

jQuery 学习目标

- ☐ 能够引入jQuery
- ☐ 能够掌握JQ和JS对象的转换
- ☐ 能够掌握JQ中事件的使用
- ☐ 能够掌握JQ中常用的选择器
- ☐ 能够使用JQ操作样式
- ☐ 能够使用JQ操作属性
- ☐ 能够使用JQ操作DOM
- ☐ 能够掌握JQ遍历

第一章-JQ知识点

知识点-JQ介绍

1.目标

- ☐ 知道什么是JQ

2.路径

1. jQuery的概述
2. jQuery的作用
3. jQuery框架的下载
4. jQuery的版本

3.讲解

3.1 jQuery的概述

jQuery是一个**优秀的javascript库**(类似Java里面的jar包)，兼容css3和各大浏览器，提供了dom、events、animate、ajax等简易的操作。并且jquery的插件非常丰富，大多数功能都有相应的插件解决方案。jquery的宗旨是 write less, do more。

简而言之：jQuery是一个js库【工具包】，使用jQuery可以简化js操作html

3.2 jQuery的作用

jQuery最主要的作用是简化js操作HTML

3.3 jQuery框架的下载

jQuery的官方下载地址: <http://www.jquery.com>

3.4 jQuery的版本

- 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, 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。
- 注:开发版本与生产版本, 命名为jQuery-x.x.x.js为开发版本, 命名为jQuery-x.x.x.min.js为生产版本, 开发版本源码格式良好, 有代码缩进和代码注释, 方便开发人员查看源码, 但体积稍大。而生产版本没有代码缩进和注释, 且去掉了换行和空行, 不方便发人员查看源码, 但体积很小

4.小结

1. jQuery是一个js库, 解决了浏览器的兼容性问题, 简化了js操作html
2. jQuery版本: 开发中主要使用1.x
 1. 压缩版: 80多kb 体积小 适合用于生产环境
 2. 未压缩版: 200多kb 体积大 有良好的代码格式 方便开发人员阅读源码和学习

案例-JQ快速入门

1.需求

页面一打开, 就加载"Hello jQuery"

2.步骤

1. 项目中添加jQuery js文件
2. 页面中引入jQuery js文件
3. 编写jQuery代码

3.实现

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <script src="js/jquery-3.3.1.min.js"></script>
</head>
<body>
  <h1>Hello sz118</h1>
  <!--
    1.将jQuery的js文件引入项目中
    2.将jQuery的js引入到要使用的页面
    3.使用jQuery
  -->
  <script>
```

```

/*
    $:jQuery $(document).ready() = onload
    js加载: onload="函数()"
    jQuery:$(document).ready(函数)
*/
/*$(document).ready(function () {
    alert("Hello jQuery");
});*/

//简易方式
$(function () {
    alert("Hello jQuery");
});
</script>
</body>
</html>

```

4.小结

1. 步骤:

1. 将jQuery的js文件添加到项目中
2. 将jQuery的js文件引入到要使用的html页面中

```
<script src="js/jquery-3.3.1.min.js"></script>
```

3. 使用jQuery

2. 注意:

✖ Uncaught ReferenceError: \$ is not defined
at 00_jQuery快速入门.html?_...d4o5a2uuc180nj1h:26

未定义\$, 表示jQuery文件在页面中没有引入!

知识点-JQ和JS对象转换【重点】

1.目标

☐ 掌握JQ和JS对象转换

2.分析

2.1对象说明

js对象: 使用js方式【document.getElementXxx()】获取的对象是js对象, js对象只能调用js的属性和方法。

jQuery对象: 使用jQuery方式【\$(选择器)】获取的对象是jQuery对象, jQuery对象只能调用jQuery中的方法。

问题: 如何向使用js对象调用jQuery方法, 使用jQuery对象调用js属性, 该怎么办?

解决: 需要进行对象的转换

jQuery本质上虽然也是js，但如果使用jQuery的属性和方法那么必须保证对象是jQuery对象而不是js方式获得的DOM对象。使用js方式获取的对象是js的DOM对象，使用jQuery方式获取的对象是jQuery对象。两者的转换关系如下

2.2转换语法

js的DOM对象转换成jQuery对象， **语法：\$(js对象)**

jQuery对象转换成js对象， **语法：jquery对象[索引]** 或 jquery对象.get(索引); 一般索引写0

3.讲解

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <script src="js/jquery-3.3.1.min.js"></script>
</head>
<body>
  <!--需求：点击按钮改变span标签中的文本内容-->
  <span id="spanId">Hello World</span><br>
  <button onclick="fun01()">改变内容【js转jQuery】</button><br>
  <button onclick="fun02()">改变内容【jQuery转js】</button><br>

  <script>
    function fun01() {
      //使用js对象js方式改变
      //1.获取span标签对象
      var span = document.getElementById("spanId");
      //2.设置span标签体内容
      //span.innerText = "Hello JQuery";

      //2.将js对象转为jQuery对象 使用jQuery对象的方式改变 语法：jq对象 = $(js对象);
      //变量命名随便起 我这里加$是为了表示该对象是jQuery对象 方便和js对象区分
      var $span = $(span);
      $span.text("Hello SZ118");
    }

    function fun02() {
      //1.获取span标签对象 使用jQuery的方式获取
      var $span = $("#spanId");
      //2.jquery对象转js对象 语法：js对象=jq对象[下标] 或 jq对象.get(下标) 下标从0开始
      var span = $span[0];
      //3.通过js对象使用js方式改变内容
      span.innerText = "Hello JS"
    }
  </script>
</body>
</html>
```

4.小结

1. js对象：使用js方式【document.getElementById()】获取的对象是js对象，js对象只能调用js的属性和方法。
2. jQuery对象：使用jQuery方式【\$(选择器)】获取的对象是jQuery对象，jQuery对象只能调用jQuery中的方法。
3. 为什么需要js对象和jQuery转换，原因就是js对象只能调用js的属性和方法，jQuery对象只能调用jQuery的方法
4. js对象转jQuery对象：jQuery对象 = \$(js对象);
5. jQuery对象转js对象：js对象 = jQuery对象[下标] ; 或 jQuery对象.get(下标);

知识点-JQ中事件的使用

1.目标

- ☐ 掌握jq中事件的使用

2.路径

1. 基本事件的使用
2. jQuery的事件绑定与解绑
3. 事件切换【链式写法】

3.讲解

3.1 基本事件的使用【重点】

- 事件在jq里面都封装成了方法. 去掉了JS里面.on 语法:

语法：jq对象.事件方法名(function(){ //处理 });

eg: 点击事件

```
btn.click(function(){});
```

- 点击事件

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <script src="js/jquery-3.3.1.min.js"></script>
</head>
<body>
  <button id="btn01">点击</button><br>
  <script>
    //基本事件使用语法：jq对象.事件方法( function(){事件处理} );
    $("#btn01").click(function () {
      alert("点我很开心! ");
    });
  </script>
</body>
</html>
```

- 获得焦点和失去焦点

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <script src="js/jquery-3.3.1.min.js"></script>
</head>
<body>
  name: <input type="text" id="name"><br>

  <script>

    //需求：输入框获得焦点和失去事件  在使用焦点时获取输入框输入内容
    $("#name").focus(function(){
      console.log("获得了焦点");
    });
    //失去焦点 获取输入框内容
    $("#name").blur(function () {
      //this表示的就是当前操作的标签对象  是一个js对象
      console.log("失去了焦点: "+this.value);
    });

  </script>
</body>
</html>
```

- 内容改变事件

```
<body>
  <select name="province" id="province">
    <option value="河南">河南</option>
    <option value="河北">河北</option>
    <option value="广东">广东</option>
    <option value="广西">广西</option>
  </select><br>

  <script>

    /*需求3：下拉列表内容改变事件*/
    $("#province").change(function () {
      console.log("内容改变了: "+this.value);
    });

  </script>
</body>
```

- 鼠标相关的事件

```

<body>
  <div id="divId" style="width: 200px;height: 200px;border: 1px solid black"></div>

  <script>
    /*需求4: 当鼠标移入div 背景颜色显示为红色 移出 背景颜色显示为绿色*/
    $("#divId").mouseover(function () {
      this.style.backgroundColor = "red";
    });
    $("#divId").mouseout(function () {
      this.style.backgroundColor = "green";
    });
  </script>
</body>

```

- 键盘相关事件

```

<body>
  <input type="text" name="name" id="name"><br>

  <script>
    /*需求5: 键盘按下松开事件*/
    $("#username").keydown(function () {
      console.log("键盘按下");
    });
    $("#username").keyup(function () {
      console.log("键盘松开");
    });
  </script>
</body>

```

语法: jq对象.事件方法(function(){事件处理代码});

3.2jQuery的事件绑定与解绑 【了解】

- 事件的绑定

jQuery对象.on(事件名称, function(){});

其中: 事件名称是jQuery的事件方法的方法名称, 例如: click、mouseover、mouseout、focus、blur等

eg: 点击事件

```

-- 基本使用   jq对象.click(function(){} )
-- 绑定发送   jq对象.on("click",function(){});

```

- 事件的解绑

jQuery对象.off(事件名称);

其中: 参数事件名称如果省略不写, 可以解绑该jQuery对象上的所有事件

- 实例代码

```

<body>
  <div id="divId" style="width: 200px;height: 200px;border: 1px solid black"></div>

  <button id="btn01">事件绑定</button>
  <button id="btn02">事件解绑</button>
  <script>

```

/*需求1：为div绑定点击事件 当点击按钮btn01时，为div绑定点击事件 绑定完成后 点击div 弹出事件绑定成功*/

```
$("#btn01").click(function () {  
    //事件处理 就是为div绑定点击事件  
    $("#divId").on("click",function () {  
        alert("事件绑定成功");  
    });  
});
```

/*需求2：为div解绑点击事件 当点击按钮btn02时，为div解绑点击事件 解绑完成后 点击div 将没有任何反应*/

```
$("#btn02").click(function () {  
    //事件处理 就是为div解绑点击事件  
    $("#divId").off("click");  
});
```

</script>

</body>

3.3事件切换

- 普通写法

```
<body>  
    <div id="divId" style="width: 200px;height: 200px;border: 1px solid black"></div>
```

<script>

/*需求：当鼠标移入div，其背景颜色为红色 移出 背景颜色为绿色*/

```
$("#divId").mouseover(function () {  
    this.style.backgroundColor = "red";  
});  
$("#divId").mouseout(function () {  
    this.style.backgroundColor = "green";  
});
```

</script>

</body>

- 链式写法

```
<body>  
    <div id="divId" style="width: 200px;height: 200px;border: 1px solid black"></div>
```

<script>

//jQuery支持链式写法

```
$("#divId").mouseover(function () {  
    this.style.backgroundColor="red";  
}).mouseout(function () {  
    this.style.backgroundColor="green";  
});
```

</script>

</body>

4.小结

1. 基本事件使用【重点】

1. 语法：jq对象.事件方法(function(){ 事件处理 });
2. 掌握的事件：click、focus、blur、change、mouseover、mouseout、keydown、keyup

2. 事件绑定和解绑【了解】

1. 绑定: jq对象.on("事件方法名称",function({}));
2. 解绑: jq对象.off("事件方法名称");
3. 链式写法 操作jq对象后返回依然是之前操作的对象

知识点-JQ动画【了解】

1.目标

- ☐ 了解JQ的动画

2.路径

1. 基本效果
2. 滑动效果
3. 淡入淡出效果

3.讲解

3.1基本效果

- 方法

方法名称	解释
show([speed],[easing],[fn])	显示元素方法
hide([speed],[easing],[fn])	隐藏元素方法
toggle([speed],[easing],[fn])	切换元素方法, 显示的使之隐藏, 隐藏的使之显示

- 参数

参数名称	解释
speed	三种预定速度之一的字符串("slow","normal", or "fast")或表示动画时长的毫秒数值(如: 1000)
easing	用来指定切换效果, 默认是"swing", 可用参数"linear"
fn	在动画完成时执行的函数, 每个元素执行一次

- 实例

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <script src="../../js/jquery-3.3.1.min.js"></script>
</head>
<body>
  <br>
  <button id="btn01">显示【show】</button>
```

```

<button id="btn02">隐藏【hide】</button>
<button id="btn03">切换【toggle】</button><br>
<button id="btn04">显示【slideDown】</button>
<button id="btn05">隐藏【slideUp】</button>
<button id="btn06">切换【slideToggle】</button><br>
<button id="btn07">显示【fadeIn】</button>
<button id="btn08">隐藏【fadeOut】</button>
<button id="btn09">切换【fadeToggle】</button><br>

<script>
    $("#btn01").click(function () {
        $("#mm").show(3000,"linear",function () {
            alert(" 图片显示完成! ");
        });
    });
    $("#btn02").click(function () {
        $("#mm").hide();
    });
    $("#btn03").click(function () {
        $("#mm").toggle();
    });

    $("#btn04").click(function () {
        $("#mm").slideDown(3000);
    });
    $("#btn05").click(function () {
        $("#mm").slideUp(3000);
    });
    $("#btn06").click(function () {
        $("#mm").slideToggle();
    });

    $("#btn07").click(function () {
        $("#mm").fadeIn(3000);
    });
    $("#btn08").click(function () {
        $("#mm").fadeOut(3000);
    });
    $("#btn09").click(function () {
        $("#mm").fadeToggle();
    });
</script>
</body>
</html>

```

3.2滑动效果

- 方法

方法名称	解释
slideDown([speed],[easing],[fn])	向下滑动方法
slideUp([speed],[easing],[fn])	向上滑动方法
slideToggle([speed],[easing],[fn])	切换元素方法，显示的使之隐藏，隐藏的使之显示

- 参数

参数名称	解释
speed	三种预定速度之一的字符串("slow","normal", or "fast")或表示动画时长的毫秒数值(如: 1000)
easing	用来指定切换效果, 默认是"swing", 可用参数"linear"
fn	在动画完成时执行的函数, 每个元素执行一次

3.3 淡入淡出效果

- 方法

方法名称	解释
fadeIn([speed],[easing],[fn])	淡入显示方法
fadeOut([speed],[easing],[fn])	淡出隐藏方法
fadeToggle([speed],[easing],[fn])	切换元素方法, 显示的使之隐藏, 隐藏的使之显示

- 参数

参数名称	解释
speed	三种预定速度之一的字符串("slow","normal", or "fast")或表示动画时长的毫秒数值(如: 1000)
easing	用来指定切换效果, 默认是"swing", 可用参数"linear"
fn	在动画完成时执行的函数, 每个元素执行一次

4.小结

1. 了解就行, 对着帮助文档玩一下

知识点-JQ选择器【重点】

1.目标

- ☐ 掌握JQ常用的选择器

2.路径

- 1. 基本选择器
- 2. 层级选择器
- 3. 属性选择器
- 4. 基本过滤选择器
- 5. 表单属性选择器

3.讲解

3.1.基本选择器【重点】

- 语法

选择器名称	语法	解释
标签选择器（元素选择器）	\$("html标签名")	获得所有匹配标签名称的元素
id选择器	\$("#id的属性值")	获得与指定id属性值匹配的元素
类选择器	\$(".class的属性值")	获得与指定的class属性值匹配的元素

3.2 层级选择器

- 语法

选择器名称	语法	解释
后代选择器（子子孙孙）	\$("A B ")	选择A元素内部的所有B元素
子选择器(儿子)	\$("A > B")	选择A元素内部的所有B子元素
兄弟选择器(弟弟)	\$("A + B")	获得A元素同级的下一个B元素
相邻选择器(所有弟弟)	\$("A ~ B")	获得A元素同级的所有B元素

3 属性选择器

- 语法

选择器名称	语法	解释
属性选择器	\$("A[属性名]")	包含指定属性的选择器
属性选择器	\$("A[属性名=值]")	包含指定属性等于指定值的选择器

4.基本过滤选择器

- 语法

选择器名称	语法	解释
首元素选择器	:first	获得选择的元素中的第一个元素
尾元素选择器	:last	获得选择的元素中的最后一个元素
非元素选择器	:not(selector)	不包括指定内容的元素
偶数选择器	:even	偶数，从 0 开始计数
奇数选择器	:odd	奇数，从 0 开始计数
等于索引选择器	:eq(index)	指定索引元素
大于索引选择器	:gt(index)	大于指定索引元素
小于索引选择器	:lt(index)	小于指定索引元素

5.表单属性选择器

- 语法

选择器名称	语法	解释
可用元素选择器	:enabled	获得可用元素
不可用元素选择器	:disabled	获得不可用元素
选中选择器	:checked	获得单选/复选框选中的元素
选中选择器	:selected	获得下拉框选中的元素

4.小结

1. 基本选择器【重点】
 - 标签选择器：\$("标签名") 选择指定的标签名称的元素 获取的标签集合
 - id选择器：\$("#id属性值") 选择指的的id属性值的元素 获取的是唯一
 - 类选择器：\$(".class属性值") 选择指定的class属性值的元素 获取的标签集合
2. 属性选择器、层级选择器、基本过滤选择器 根据帮助文档会使用即可

知识点-JQ操作样式

1.目标

- ☐ 掌握JQ操作样式

2.讲解

API方法	解释
css(name)	获取CSS样式
css(name,value)	设置CSS样式
addClass(className)	添加CSS类样式
removeClass(className)	移除CSS类样式

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <script src="../js/jquery-3.3.1.min.js"></script>
  <style>
    .color{
      background-color: red;
    }
  </style>
</head>
<body>
  <div id="divId" style="width: 200px;height: 200px;border: 1px solid black"></div>
  <button id="btn01">获取样式</button>
  <button id="btn02">设置样式</button>

  <script>
    //通过css()方法设置获取的实际上是行内样式
    /*需求： 点击btn01获取div的指定css样式  点击btn02设置div的css样式*/
    $("#btn01").click(function () {
      console.log("width: "+ $("#divId").css("width"));
    });

    $("#btn02").click(function () {
      //$("#divId").css("width", "500px");
      $("#divId").css({"width": "500px", "height": "500px"});
    });

    //需求： 当鼠标移入div时 设置背景颜色为红色，移出恢复原样
    $("#divId").mouseover(function () {
      $(this).addClass("color");
    }).mouseout(function () {
      $(this).removeClass("color");
    });
  </script>
</body>
</html>
```

3.小结

1. 获取样式: jq对象.css(name);
2. 设置样式: jq对象.css(name,value);
3. 添加类样式: jq对象.addClass(className);
4. 移除类样式: jq对象.removeClass(className);

知识点-JQ操作属性

1.目标

- ☐ 掌握JQ操作属性

2.讲解

API方法	解释
attr(name,[value])	获得/设置属性的值
prop(name,[value])	获得/设置属性的值(checked, selected)

- attr与prop的注意问题
 - attr与prop是以1.6为界限
 - checked 和 selected 建议 使用prop获取
 - 其他使用attr获取

3.小结

1. 获取元素属性值: attr("属性名") 或 prop("属性名")
2. 设置元素属性值: attr("属性名","属性值") 或 prop("属性名","属性值")
3. **一般获取或设置选中状态 使用prop()方法**

知识点-使用JQ操作DOM

1.目标

- ☐ 掌握使用JQ操作DOM

2.路径

1. jQuery对DOM树中的文本和值进行操作
2. jQuery创建,插入节点
3. jQuery移除节点

3.讲解

3.1jQuery对DOM树中的文本和值进行操作

- API

API方法	解释
val([value]) js中value	获得/设置输入框的值
text([value]) js中innerText	获得/设置元素的文本内容 不会解析html标签
html([value]) js中innerHTML	获得/设置元素的标签体内容 会解析html标签

- 解释

val() 获得标签里面value属性的值 value属性的封装
val("...") 给标签value属性设置值

html() 获得标签的内容，如果有标签，一并获得。
html("...") 设置html代码，如果有标签，将进行解析。支持标签 封装了innerHTML

text() 获得标签的内容，如果有标签，忽略标签。
text("...") 设置文本，如果含有标签，把标签当做字符串。不支持标签

- val()和val(...)
- text()和html()
- text(...)和html(...)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <script src="js/jquery-3.3.1.min.js"></script>
</head>
<body>
  <span>hello world</span>
  <input type="text" id="name" name="name" value="zs"> <br>
  <button id="btn01">获取文件和价值</button>
  <button id="btn02">设置文本和价值</button>
  <br>

  <!--
    val():获取或设置输入框的值
    text():获取或设置标签体中的文本 不会解析html标签
    html():获取或设置标签体中的文本 会解析html标签
    使用技巧:
      不传参表示获取，传递参数表示设置
  -->
  <script>

    $("#btn01").click(function () {
      //获取输入框的值 等价于js中的value
      var value = $("#name").val();
      console.log(value);
```



```

        //获取标签体中的文本
        console.log($("#span").text());
        console.log($("#span").html());
    });

    $("#btn02").click(function () {
        //设置输入框的值
        $("#name").val("张三是个大坏蛋");
        //设置标签体中的文本
        //$("#span").text("<font color='red'>Hello SZ118</font>");
        //设置标签体中的文本 html()
        $("#span").html("<font color='red'>Hello SZ118</font>");
    });
</script>
</body>
</html>

```

3.2jQuery创建,插入

- API

API方法	解释
\$("#A")	创建A元素对象
append(element)	添加成最后一个子元素，两者之间是父子关系
prepend(element)	添加成第一个子元素，两者之间是父子关系
appendTo(element)	添加到父元素的内部最后面 B.appendTo(A) = A.append(B)
prependTo(element)	添加到父元素的内部最前面 B.prependTo(A) = A.prepend(B)
before(element)	添加到当前元素的前面，两者之间是兄弟关系
after(element)	添加到当前元素的后面，两者之间是兄弟关系

- 内部插入:父节点/子节点之间操作

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <script src="js/jquery-3.3.1.min.js"></script>
</head>
<body>

    <ul>
        <li>猪八戒</li>
    </ul>

    <script>
        //1.创建元素
        var li1 = $("<li>孙悟空</li>");
        var li2 = $("<li>沙悟净</li>");
        var p1 = $("<p>唐三藏</p>");
        var p2 = $("<p>白龙马</p>");
    </script>

```

```

//2.元素插入
//2.1: 内部插入 A.prepend(B):表示将B插入到A的内部最前面
$("#u1").prepend(li1);
//2.2: 内部插入 A.append(B):表示将B插入A的内部最后面
$("#u1").append(li2);

//3.外部插入
//3.1: 外部插入 A.before(B):表示将B插入到A的前面 A和B同级
$("#u1").before(p1);
//3.2: 外部插入 A.after(B):表示将B插入到A的后面 A和B同级
$("#u1").after(p2)

//4.移除元素
//$("#u1").remove();
$("#u1").empty();

</script>
</body>
</html>

```

- 外部插入(了解): 兄弟节点之间操作

```

after    a.after(c);      哥哥找弟弟 八戒找沙师弟
        <a></a><c></c>
before   a.before(c);   弟弟找哥哥 八戒找孙悟空
        <c></c><a></a>

```

3.3jQuery移除节点(对象)【了解】

- API

API方法	解释
remove()	删除指定元素(自己移除自己)
empty()	清空指定元素的所有子元素(清空元素内容)

4.小结

4.1操作值和内容

- 操作输入框的值
 - val(): 获取输入框的值
 - val(...): 设置输入框的值
- 操作标签体的文本
 - text(): 获取或设置标签体的文本 不会解析html标签
 - html(): 获取或设置标签体的文本 会解析html标签

4.2创建和添加节点

- 创建节点 `$("A")`
- 插入节点
 - 内部插入: `prepend()`、`append()`
 - 外部插入: `before()`、`after()`

4.3移除节点

- `remove`: 自杀
- `empty`: 清空节点标签体中的内容

知识点-JQ遍历【重点】

1.目标

- ☐ 掌握JQ遍历

2.路径

1. 复习JS方式遍历
2. jquery对象方法遍历
3. jquery的全局方法遍历
4. jQuery3.0新特性for of语句遍历

3.讲解

3.1 复习JS方式遍历

- 语法

```
for(var i=0;i<元素数组.length;i++){  
    元素数组[i];  
}
```

- eg

```
var array = [ "张三", "李四", "王五" ];  
//js方式遍历  
for(i=0;i<array.length;i++){  
    console.log(array[i]);  
}
```

3.2 jquery对象方法遍历【重点】

- 语法

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

其中:(参数名字随便取的)

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

- eg

```
//方式一: jq对象.each(function(index,element){}); 两个参数名称可以随便起 【推荐使用】
console.log("jQuery方式一遍历: ");
$(array).each(function(index,element){
    console.log(index+" --> "+element);
});
```

3.3 jquery的全局方法遍历

- 语法

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

其中,

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

- eg

```
//方式二: $.each(jq对象,function(index,element){});
console.log("jQuery方式二遍历: ");
$.each($(array),function (index,element) {
    console.log(index+" --> "+element);
})
```

3.4 jQuery3.0新特性for of语句遍历

- 语法

```
for(变量 of jquery对象){
    变量;
}
```

其中,

变量: 定义变量依次接受jquery数组中的每一个元素

jquery对象: 要被遍历的jquery对象

- eg

```
//方式三: jQuery3.0新特性 for(变量 of jq对象){}
console.log("jQuery方式三遍历: ");
for(element of array){
    console.log(element);
}
```

4.小结

1. jq对象方法遍历【重点】

```
jq对象.each(function(i,ele){

});
```

参数1: 下标

参数2: 下标对应的元素

参数名字随便取

2. 全局方法遍历

```
$.each(jq对象,function(i,ele){  
  
});
```

参数1: 下标

参数2: 下标对应的值

参数名字随便取

3. jq3.0新特性

```
for(ele of jq对象){  
  
}
```

第二章-JQ案例

案例一:使用JQuery完成页面定时弹出广告

一,需求分析



- 进入页面3s后弹出广告,3s后广告隐藏

二,思路分析

1. 在页面头部添加一个div, 作为广告展示区域, 并且先隐藏

```
<!--广告展示div-->
<div id="divId" style="width: 100%;height: 100%;display: none">
  
</div>
```

2. 设置定时函数 `setTimeout(3000,"showAd()")`
3. 编写`showAd()`函数

```
function showAd(){
  //1.显示广告
  //2.3s之后广告隐藏
  setTimeout(3000,"hideAd()")
}
```

4. 编写`hideAd()`函数

三,代码实现

```
<div id="divId" style="width: 100%;height: 100%;display: none">
  
</div>
</div>
<script>
  //1.设置定时函数
  setTimeout("showAd()",3000);
  //2.编写showAd函数
  function showAd() {
    /*//1.显示广告
    $("#divId").show(3000);
    //2.3s之后广告隐藏
    setTimeout("hideAd()",6000);*/

    //3000表示显示广告图片需要3000ms, function表示图片显示完成之后做某个事情
    $("#divId").show(3000,function () {
      setTimeout("hideAd()",3000);
    });
  }
  //3.编写hideAd函数
  function hideAd() {
    $("#divId").hide();
  }
</script>
```

四,小结

1. 页面头部创建div, 作为广告显示区域
2. 设置定时任务`setTimeout()`, 页面打开后3秒显示广告图片
3. 编写`showAd()`函数
4. 编写`hideAd()`函数

案例二:使用jQuery完成表格的隔行换色

一,需求分析

<input type="checkbox"/>	商品名称	商品价格	商品数量	操作
<input type="checkbox"/>	MAC	18000	10	删除 修改
<input type="checkbox"/>	MAC	18000	10	删除 修改
<input type="checkbox"/>	MAC	18000	10	删除 修改
<input type="checkbox"/>	MAC	18000	10	删除 修改

二,思路分析

1. 使用基本过滤选择器，通过:even设置偶数行背景颜色为蓝色blue
2. 使用基本过滤选择器，通过:odd设置奇数行背景颜色为粉色pink

三,代码实现

```
<script>
    //- 筛选出奇数的tr， 设置一个背景色
    $("tr:odd").css("backgroundColor","pink");
    //- 筛选出偶数的tr， 设置一个背景色
    $("tr:even").css("backgroundColor","blue");
</script>
```

四,小结

1. 基本过滤选择器(:even :odd)
2. 操作样式 css(css属性,css属性值)

案例三:使用JQuery完成复选框的全选效果

一,需求分析

<input checked="" type="checkbox"/>	商品名称	商品价格	商品数量	操作
<input checked="" type="checkbox"/>	Mac	18000	10	删除 修改
<input checked="" type="checkbox"/>	Mac	18000	10	删除 修改
<input checked="" type="checkbox"/>	Mac	18000	10	删除 修改
<input checked="" type="checkbox"/>	Mac	18000	10	删除 修改

二.思路分析

1. 准备页面
2. 获取到父复选框的选中状态【注意：给父复选框设置一个id="all"】
3. 将父复选框的选中状态设置给子复选框【注意：给子复选框设置一个class="itemSelect"】
4. 给父复选框设置点击事件

```
$("#all").click(function(){  
    //1.获取父复选框的选中状态 使用prop()  
    //2.将父复选框的选中状态赋值给子复选框 使用prop()  
});
```

三,代码实现

- js代码

```
<script>  
    $("#all").click(function(){  
        //1.获取父复选框的选中状态 使用prop()  
        var checked = $(this).prop("checked");  
        //2.将父复选框的选中状态赋值给子复选框 使用prop()  
        $(".itemSelect").prop("checked",checked);  
    });  
</script>
```

四,小结

1. 触发点：为全选复选框设置点击事件
2. 实现：获取父复选框的选中状态 **赋值** 给子复选框

案例四:使用JQuery完成省市联动效果

一,需求分析

籍贯:

-请选择-

-请选择-

深圳

广州

东莞

惠州

二,思路分析

1. 准备页面, 准备两个select下拉列表, 分别存放省市数据
2. 准备一个二维数据, 存放省份对应城市数据

```
var citys = [ ["郑州", "信阳", "洛阳", "南阳", "开封"], ["广州", "深圳", "珠海", "中山", "韶关"],  
["武汉", "襄阳", "宜昌", "黄冈", "仙桃"] ];
```

3. 为省份下拉列表绑定内容改变事件

```
$("#province").change(function(){  
    //1. 获取省份下拉列表的值value  
    //2. 根据省份列表的value去二维数组中获取对应的城市数据  
    //3. 遍历城市数据 【1. 创建option元素 2. 插入到城市下拉列表中】  
});
```

三,代码实现

- js代码

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>Title</title>  
    <script src="../../js/jquery-3.3.1.min.js"></script>  
</head>  
<body>  
    籍贯: <select name="province" id="province">  
        <option value="-1">请选择</option>  
        <option value="0">河南</option>  
        <option value="1">广东</option>  
        <option value="2">湖北</option>  
    </select>  
    <select name="city" id="city">  
        <option value="-1">请选择</option>  
    </select>  
  
    <script>  
        var citys = [ ["郑州", "信阳", "洛阳", "南阳", "开封"], ["广州", "深圳", "珠海", "中山", "韶  
关"], ["武汉", "襄阳", "宜昌", "黄冈", "仙桃"] ];  
        $("#province").change(function () {  
            //0. 当每次选择省份改变之后需要清空城市下拉列表内容  
            //$("#city").empty();  
            $("#city").html("<option>请选择</option>");  
  
            //1. 获取省份下拉列表的值value  
            var value = $(this).val();  
            //2. 根据省份列表的value去二维数组中获取对应的城市数据  
            var cityes = citys[value];  
            //3. 遍历城市数据 【1. 创建option元素 2. 插入到城市下拉列表中】  
            $(cityes).each(function(index, element){  
                var option = $("<option>"+element+"</option>");  
                $("#city").append(option);  
            });  
        });  
    </script>
```

```
</body>
</html>
```

四.小结

1. 内容改变事件 change()
2. 遍历 jq对象.each(function(i,ele){})
3. 操作dom a.append(c)
4. 思路
 - 给省份的select设置内容改变事件
 - 获得省份的值
 - 根据省份的值 获得对应的城市的数据
 - 遍历城市的数据, 创建option标签元素
 - 插入到城市的select里面

扩展案例_表格换色

1.需求

<input type="checkbox"/>	商品名称	商品价格	商品数量	操作
<input type="checkbox"/>	Mac	18000	10	删除修改
<input type="checkbox"/>	Mac	18000	10	删除修改
<input type="checkbox"/>	Mac	18000	10	删除修改
<input type="checkbox"/>	Mac	18000	10	删除修改
<input type="checkbox"/>	Mac	18000	10	删除修改

- 鼠标进入当前行tr的时候背景颜色变为red
- 鼠标离开当前行tr的时候背景颜色变为原来的颜色

2.技术

- 选择器【基本过滤选择器 :even :odd】
- 事件(mouseover() mouseout())
- css()

3.思路

1. 给tr设置鼠标移入事件

```
$( "tr" ).mouseover( function() {
    //1. 获取当前行之前的颜色
    //2. 设置当前行背景颜色为红色
});
```

2. 给tr设置鼠标移出事件

```
$("#tr").mouseout(function(){  
    //1.设置当前行背景颜色为原来的颜色  
});
```

4.代码

```
<script>  
var oldColor;  
$("#tr").mouseover(function(){  
    //1.获取当前行之前的颜色  
    oldColor = $(this).css("backgroundColor");  
    //2.设置当前行背景颜色为红色  
    $(this).css("backgroundColor", "red");  
}).mouseout(function () {  
    //3.移出设置当前行的颜色为原来的颜色  
    $(this).css("backgroundColor", oldColor);  
});  
</script>
```

扩展案例_电子时钟

1.需求

2019/12/17 16:06:10

2.思路

1. 准备页面，创建一个标签（h1、p、span）用于存放当前时间
2. 设置定时函数，每隔1s重新获取一下时间

```
setInterval("myClock()", 1000);  
  
function myClock(){  
    //1.获取当前时间  
    //2.将当前时间设置到标签体中  
}
```

3.实现

```

<h1 id="curTime"></h1>
<script>
    // 2. 设置定时函数，间隔1s调用一次
    setInterval("clock()",1000);
    //3.编写时钟函数 获取当前时间
    function clock() {
        //3.1:获取当前时间
        var dateStr = new Date().toLocaleString();
        //3.2:将当前时间放入到h1标签中显示
        $("#curTime").text(dateStr);
    }
</script>

```

本章小结

1. jQuery概念: jQuery是js的一个库，类似于java中的jar包，封装了js中的属性方法，解决了一些浏览器兼容性问题，简化js操作html。
2. jQuery入门

```

<html>
  <head>

  </head>
  <body>

    <script src="jQuery的js文件路径"></script>
    <script>
        //jQuery代码 操作Html
    </script>
  </body>
</html>

```

3. jQuery报错: js文件没有引入

```

✖ Uncaught ReferenceError: $ is not defined
    at myClock (02_电子时钟.html?_ijt=g4...rdfhsfqv02birvhk:17)
    at <anonymous>:1:1

```

4. **jQuery对象和js对象转换**

js对象: 使用js方式【document.getElementById()】获取的对象，称为js对象，只能调用js中的属性和方法。

jQuery对象: 使用jQuery方式【\$("选择器")】获取到的对象，称为jQuery对象，只能调用jQuery中的方法。

js对象-->jQuery对象: var jq对象=\$(js对象);

jQuery对象-->js对象: var js对象 = jq对象[下标]; 或者 var js对象=jq对象.get(下标);

5. jQuery事件的使用

1. **基本事件使用 jq对象.事件方法名(function(){ //处理 });**
2. 基本事件: click、focus、blur、change、mouseover、mouseout、keydown、keyup
3. 事件的绑定和解绑
 1. 绑定: jq对象.on("事件方法名",function({});

2. 解绑: jq对象.off("事件方法名");
4. 事件的切换【jQuery链式写法】
 1. jq对象.mouseover().mouseout();
6. jQuery动画【了解】
 1. show|hide|toggle
 2. slideDown|slideUp|slideToggle
 3. fadeIn|fadeOut|fadeToggle
7. jQuery选择器
 1. **基本选择器【标签选择器、id选择器、类选择器】**
 2. 层次选择器【后代选择器、子选择器、兄弟选择器、同辈选择器】
 3. 属性选择器
 4. 基本过滤选择器【:first :last :even :odd :eq(下标)】
8. jQuery操作样式
 1. css(name): 获取指定的css样式属性值
 2. css(name,value): 设置css样式
 3. addClass(类样式名称): 设置类样式
 4. removeClass(类样式名称): 移除类样式
9. jQuery操作属性
 1. attr(name)|prop(name): 获取指定属性
 2. attr(name,value)|prop(name,value): 设置属性
 3. 单选框复选框以及下拉列表的选中状态属性的获取和设置一般使用prop()
10. jQuery操作标签文本和值
 1. val(): 获取或设置输入框的值
 2. text(): 获取或设置标签体中的文本 不会解析html标签
 3. html(): 获取或设置标签体中的html内容 会解析html标签
11. jQuery的DOM操作
 1. 创建节点: var node = \$("标签内容");
 2. 插入节点:
 1. 内部插入
 1. A.prepend(B): 在A中插入B, 将B插入到最前面
 2. A.append(B): 在A中插入B, 将B插入到最后面
 2. 外部插入
 1. A.before(B): 将B插入到A的前面 同级插入
 2. A.after(B): 将B插入到A的后面 同级插入
 3. 元素移除
 1. remove(): 删除当前元素
 2. empty: 清空当前元素内容
12. jQuery遍历
 1. **jq对象方法遍历: jq对象.each(function(index,element){});**
 2. jq全局方法遍历: \$.each(jq对象,function(index,element){});
 3. jq3.0新特性: for(迭代变量 of jq对象){}
13. 练习:
 1. jq和js对象转换
 2. jQuery基本事件使用
 3. jQuery基本选择器
 4. jQuery对象方法遍历
 5. 案例...

