

## 1. jQuery概述

jQuery可以实现写的更少而做的更多。它是一个由js函数组成的函数库。jQuery的本质也是js。

## 2. jQuery版本支持

jQuery分为很多版本, 还分为未压缩版和压缩版, 根据需要选择对应的版本进行下载!

1.x 支持常用的浏览器和IE6+

2.x 支持常用的浏览器和IE9+

3.x 支持常用的浏览器和IE9+

**注意: jQuery不兼容老版本.** 因为jQuery升级除了会做一些内部优化之外, 还会删除以前的一些代码, 比如删除一些方法. 或者是添加一些新的方法. 所以在升级之后, 以前的代码可能会无法执行。

## 3. jQuery的优势

- a. 可以通过css选择器选中页面中的元素
- b. 动态创建HTML元素
- c. 便捷的获取文档就绪事件
- d. 便捷的修改css样式

## 4. jQuery的引入

在head标签下书写一个script标签, 通过src属性引入外部的jQuery函数库。

```
<!--引入jQuery函数库-->  
<script type="text/javascript" src="jquery-1.4.2.js"></script>
```

## 5. jQuery语法

引入jQuery函数库后, 在页面中就可以使用\$来操作jQuery对象。它的实际操作方式为jQuery(""), 书写上可以把它简化\$("")

jQuery <==> \$("")

### a. 操作css选择器

\$("#test").text("abc"); //表示将页面id为test的元素选中, 并且设置其中的text文本内容。

### b. 文档就绪事件

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

});

< === >

\$().ready(function(){

});

```
< === >
$(function(){
});
```

## 6. js对象和jQuery对象相互转换

### a. js对象转换为jQuery对象:

将js对象使用\$()包起来，这样js对象就会变为一个jQuery对象，这时只能使用jQuery对象身上的api。

### b. jQuery对象转换为js对象:

jQuery选中的元素可以看做是将这些元素放入一个数组中，根据下标获取某一个元素，就获取数组中的一个js对象。

```
$("div")[0];
$("div").get(0);
```

代码测试:

```
//1.js转换为jQuery
/*$(function(){
    var div = document.getElementById("test");//js对象
    //转换为jQuery
    alert($(div).text());
});*/

//2.jQuery转换为js
$(function(){
    $("div");//jQuery对象
    var d = $("div")[0];//转换为js
    //$("div").get(0)//转换为js
    alert(d.innerText);
});
```

## 7. 定义一个变量

```
$a = $("div");
```

```
$b = document.getElementsByTagName("div");
```

使用\$xxx定义一个变量只是为增强代码可读性，并不是只能在jQuery对象身上使用这样的对象。js同样可以。

## 1. 基本选择器 ( selector案例一 )

元素名选择器, 类选择器, id选择器, 多元素选择器, \*选择器

### a. 元素名选择器

`$("div")` – 匹配所有的div元素

### b. class选择器

`$(".c1")` – 匹配所有class值为c1 的元素

`$("div.c1")` – 匹配所有class值为c1的div元素

### c. id选择器

`$("#d1")` – 匹配所有id值为d1的元素

`$("div#d1")` – 匹配所有id值为d1的div元素

### d. \*号匹配符

`$("*")` – 匹配所有的元素

### e. 多元素选择器

`$("div,span,#d1,.c1")` – 匹配所有的div/span元素以及id值为d1的元素和class值为c1的元素。。。

## 2. 层级选择器 ( selector案例二 )

如果想通过DOM元素之间的层次关系来获取特定元素。例如子元素、兄弟元素等。则需要通过层次选择器。

`$("div span")` – 匹配div下所有的span元素

`$("div>span")` – 匹配div下所有的span子元素

`$("div+span")` – 匹配div后面紧邻的span兄弟元素

`$("div~span")` – 匹配div后面所有的span兄弟元素

## 3. 基本过滤选择器 ( selector案例三 )

过滤选择器主要是通过特定的过滤规则来筛选出所需的DOM元素, 该选择器都以 ":" 开头

`$("div:first")` – 匹配所有div中的第一个div元素

`$("div:last")` – 匹配所有div中的最后一个div元素

`$("div:even")` – 匹配所有div中索引值为偶数的div元素, 0开始

`$("div:odd")` – 匹配所有div中索引值为奇数的div元素, 0开始

`$("div:eq(n)")` – 匹配所有div中索引值为n的div元素, 0开始

`$("div:lt(n)")` – 匹配所有div中索引值小于n的div元素, 0开始

`$("div:gt(n)")` – 匹配所有div中索引值大于n的div元素, 0开始

`$("div:not(.one)")` – 匹配所有class值不为one的div元素

## 4. 内容选择器 ( selector案例四 )

`$("div:contains('abc')")` – 匹配所有div中包含abc内容的div元素

如: `<div>xxxabcxx</div>`

`$("div:has(p)")` – 匹配所有包含p元素的div元素

如: `<div> <p> </p> </div>`

`$("div:empty")` – 匹配所有内容为空的div元素

如: `<div> </div>`

`$("div:parent")` – 匹配所有内容不为空的div元素

如: `<div>xxxxx</div>`

## 5. 可见选择器

`$("div:hidden")` – 匹配所有隐藏的div元素

`$("div:visible")` – 匹配所有可见的div元素

## 6. 属性选择器 ( selector案例五 )

`$("div[id]")` – 匹配所有具有id属性的div元素

`$("div[id='d1']")` – 匹配所有具有id属性并且值为d1的div元素

`$("div[id!='d1']")` – 匹配所有id属性值不为d1的div元素

`$("div[id^='d1']")` – 匹配所有id属性值以d1开头的div元素

`$("div[id$='d1']")` – 匹配所有id属性值以d1结尾的div元素

## 7. 子元素选择器

`$("div:nth-child(n)")` – n从1开始, 匹配div中第n个子元素。

`$("div:first-child")` – 匹配div中第1个子元素。

`$("div:last-child")` – 匹配div中最后一个子元素。。。

## 8. 表单选择器 ( selector案例六 )

`$(":input")` – 匹配所有的input文本框、密码框、单选框、复选框、select框、textarea、button。

`$(":password")` – 匹配所有的密码框

`$(":radio")` – 匹配所有的单选框

`$(":checkbox")` – 匹配所有的复选框

`$(":checked")` – 匹配所有的被选中的单选框/复选框/option

`$("input:checked")` – 匹配所有的被选中的单选框/复选框

`$(":selected")` – 匹配所有被选中的option选项

## 1. 文档操作

parent()

`$("#d1").parent()` – 获取id为d1元素的父元素

`<div>`

`<div id="d1">`

parents()

`$("#d1").parents()` – 获取id为d1元素的祖先元素

`$("#d1").parents("tr")` – 获取id为d1元素的tr祖先元素

next()

`$("div").next()` – 获取所匹配元素后面紧邻的兄弟元素

`$("div").next("span")` – 获取所匹配元素后面紧邻的span兄弟元素

nextAll()

`$("div").nextAll()` – 获取所匹配元素后面所有的兄弟元素

`$("div").nextAll("span")` – 获取所匹配元素后面所有的span兄弟元素

prev()

`$("div").prev()` – 获取所匹配元素前面紧邻的兄弟元素

`$("div").prev("span")` – 获取所匹配元素前面紧邻的span兄弟元素

prevAll()

`$("div").prevAll()` – 获取所匹配元素前面所有的兄弟元素

`$("div").prevAll("span")` – 获取所匹配元素前面所有的span兄弟元素

siblings()

`$("div").siblings()` – 获取所匹配元素前后所有的兄弟元素

`$("div").siblings("span")` – 获取所匹配元素前后所有的span兄弟元素

append()

`$("div").append("<span> </span>")` – 为所匹配元素追加一个span子元素

remove()

`$("div").remove()` – 删除所匹配元素

html()

\$("#div").html() – 获取所匹配元素的html内容

\$("#div").html("xxx") – 为所匹配元素设置html内容

text()

\$("#div").text() – 获取所匹配元素的文本内容

\$("#div").text("xxx") – 为所匹配元素设置文本内容

attr()

\$("#div").attr("id") – 获取所匹配元素的id属性值

\$("#div").attr("id", "d2") – 为所匹配元素设置id属性

css

\$("#div").css("width") – 获取所匹配元素的width样式属性值

\$("#div").css("width", "200px") – 为所匹配元素设置width样式属性

\$("#div").css({ "width": "200px", "color": "red", "font-size": "24px" }); – 为所匹配元素设置width样式属性

## 2. 事件

click()

\$("#div").click(function() {}) – 为所匹配元素绑定点击事件

blur()

\$("#input").blur(function() {}) – 为所匹配元素绑定失去输入焦点事件

focus()

\$("#input").focus(function() {}) – 为所匹配元素绑定获得输入焦点事件

change()

\$("#select").change(function() {}) – 为所匹配元素绑定选项切换事件

ready()

\$(document).ready(function() {}) – 文档就绪事件

其作用相当于: window.onload = function() {}

简写形式为:

`$(function){}` – 在整个文档加载完成后立即执行

### 3. 效果

`show()`

`$("#div").show()` – 将隐藏元素设置为显示(底层操作的是display);

`hide()`

`$("#div").hide()` – 将显示元素设置为隐藏(底层操作的是display);

`toggle()`

`$("#div").toggle()` – 切换元素的可见状态, 如果元素显示则设置为隐藏, 如果元素隐藏则设置为可见.

### ( Exercise案例 )