July16 jQuery

1 jQuery操作DOM

1.1 创建对象

语法：

$(“创建的完整标记”);

示例：

1.创建一对div元素

var $div = $(“<div></div>”);

$div.html(“<h1>…</h1>”);

$div.attr(“id”,”container”);

2.创建一个图像标记

var $(“<img>”);

$img.attr(“src”,”img/a.jpg”);

1.2 插入元素

**1.2.1内部插入**

作为子元素插入到网页中

1.$obj.append($new);

将$new元素追加到$obj元素的尾部（子元素）

2.$obj.prepend($new);

将$new元素添加到$obj元素的头部（第一个子元素）



**1.2.2外部插入**

作为兄弟元素插入到网页中

1.$obj.before($new)

将$new作为$obj的上一个兄弟元素插入进来



2.$obj.after($new)

将$new作为$obj的下一个兄弟元素插入进来

1.3 删除节点

|  |  |  |
| --- | --- | --- |
|  | 语法 | 作用 |
| remove(); | $obj.remove(); | 删除$obj元素 |
| remove(“selector”) | $obj.remove(“selector”); | 将满足selector选择器的元素删除出去 |
| empty | $obj.empty() | 清空$obj内的内容 |



1.4 遍历节点

1.children()/children(selector)

作用：

获取某对象中的子元素（只考虑元素节点）或满足selector选择器的子元素（只考虑元素节点）。

注意：

只能获取子元素，不能获取后代元素

2.next/next(selector)

作用：

获取某对象的下一个兄弟元素/满足selector的下一个兄弟元素

3.prev()/prev(selector)

作用：

获取某对象的上一个兄弟元素/满足selector的上一个兄弟元素

4.siblings()/siblings(selector)

作用：

获取某对象的所有兄弟元素/满足selector的兄弟元素

5.find(selector)

作用：

查找满足selector选择器的所有后代元素

6.parent()

作用：

查找某对象的父元素

2 事件参数对象

2.1 获取事件对象

1.

<button onclick=”btnClick(event)”></button>

function btnClick(event){

}

2.

<button id = “btnTest”></button>

var btn = document.getElementById(“btnTest”);

btn.onclick = function(event){

event:表示当前时间的事件对象

}

2.2事件对象属性

1.offsetX

获取鼠标操作的点的横坐标值

2.offsetY

获取鼠标操作的点的纵坐标值

3.target

获取事件源（事件所在的元素）



2.3 this

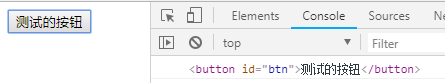
1.this在全局函数中，永远都表示的是window对象

2.this在局部函数中，表示都是函数的所有者

btn.onclick = function(){

this:表示的就是函数的所有者btn

}





3 jQuery事件处理

3.1 页面加载后的执行

在DOM树加载完成后就要执行的操作

1.

$(document).redy(function(){

//页面的初始化操作

//DOM树加载完毕后就开始运行

});

2.

$().ready(function(){

});

3.

$(function(){

});



3.2 jQuery的事件绑定

1.方式一

$obj.bind(“事件名称”,事件处理函数);

示例：

$obj.bind(“click”,function(){

alert(“单击事件”);

});

2.方式二

$obj.事件名称(事件处理函数)

示例：

$obj.click(function(){

alert(“单击事件”);

});

注意：

在事件处理函数中，允许使用this来表示触发当前事件的DOM对象





4 jQuery事件

4.1 事件对象-event

在绑定事件的时候，允许传递event参数来表示事件对象

1.

$obj.bind(“click”,function(event){

//event表示的就是事件对象

});

2.

$obj.click(function(event){

//event表示的就是事件对象

});

event的使用方式与原生JS事件中的事件对象一致

1.offsetX

获取鼠标操作的点的横坐标值

2.offsetY

获取鼠标操作的点的纵坐标值

3.target

获取事件源（事件所在的元素）

4.event.which

键位码或ASCII码

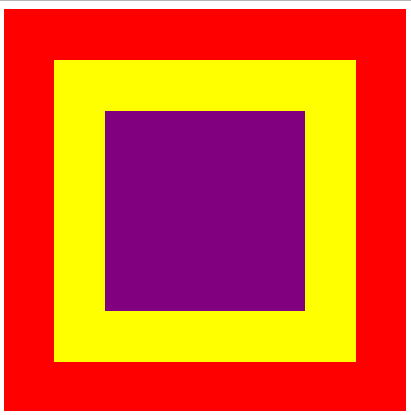
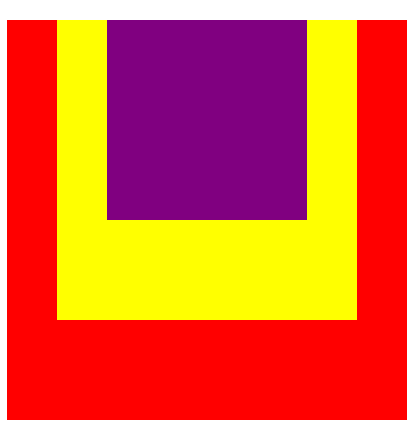




4.2 事件冒泡

1.什么是事件冒泡？

在执行子元素的事件时，同时把父元素、祖先元素的对应事件也给执行了





2.阻止事件冒泡

event.stopPropagation()



5 jQuery动画

5.1 基本显示/隐藏

语法：

$obj.show()/$obj.show(执行的时间)

$obj.hide()/$obj.hide(执行时间)

5.2 滑动显示/隐藏

语法：

显示$obj.slideDown()/$obj.slideDown(执行时间)

隐藏$obj.slideUp()/$obj.slideUp(执行时间)

5.3 淡入淡出显示/隐藏

语法：

显示$obj.fadeIn()/$obj.fadeIn(执行时间)

隐藏$obj.fadeOut()/$obj.fadeOut(执行时间)



==========================





Jquery遍历操作

方法：

each()

1.循环遍历jQuery数组

$obj.each(function(i循环出元素下标,obj循环出来的元素DOM){

});

2.循环遍历JS数组（原生）

①将原生数组通过工厂函数$()转换为Jquery数组

②$.each(原生数组,function(i,obj){

i：遍历出来的下标

obj：循环出来的元素

});