2018年3月14日19:44:57

1. jQuery是一个集JavaScript、css、Dom、Ajax于一体的强大框架体系，它的主旨是：以更少的代码，实现更多的功能
2. **搭建jQuery的开发环境**

需要到jQuery的官网下载jquery的库文件，是一个.js文件。

现在在官网下载的最新版本是jquery-3.3.1.min.js，而且这个文件只有85kb大小。不用奇怪，只需要这么大的文件就可以了。Jquery还提供一个ui库，别的人基于它开发了一个jquery-easyui，这个库就比较大了。

下载好这个js文件后，把它加载到工程下面，在<script>标签的src里面指定路径即可。

比如你在html的当前路径下建了一个Jsscript文件夹，里面放js文件，这导入方法是：

<script src="JScript/jquery-3.3.1.min.js" type="text/javascript"></script>

1. 在play framework的默认项目下，自带了jquery的库文件，在public/javascript文件夹下，public文件夹下的文件都是静态文件，也就是说可以在系统里面随意调用，即可以在html里面调用，也可以在java语言里面调用。

<script language=*"javascript"* type=*"text/javascript"* src=*"@routes.Assets.at("*javascripts/jquery-1.9.0.min.js")"></script>

1. 第一个jquery程序

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>test jQuery</title>

<script language=*"javascript"* type=*"text/javascript"* src=*"@routes.Assets.at("*javascripts/jquery-1.9.0.min.js")"></script>

<style type=*"text/css"*>

**div**{

padding:*8px 0px*;

font-size:*12px*;

text-align:*center*;

border:*solid 1px #888*;

color:*red*;

}

</style>

<script type=*"text/javascript"*>

$(document).ready(**function**(){

$("div").html("您好！欢迎来到jQuery的精彩世界。");

});

</script>

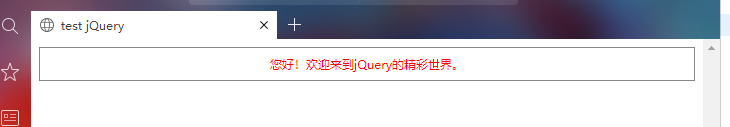
</head>

<body>

<div></div>

</body>

</html>



1. $(document).ready(function(){})可以简写为$(function(){})

而且这个$(function(){})可以写多个

1. 页面元素的选择，功能函数的前缀都需要使用美元符号$

7． jQuery访问DOM对象

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>test jQuery</title>

<script language="javascript" type="text/javascript" src="@routes.Assets.at("javascripts/jquery-1.9.0.min.js")"></script>

<script type="text/javascript">

$(function(){

var tDiv=$("#Tmp");

var oDiv=$("#Out");

var cDiv=tDiv.html();

oDiv.html(cDiv);

});

</script>

</head>

<body>

<div id="Tmp">测试文本333</div>

<div id="Out"></div>

</body>

</html>

功能是给第二个div设置第一div的内容

javascript中访问dom对象的方法是getElementById

var tDiv = document.getElementById(“temp”);

在jQuery中访问dom的方法是：

var tDiv=$("#Tmp");

注意这里的代码要写到function函数里面，也要用$符号

$(function(){

});

1. jquery操作css

之前用到了jQuery中的addClass来设置样式，还可以用toggleClaass来切换样式

$(function(){

$(".content").html("你好,欢迎来到jQuery的精彩世界。");

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

$(this).toggleClass("curcol")

.next(".content").css("display","block");

})

})

点击会切换样式，再次点击回回到原来的样式

1. jQuery的选择器类型

jQuery的选择器分为四大类：基本选择器、层次选择器、过滤选择器、表单选择器

在过滤选择器中又可以分为：简单过滤选择器、内容过滤选择器、可见性过滤选择器、属性过滤选择器、子元素过滤选择器、表单对象属性过滤选择器。

1. 基本选择器

是使用最频繁的选择器，它由元素ID、Class、元素名、多个选择符组成。

<script type="text/javascript">

$(function(){ //ID匹配元素

$("#divOne").css("display","block");

})

$(function(){ //元素名匹配元素

$("div span").css("display","block");

})

$(function(){ //类匹配元素

$(".clsFrame .clsOne").css("display","block");

})

$(function(){ //匹配所有元素

$("\*").css("display","block");

})

$(function(){ //合并匹配元素

$("#divOne,span").css("display","block");

})

<div class="clsFrame">

<div id="divOne">ID</div>

<div class="clsOne">CLASS</div>

<span>SPAN</span>

</div>

1. 层次选择器

主要用于匹配标签里面的标签，也就是包括后代、父子、相邻、兄弟关系。

<script type="text/javascript">

$(function(){ //匹配后代元素

$("#divMid").css("display","block");

$("div span").css("display","block");

})

$(function(){ //匹配子元素

$("#divMid").css("display","block");

$("div>span").css("display","block");

})

$(function(){ //匹配后面元素

$("#divMid + div").css("display","block");

$("#divMid").next().css("display","block");

})

$(function(){ //匹配所有后面元素

$("#divMid ~ div").css("display","block");

$("#divMid").nextAll().css("display","block");

})

$(function(){ //匹配所有相邻元素

$("#divMid").siblings("div").css("display","block");

})

</script>

<body>

<div class="clsFraA">Left</div>

<div class="clsFraA" id="divMid">

<span class="clsFraP" id="Span1">

<span class="clsFraC" id="Span2"></span>

</span>

</div>

<div class="clsFraA">Right\_1</div>

<div class="clsFraA">Right\_2</div>

</body>

1. 简单过滤选择器

过滤选择器根据某类过滤规则进行元素匹配，书写时以：开头。

<script type="text/javascript">

$(function(){ //增加第一个元素的类别

$("li:first").addClass("GetFocus");

})

$(function(){ //增加最后一个元素的类别

$("li:last").addClass("GetFocus");

})

$(function(){ //增加去除所有与给定选择器匹配的元素类别

$("li:not(.NotClass)").addClass("GetFocus");

})

$(function(){ //增加所有索引值为偶数的元素类别,从0开始计数

$("li:even").addClass("GetFocus");

})

$(function(){ //增加所有索引值为奇数的元素类别,从0开始计数

$("li:odd").addClass("GetFocus");

})

$(function(){ //增加一个给定索引值的元素类别,从0开始计数

$("li:eq(1)").addClass("GetFocus");

})

$(function(){ //增加所有大于给定索引值的元素类别,从0开始计数

$("li:gt(1)").addClass("GetFocus");

})

$(function(){ //增加所有小于给定索引值的元素类别,从0开始计数

$("li:lt(4)").addClass("GetFocus");

})

$(function(){ //增加标题类元素类别

$("div h1").css("width","238");

$(":header").addClass("GetFocus");

})

$(function(){

animateIt(); //增加动画效果元素类别

$("#spnMove:animated").addClass("GetFocus");

})

function animateIt() { //动画效果

$("#spnMove").slideToggle("slow", animateIt);

}

</script>

12． 内容过滤选择器

根据元素中的文字内容或所包含的子元素特征获取元素

<script type="text/javascript">

$(function(){ //显示包含给定文本的元素

$("div:contains('A')").css("display","block");

})

$(function(){ //显示所有不包含子元素或者文本的空元素

$("div:empty").css("display","block");

})

$(function(){ //显示含有选择器所匹配的元素的元素

$("div:has(span)").css("display","block");

})

$(function(){ //显示含有子元素或者文本的元素

$("div:parent").css("display","block");

})

</script>

1. 可见性过滤选择器

可见性过滤选择器根据元素是否可见的特征获取元素。

<script type="text/javascript">

$(function(){ //增加所有可见元素类别

$("span:hidden").show();

$("div:visible").addClass("GetFocus");

})

$(function(){ //增加所有不可见元素类别

$("span:hidden").show().addClass("GetFocus");

})

</script>

1. 属性过滤选择器

<script type="text/javascript">

$(function(){ //显示所有含有id属性的元素

$("div[id]").show(3000);

})

$(function(){ //显示所有属性title的值是"A"的元素

$("div[title='A']").show(3000);

})

$(function(){ //显示所有属性title的值不是"A"的元素

$("div[title!='A']").show(3000);

})

$(function(){ //显示所有属性title的值以"A"开始的元素

$("div[title^='A']").show(3000);

})

$(function(){ //显示所有属性title的值以"C"结束的元素

$("div[title$='C']").show(3000);

})

$(function(){ //显示所有属性title的值中含有"B"的元素

$("div[title\*='B']").show(3000);

})

$(function(){ //显示所有属性title的值中含有"B"且属性id的值是"divAB"的元素

$("div[id='divAB'][title\*='B']").show(3000);

})

</script>

1. 子元素过滤选择器

<script type="text/javascript">

$(function(){ //增加每个父元素下的第2个子元素类别

$("li:nth-child(2)").addClass("GetFocus");

})

$(function(){ //增加每个父元素下的第一个子元素类别

$("li:first-child").addClass("GetFocus");

})

$(function(){ //增加每个父元素下的最后一个子元素类别

$("li:last-child").addClass("GetFocus");

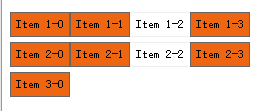
})

$(function(){ //增加每个父元素下只有一个子元素类别

$("li:only-child").addClass("GetFocus");

})

</script>



注意是增加当前元素的父元素的下的子元素。比如，当前是li，父元素是ul，ul的下面是li。

:nth-child(eq|even|odd|index)

1. 表单对象属性过滤选择器

表单对象属性过滤选择器通过表单中的某对象属性特征获取该元素，如enable、disable、checked、selected

<script type="text/javascript">

$(function(){ //增加表单中所有属性为可用的元素类别

$("#divA").show(3000);

$("#form1 input:enabled").addClass("GetFocus");

})

$(function(){ //增加表单中所有属性为不可用的元素类别

$("#divA").show(3000);

$("#form1 input:disabled").addClass("GetFocus");

})

$(function(){ //增加表单中所有被选中的元素类别

$("#divB").show(3000);

$("#form1 input:checked").addClass("GetFocus");

})

$(function(){ //显示表单中所有被选中option的元素内容

$("#divC").show(3000);

$("#Span2").html("选中项是："+$("select option:selected").text());

})

</script>

</head>

<body>

<form id="form1" style="width:241px">

<div id="divA">

<input type="text" value="可用文本框" class="clsIpt" />

<input type="text" disabled="disabled" value="不可用文本框" class="clsIpt" />

</div>

<div id="divB">

<input type="checkbox" checked="checked" value="1" /> 选中

<input type="checkbox" value="0" /> 未选中

</div>

<div id="divC">

<select multiple="multiple">

<option value="0">Item 0</option>

<option value="1" selected="selected">Item 1</option>

<option value="2">Item 2</option>

<option value="3" selected="selected">Item 3</option>

</select>

<span id="Span2"></span>

</div>

</form>

</body>

1. jquery中的show方法

$(*selector*).show(*speed*,*callback*)

如果元素被隐藏，则显示这些元素，可以参数可选，规定元素从隐藏到完全可见的速度。



1. 表单选择器

<script type="text/javascript">

$(function(){ //显示Input类型元素的总数量

$("#form1 div").html("表单共找出 Input 类型元素:"+$("#form1 :input").length);

$("#form1 div").addClass("div");

})

$(function(){ //显示所有文本框对象

$("#form1 :text").show(3000);

})

$(function(){ //显示所有密码框对象

$("#form1 :password").show(3000);

})

$(function(){ //显示所有单选按钮对象

$("#form1 :radio").show(3000);

$("#form1 #Span1").show(3000);

})

$(function(){ //显示所有复选框对象

$("#form1 :checkbox").show(3000);

$("#form1 #Span2").show(3000);

})

$(function(){ //显示所有提交按钮对象

$("#form1 :submit").show(3000);

})

$(function(){ //显示所有图片域对象

$("#form1 :image").show(3000);

})

$(function(){ //显示所有重置按钮对象

$("#form1 :reset").show(3000);

})

$(function(){ //显示所有按钮对象

$("#form1 :button").show(3000);

})

$(function(){ //显示所有文件域对象

$("#form1 :file").show(3000);

})

</script>

</head>

<body>

<form id="form1">

<textarea class="txt"> TextArea</textarea>

<select><option value="0"> Item 0</option></select>

<input type="text" value="Text" class="txt"/>

<input type="password" value="PassWord" class="txt"/>

<input type="radio" /><span id="Span1"> Radio</span>

<input type="checkbox" /><span id="Span2"> CheckBox</span>

<input type="submit" value="Submit" class="btn"/>

<input type="image" title="Image" src="Images/logo.gif" class="img"/>

<input type="reset" value="Reset" class="btn"/>

<input type="button" value="Button" class="btn"/>

<input type="file" title="File" class="txt"/>

<div id="divShow"></div>

</form>

</body>

2018年3月15日22:22:55

1. jquery操作DOM

元素属性操作

获取元素属性的语法格式：

attr（name）

例：var strAlt=$(“img”).attr(“src”)

设置元素属性：

attr（key，value）

$("img").attr("src", "Images/img01.jpg");//设置src属性

$("img").attr("title", "这是一幅风景画");//设置title属性

$("img").attr({ src: "Images/img02.jpg", title: "这是一幅风景画" });//同时设置二个属性

$("img").addClass("clsImg");//增加样式

$("span").html("加载完毕");//显示加载状态

attr()方法还可以绑定一个function()函数，通过该函数返回的值作为元素的属性值，格式如下：  
attr(key,function())

使用function随机返回一张图片

<script type="text/javascript">

$(function() {

$("img").attr("src", function() { return "Images/img0" + Math.floor(Math.random() \* 2 + 1) + ".jpg" }); //设置src属性

$("img").attr("title", "这是一幅风景画"); //设置title属性

$("img").addClass("clsImg"); //增加样式

})

</script>

1. 获取和设置元素

在jquery中，操作元素内容的方法包括html()和text()

<script type="text/javascript">

$(function() {

var strHTML = $("#divShow").html();//获取HTML内容

var strText = $("#divShow").text();//获取文本内容

$("#divHTML").html(strHTML);//设置HTML内容

$("#divText").text(strText);//设置文本内容

})

</script>

</head>

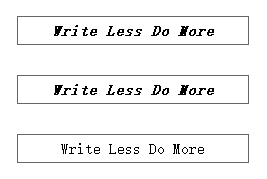
<body>

<div id="divShow"><b><i>Write Less Do More</i></b></div>

<div id="divHTML"></div>

<div id="divText"></div>

</body>



2018年3月15日23:41:51

1. $(this).next('ul').slideUp();

在层次选择器里面有个next()方法可以选择紧邻的下一个元素，pre()选择前一个元素。

这里还可以设置选择的类型，这里参数ul表示是列表。

1. $('dd').find('ul').slideUp();

find表示选择dd标签下的所有ul标签。