[July16 jQuery 1](#_Toc519708257)

[1 什么是JavaQuery 1](#_Toc519708258)

[1.1 jQuery的版本 1](#_Toc519708259)

[2 使用jQuery 1](#_Toc519708260)

[2.1 引入jQuery文件 1](#_Toc519708261)

[3 jQuery重点 1](#_Toc519708262)

[3.1 什么是jQuery对象 1](#_Toc519708263)

[3.2 工厂函数 1](#_Toc519708264)

[3.3 DOM对象和jQuery对象之间的转换 2](#_Toc519708265)

[4 jQuery基本选择器 3](#_Toc519708266)

[5 jQuery层级选择器 5](#_Toc519708267)

[5.1 后代选择器 5](#_Toc519708268)

[6 过滤选择器 6](#_Toc519708269)

[6.1 基本选择器 6](#_Toc519708270)

[6.2 属性选择器 6](#_Toc519708271)

[7 子元素过滤选择器 6](#_Toc519708272)

[8 jQuery操作DOM 7](#_Toc519708273)

[8.1 基本操作 7](#_Toc519708274)

[8.2 属性操作 8](#_Toc519708275)

[8.3 样式操作 8](#_Toc519708276)

jqJuly16 jQuery

1 什么是JavaQuery

jQuery是一个轻量级的JS库，是一个被封装好的JS文件，提供了更为简便的元素操作方式

核心理念：Write Less Do More

1.1 jQuery的版本

jQuery 2.x 不再兼容IE6\7\8

jQuery1.x jQuery 1.11.x

2 使用jQuery

2.1 引入jQuery文件

<script src=’jquery-1.11.3.js’></script>

注意：引入文件的操作必须要放在其他的jquery操作之前

3 jQuery重点

3.1 什么是jQuery对象

jQuery对象是由jQuery对页面元素封装后的一种体现

jQuery中所提供的所有操作只针对jQuery对象，其他对象（DOM）不能使用

3.2 工厂函数

function $(id){

return document.getElementById(id);

}

作用：能够将页面元素加工成jQuery对象

能够将DOM对象转换成jQuery对象

3.3 DOM对象和jQuery对象之间的转换

DOM对象：不能用jQuery提供的操作

jQuery对象：不能使用DOM提供的操作

1.将DOM对象转换为jQuery对象

语法：

var 变量 = $(DOM对象);

注意：

所有的jQuery对象在起名的时候，最好在变量名称前+$,

var d1 = document.getElementById(‘d1’);

var $d1 = $(d1);

d1:是DOM对象

$d1:是jQuery对象

2.将jQuery对象转换为DOM对象

1. var dom对象 = jQuery对象[下标];

var $d1 = $(‘#d1’);

var d1 = $d1[0];

var $redColor = $(‘.redColor’);

var r1 = $redColor[1];

2.var dom对象 = jQuery对象.get([下标])



4 jQuery基本选择器

|  |  |  |
| --- | --- | --- |
| 选择器 | 语法 | 返回值 |
| id选择器 | $(‘#id’) | 封装了包含指定id值的jquery数组 |
| 类选择器 | $(‘.className’) | 页面中指定className的所有指定元素 |
| 元素选择器 | $(‘element’) | 返回页面中指定的所有元素 |
| 群组/符号选择器 | $(‘select1,select2,…’) | 满足函数内所有选择器的元素们 |
| 通用选择器 | $(‘#’) | 返回页面中所有的元素 |

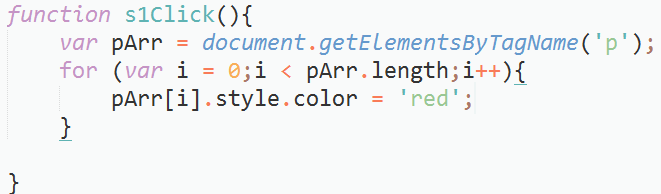
获取页面上的元素们，返回值都是由JQ对象所组成的数组

语法：

$(“选择器”);



===============================原生JavaScript的方法==================





5 jQuery层级选择器

5.1 后代选择器

|  |  |  |
| --- | --- | --- |
| 选择器种类 | 语法 | 作用 |
| 后代选择器 | $(‘selector1 selector2’) |  |
| 子代选择器 | $(‘selector1>selector2’) |  |
| 相邻兄弟选择器 | $(‘selector1 + selector2’) | 匹配跟在selector1后面且满足selector2选择器的元素(后面一个) |
| 通用兄弟选择器 | $(‘selector1~selector2’) | 匹配selector1后面所有满足selector2的元素 |



6 过滤选择器

6.1 基本选择器

|  |  |  |
| --- | --- | --- |
|  |  |  |
| :first | 只能匹配到第一个元素 | $(‘p:first’); |
| :last | 只能匹配到最后一个元素 |  |
| :not | 将满足selector选择器的元素排除在外 | $(‘p:not(:first)’); |
| :odd | 匹配第偶数个元素（奇数下标） |  |
| :even | 匹配第奇数个元素（偶数下标） |  |
| :eq(index) | 匹配下标等于index的元素 |  |
| :gt(index) | 匹配下标大于index的元素 |  |
| :lt(index) | 匹配下标小于index的元素 |  |



6.2 属性选择器

通过元素的属性及其值来匹配页面的元素

|  |  |  |
| --- | --- | --- |
|  | 作用 | 示例 |
| [attribute] | 匹配包含指定属性的元素 | div[id]匹配有id属性的div |
| [attribute=value] | 匹配attribute属性值为value的元素 | $(“div[id=main]”)->  $(“#main”) |
| [attribute != value] | 匹配attribute属性不是value的元素 |  |
| [attribute ^= value] | 匹配attribute属性值是以value开头的字符 | div[class^=col] |
| [attribute $= value] | 匹配attribute属性值是以value结尾的元素 | div[class$=col] |
| [attribute \*= value] | 匹配attribute属性值中包含value字符的元素 |  |



7 子元素过滤选择器

|  |  |  |
| --- | --- | --- |
| :first-child | 匹配属于其父元素中的首个子元素  （可能匹配出来多个） | $(‘ul li:first-child’);每个ul中的第一个li |
| :first |  | $(‘li:first’); 第一个li |
| :nth-chiild(n) | 匹配属于其父元素中的第n个子元素 |  |
| :last-child | 匹配属于父元素中最后一个子元素 |  |



8 jQuery操作DOM

8.1 基本操作

html()

作用：

获取或设置jQuery对象中的HTML内容，等同于DOM中的innerHTML属性



text()

作用：

获取或设置jQuery对象中的文本内容，等同于DOM中的innerText属性

val()

作用：

获取或设置jQuery对象的value属性值（针对表单控件）



等同于DOM中的value属性

示例：

<input type=’text’ id=’name’>

console.log($(‘#uname’).val());//获取uname的值并打印

$(‘#uname’).val(‘Mary’);//设置uname的值

练习：

1.创建一个文本框并定义id

2.创建一个按钮，显示文字为‘显示’

3.创建一个div并定义id

4.点击按钮时，将文本框中的内容获取出来，构建成一级标题的样式显示在div中



8.2 属性操作

1.attr()

作用：读取

|  |  |  |
| --- | --- | --- |
| 属性 | 作用 | 示例 |
| attr() | 读取或设置jQuery对象的属性 | $obj.attr(‘id’);  读取$obj对象的id属性值  $obj.attr(‘id’,’container’);  设置$obj对象的id属性值container |
| removeAttr(‘attrName’) | 删除jQuery对象的attrName属性 | $obj.removeAttr(‘class’);  从$obj对象中将class属性移除出去 |

8.3 样式操作

|  |  |  |
| --- | --- | --- |
| 样式操作 | 作用 | 示例 |
| attr(‘class’,’className’) | 使用attr()动态绑定class属性值 | $obj.attr(‘class’,’className’); |



|  |  |  |
| --- | --- | --- |
| addClass(‘className’) | 将className类选择器追加到jQuery对象class值之后 | 返回当前的jQuery对象  $obj.addClass(‘bigFont’);  $obj.addClass(‘red’);  $obj.addClass(‘bigFont’).addClass(‘red’); |



|  |  |  |
| --- | --- | --- |
| removeClass(‘className’) | 删除所有的类选择器 |  |



|  |  |
| --- | --- |
| toggleClass(‘className’) | 元素如果具备className类选择器，则删除；  元素如果没有className类选择器，则添加； |

显示隐藏



|  |  |  |
| --- | --- | --- |
| css(‘属性’,’值’) | 为元素设置某css的属性及其值 | obj.css(‘background-color’,’red’); |
| css(JSON)对象 | 为元素设置一组css属性及其值 | obj.css({‘color’:’red’,  ’font-size’:’12px’}); |