jQuery的DOM操作

DOM分类：DOM核心（getElmentById()...），HTML DOM，CSS-DOM

# DOM核心

## 文档处理

### 查找节点

|  |
| --- |
| **#查找元素节点**  使用选择器查找：var $li = $("ul li:eq(1)");//查找ul里第二个<li>节点  **#查找属性节点**  使用jQuery对象的attr("属性名")获取：  var p\_txt=$("p").attr("title");//获取p节点的title属性 |

### 创建节点($(html))

|  |
| --- |
| **#创建元素节点**  var $li = $("<li></li>");//创建li元素  **#创建文本节点**  var $li = $("<li>text</li>");//创建li元素  **#创建属性节点**  var $li = $("<li title="attr">text</li>");//创建li元素 |

### 插入节点

|  |
| --- |
| **#内部插入**  append(content)：向每个匹配的元素内部追加内容，添加内容为最后一个子节点  appendTo(content)：将所有匹配的元素最佳到另一个指定元素的集合中  prepend(content)：向每一匹配的元素内部前置内容  prependTo(content)：向所有匹配的元素前置到另一个指定的元素集合中  **#外部插入**  after(content):在每个匹配的元素之后插入内容。  before(content)：在每个匹配的元素之前插入内容。  insertAfter(content)：将所有匹配的元素插入到另一个指定的元素集合后面  insertBefore(content)：将所有匹配的元素插入到另一个指定集合的前面。 |

### 删除节点

|  |
| --- |
| #remove([选择器]),  删除该节点所包含的所有后代节点(整个dom节点)，返回一个已被删除节点引用。也可以接收一个选择器，选择指定的子元素。  var $li=$("ul li:eq(1)").remove();  $li.appendTo("ul");  #detach()，从DOM文档中删除所有匹配的元素。和remove相同，不同的是，detach()保存了所有jquery数据和被移走的元素的相关联。  #empty()：删除匹配的元素集合中所有的子节点。 |

### 复制节点

|  |
| --- |
| clone()：复制匹配的DOM元素并且选中副本  clone(true)：复制匹配的DOM元素以及所有的事件处理并且返回这些副本  $("ul li").click(function){  $(this).clone().appendTo("ul");  }): |

### 替换节点

|  |
| --- |
| replaceWith(content)：将所有匹配的元素替换成指定的HTML或者DOM元素  replaceAll(selector)：一指定的HTML或者DOM元素来替换选择器的元素。和replaceWith方法作用一样，只是颠倒了顺序。  $("p").replaceWith("<b>hello</b>");  $("<b>hello</b>").replaceAll("p"); |

### 包裹节点

|  |
| --- |
| wrap(html/dom):将所有匹配元素分别用指定元素包裹起来  unwrap(dom):去除包裹元素  wrapAll(html/ dom): 将所有匹配元素用一个指定元素包裹起来  wrapInner(html/ dom):将每一个匹配的元素的子内容包括文本节点用指定元素包裹起来  $("strong").wrap("<b></b>");//用<b>元素把<strong>元素包裹起来  $("strong").wrapInner("<b></b>");<Strong>里面的子元素用<b>包裹  $("strong").wrapAll("<b></b>");< strong >集合通过<b >元素包裹起来 |

## 属性

### 属性

|  |
| --- |
| **#获取属性**  attr(name)：获取指定属性名的属性值  $("p").attr("p")；  **#设置属性**  attr(key,value)：设置单个属性值/名 attr(key:value,...)：设置多个属性名/值  $("p").attr("title","myTitle");  $("p").attr("key1":"value1,"key2":"value2");  **#删除属性**  removeAttr(name)  $("P").removeAttr("title");  注意：对于boolean类型的属性disabled,checked,selected使用prop()方法进行操作。 |

### 样式属性

|  |
| --- |
| #获取和设置样式  attr("class")  **#追加样式**  addClass(class)  $("p").addClass("another");  **#移除样式**  removeClass()：删除所有的class  removeClass(class):删除指定的class  $("p").removeClass("anthor");  **#切换样式**  toggleClass(class)：如果存在（不存在）class，就删除（添加）一个类  $("p").toggleClass("anthor");  **#判断是否存在某样式**  hasClass(class)  $("p").hasClass("anthor"); |

### HTML值

|  |
| --- |
| #html()方法，此方法类似于javascript中的innnerHTML属性  **#获取HTML值**  $("p").html();//<strong>..</strong>  **#设置HTML值**  $("p").html("<strong>...</strong>); |

### 文本值

|  |
| --- |
| #text方法类似于javascript中的innerText属性，可以用来获取或者设置文本内容  **#获取文本内容**  $("p").text(); //选择器DOM对象的文本内容  **#设置文本内容**  $("p").text("hello"): |

### val值

|  |
| --- |
| #val()方法类似于javascript中的value值，可以用来设置和获取元素的值。可以获取文本框，单元框，下拉框，多选框（返回数组）等的值。  **#获取value值(直接获取select值，不能直接checkbox,radio的值)**  $("#single").val();  **#获取radio的值，**  $(":radio[name='c']:checked").val();  **#获取checkbox的值，只能**  $.each($(":checkbox[name='c']:checked"),function(){ this.value  });  **#设置value值，是表单选中该值(单选，多选，多下拉框都应该使用数组)**  $(":chekbox").val(["check2","check3"]; |

## 查找和遍历节点

### children()方法

|  |
| --- |
| #children方法用于取得匹配元素的子元素集合。  var $ul = $("ul").children();  alert( $ul.length ); // <p>元素下有3个子元素  for(var i=0;i< $ul.length;i++){  alert( $ul[i].innerHTML );//苹果，...  }  注意：children()只考虑子元素，不考虑后代元素。find查找的是所有的后代。 |

### next方法

|  |
| --- |
| #next()方法用于取得匹配元素后面紧邻的同辈元素。  var $p1=$("p").next();  注意：nextAll()用于获取匹配元素后面所有同辈元素，返回的是数组 |

### prev方法

|  |
| --- |
| #prev()方法用于取得匹配元素前面紧邻的同辈元素  var $ul = $("ul").prev();  注意：prevAll()获取的是前面同辈元素的集合。 |

### siblings方法

|  |
| --- |
| #siblings方法用于取得匹配元素前后所有的同辈元素 |

### closest()方法

该方法用于取得最近的匹配元素。从元素本身开始，逐级向上级元素匹配，并返回最先匹配的祖先元素。

parent()：获取集合中每个匹配元素的父级元素。

parents()：返回集合中每个匹配元素的祖先元素。找到一个父节点还会继续查找父节点，最后返回多个父节点。

注意：它们都可以加入选择器。

### 常用筛选方法

|  |
| --- |
| find(selector) 查找子节点  eq(index)  filter(expr)  is(selector)：  not(expr) |

# CSS-DOM操作

## 样式css

|  |
| --- |
| #获取样式属性  css(name);//无论是外部导入的属性还是内联的属性  $('p:eq(1)').text( $('.first').css("font-size") );  var value = $('.first').css(['width','height']);  #设置样式  css(name,value):给所有匹配的元素的一个样式属性设置值  css(proerties):给所有匹配的元素的多个样式属性设置值。  注意：.addClass()方法是通过增加class名的方式，那么这个样式是在外部文件或者内部样式中先定义好的，等到需要的时候在附加到元素上  通过.css()方法处理的是内联样式，直接通过元素的style属性附加到元素上的 |

## 定位

|  |
| --- |
| # offset()方法，用于获取元素当前视窗的相对偏移量  var offset=$("p").offset();  var left = offset.left;  var top = offset.top;  #position()方法获取元素相对于最近的position样式属性为relative或者absolute的祖父节点的相对偏移  var offset=$("p". position();  var left = position().left;  var top = position.top;  #scrollTop()方法和srollLeft()方法：获取滚动条距顶端距离和距左侧距离  $("p"). scrollTop();  $("p"). srollLeft();  #设置滚动条距顶端的距离和距左侧的距离  $("p"). scrollTop(300);  $("p"). srollLeft(300); |

## 高宽

|  |
| --- |
| #height(),width()获取元素的宽度和高度值默认的单位是px  $("p").height();  $("p").width();  #设置宽度和高度px和em()  $("p).height(100);  $("p).height("10em");  注意：  width():获取width的长度  innerWidth():获取padding+width  outterWidth():获取bodder+padding+width  outterWidth(true):获取margin+bodder+padding+width |

# jQuery常用方法

## each()遍历方法

|  |
| --- |
| **#each方法的使用**  $.each(["Aaron", "慕课网"], function(index, value) {  //index是索引,也就是数组的索引  //value就是数组中的值了  //this表示当前dom对象  }); |

## inArray()方法

|  |
| --- |
| **#inArray()函数**用于在数组中搜索指定的值，并返回其索引值。如果数组中不存在该值，则返回 -1。  jQuery.inArray( value, array ,[ fromIndex ] )  用法非常简单，传递一个检测的目标值，然后传递原始的数组，可以通过fromIndex规定查找的起始值，默认数组是0开始  $.inArray(5,[1,2,3,4,5,6,7]) //返回对应的索引：4 |

## trim()方法

|  |
| --- |
| **#jQuery.trim()函数用于去除字符串两端的空白字符**  移除字符串开始和结尾处的所有换行符，空格(包括连续的空格)和制表符（tab），如果这些空白字符在字符串中间时，它们将被保留，不会被移除 |

## index()方法

|  |
| --- |
| **#jQuery.** index()**函数用于获取节点在兄弟索引中的位置（从0开始）** |