

一、jQuery简介

1. 介绍
2. 使用
 - 1) 引入
 - 2) 工厂函数 - `$()`
 - 3) 原生JS对象与jQuery对象
 - 4) jQuery获取元素
 - 5) 操作元素内容
 - 6) 操作标签属性
 - 7) 操作标签样式
 - 8) 根据层级结构获取元素
 - 9) 元素的创建,添加,删除
 - 10) jQuery事件处理

一、jQuery简介

1. 介绍

jQuery是JS的工具库，对原生JS中的DOM操作、事件处理、包括数据处理和Ajax技术等进行封装,提供更完善,更便捷的方法。

2. 使用

1) 引入

先引入jquery文件，才能使用jquery语法

2) 工厂函数 - `$()`

"`$()`"函数用于获取元素节点，创建元素节点或将原生JS对象转换为jquery对象,返回 jQuery 对象。jQuery 对象实际是一个类数组对象，包含了一系列 jQuery 操作的方法。
例：

```
//$()获取元素节点, 需传入字符串的选择器
$("h1")
$("#d1")
$(".c1")
$("body, h1, p")
```

3) 原生JS对象与jQuery对象

原生JS对象与jQuery对象的属性和方法不能混用。可以根据需要，互相转换：

1. 原生JS转换jQuery对象
`$(原生对象)`，返回 jQuery 对象
2. jQuery对象转换原生JS对象
 - 方法一：根据下标取元素,取出即为原生对象
`var div = $("div")[0];`

- 方法二：使用jQuery的get(index)取原生对象
var div2 = \$("div").get(0);

4) jQuery获取元素

jQuery通过选择器获取元素，\$("选择器")

选择器分类：

1. 基础选择器

标签选择器：\$("div")
ID 选择器：\$("#d1")
类选择器：\$(".c1")
群组选择器：\$("body,p,h1")

2. 层级选择器

后代选择器：\$("div .c1")
子代选择器：\$("div>span")
相邻兄弟选择器：\$("h1+p") 匹配选择器1后的第一个兄弟元素,同时要求兄弟元素满足选择器2
通用兄弟选择器：\$("h1~h2") 匹配选择器1后所有满足选择器2的兄弟元素

3. 过滤选择器

需要结合其他选择器使用。

:first
匹配第一个元素 例: \$("p:first")
:last
匹配最后一个元素 例: \$("p:last")
:odd
匹配奇数下标对应的元素
:even
匹配偶数下标对应的元素
:eq(index)
匹配指定下标的元素
:lt(index)
匹配下标小于index的元素
:gt(index)
匹配下标大于index的元素
:not(选择器)
否定筛选,除()中选择器外,其他元素

4. 属性选择器

属性选择器以[]为标志.

1. [attrName]
匹配包含指定属性的元素
2. [attrName=value]/[attrName="value"]
匹配属性名=属性值的元素
3. [attrName^=value]
匹配属性值以指定字符开头的元素
4. [attrName\$=value]
匹配属性值以指定字符结尾的元素
5. [attrName*=value]
匹配属性值包含指定字符的元素

5. 子元素过滤选择器

```
:first-child
    匹配第一个子元素
:last-child
    匹配最后一个子元素
:nth-child(n)
    匹配第n个子元素(n从1开始计数)
```

5) 操作元素内容

```
html() //设置或读取标签内容,等价于原生innerHTML,可识别标签语法
text() //设置或读取标签内容,等价于innerText,不能识别标签
val()  //设置或读取表单元素的值,等价于原生value属性
```

6) 操作标签属性

1. attr("attrName","value")
设置或读取标签属性
2. prop("attrName","value")
设置或读取标签属性
注意:在设置或读取元素属性时,attr()和prop()基本没有区别;但是在读取或设置表单元素(按钮)的选中状态时,必须用prop()方法,attr()不会监听按钮选中状态的改变,只看标签属性checked是否书写
3. removeAttr("attrName")
移除指定属性

7) 操作标签样式

1. 为元素添加id/class属性,对应选择器样式
2. 针对类选择器,提供操作class属性值的方法

```
addClass("className") //添加指定的类名
removeClass("className")//移除指定的类型,如果参数省略,表示清空class属性值
toggleClass("className")//结合用户行为,实现动态切换类名.如果当前元素存在指定类名,则移除;不存在则添加
```

3. 操作行内样式

```
css("属性名","属性值") //设置行内样式
css(JSON对象)           //设置一组CSS样式
/*
JSON对象:常用数据传输格式
语法:
{
    "width":"200px",
    "height":"200px",
    "color":"red"
}
*/
```

8) 根据层级结构获取元素

1. parent()
返回父元素

2. `parents('selector')`
返回满足选择器的祖先元素
3. `children()/children("selector")`
返回所有直接子元素/返回满足选择器的直接子元素
4. `find("selector")`
返回所有的后代元素(包含直接与间接)
5. `next()/next("selector")`
返回下一个兄弟元素/返回下一个兄弟元素,必须满足选择器
6. `prev()/prev("selector")`
返回前一个兄弟元素/返回前一个兄弟元素,要求满足选择器
7. `siblings()/siblings("selector")`
返回所有的兄弟元素/满足选择器的所有兄弟元素

9) 元素的创建,添加,删除

1. 创建
使用`$("标签语法")`, 返回创建好的元素

```
var div = $( "<div></div>" ); //创建元素
div.html( "动态创建" ).attr( "id", "d1" ).css( "color", "red" ); //链式调用, 设置内容和属性
var h1 = $( "<h1 id='d1'>一级标题</h1>" ); //创建的同时设置内容, 属性和样式
```

2. 添加至页面
 - 1) 作为子元素添加

```
$obj.append(newObj); //在$obj的末尾添加子元素newObj
$obj.prepend(newObj); //作为第一个子元素添加至$obj中
```

- 2) 作为兄弟元素添加

```
$obj.after(newObj); //在$obj的后面添加兄弟元素
$obj.before(newObj); //在$obj的前面添加兄弟元素
```

- 3) 移除元素

```
$obj.remove(); //移除$obj
```

10) jQuery事件处理

1. 文档加载完毕
原生JS 方法: `window.onload`
jQuery:

```
//语法一
$(document).ready(function (){
    //文档加载完毕后执行
})
//语法二
$.ready(function (){
    //文档加载完毕后执行
})
//语法三
$(function(){
    //文档加载完毕后执行
})
```

区别:

原生onload事件不能重复书写, 会产生覆盖问题; jquery中对事件做了优化,可以重复书写ready方法,依次执行

2. 事件绑定方式

事件名称省略 on 前缀

```
//on("事件名称", function)
$("div").on("click", function(){});
//bind("事件名称", function)
$("div").bind("click", function(){});
//事件名作为方法名
$("div").click(function(){});
```

3. this表示事件的触发对象, 在jquery中可以使用, 注意转换类型。this为原生对象只能使用原生的属性和方法, 可以使用\$(this)转换为jquery对象, 使用jquery方法。