

jQuery

最流行的JavaScript程序库

jQuery的用途

- 访问和操作DOM元素
- 控制页面样式
- 对页面样式进行处理
- 扩展新的jQuery插件
- 与Ajax技术完美结合

* DOM : 文件对象模型 (Document Object Model) 是一种理念 , 一种思想 , 一种使 Web 开发人员可以访问HTML元素的功能 , 不是具体的方法。

* Ajax : 异步JavaScript和XML (Asynchronous Javascript And XML) , 是指一种创建交互式网页应用的网页开发技术 ; 是一种用于创建快速动态网页的技术 ; 是一种在无需重新加载整个网页的情况下 , 能够更新部分网页的技术。

jQuery的优势

- 体积小 , 压缩后只有 100KB 左右
- 强度的选择器
- 出色的 DOM 封装
- 可靠的事件处理机制
- 出色的浏览器兼容性
- 使用隐式迭代简化编程
- 丰富的插件支持

jQuery使用

- 引入jQuery库文件

直接在HTML文件 body 标签内容的地步调用即可

```
<script type="text/javascript" src="plug/jquery-1.11.0.js"></script>
```

- 使用jQuery

1. 使用jQuery弹出提示框

```
<script>
    $(document).ready(function(){
        alert("Hello jQuery");
    });
</script>
```

2. 语法

`$(选择器).方法名();`

- `$()` :工厂函数，将DOM对象转化为jQuery对象
- 选择器：需要操作的DOM元素
- 方法名()：jQuery中提供的方法，包括操作元素和事件。

3. DOM元素的操作

`var $input = $('#userName');` 表示获取id为userName的jQuery对象

4. 将DOM对象转换为jQuery对象

```
var userName = document.getElementById('userName'); //获得DOM对象
var $userName = $(userName); //获得jQuery对象
```

5. 通过选择器直接获得jQuery对象

`var $userName = $('#userName');` //直接获得id为userName的jQuery对象

6. 通过jQuery对象获得DOM对象

`var $userName = $('#userName');` //直接获得id为userName的jQuery对象

1. `var(userName = $userName[0]);` //通过数组方式获得
2. `var(userName = $userName.get(0));` //通过get方式获得

jQuery选择器

1. 基本选择器

名称	语法构成	描述	示例
标签选择器	element	根据给定的标签名匹配元素	<code>\$("h2")</code> 选取所有h2元素
类选择器	.class	根据给定的class匹配元素	<code>\$(".title")</code> 选取所有class为title的元素
ID选择器	#id	根据给定的id匹配元素	<code>\$("#title")</code> 选取id为title的元素
并集选择器	selector1,selector2,...,selectorN	将每一个选择器匹配的元素合并后一起返回	<code>\$(".div,p,.title")</code> 选取所有div、p和拥有class为title的元素
交集选择器	element.class或element#id	匹配指定class或id的某元素或元素集合	<code>\$(".h2.title")</code> 选取所有拥有class为title的h2元素
全局选择器	*	匹配所有元素	<code>\$("*")</code> 选取所有元素

2. 层次选择器

名称	语法构成	描述	示例
后代选择器	ancestor descendant	选取ancestor元素里的所有descendant（后代）元素	<code>\$("#menu span")</code> 选取#menu下的 元素
子选择器	parent>child	选取parent元素下的child（子）元素	<code>\$("#menu>span")</code> 选取#menu的子元素
相邻元素选择器	prev+next	选取紧邻prev元素之后的next元素	<code>\$(".h2+dl")</code> 选取紧邻 <h2>元素之后的同辈元素<dl></dl></h2>
同辈元素选择器	prev~siblings	选取prev元素之后的所有siblings元素	<code>\$(".h2~dl")</code> 选取 <h2>元素之后所有的同辈元素<dl></dl></h2>

3. 属性过滤选择器

名称	语法构成	描述	示例
属性过滤选择器	[attribute]	选取包含给定属性的元素	<code>\$("[href]")</code> 选取含有href属性的元素
	[attribute=value]	选取等于给定属性是某个特定值的元素	<code>\$("[href ='#']")</code> 选取href属性值为“#”的元素
	[attribute!=value]	选取不等于给定属性是某个特定值的元素	<code>\$("[href != '#']")</code> 选取href属性值不为“#”的元素
	[attribute^=value]	选取给定属性是以某些特定值开始的元素	<code>\$("[href^='en']")</code> 选取href属性值以en开头的元素
	[attribute\$=value]	选取给定属性是以某些特定值结尾的元素	<code>\$("[href\$='jpg']")</code> 选取href属性值以.jpg结尾的元素
	[attribute*=value]	选取给定属性是以包含某些值的元素	<code>\$("[href*='txt']")</code> 选取href属性值中含有txt的元素
	[selector] [selector2] [selectorN]	选取满足多个条件的复合属性的元素	<code>\$("li[id][title=新闻要点]")</code> 选取含有id属性和title属性为新闻要点的 元素

4. 基本过滤选择器

名称	语法构成	描述	示例
基本过滤选择器	:first	选取第一个元素	<code>\$(" li:first")</code> 选取所有元素中的第一个元素
	:last	选取最后一个元素	<code>\$(" li:last")</code> 选取所有元素中的最后一个元素
	:even	选取索引是偶数的所有元素 (index从0开始)	<code>\$(" li:even")</code> 选取索引是偶数的所有元素
	:odd	选取索引是奇数的所有元素 (index从0开始)	<code>\$(" li:odd")</code> 选取索引是奇数的所有元素
	:eq(index)	选取索引等于index的元素 (index从0开始)	<code>\$("li:eq(1)")</code> 选取索引等于1的元素
	:gt(index)	选取索引大于index的元素 (index从0开始)	<code>\$(" li:gt(1)")</code> 选取索引大于1的元素 (注: 大于1, 不包括1)
	:lt(index)	选取索引小于index的元素 (index从0开始)	<code>\$("li:lt(1)")</code> 选取索引小于1的元素 (注: 小于1, 不包括1)

5. 可见性过滤选择器

名称	语法构成	描述	示例
可见性过滤选择器	:visible	选取所有可见的元素	<code>\$(":visible")</code> 选取所有可见的元素
	:hidden	选取所有隐藏的元素	<code>\$(":hidden")</code> 选取所有隐藏的元素

6. 表单过滤选择器

名称	语法构成	描述	示例
表单过滤选择器	:input	匹配所有 input, textarea, select 和 button 元素	<code>\$(":input")</code> 选取所有的表单元素
	:checked	匹配所有选中的被选中元素(复选框、单选框等, 不包括select中的option)	<code>\$("[type=checkbox]:checked")</code> 选取所有选中的复选框元素
	:selected	匹配所有选中的option元素	<code>\$("select:selected")</code> 匹配所有选中的下拉框
	:enabled	匹配所有可用元素	<code>\$(":enabled")</code>
	:disabled	匹配所有不可用元素	<code>\$("button:disabled")</code>