jQuery

## jQuery基础

### 入口函数

#### 原生js和jQuery的区别

1、原生js和jQuery入口函数的加载模式不同：

原生JS会等到DOM元素加载完毕，并且图片也加载完毕才会执行；jQuery会等到DOM元素加载完毕，但不会等到图片也加载完毕就会执行

2、原生js如果编写了多个入口函数，后面编写的会覆盖前面编写的；jQuery中编写多个入口函数，后面的不会覆盖前面的

#### 解决冲突问题

推荐的入口函数书写方式：$(function(){

})

当$和其他冲突：

1：释放$的使用权，jQuery.noConflict();释放操作必须在编写其他jQuery代码前

2：自定义一个访问符号如var nn=jQuery.noConflict();

### 核心函数

$(); 就代表调用jQuery的核心函数

1. 接收一个函数
2. 接收一个字符串

2.1接收一个字符串选择器，返回一个jQuery对象，对象**保存了找到的DOM**元素

2.2接收一个代码片段，返回一个jQuery对象，对象保存**创建的DOM**元素

1. 接收一个DOM元素（会被包装成jQuery对象返回给我们）

jQuery对象是一个伪数组（由0到length-1的属性，有length属性）

### 静态方法

静态方法直接添加到类上，通过类名调用，实例方法通过类的实例调用

添加静态方法：类名.staticMethod();

添加实例方法：类名.prototype.instanceMethod();

#### each方法

1、原生的each：数组名.forEach(function(value,index){

})

第一个参数：遍历到的元素；第二个参数：当前遍历到的索引

注意原生的forEach方法只能遍历数组，不能遍历伪数组

2、jQuery的each静态方法遍历数组：

第一个参数：当前遍历到的索引；第二个参数：遍历到的元素（与原生的顺序不同）

**注意jQuery的each方法可以遍历伪数组**

#### map方法

1、利用原生JS的map方法遍历：

数组名.map(function(value,index,array){

})

第一个参数：当前遍历到的元素

第二个参数：当前遍历到的索引

第三个元素：当前被遍历的数组

注意点：和原生的forEach一样，不能遍历伪数组

2、jQuery里的map方法

$.map(数组名,function(value,index){

})

第一个参数：要遍历的数组

第二个参数：每遍历一个元素后执行的回调函数

回调函数的参数：第一个参数：遍历到的元素；第二个参数：遍历到的索引

注意点：和jQuery中的each静态方法一样，可以遍历伪数组

jQuery中的each和map方法区别：

1. each静态方法默认的返回值是遍历谁就返回谁，map默认返回的是空数组。
2. map静态方法可以在回调函数中通过return对遍历的数组进行处理，然后生成一个新的数组返回

#### 其他静态方法

1、var res = $.trim(str);去掉字符串两端的空格

参数：需要去除空格的字符串

返回值：去除空格之后的字符串

2、$.isWindow()判断传入的对象是否是window对象

返回值是true/false

3、$.isArray()判断传入的对象是否是数组

4、$.isFunction()判断传入的对象是否是函数

注意点：jQuery框架本质上是一个匿名函数

1. holdReady方法 ：暂停入口函数的执行$.holdReady(true);

### 选择器

#### 内容选择器

1. empty：找到既没有文本内容也没有子元素的指定元素，例：

var $div = $("div:empty");

1. parent:找到有文本内容或有子元素的指定元素。例：var $div = $("div:parent");
2. contains：找到包含文本内容的元素。例var $div = $("$div:contains('我是div')");
3. has：找到包含指定子元素的元素。例var $div = $("div:has('span')");

### 属性操作

#### 属性的定义及操作

定义：对象身上保存的变量

如何操作：对象.属性名; / 对象[“属性名”];

#### 属性节点

在标签中添加的属性就是属性节点

在浏览器中找到span这个DOM元素之后，展开看到的都是属性，在attributes属性中保持的所有内容都是属性节点

操作属性节点：DOM元素.setAttribute(“属性名称”,”值”);

DOM元素.getAttribute(“name”);

属性和属性节点的区别：任何对象都有属性，但是只有DOM对象才有属性节点

#### attr方法

attr(name|pro|key,val|fn)作用：获取或设置属性节点的值，可以传递一个参数，也可以传递两个参数。传递一个代表获取属性节点的值，传递两个参数，代表设置属性节点的值

**注意点：**

如果是获取：只会返回第一个元素指定的属性节点的值

如果是设置：找到多少个元素就会设置多少个元素，不存在的属性节点会自动新增一个

removeAttr(name)删除属性节点：会删除所有找到元素指定的属性节点，若想删除多个属性节点，则各节点间以空格隔开

示例：$(“属性节点”).removeAttr(属性节点1 属性节点2);

#### prop方法

prop方法的特点和attr方法一样，removeProp方法特点和removeProp方法一样

注意：

prop方法不仅可以操作属性，还可以操作属性节点

官方推荐在操作属性节点时具有true和false属性的属性节点，如checked、selected或者disabled使用prop()，其他的使用attr()

#### Jquery操作类相关的方法

1. addClass(class | fu)

作用：添加一个类，如果添加多个，多个类名间用空格隔开

$(“要添加类的元素”).addClass(“类1 类2”);

1. removeClass(class | fu)和add用法相似
2. toggleClass切换类：有就删除，没有就添加

#### HTML代码/文本/值

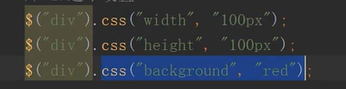
1. html([val | fn])和原生JS的innerHTML一模一样

如：$("div").html("<p>我是段落</p>")

1. text([val | fn])和原生JS 的innerText一样
2. val(“要设置的value值”)和过去的input里的value一样

#### Jquery操作css样式的方法

1. 逐个设置



1. 链式设置



注意：如果大于三步，建议分开

1. 批量设置





1. Width()获取设置的元素宽度



1. offset获取元素距离窗口（浏览器）的偏移位
2. position()获取元素举例定位元素的偏移位

注：position方法只能获取不能设置，可以通过.css()修改

1. scrollTop()



获取网页滚动的偏移位(为了浏览器的兼容性)：

$(“body”).scrollTop()+$(“html”).scrollTop();

设置网页滚动偏移位(保证兼容性)$(“html,body”).scrollTop(偏移位);

## 1

### 2

#### 3

##### 4

###### 5