# 第7阶段 jQuery基础

## 7.1 jQuery简介及语法

1. 课程概要：认识jQuery、引入jQuery、jQuery语法。
2. jQuery是一个JavaScript**函数库**，jQuery库可以通过一行简单的标记被添加到网页中。

jQuery对JavaScript进行封装，使用起来更加方便，并且jQuery在封装JavaScript时考虑了浏览器兼容问题。

注意：深刻理解**函数库**的含义，$(选择器).封装函数()为jQuery函数库中函数的基本(最小)单位。$(selector)和action()相互依存、不可分隔，否则不能成功调用jQuery函数库中的函数。使用jQuery的过程即是**调用**jQuery函数库中**函数**的过程。

1. jQuery库包含的功能：HTML**元素选取**、HTML**元素操作**、**CSS操作**、HTML**事件函数**、JavaScript**特效和动画**、HTML **DOM遍历和修改**、**AJAX**、**Utilities**。
2. AJAX：在不刷新页面的情况下，修改页面的内容，将服务器所请求的数据直接显示在页面上，提高用户体验。
3. 引入jQuery的方法：
4. 在网页中添加jQuery

1)从jquery.com下载jQuery库

2)从CDN中载入jQuery，如从Google中加载jQuery

1. :百度和新浪的服务器存有jQuery

例如：<http://libs.baidu.com/jquery/1.10.2/jquery.min.js>

b):谷歌和微软的服务器存有jQuery

例如：[http://ajax.googleapis.com/ajax/libs/jquery/jquery.min.js](http://ajax.gppgleapis.com/libs/jquery/jquery.min.js)

1. 版本介绍

v1.11.1:1.x支持IE6+

v2.1.1 :2.x支持IE9+

1. jQuery语法

基础语法：$(selector).action()

美元符号定义jQuery

选择符(selector)“查询”和“查找”HTML元素

jQuery的action()执行对元素的操作

$(this).hide()-隐藏当前元素

$(“p”).hide()-隐藏所有段落

举例：<script>

$(document).ready(function(){ //文档加载完毕。

// alert("文档加载完毕")

$('p').click(function(){

$(this).hide();

})

})

</script>

注意：$(selector).action(function(){}/空)——①利用$()方法定位到元素之后一定要加上jQuery已封装的函数，即：jQuery函数库中$(selector)不是一个独立的函数，$(selector).action()是最小的函数单位。②$(selector).action()括号中有两种情况，一种是空，一种是function(){}。

即：jQuery函数库的基本单位——**$(选择器).封装函数()**

## 7.2 jQuery选择器和事件

1. jQuery选择器

(1)元素选择器 (2)ID选择器 (3)类选择器

<p>p1</p>

<p id="pid">p2</p>

<p class="pclass">p3</p>

<button>click me</button>

<script>

$(document).ready(function(){

$("button").click(function(){

$("p").text("P1元素被修改了")

//$("元素名称")——元素选择器

$("#pid").text("P2元素被修改了")

//$("#id名称")——ID选择器

$(".pclass").text("P3元素被修改了")

//$(".class名称")——类选择器

});

});

</script>

注意：①jQuery中三种选择器——元素选择器:$("元素名称"); ID选择器—$("#id名称");类选择器—$(".class名称")。②$(selector).text()—text函数为修改元素中文本内容。

2. jQuery事件

1. 常用事件方法; (2)绑定事件; (3)解除绑定事件;
2. 事件的目标; (5)事件的冒泡; (6)自定义事件;
3. 常用的事件方法

点击事件、单机事件、双击事件、鼠标移入事件、鼠标移出事件、获取焦点事件、失去焦点事件……

<script>

$(document).ready(function(){

// $("button").click(function(){ //单机事件

// $("button").dblclick(function(){ //双击事件

// $("button").mouseenter(function(){//鼠标移入事件

$("button").mouseleave(function(){//鼠标移出事件

$(this).hide();

//不可以写this.hide(),因为hide()方法为jQuery库封装的方法，也不可以写$("this").hide()

})

})

</script>

注意：$(selector).click()为jQuery封装的单机事件方法(函数);

$(selector).dblclick()为jQuery封装的双击事件方法(函数);

$(selector).mouseenter()为jQuery封装的鼠标移入事件方法(函数);

$(selector).mouseleave()为jQuery封装的鼠标移出事件方法(函数)。

4. 绑定事件与解除绑定事件

<button id="btn">Click me</button>

<script>

$(document).ready(function(){

/\*$("#btn").click(function(){

alert("hello")

})\*/

/\*$("#btn").bind("click",demo1)

$("#btn").bind("click",demo2)

// $("#btn").unbind("click",demo1)//解除demo1。

$("#btn").unbind("click")//全部解除。\*/

$("#btn").on("click",demo1)

$("#btn").on("click",demo2)

$("#btn").off("click",demo1)//解除demo1。

// $("#btn").off("click")//全部解除。

});

function demo1(){

console.log("hello1");

};

function demo2(){

console.log("hello2");

};

</script>

</body>

注意：①绑定事件bind等价于on,推荐使用on;解除绑定unblind等价于off,推荐使用off。②jQuery函数库封装了$(selector).on()函数与$(selector).off()函数。

1. 事件目标与事件冒泡

<script>

$(document).ready(function(){

$("body").on("click",function(e){

console.log(e);

});

$("div").on("click",function(e){

console.log(e);

e.stopPropagation();

//阻止**父级**事件冒泡。

// e.stopImmediatePropagation();

//阻止**所有的(父级和自己)**事件冒泡。

});

$("div").on("click",function(e){

console.log(e);

});

});

</script>

1. 自定义事件对象

<button id="btn">按钮</button>

<script>

$(document).ready(function(){

$("#btn").click(function(){

var e=jQuery.Event("myEvent");//自定义事件对象

$("#btn").trigger(e);

});

$("#btn").bind("myEvent",function(e){

//绑定自定义事件

console.log(e);

});

});

</script>

问题：自定义事件对象如何触发？

## 7.3 jQuery效果之隐藏与显示、淡入淡出、滑动、回调

1. 课程概要：jQuery隐藏/显示;jQuery淡入淡出;jQuery滑动;jQuery动画;jQuery停止动画;jQuery Callback;jQuery Chaining;
2. jQuery效果—隐藏/显示

例题1：

<p>这是一个P元素</p>

<button id="btn1">hide</button>

<button id="btn2">show</button>

<button id="btn3">toggle</button>

<script>

$(document).ready(function(){

$("#btn1").click(function(){

$("p").hide(1000);

});

$("#btn2").click(function(){

$("p").show(1000);

});

$("#btn3").click(function(){

$("p").toggle(1000);

});

});

</script>

注意:$(selector).hide(毫秒数,function(){})函数;

$(selector).show(毫秒数,function(){})函数;

$(selector).toggle(毫秒数,function(){})函数。

例题2：

<script>

$(document).ready(function(){

for (var n=0;n<6;n++) {

$("<div>").appendTo(document.body);

};

$("div").click(function(){

$(this).hide(1000,function(){

$(this).remove();

});

});

});

</script>

注意：①toggle—开关、触发器。②$("<元素>")表示创建元素，$("元素")表示选择元素。

1. jQuery效果—淡入/淡出

<button id="btn1">淡入</button>

<button id="btn2">淡出</button>

<button id="btn3">淡入/淡出</button>

<button id="btn4">透明度</button>

<div id="div" style="width: 100px;display: none;height: 100px;background: #FF8C00;"></div>

<script>

$(document).ready(function(){

$("#btn1").on("click",function(){

$("#div").fadeIn(1000)

});

$("#btn2").on("click",function(){

$("#div").fadeOut(1000)

});

$("#btn3").on("click",function(){

$("#div").fadeToggle(1000)

});

$("#btn4").on("click",function(){

$("#div").fadeTo(1000,0.5)

});

});

</script>

注意：淡入—fadeIn(毫秒数);淡出—fadeOut(毫秒数);

开关—fadeToggle(毫秒数);淡到—fadeTo(毫秒数,透明度)

1. jQuery效果—滑动

<div id="div1">出现</div>

<div id="div2">隐藏</div>

<div id="div3">出现/隐藏</div>

<div id="div4">Hello World</div>

<script>

$(document).ready(function(){

$("#div1").click(function(){

$("#div4").**slideDown(1000)**

});

$("#div2").click(function(){

$("#div4").**slideUp(1000)**

});

$("#div3").click(function(){

$("#div4").**slideToggle(1000)**

});

});

</script>

5. jQuery效果—回调

回调——每一个**动画执行完毕之后**所能执行的一个方法。

<button id="btn">按钮</button>

<h1>wenzi fdsafdsafdsa fdsaf dsaf </h1>

<script>

$(document).ready(function(){

$("#btn").click(function(){

/\***$("h1").hide(1000,function(){**

alert("隐藏结束")

});\*/ **$("h1").css("color","red").slideUp(1000).slideDown(1000);**

});

});

</script>

注意：**两种回调方式。**

## 7.4 jQuery HTML之捕获、设置、添加、删除

1. 课程概要:jQuery捕获;jQuery设置;jQuery添加元素;jQuery删除元素。

2. jQuery HTML—捕获

<button id="btn">按钮</button>

<p id="pid">this is a webpage<a id="aid" href='事件的方法.html'>a标签</a></p>

<input type="text" id="ipt" value="jikexueyuan" />

<script>

$(document).ready(function() {

$("#btn").click(function() {

//alert(**$("#pid").text()**);//只能获取具体内容。

//alert(**$("#pid").html()**);//可以获取标签。

//alert($("#ipt").val());

alert(**$("#aid").attr("href")**);

});

});

</script>

3. jQuery HTML—设置

<p id="p1">Hello world</p>

<button id="btn1">按钮</button>

<input type="text" id="ipt" value="hello"/>

<script>

$(document).ready(function(){

$("#btn1").click(function(){

//  **$("#p1").text("jikexuyuan");**

//修改文本内容。

// **$("#p1").html("<b>jikexueyuan</b>")**

//修改标签内容。

// $("#ipt").val("world")

//修改value属性值。

// **$("#ipt").attr("value","World")**

//修改一个属性值。

// **$("#ipt").attr({"value":"World","type":"password"})** //修改多个属性值。

**$("#p1").text(function(i,j){**

//回调函数，有两个参数:元素下标与原内容(可省略)。

return "元素下标："+i+" 原内容："+j;

//返回新内容。

});

});

});

</script>

4. jQuery HTML—添加元素

<p id="pid">Hello world</p>

<button id="btn">插入</button>

<script>

$(document).ready(function(){

$("#btn").click(function(){

// **$("#pid").append("结尾插入内容")**

// **$("#pid").prepend("开头插入内容")**

// **$("#pid").before("之前插入内容")**

// **$("#pid").after("之后插入内容")**

/\*三种方法:html jQuery DOM(javaScript)\*/

var tex1="<h1>iwen1</h1>"

var tex2=$("<h1>").text("iwen2");

var tex3=document.createElement('h1');

tex3.innerHTML="iwen3";

**$("#pid").append(tex1,tex2,tex3);**

});

});

注意：append——在被选中元素的结尾插入内容; prepend——在被选中元素的开头插入内容;before——在被选中元素之前插入内容;after——在被选中元素的之后插入内容。

5. jQuery HTML—删除元素

<p>p1</p>

<button id="btn">删除</button>

<script>

$(document).ready(function(){

$("#btn").click(function(){

// **$("p").remove();//移除元素**

**$("p").empty(); //清空元素**

});

});

</script>

## 7.5 jQuery CSS操作及jQuery的盒子模型

1. 课程概要：jQuery方法;jQuery盒子模型
2. jQuery CSS-jQuery CSS方法

<style>

div{width: 100px;height: 100px;}

.div1{background: #F80;}

</style>

<script src="js/jquery-2.1.1.js"></script>

</head>

<body>

<div></div>

<script>

$(document).ready(function(){

//  **$("div").css("width","100px")**

// **$("div").css({"height":"100px","background":"red"})**

$("div").addClass("div1");

// $("div").attr("class","div1");

$("div").click(function(){

// $(this).removeClass("div1");//this不可加引号。

// $(this).attr("class","");//this不可加引号。

**$(this).toggleClass("div1");**

})

});

</script>

</body>

3. jQuery CSS-jQuery盒子模型

在jQuery中——Width/Height() 指 content的内容尺寸;

innerWidth/Height() 指content+padding的内边距尺寸;

outerWidth/Height() 指content+padding+border的边框尺寸;

outerWidth/Height(true)指content+padding+border+margin外边距尺寸。

## 7.6 jQuery之元素的遍历与元素的过滤

1. 课程概要：jQuery遍历;jQuery过滤。
2. jQuery 遍历之向下遍历

<script>

$(document).ready(function(){

$("#div1").children().css({"border":"3px solid black"})

// $("#div1").children("#div2").css({"border":"3px solid black"})

// $("#div1").find("p").css({"border":"3px solid black"})

})

</script>

注意：children()——定位到所有**子**元素;children("元素")——定位到**子**元素1;find("元素")——定位到**内层**元素1;find()——错误写法。

3. jQuery 遍历之向上遍历

<script>

$(document).ready(function(){

// $("a").parent().css({"border":"3px solid black"})

//一个父元素

//$("a").parents().css({"border":"3px solid black"})

//所有外层元素

//$("a").parents("p").css({"border":"3px solid black"})

//某一个外层元素

// $("a").parentsUntil("#div2").css({"border":"3px solid black"})

//父级元素到某一外层元素。

$("a").parentsUntil().css({"border":"3px solid black"})

//所有外层元素。

})

</script>

注意：parent()——一个父级元素;

parents()——所有外层元素;parents("元素")——某一外层元素;

parentsUntil("元素")——到某一外层元素;parentsUntil()——所有外层元素

4. 同级遍历

在CSS中，\*代表所有元素，可与后代选择器结合使用。

1. 过滤

## 7.7 jQuery AJAX之异步访问和加载片段

1. 异步访问

## 7.8 jQuery的扩展与noConflict

方法一：$.noConflict();

此时$不再是jQuery的简写，使用jQuery替换所有的$。

方法二：var myjq=$.noConflict();

此时myjq是jQuery的简写，使用myjq替换所有的$。

## 7.10 jQuery瀑布流

# 第8阶段 jQuery UI基础

## 8.1 jQuery UI下载与使用

1. 课程概要——认识jQuery UI; 下载jQuery UI; jQuery UI基本使用

2. 认识jQuery UI

jQuery UI是以jQuery为基础的JavaScript网页用户界面的开源库。包含底层交互、动画、特效和可更换主题的可视控件。我们可以直接用它来构建具有很好交互性的web应用程序。

jQuery UI包含许多维持状态的小部件(Widget)，因此，它与典型的jQuery插件使用模式略有不同。所有的jQuery UI小部件(Widget)使用相同的模式，所以只要学会使用其中一个，就知道如何使用其他小部件(Widget)。

支持版本:IE6.0+,Firefox3+,Safari3.1+,Opera9.6+和Google Chrome

3. jQuery UI不分版本，而jQuery 分为1.X和2.X,2.X只支持IE9.0以上版本，因此体积小。

4. jQuery UI主要分为3个部分：交互、小部件、和效果库。

1. 交互(Intearctions)

交互部件是一些与鼠标交互相关的内容，包括Draggable、Droppable、Resizable、Selectable和Sortable等。

(2)小部件(Widgets)

主要是一些界面的扩展，包括AutoComplete、ColorPicker、Dialog、Slider、Tabs、ProgressBar、Spinner等。

1. 效果(Effects)

用于提供丰富的动画效果，包括：Add Class、Color Animation、Toggle等。

## 8.2 jQuery UI Interactions

$("selector").draggable(); —— 自元素可拖动

$("selector").droppable(); —— 自元素可添加放置(drop)事件

$("selector").resizable(); —— 自元素可改变尺寸

$("selector").selectable(); —— 子元素可被选中

$("selector").sortable(); —— 子元素可调换位置

## 8.3 jQuery UI Widgets

$("selector").accordion(); —— 手风琴控件

$("selector").autocomplete(); —— 搜索框控件

$("selector").button(); —— 按钮控件

$("selector").dialog(); —— 对话框控件

$("selector").datepicker(); —— 日期控件

$("selector").progressbar(); —— 进度条控件

$("selector").menu(); —— 菜单控件

$("selector").slider(); —— 进度条控件

$("selector").spinner(); —— 数值文本框控件

$("selector").tabs(); —— **选项卡**控件

## 8.4 jQuery UI Effects

(1)$("selector").animate(paras [ , duration ] [ , easing ] [ , complete ] ) ——动态改变包括颜色在内的CSS样式

例如: $("selector").animate({"backgroundColor":"red",

"borderColor":"yellow","color":"blue"},3000)

注意:①引入jQuery UI，则animate动画可以动态修改背景颜色。若属性名只书写background，则背景颜色无法动态修改。

(2)$("selector").addClass( className [,duration ] [,easing ] [, complete ] ) ——动态添加包括颜色在内的CSS样式

(3)$("selector").removeClass(className[,duration] [, easing ] [, complete ] ) ——动态移除包括颜色在内的CSS样式

注意:①引入jQuery UI，则addClass可以给元素动态添加类名，产生动画效果。②引入jQuery UI，则removeClass可以给元素动态移除类名，产生动画效果。

(4)$("selector").toggleClass(className[,switch] [, duration ] [, easing ] [, complete ] )

注意:引入jQuery UI，则toggleClass可以给元素动态切换类名，产生动画效果。

# 第9阶段 jQuery Mobile基础

## 9.1 了解 jQuery Mobile

1. Jquery Mobile是jQuery在手机和平板设备上的版本。
2. jQuery Mobile事件：jQuery Mobile也提供了针对移动端浏览器的事件：

触摸事件-当用户触摸屏幕时触发

滑动事件-当用户上下滑动时触发

定位事件-当设备水平或者垂直翻转时触发

页面事件-当页面显示、隐藏、创建、加载或未加载时触发。