

jQuery概述

前方高能：本章偏重于记忆的内容可能会比较多，请保持专注。

一、jQuery介绍

1.1 概念

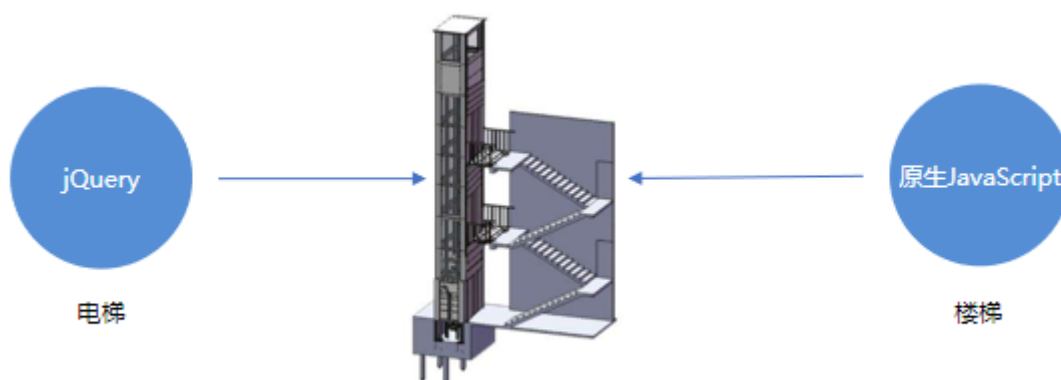
jQuery 在 2006 年 1 月由美国人 John Resig 在纽约发布，吸引了来自世界各地的众多 JavaScript 高手加入，由 Dave Methvin 率领团队进行开发。如今，jQuery已经成为最流行的 javascript 框架，在世界前 10000 个访问最多的网站中，有超过 55%在使用 jQuery。



jQuery是一个开源的轻量级JavaScript代码库，它封装了JavaScript常用的功能，提供一种简便的JavaScript设计模式，优化DOM操作、事件处理、动画设计和Ajax交互。

j 就是 JavaScript； Query 查询； 意思就是查询js，把js中的DOM操作做了封装，我们可以快速的查询使用里面的功能。

简单理解：jQuery就是一个JS 文件，里面对我们原生js代码进行了面向对象封装，这样我们可以快速高效的使用这些封装好的功能了。



1.2 宗旨

jQuery设计的宗旨是“write Less, Do More”，即倡导写更少的代码，做更多的事情。

1.3 思想

jQuery的思想是通过选择器查找对应jQuery节点对象，通过调用jQuery对象的属性或方法来实现对节点的操作

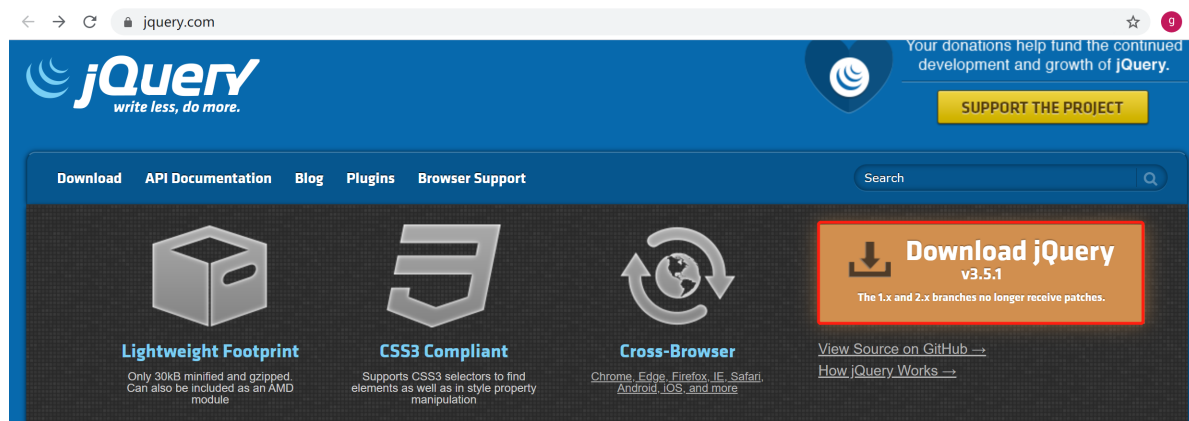
1.4 优势

- 轻量级。核心文件才几十kb，不会影响页面加载速度
- 出色的浏览器兼容性，基本兼容了现在所有主流的浏览器
- 对事件、样式、动画支持，大大简化了DOM操作
- 完善的Ajax支持
- 链式操作方式
- 隐式迭代
- 丰富的插件支持
- 免费、开源

二、jQuery使用

2.1 下载

jQuery的官网：<https://jquery.com/>，即可下载最新版本。



所有版本的下载：<https://code.jquery.com/>

- 版本介绍：

1x：兼容 IE 678 等低版本浏览器，官网不再更新

2x：不兼容 IE 678 等低版本浏览器，官网不再更新

3x：不兼容 IE 678 等低版本浏览器，是官方主要更新维护的版本

- jquery.js（开发版）完整无压缩版本，主要用于测试、学习和开发
- jquery.min.js（生产版）经过压缩的版本，主要应用于产品和项目

2.2 引入

在HTML页面中引入jQuery文件

```
<script src="js/jquery-3.3.1.min.js" type="text/javascript"></script>
```

2.3 Hello,world

```
$("#h1").html("Hello world!");
```

- \$符号是jQuery的一个简写形式，是jQuery中的顶级对象
- \$()就是jQuery中的函数，它的功能是获得()中指定的标签元素
- \$()方法等价于jQuery()方法

2.4 入口函数

```
// 第一种：
$(document).ready(function(){
    ... // 此处是页面DOM加载完成的入口
});

// 第二种：
$(function () {
    ... // 此处是页面 DOM 加载完成的入口
});
```

总结：

1. 等着 DOM 结构渲染完毕即可执行内部代码，不必等到所有外部资源加载完成，可重复使用多次
2. 相当于原生 js 中的 DOMContentLoaded，jQuery 帮我们完成了封装。
3. 原生 js 中的 onload 事件需要等到所有图片、css、图片等下载完才能执行，且只能执行一次。

```
$(document).ready(function() {
    $("#div").css("color", "red");
})

$(function() {
    $("#ul").css("color", "blue");
})
```

2.5 DOM对象与jQuery对象

使用 jQuery 方法和原生JS获取的元素是不一样的，总结如下：

1. 用原生JS 获取来的对象就是 DOM 对象,可以使用DOM下的属性和方法
2. jQuery 方法获取的元素就是 jQuery 对象,可以使用jQuery中的属性和方法
3. jQuery对象和DOM对象的属性和方法不可以混合使用
4. jQuery对象和DOM对象可以相互转换

```
/**
 *1.DOM对象转换成jQuery对象，方法只有一种：
 *对于一个DOM对象， 只需要用$( )把DOM对象包装起来，就可以获得一个jQuery对象了
 *jQuery对象=$(DOM对象)
 */
var box = document.getElementById('box'); //DOM对象
var jqueryObject = $(box); //jQuery对象
```

```
/**
 *2. jQuery对象转为DOM对象, 有两种方法:
 */
// 方法一: jQuery对象[索引值]
var domObject1 = $('div')[0];
// 方法二: jQuery对象.get(索引值)
var domObject2 = $('div').get(0);
```

总结: 实际开发比较常用的是把DOM对象转换为jQuery对象, 这样能够调用功能更加强大的jQuery中的方法。

2.5 jQuery对象元素的个数

```
$("div").length;
```

2.6 jQuery对象元素的索引

```
$("div").index();
```

2.7 jQuery对象元素的遍历

```
$("div").each(function(index, element){
    // index表示元素索引
    // element表示元素对象 (和this相同)
});
```

三、jQuery选择器

3.1 基础选择器

名称	用法	描述
id选择器	<code>\$("#id")</code>	获取指定id的元素
类选择器	<code>\$(".class")</code>	获取类选择器对应所有元素
标签选择器	<code>\$(div)</code>	获取标签对应所有元素
通配选择器	<code>\$(*)</code>	获取通配的所有元素
并集选择器	<code>\$(div,h1,p)</code>	获取选择器匹配到的所有元素

3.2 层级选择器

名称	用法	描述
后代选择器	<code>\$(div p)</code>	获取div下所有的后代元素中的p元素
子选择器	<code>\$(div>p)</code>	获取div下所有子元素中的p元素(亲儿子)
下一兄弟选择器	<code>\$(div+p)</code>	获取紧邻div元素的下一个p元素
后面所有兄弟选择器	<code>\$(div~p)</code>	获取div元素后的所有p元素

3.3 基本过滤选择器

语法	用法	描述
:first	\$(<code>"li:first"</code>)	获取第一个li元素
:last	\$(<code>"li:last"</code>)	获取最后一个li元素
:eq(index)	\$(<code>"li:eq(2)"</code>)	获取对应索引的li元素,索引从0开始
:odd	\$(<code>"li:odd"</code>)	获取索引号为奇数的元素
:even	\$(<code>"li:even"</code>)	获取索引号为偶数的元素
:not(selector)	\$(<code>"li:not(:first)"</code>)	获取除了选择器匹配到的元素之外的所有元素

3.4 子元素过滤选择器

语法	用法	描述
:first-child	\$(<code>"li:first-child"</code>)	获取第一个li元素
:last-child	\$(<code>"li:last-child"</code>)	获取最后一个li元素
:nth-child(index)	\$(<code>"li:nth-child(1)"</code>)	获取对应索引的li元素,索引从1开始
:first-of-type	\$(<code>"li:first-of-type"</code>)	获取第一个li元素
:last-of-type	\$(<code>"li:last-of-type"</code>)	获取最后一个li元素
:nth-of-type(index)	\$(<code>"li:nth-of-type(1)"</code>)	获取对应索引的li元素,索引从1开始

3.5 属性过滤选择器

名称	用法	描述
[attribute]	\$(<code>"div[class]"</code>)	获取包含给定属性的元素
[attribute=value]	\$(<code>"div[class='red']"</code>)	获取给定的属性是某个特定值的元素
[attribute!=value]	\$(<code>"div[class!='red']"</code>)	获取属性不等于特定值的元素
[attribute^=value]	\$(<code>"div[class^='red']"</code>)	获取给定的属性是以某些值开始的元素
[attribute\$=value]	\$(<code>"div[class\$='red']"</code>)	获取给定的属性是以某些值结尾的元素
[attribute*=value]	\$(<code>"div[class*='red']"</code>)	获取给定的属性是以包含某些值的元素

3.6 表单过滤选择器

语法	用法	描述
:input	\$(":input")	获取所有 input, textarea, select 和 button 元素
:text	\$(":text")	获取所有的单行文本框
:password	\$(":password")	获取所有密码框
:radio	\$(":radio")	获取所有单选按钮
:checkbox	\$(":checkbox")	获取所有复选框
:submit	\$(":submit")	获取所有提交按钮
:reset	\$(":reset")	获取所有重置按钮
:button	\$(":button")	获取所有按钮
:file	\$(":file")	获取所有文件域
:enabled	\$(":enabled")	获取所有可用元素
:disabled	\$(":disabled")	获取所有不可用元素
:checked	\$(":checked")	获取所有被选中的(复选框、单选框)元素
:selected	\$(":selected")	获取所有select下拉菜单选中的元素

3.7 可见性过滤选择器

语法	用法	描述
:hidden	\$(":hidden")	获取所有不可见元素，或者type为hidden的input元素
:visible	\$(":visible")	获取所有的可见元素

3.8 内容过滤选择器

语法	用法	描述
:contains(text)	\$("div:contains('Rose')")	获取包含给定文本的元素
:has	\$("div:has(p)")	获取含有选择器所匹配的子元素的元素
:empty	\$("div:empty")	获取所有不包含子元素或者文本的空元素
:parent	\$("div:parent")	获取含有子元素或者文本的元素

四、jQuery对象筛选

语法	用法	描述
children([expr])	<code>\$("div").children('p')</code>	获取所有匹配到的子元素（ 亲儿子 ）
find([expr])	<code>\$("div").find('p')</code>	获取所有匹配到的后代元素
next([expr])	<code>\$("#li3").next("li")</code>	获取紧邻的下一个匹配到的同辈元素
nextall([expr])	<code>\$("#li3").nextAll()</code>	获取当前元素之后所有匹配到的同辈元素
prev([expr])	<code>\$("#li3").prev()</code>	获取紧邻的上一个匹配到同辈元素
prevall([expr])	<code>\$("#li3").prevAll()</code>	获取当前元素之前所有匹配到的同辈元素
siblings([expr])	<code>\$("#li3").siblings()</code>	获取所有匹配到同辈元素
parent([expr])	<code>\$("#li3").parent()</code>	获取匹配到的父元素
closest([expr])	<code>\$("#li3").closest('div')</code>	获取匹配到的最近的祖先元素
offsetParent()	<code>\$("#li3").offsetParent()</code>	获取第一个已经定位的祖先元素，没有直接为html元素
parents([expr])	<code>\$("#li3").parents()</code>	获取匹配到的祖先元素集合（不包含document根元素）
hasClass(class)	<code>console.log(\$("#li3").hasClass('red'))</code>	检查当前的元素是否含有某个特定的类选择器，如果有，则返回true
eq(index)	<code>\$("li").eq(-1)</code>	获取指定位置元素,这个元素的位置是从0算起，可以是负值
slice(start, [end])	<code>\$("li").slice(-5, -2)</code>	获取一个匹配的子集，不包含结束位置的元素 start: 开始选取子集的位置。第一个元素是0.如果是负数，则可以从集合的尾部开始选起。

五、案例：淘宝精品选项卡



HTML部分:

```
<script src="jquery-3.5.1.min.js"></script>
<div class="box">
  <ul class="tab">
    <li><a href="#">男靴</a></li>
    <li><a href="#">连衣裙</a></li>
    <li><a href="#">毛衣</a></li>
    <li><a href="#">男棉服</a></li>
    <li><a href="#">女包</a></li>
    <li><a href="#">男毛衣</a></li>
    <li><a href="#">男棉服</a></li>
    <li><a href="#">牛仔裤</a></li>
    <li><a href="#">铅笔裤</a></li>
  </ul>
  <div class="content">
    
    
    
    
    
    
    
    
    
  </div>
</div>
```

CSS部分:

```
* {
  margin: 0;
```



```
padding: 0;
box-sizing: border-box;
font-size: 12px;
}

ul {
  list-style: none;
}

.box {
  position: absolute;
  left: 50%;
  top: 50%;
  transform: translate(-50%, -50%);
  width: 252px;
  height: 252px;
  border: 1px solid pink;
}

.tab {
  float: left;
  width: 50px;
}

.tab li {
  width: 50px;
  height: 28px;
  border-bottom: 1px solid pink;
  text-align: center;
  line-height: 28px;
  background: linear-gradient(#fff 20%, pink 80%, #fff);
}

.tab li:hover {
  background: linear-gradient(#fff 20%, rgb(235, 22, 57) 80%, #fff);
}

.tab li:last-of-type {
  height: 26px;
  border-bottom: 0;
}

.tab li a {
  text-decoration: none;
  color: black;
}

.content {
  position: relative;
  float: right;
  width: 200px;
}

.content img {
  position: absolute;
  top: 0;
  left: 0;
}
```

```
.content img:first-of-type {  
    z-index: 1;  
}
```

javaScript部分:

```
$(function() {  
    $(".tab li").mouseover(function() {  
        // 得到li的索引  
        var index = $(this).index();  
        // 链式编程  
        $(".content img").eq(index).fadeIn().siblings().fadeOut();  
    })  
})
```

作业:

- 一、谈谈对jQuery的理解
- 二、如何使用jQuery
- 三、jQuery对象和DOM对象的相互转换
- 四、jQuery对象的遍历
- 五、举例说明jQuery选择器的用法
 1. 基础选择器
 2. 层级选择器
 3. 基本过滤选择器
 4. 属性过滤选择器
 5. 表单过滤选择器
 6. 可见性过滤选择器
 7. 内容过滤选择器
- 六、举例说明jQuery对象常用的筛选方法
- 七、使用jQuery实现隔行换色
- 八、使用jQuery实现淘宝精品选项卡