

今日内容

1. JQuery 基础:

1. 概念
2. 快速入门
3. JQuery对象和JS对象区别与转换
4. 选择器
5. DOM操作
6. 案例

JQuery 基础:

1. 概念: 一个JavaScript框架。简化JS开发

- * jQuery是一个快速、简洁的JavaScript框架，是继Prototype之后又一个优秀的JavaScript代码库（或JavaScript框架）
 - * jQuery设计的宗旨是“**write Less, Do More**”，即倡导写更少的代码，做更多的事情。
 - * 它封装JavaScript常用的功能代码，提供一种简便的JavaScript设计模式，优化HTML文档操作、事件处理、动画设计和Ajax交互。
- * JavaScript框架：本质上就是一些js文件，封装了js的原生代码而已

2. 快速入门

1. 步骤:

1. 下载JQuery

- * 目前jQuery有三个大版本：

1.x: 兼容ie678，使用最为广泛的，官方只做BUG维护，
功能不再新增。因此一般项目来说，使用1.x版本就可以了，
最终版本: 1.12.4 (2016年5月20日)

2.x: 不兼容ie678，很少有人使用，官方只做BUG维护，
功能不再新增。如果不考虑兼容低版本的浏览器可以使用2.x，
最终版本: 2.2.4 (2016年5月20日)

3.x: 不兼容ie678，只支持最新的浏览器。除非特殊要求，
一般不会使用3.x版本的，很多老的jQuery插件不支持这个版本。
目前该版本是官方主要更新维护的版本。最新版本: 3.2.1 (2017年3月20日)

- * jquery-xxx.js 与 jquery-xxx.min.js区别：

1. jquery-xxx.js: 开发版本。给程序员看的，有良好的缩进和注释。体积大一些
2. jquery-xxx.min.js: 生产版本。程序中使用，没有缩进。体积小一些。程序加载更快

2. 导入JQuery的js文件：导入min.js文件

3. 使用

```
var div1 = $("#div1");
alert(div1.html());
```

3. JQuery对象和JS对象区别与转换

1. JQuery对象在操作时，更加方便。
2. JQuery对象和js对象方法不通用的。
3. 两者相互转换

* jq --> js : jq对象[索引] 或者 jq对象.get(索引)
* js --> jq : \$(js对象)

4. 选择器：筛选具有相似特征的元素(标签)

1. 基本操作学习：

1. 事件绑定

```
//1. 获取b1按钮
$("#b1").click(function(){
    alert("abc");
});
```

2. 入口函数

```
$(function () {
});
```

window.onload 和 \$(function) 区别

- * window.onload 只能定义一次，如果定义多次，后边的会将前边的覆盖掉
- * \$(function)可以定义多次的。

3. 样式控制：css方法

```
// $("#div1").css("background-color","red");
$("#div1").css("backgroundColor","pink");
```

2. 分类

1. 基本选择器

1. 标签选择器（元素选择器）

- * 语法： \$("html标签名") 获得所有匹配标签名称的元素

2. id选择器

- * 语法： \$("#id的属性值") 获得与指定id属性值匹配的元素

3. 类选择器

- * 语法： \$(".class的属性值") 获得与指定的class属性值匹配的元素

4. 并集选择器：

- * 语法： \$("选择器1,选择器2....") 获取多个选择器选中的所有元素

2. 层级选择器

1. 后代选择器

- * 语法： \$("A B ") 选择A元素内部的所有B元素

2. 子选择器

- * 语法： \$("A > B") 选择A元素内部的所有B子元素

3. 属性选择器

1. 属性名称选择器

- * 语法： \$("A[属性名]") 包含指定属性的选择器

2. 属性选择器

- * 语法： \$("A[属性名='值']") 包含指定属性等于指定值的选择器

3. 复合属性选择器

- * 语法： \$("A[属性名='值'][...]) 包含多个属性条件的选择器

4. 过滤选择器

1. 首元素选择器

- * 语法： :first 获得选择的元素中的第一个元素

2. 尾元素选择器

- * 语法： :last 获得选择的元素中的最后一个元素

3. 非元素选择器

- * 语法： :not(selector) 不包括指定内容的元素

4. 偶数选择器

- * 语法： :even 偶数，从 0 开始计数

5. 奇数选择器

- * 语法： :odd 奇数，从 0 开始计数

- 6. 等于索引选择器
 - * 语法: `:eq(index)` 指定索引元素
- 7. 大于索引选择器
 - * 语法: `:gt(index)` 大于指定索引元素
- 8. 小于索引选择器
 - * 语法: `:lt(index)` 小于指定索引元素
- 9. 标题选择器
 - * 语法: `:header` 获得标题 (`h1~h6`) 元素, 固定写法
- 5. 表单过滤选择器
 - 1. 可用元素选择器
 - * 语法: `:enabled` 获得可用元素
 - 2. 不可用元素选择器
 - * 语法: `:disabled` 获得不可用元素
 - 3. 选中选择器
 - * 语法: `:checked` 获得单选/复选框选中的元素
 - 4. 选中选择器
 - * 语法: `:selected` 获得下拉框选中的元素
- 5. DOM操作
 - 1. 内容操作
 - 1. `html()`: 获取/设置元素的标签体内容 `<a>内容` --> `内容`
 - 2. `text()`: 获取/设置元素的标签体纯文本内容 `<a>内容` --> 内容
 - 3. `val()`: 获取/设置元素的`value`属性值
 - 2. 属性操作
 - 1. `attr()`: 获取/设置元素的属性
 - 2. `removeAttr()`: 删除属性
 - 3. `prop()`: 获取/设置元素的属性
 - 4. `removeProp()`: 删除属性
 - * `attr`和`prop`区别?
 - 1. 如果操作的是元素的固有属性, 则建议使用`prop`
 - 2. 如果操作的是元素自定义的属性, 则建议使用`attr`
 - 2. 对`class`属性操作
 - 1. `addClass()`: 添加`class`属性值
 - 2. `removeClass()`: 删除`class`属性值
 - 3. `toggleClass()`: 切换`class`属性
 - * `toggleClass("one")`:
 - * 判断如果元素对象上存在`class="one"`, 则将属性值`one`删除掉。 如果元素对象上不存在`class="one"`
 - 4. `css()`:
 - 3. CRUD操作:
 - 1. `append()`: 父元素将子元素追加到末尾
 - * 对象`1.append(对象2)`: 将对象`2`添加到对象`1`元素内部, 并且在末尾
 - 2. `prepend()`: 父元素将子元素追加到开头
 - * 对象`1.prepend(对象2)`: 将对象`2`添加到对象`1`元素内部, 并且在开头
 - 3. `appendTo()`:
 - * 对象`1.appendTo(对象2)`: 将对象`1`添加到对象`2`内部, 并且在末尾
 - 4. `prependTo()`:
 - * 对象`1.prependTo(对象2)`: 将对象`1`添加到对象`2`内部, 并且在开头
 - 5. `after()`: 添加元素到元素后边
 - * 对象`1.after(对象2)`: 将对象`2`添加到对象`1`后边。对象`1`和对象`2`是兄弟关系

6. `before()`:添加元素到元素前边

* 对象`1.before(对象2)`: 将对象2添加到对象1前边。对象1和对象2是兄弟关系

7. `insertAfter()`

* 对象`1.insertAfter(对象2)`: 将对象2添加到对象1后边。对象1和对象2是兄弟关系

8. `insertBefore()`

* 对象`1.insertBefore(对象2)`: 将对象2添加到对象1前边。对象1和对象2是兄弟关系

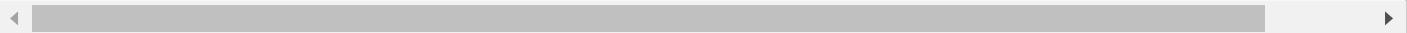
9. `remove()`:移除元素

* 对象`.remove()`:将对象删除掉

10. `empty()`:清空元素的所有后代元素。

* 对象`.empty()`:将对象的后代元素全部清空，但是保留当前对象以及其属性节点

6. 案例



今日内容：

1. JQuery 高级

1. 动画
2. 遍历
3. 事件绑定
4. 案例
5. 插件

JQuery 高级

1. 动画

1. 三种方式显示和隐藏元素

1. 默认显示和隐藏方式

1. `show([speed],[easing],[fn])`

1. 参数:

1. `speed`: 动画的速度。三个预定义的值("slow", "normal", "fast")或表示动画时长的毫秒数值(如: 1000)

2. `easing`: 用来指定切换效果, 默认是"swing", 可用参数"linear"

* `swing`: 动画执行时效果是 先慢, 中间快, 最后又慢

* `linear`: 动画执行时速度是匀速的

3. `fn`: 在动画完成时执行的函数, 每个元素执行一次。

2. `hide([speed],[easing],[fn])`

3. `toggle([speed],[easing],[fn])`

2. 滑动显示和隐藏方式

1. `slideDown([speed],[easing],[fn])`

2. `slideUp([speed],[easing],[fn])`

3. `slideToggle([speed],[easing],[fn])`

3. 淡入淡出显示和隐藏方式

1. `fadeIn([speed],[easing],[fn])`

2. `fadeOut([speed],[easing],[fn])`

3. `fadeToggle([speed],[easing],[fn])`

2. 遍历

1. js的遍历方式

* `for`(初始化值;循环结束条件;步长)

2. jq的遍历方式

1. `jq对象.each(callback)`

1. 语法:

`jquery对象.each(function(index,element){});`

* `index`: 就是元素在集合中的索引

* `element`: 就是集合中的每一个元素对象

* `this`: 集合中的每一个元素对象

2. 回调函数返回值:

* `true`: 如果当前`function`返回为`false`, 则结束循环(`break`)。

* `false`: 如果当前`function`返回为`true`, 则结束本次循环, 继续下次循环(`continue`)

```
2. $.each(object, [callback])
3. for..of: jquery 3.0 版本之后提供的方式
   for(元素对象 of 容器对象)
```

3. 事件绑定

1. jquery标准的绑定方式

- * jq对象.事件方法(回调函数);
- * 注: 如果调用事件方法, 不传递回调函数, 则会触发浏览器默认行为。
- * 表单对象.submit();//让表单提交

2. on绑定事件/off解除绑定

- * jq对象.on("事件名称",回调函数)
- * jq对象.off("事件名称")
- * 如果off方法不传递任何参数, 则将组件上的所有事件全部解绑

3. 事件切换: toggle

- * jq对象.toggle(fn1,fn2...)
- * 当单击jq对象对应的组件后, 会执行fn1.第二次点击会执行fn2.....

* 注意: 1.9版本 .toggle() 方法删除, **jQuery Migrate** (迁移) 插件可以恢复此功能。

```
<script src="../js/jquery-migrate-1.0.0.js" type="text/javascript" charset="utf-8"></script>
```

4. 案例

1. 广告显示和隐藏

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>广告的自动显示与隐藏</title>
<style>
#content{width:100%;height:500px;background:#999}
</style>

<!--引入jquery-->
<script type="text/javascript" src="../js/jquery-3.3.1.min.js"></script>
<script>
/*
需求:
1. 当页面加载完, 3秒后。自动显示广告
2. 广告显示5秒后, 自动消失。

```

分析:

1. 使用定时器来完成。setTimeout (执行一次定时器)
2. 分析发现JQuery的显示和隐藏动画效果其实就是控制display
3. 使用 show/hide方法来完成广告的显示

```
*/
```

```
//入口函数, 在页面加载完成之后, 定义定时器, 调用这两个方法
$(function () {
    //定义定时器, 调用adShow方法 3秒后执行一次
    setTimeout(adShow,3000);
    //定义定时器, 调用adHide方法, 8秒后执行一次
    setTimeout(adHide,8000);
});
```

```

//显示广告
function adShow() {
    //获取广告div，调用显示方法
    $("#ad").show("slow");
}

//隐藏广告
function adHide() {
    //获取广告div，调用隐藏方法
    $("#ad").hide("slow");
}

</script>
</head>
<body>
<!-- 整体的DIV -->
<div>
    <!-- 广告DIV -->
    <div id="ad" style="display: none;">
        
    </div>

    <!-- 下方正文部分 -->
    <div id="content">
        正文部分
    </div>
</div>
</body>
</html>

```

2. 抽奖

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>jquery案例之抽奖</title>
    <script type="text/javascript" src="../js/jquery-3.3.1.min.js"></script>

    <script language='javascript' type='text/javascript'>

```

/*

分析：

1. 给开始按钮绑定单击事件
 - 1.1 定义循环定时器
 - 1.2 切换小相框的src属性
 - * 定义数组，存放图片资源路径
 - * 生成随机数。数组索引

2. 给结束按钮绑定单击事件

1.1 停止定时器

1.2 给大相框设置src属性

```
/*
var imgs = ["../img/man00.jpg",
            "../img/man01.jpg",
            "../img/man02.jpg",
            "../img/man03.jpg",
            "../img/man04.jpg",
            "../img/man05.jpg",
            "../img/man06.jpg",
];
var startId;//开始定时器的id
var index;//随机角标
$(function () {
    //处理按钮是否可以使用的效果
    $("#startID").prop("disabled",false);
    $("#stopID").prop("disabled",true);

    //1. 给开始按钮绑定单击事件
    $("#startID").click(function () {
        // 1.1 定义循环定时器 20毫秒执行一次
        startId = setInterval(function () {
            //处理按钮是否可以使用的效果
            $("#startID").prop("disabled",true);
            $("#stopID").prop("disabled",false);

            //1.2生成随机角标 0-6
            index = Math.floor(Math.random() * 7); //0.000--0.999 --> * 7 --> 0.0-----6.99
            //1.3设置小相框的src属性
            $("#img1ID").prop("src",imgs[index]);

        },20);
    });

    //2. 给结束按钮绑定单击事件
    $("#stopID").click(function () {
        //处理按钮是否可以使用的效果
        $("#startID").prop("disabled",false);
        $("#stopID").prop("disabled",true);

        // 1.1 停止定时器
        clearInterval(startId);
        // 1.2 给大相框设置src属性
        $("#img2ID").prop("src",imgs[index]).hide();
        //显示1秒之后
        $("#img2ID").show(1000);
    });
});
```

```
</script>

</head>
<body>




</div>




</div>


<input
    id="startID"
    type="button"
    value="点击开始"
    style="width:150px; height:150px; font-size:22px">


<input
    id="stopID"
    type="button"
    value="点击停止"
    style="width:150px; height:150px; font-size:22px">

</body>
</html>


```

5. 插件：增强JQuery的功能

1. 实现方式：

1. `$.fn.extend(object)`
* 增强通过Jquery获取的对象的功能 `$("#id")`
2. `$.extend(object)`
* 增强JQeury对象自身的功能 `$/jQuery`