确的？（单选）
 - A. `visible`
 - B. `hidden`

- C. visible()
 - D. hidden()
8. 如果需要匹配包含文本的元素，用下面哪种函数来实现？（单选）
- A. text()
 - B. contains()
 - C. input()
 - D. attr(name)
9. 现有一个表格，如果想要匹配表格的偶数行，用_____实现，奇数行用_____实现。
10. 如果想要找到一个表格的指定行数的元素，可以使用下面哪个函数？（单选）
- A. text()
 - B. get()
 - C. eq()
 - D. contents()
11. 在一个表单里，想要找到指定元素的第一个元素用_____实现，第二个元素用_____实现。
12. 下面哪种不属于 jQuery 的筛选？（单选）
- A. 过滤
 - B. 自动
 - C. 查找
 - D. 串联
13. 下面哪几种属于 jQuery 文档处理？（多选）
- A. 包裹
 - B. 替换
 - C. 删除
 - D. 内部和外部插入
14. 如果想在指定的元素前添加内容，下面哪个可以实现？（单选）
- A. append(content)
 - B. appendTo(content)
 - C. insertAfter(content)
 - D. after(content)
15. 在 jQuery 中，用一个表达式来检查当前选择的元素集合，使用_____来实现，如果这个表达式失效，则返回_____值。
16. 在 jQuery 中，如果想从 DOM 中删除所有匹配的元素，下面哪一个正确？（单选）
- A. delete()
 - B. empty()
 - C. remove()
 - D. removeAll()

17. 在 jQuery 中, 想要给第一个指定的元素添加样式, 下面哪一个正确? (单选)
- A. first
 - B. eq(1)
 - C. css(name)
 - D. css(name,value)
18. 在编写页面时, 如果想要获取指定元素在当前窗口的相对偏移, 用_____来实现, 该方法的返回值有两个属性, 分别是_____和_____。
19. 在 jQuery 中, 如果想要获取当前窗口的宽度值, 下面哪个可以实现? (单选)
- A. width()
 - B. width(val)
 - C. width
 - D. innerWidth()
20. 在一个表单中, 如果将所有的 div 元素都设置为绿色, 实现代码是_____。
21. 为每个元素的指定事件 (如 click) 绑定一个事件处理函数, 下面哪个可以实现? (单选)
- A. trigger(type)
 - B. bind(type)
 - C. one(type)
 - D. bind
22. 在 jQuery 中, 鼠标移动到一个指定的元素上时会触发指定的方法, 实现该操作的是_____。
23. 下面哪几个不属于 jQuery 的事件处理? (多选)
- A. bind(type)
 - B. click()
 - C. change()
 - D. one(type)
24. 在一个表单中, 如果想给输入框添加一个输入验证, 哪个事件可以实现? (单选)
- A. hover(over,out)
 - B. keypress(fn)
 - C. change()
 - D. change(fn)
25. 当一个文本框中的内容被选中时, 想要执行指定的方法, 使用哪个事件? (单选)
- A. click(fn)
 - B. change(fn)
 - C. select(fn)
 - D. bind(fn)
26. 在 jQuery 中, 想让一个元素隐藏, 用_____实现, 显示隐藏的元素用_____实现。

27. 在一个表单中, 用 600 毫秒缓慢地将段落滑上, 用_____来实现。
28. 如果想要自定义一个动画, 用_____函数来实现。
29. 想要实现通过远程 http-get 请求载入信息功能的是哪个事件? (单选)
- A. \$.ajax()
 - B. load(url)
 - C. \$.get(url)
 - D. \$.getScript(url)
30. 下面不属于 AJAX 事件的是哪个? (单选)
- A. ajaxComplete(callback)
 - B. ajaxSuccess(callback)
 - C. \$.post(url)
 - D. ajaxSend(callback)
31. 彻底将 jQuery 变量还原, 可以使用_____函数实现。
32. 在一个表单中, 查找所有选中的 input 元素, 可以用 jQuery 中的_____来实现。
33. 下面哪几种是 jQuery 中表单的对象属性? (多选)
- A. :checked
 - B. :enabled
 - C. :hidden
 - D. :selected
 - E. :file
 - F. :disabled
34. 在 jQuery 中如果将一个“名/值”形式的对象设置为所有指定元素的属性, 可以用_____实现。
35. 在 jQuery 中指定一个类, 如果存在就执行删除功能, 如果不存在就执行添加功能, 下面哪一个可以直接完成该功能? (单选)
- A. removeClass()
 - B. deleteClass()
 - C. toggleClass(class)
 - D. addClass()
36. 在 jQuery 中想要找到所有元素的同辈元素, 下面哪一个可以实现? (单选)
- A. eq(index)
 - B. find(expr)
 - C. siblings([expr])
 - D. next()
37. 请选择表达式[typeof null, null instanceof Object]的运算结果。 (单选)
- A. ["object", false]
 - B. [null, false]