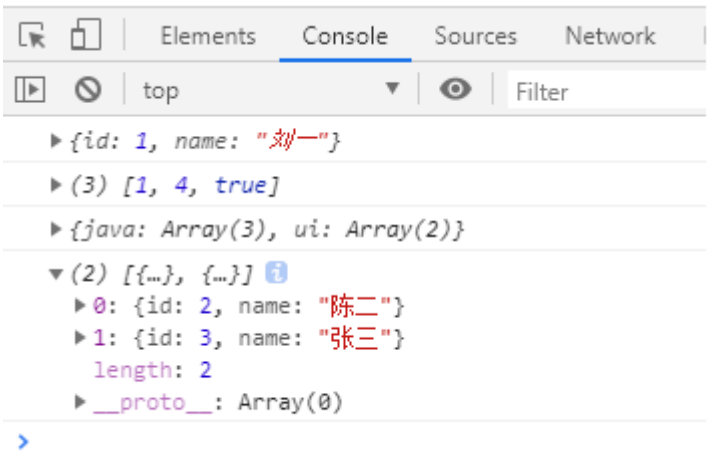
JavaScript高级

今日内容

> 1. 掌握JSON对象的使用
> 2. 掌握JS的定时器对象的使用
> 3. 会使用dom对象获取元素对象
> 4. 会使用dom对象操作元素内容
> 5. 会使用dom对象操作元素属性
> 6. 会使用dom对象操作css样式
> 7. 能够完成JS综合案例

第一章 JSON数据

1.1-JSON基础语法




```
<!DOCTYPE html>
<html lang="zh">
<head>
  <meta charset="UTF-8">
  <title>JSON基础语法</title>
</head>
<body>
<!--
JSON 语法规则
    数据:{属性:值,属性:值}
    键值:键值之间是冒号,属性值与属性值之间是逗号
    对象:用{}表示
    数组:用[]表示
```

```
JSON对象使用中常见的具体形式
    1. 对象类型
        格式 {属性:值,属性:值}
    2. 数组/集合类型
        格式 [值1,值2,值3]
    3. 混合类型
        对象中属性的值是数组
            {属性:[值1,值2,值3],属性:[值1,值2,值3]}
        数组中的值是对象
            [{属性:值,属性:值},{属性:值,属性:值}]

-->

<script>
        //1. JS中的 Object有两种格式
    // object形式
    let stu = new Object();//创建一个js对象,js对象的属性想要直接加上
    stu.id = 1;
    stu.name = "刘一";
    stu.age = 18;
    console.log(stu);//{id: 1, name: "刘一", age: 18}
    console.log(typeof stu);//object
    //JSON(JavaScript Object Notation) JS对象符号:js对象的表示形式
    // {属性:值,属性:值}  后来又称为了一种数据格式
    //属性:字符串类型
    //值:任意
    //json形式
    let per = {id:2,name:"陈二",age:16};
    console.log(per);//{id:2,name:"陈二",age:16}
    console.log(typeof per);//object


  // 2. 数组/集合类型
  let arr = [1,4,true];
  console.log(arr);//[1,4,true]


  // 3. 混合类型
  let school = {java:[118,119,120],ui:[100,101]};
  console.log(school);//{java:[118,119,120],ui:[100,101]}
  let students = [{id:2,name:"陈二"},{id:3,name:"张三"}];
  console.log(students);//[{id:2,name:"陈二"},{id:3,name:"张三"}]
</script>
</body>
</html>
```

1.2-JSON格式转换


```
<!DOCTYPE html>
<html lang="zh">
<head>
  <meta charset="UTF-8">
  <title>JSON格式转换</title>
</head>
<body>
<!--
JSON对象与字符串转换的相关函数
    1. JSON.stringify(object) 把对象转为字符串
    2. JSON.parse(string) 把字符串转为对象
-->

<script>

  //1. JSON.stringify(object) 把对象转为字符串
  let stu = {id:1,name:"刘一"};
  console.log(stu);
  console.log("stu="+stu);
  let stuStr = JSON.stringify(stu);//把stu对象转为字符串
  console.log(stuStr);
  console.log(stuStr.name);//字符串没有name属性 undefined
  console.log("stuStr="+stuStr);

  //2. JSON.parse(string) 把字符串转为对象

  let stuObj = JSON.parse(stuStr);//把字符串转为对象类型
  console.log(stuObj);
  console.log("stuObj=" + stuObj);
  console.log(stuObj.name);//对象才有name属性


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

第二章 BOM对象

2.1- BOM简介

> BOM(browser Object Model)浏览器对象模型
>
> JS把浏览器抽象成为一个window对象,运行我们使用js来模拟浏览器的行为.

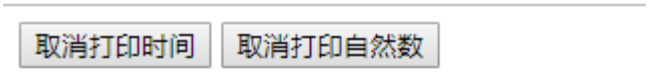
Window 对象

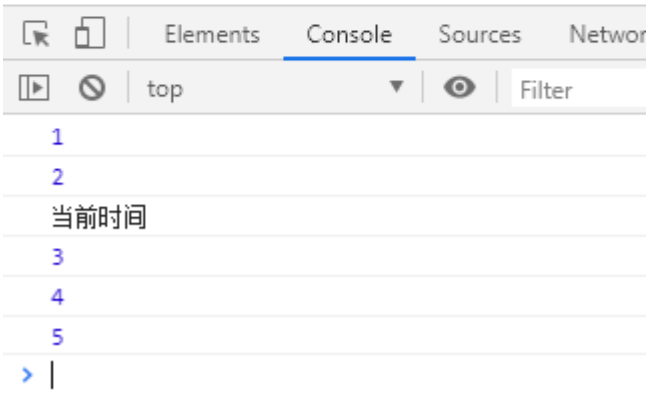
2.3-JS三个弹框


```
<!DOCTYPE html>
```

```
<html lang="zh">
<head>
    <meta charset="UTF-8">
    <title>JS三个弹框</title>
</head>
<body>
<!--
JS三个弹框
    查询手册完成
    1. 警告用户网站内容可能不安全

    2. 让用户确认是否删除所有数据

    3. 提示用户输入银行卡密码



-->

<script >

    // 1. 警告用户网站内容可能不安全
    alert("您浏览器的内容不健康\n请保护您的眼睛!");
    // 2. 让用户确认是否删除所有数据
    let boo = confirm("您确定要删除所有数据吗?");
    console.log(boo);//确定 true 取消 false
    // 3. 提示用户输入银行卡密码
    let str = prompt("请输入密码",123);
    console.log(str);//确定 用户输入的内容 取消 null

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

2.4-JS两个定时器

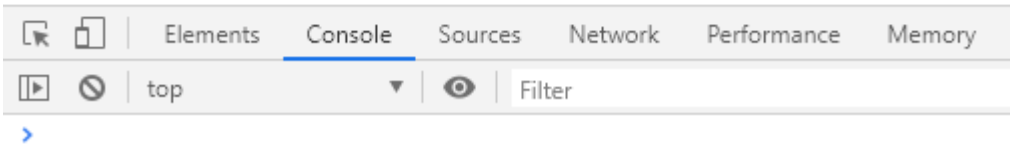





```
<!DOCTYPE html>
<html lang="zh">
<head>
    <meta charset="UTF-8">
    <title>JS两个定时器</title>
</head>
<body>

<!--
JS两个定时器
    1. 定时5秒之后在控制台打印当前时间
        setTimeout(函数,毫秒数) 创建一个定时器对象,定时`毫秒数`之后,调用一次`函数`

    2. 点击按钮取消打印时间
        clearTimeout(定时器) 销毁定时器

    3. 每隔2秒在控制台打印递增自然数
        setInterval(函数,毫秒数) 创建一个定时器对象,间隔`毫秒数`调用`函数`,直到永远

    4. 点击按钮取消打印自然数
        clearInterval(定时器) 销毁定时器


总结:
        1. setTimeout执行一次  setInterval执行无数次
        2. setTimeout销毁使用clearTimeout    setInterval销毁使用clearInterval
```

```
-->

<button onclick="myclearTimeout()">取消打印时间</button>
<button onclick="myclearInterval()">取消打印自然数</button>

<script >
    // 1. 定时5秒之后在控制台打印当前时间
    function myf1(){
        console.log("当前时间");
    }
    let timeout = setTimeout(myf1,5000);//创建一个定时器对象,5秒后调用一次myf1

    // 2. 点击按钮取消打印时间
    function myclearTimeout(){
        clearTimeout(timeout);//销毁定时器
    }

    // 3. 每隔2秒在控制台打印递增自然数
    let k =1;
    function myf2(){
        console.log(k++);
    }

    let interval = setInterval(myf2,2000);//创建一个定时器对象,每隔2秒调用一次myf2,直到永远

    // 4. 点击按钮取消打印自然数
    function myclearInterval(){
        clearInterval(interval);//销毁定时器
    }

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

2.5-location地址




```
<!DOCTYPE html>
<html lang="zh">
<head>
    <meta charset="UTF-8">
    <title>location地址</title>
</head>
<body>
<!--
location地址
    1. 获取当前浏览器地址
        location.href

    2. 刷新当前页面
        location.reload();

    3. 跳转页面
        location.href = "地址" ;

-->

<button onclick="addr()">获取当前浏览器地址</button>
<button onclick="refresh()">刷新当前页面</button>
<button onclick="jump()"> 跳转页面(必用)</button>


<script >
```

```
    // 1. 获取当前浏览器地址
    function addr(){
        console.log(location.href);//获取当前浏览器地址
    }

    // 2. 刷新当前页面
    function refresh(){
        location.reload();//刷新当前页面
    }

    // 3. 跳转页面(必用)
    function jump(){
        location.href = "https://www.jd.com";//跳转页面(必用)
    }

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

第三章 DOM对象(重点掌握)

3.1-dom简介

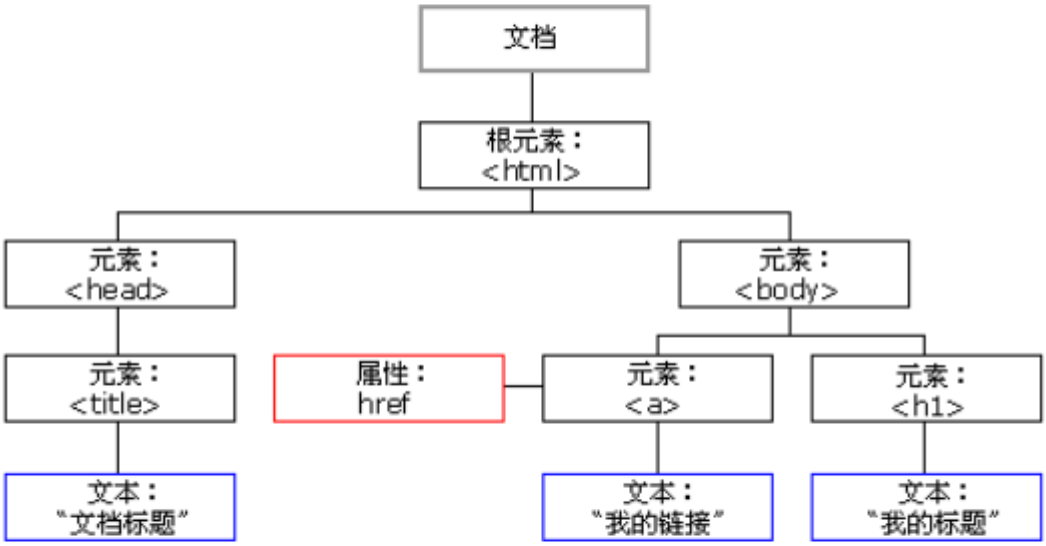
> DOM(Document Object Model) 页面文档对象模型
>
> JS把页面抽象成为一个对象,允许我们使用js来操作页面

HTML DOM （文档对象模型）

当网页被加载时，浏览器会创建页面的文档对象模型（Document Object Model）。

HTML DOM 模型被构造为对象的树。

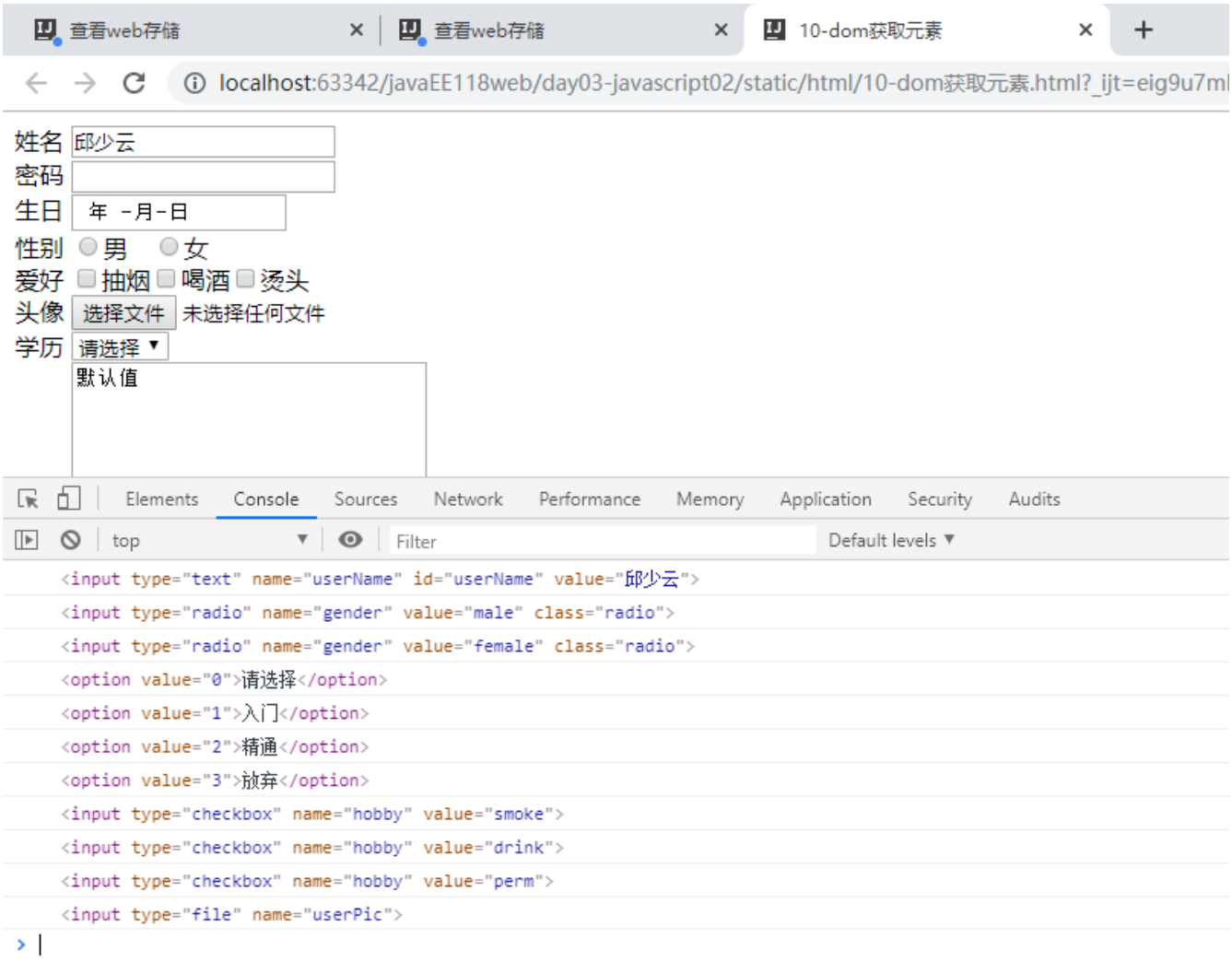
HTML DOM 树



通过可编程的对象模型，JavaScript 获得了足够的能力来创建动态的 HTML。

- JavaScript 能够改变页面中的所有 HTML 元素
- JavaScript 能够改变页面中的所有 HTML 属性
- JavaScript 能够改变页面中的所有 CSS 样式
- JavaScript 能够对页面中的所有事件做出反应

3.2-dom获取元素

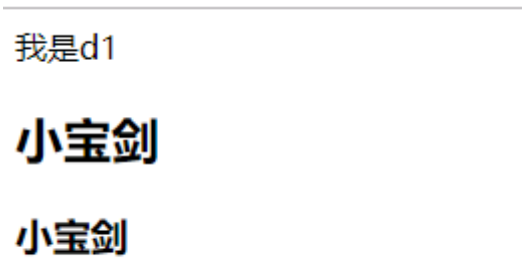



```
<!DOCTYPE html>
<html lang="zh">
<head>
    <meta charset="UTF-8">
    <title>dom获取元素</title>
</head>
<body>
<!--
dom获取元素
    第一种(掌握):es6提供了强大的根据CSS选择器获取元素的接口
        document.querySelector(CSS选择器) 根据css选择器获取一个元素对象
        document.querySelectorAll(CSS选择器) 根据css选择器获取元素对象数组集合
    根据第一种语法,完成如下功能:
    1.根据Id选择器获取一个元素
    2.根据类选择器获取一个(多个)元素
    3.根据标签选择器获取一个(多个)元素
    4.根据name属性选择器获取一个(多个)元素
    5.根据层级关系选择器获取一个(多个)元素

    第二种(了解):es5之前也有获取元素的接口
        1. 根据id获取一个元素 document.getElementById(ID) === querySelector(#ID)
```

```
        2. 根据class获取多个元素 document.getElementsByClassName(class) === querySelectorAll(.class)
        3. 根据标签名称获取多个元素 document.getElementsByTagName(标签) === querySelectorAll(标签)
        4. 根据name属性获取多个元素 document.getElementsByName('name值') === querySelectorAll([name=name值])

小总结:
    es6接口获取元素功能非常强大,而且接口非常易用.
    es5接口功能非常局限,而且接口使用复杂.


-->
<form action="#" method="get">
    姓名 <input type="text" name="userName" id="userName" value="邱少云"/> <br/>
    密码 <input type="password" name="userPass"> <br/>
    生日 <input type="date" name="userBirth"><br/>
    性别 <input type="radio" name="gender" value="male" class="radio">男 <input type="radio" name="gender" value="female" class="radio"/>女<br/>
    爱好 <input type="checkbox" name="hobby" value="smoke">抽烟<input type="checkbox" name="hobby" value="drink">喝酒<input type="checkbox" name="hobby" value="perm">烫头<br/>
    头像 <input type="file" name="userPic"><br/>
    学历
    <select name="userEdu" >
        <option value="0">请选择</option>
        <option value="1">入门</option>
        <option value="2">精通</option>
        <option value="3">放弃</option>

    </select><br/>
    简介
    <textarea name="userIntro"  cols="30" rows="10">默认值</textarea><br/>
    <input type="reset" value="清空按钮"/>
    <input type="submit" value="提交按钮"/><br/>
    <input type="button" value="普通按钮">
    <button>专业按钮</button><button>&times;</button>

</form>

<script >
    // 1.根据Id选择器获取一个元素
    let userName = document.querySelector("#userName");//获取id=userName的标签对象
    console.log(userName);//<input type="text" name="userName" id="userName" value="邱少云"/>
    // 2.根据类选择器获取一个(多个)元素
    let radios = document.querySelectorAll(".radio");//获取class=radio的标签对象数组
    for(let r of radios){
        console.log(r);//单选框对象
    }
    // 3.根据标签选择器获取一个(多个)元素
    let options = document.querySelectorAll("option");//获取所有的option标签对象数组
    for(let o of options){
        console.log(o);//每个option标签对象
```

```
    }
    // 4.根据name属性选择器获取一个(多个)元素
    let hobbys = document.querySelectorAll('input[name="hobby"]');//获取name=hobby的input标签对象数组
    for(let h of hobbys){
        console.log(h);//每个复选框标签对象
    }
    // 5.根据层级关系选择器获取一个(多个)元素
    let userPic = document.querySelector('form input[name="userPic"]');//获取文件选择框
    console.log(userPic);//<input type="file" name="userPic">




</script>

</body>
</html>
```

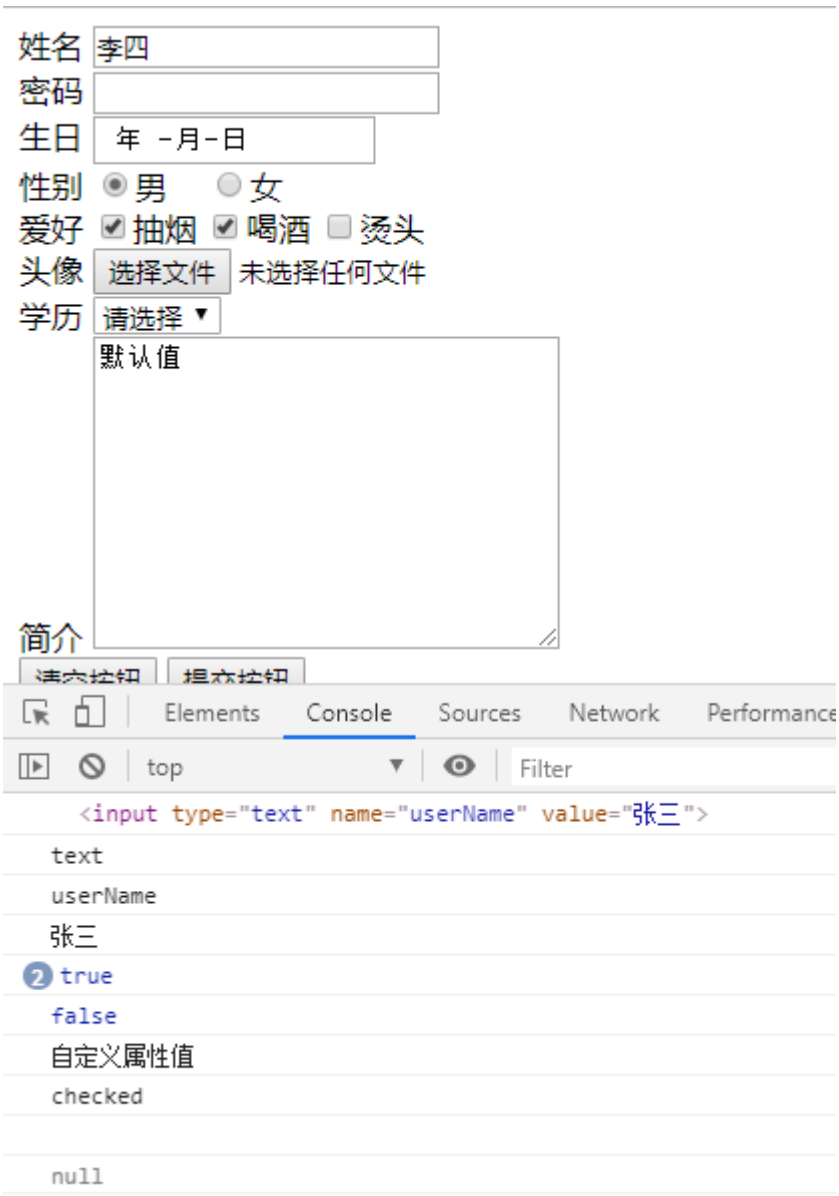
3.3-dom操作内容




```
<!DOCTYPE html>
<html lang="zh">
<head>
    <meta charset="UTF-8">
    <title>dom操作内容</title>
</head>
<body>
<!--
dom操作内容
    1. element.innerText; 获取或者修改元素的纯文本内容
    2. element.innerHTML; 获取或者修改元素的html内容
    3. element.outerHTML; 获取或者修改包含自身的html内容

总结:
    1. innerText 获取的是纯文本  innerHTML获取的是所有html内容
    2. innerText 设置到页面中的是纯文本   innerHTML设置到页面中的html会展示出外观效果
    3. innerHTML不包含自身 outerHTML包含自身的html内容
```

```
-->

<div id="d1">我是d1<br/></div>
<div id="d2">我是d2</div>
<div id="d3">我是d3</div>

<script>
    // 1. 向body中追加html内容
    document.write("<h1>大宝剑</h1>");//向body中追加html内容
    // 2. 获取元素的纯文本,html内容对比
    let d1 = document.querySelector("#d1");//获取d1标签对象
    console.log(d1.innerText);//我是d1
    console.log(d1.innerHTML);//我是d1<br/>

    // 3. 修改元素的纯文本,html内容对比
    let d2 = document.querySelector("#d2");//获取d1标签对象
    d2.innerText = "<h2>小宝剑</h2>";//修改d1的纯文本内容
    d2.innerHTML = "<h2>小宝剑</h2>";//修改d1的html内容

    // 4. 获取或修改包含元素自身的html内容(了解)
    let d3 = document.querySelector("#d3");//获取d1标签对象
    d3.outerHTML = "<h3>小宝剑</h3>";//修改d1的html内容

</script>

</body>
</html>
```

3.4-dom操作属性

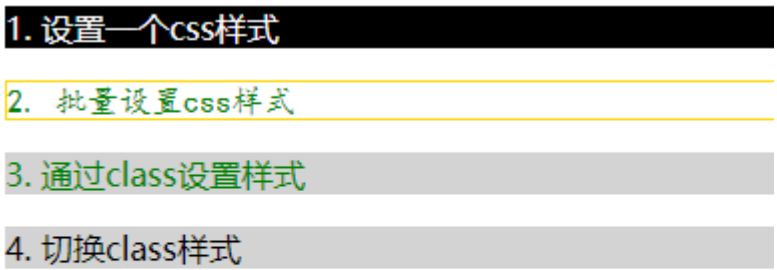


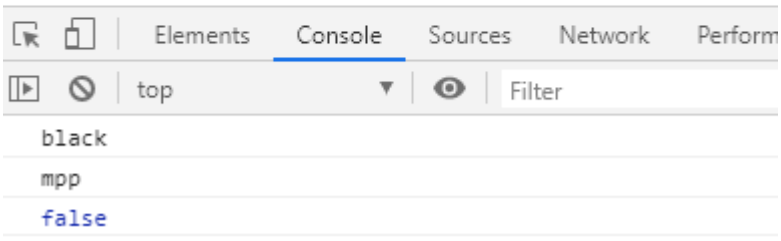

```
<!DOCTYPE html>
<html lang="zh">
<head>
    <meta charset="UTF-8">
    <title>dom操作属性</title>
</head>
<body>
<!--
dom操作属性
    1. 获取文本框的值,单选框或复选框的选中状态
        语法: element.properties 获取或者修改元素对象的原生属性
    2. 给元素设置自定义属性
        语法: element.setAttribute(属性名,属性值) 给元素设置一个属性值,可以设置原生和自定义
    3. 获取元素的自定义属性值
        语法: element.getAttribute(属性名) 获取元素的一个属性值,可以获取原生和自定义
    4. 移除元素的自定义属性
        语法: element.removeAttribute(属性名)
```

```
-->
<form action="#" method="get">
    姓名 <input type="text" name="userName" value="张三"  /> <br/>
    密码 <input type="password" name="userPass" > <br/>
    生日 <input type="date" name="userBirth"><br/>
    性别 <input type="radio" name="gender" value="男" checked="checked">男 
    <input type="radio" name="gender" value="女"/>女<br/>
    爱好 <input type="checkbox" name="hobby" value="smoke"  checked="checked"/>抽烟
    <input type="checkbox" name="hobby" value="drink" checked/>喝酒
    <input type="checkbox" name="hobby" value="perm"/>烫头<br/>
    头像 <input type="file" name="userPic"><br/>
    学历
    <select name="userEdu" >
        <option value="0">请选择</option>
        <option value="1">入门</option>
        <option value="2">精通</option>
        <option value="3">放弃</option>

    </select><br/>
    简介
    <textarea name="userIntro"  cols="30" rows="10">默认值</textarea><br/>
    <input type="reset" value="清空按钮"/>
    <input type="submit" value="提交按钮"/><br/>
    <input type="button" value="普通按钮">
    <button>专业按钮</button><button>&times;</button>


</form>
<script >
    // 1. 获取文本框的值,单选框或复选框的选中状态
    let userName = document.querySelector('input[name="userName"]');//获取文本框
    console.log(userName);//<input type="text" name="userName" value="张三" />
    console.log(userName.type);//text
    console.log(userName.name);//userName
    console.log(userName.value);//张三
    userName.value = "李四";//修改标签对象的value值


    let hobbys = document.querySelectorAll('input[name="hobby"]');//获取所有的复选框元素数组
    for(let h of hobbys){
        console.log(h.checked);//选中 true 未选中 false
    }


    // 2. 给元素设置自定义属性
    // console.log(userName.msg);//undefined
    userName.setAttribute("data-msg","自定义属性值");//添加一个属性值
    // 3. 获取元素的自定义属性值
    let msg = userName.getAttribute("data-msg");//获取一个属性值
    console.log(msg);//自定义属性值
    // 4. 移除元素的自定义属性
    userName.removeAttribute("data-msg");//移除一个属性值
```

```
    //不要使用的方式(错误用法):
    for(let hb of hobbys){
        console.log(hb.getAttribute("checked"));//选中 "checked" "" 未选中 null
    }

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

3.5-dom操作样式



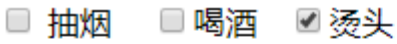


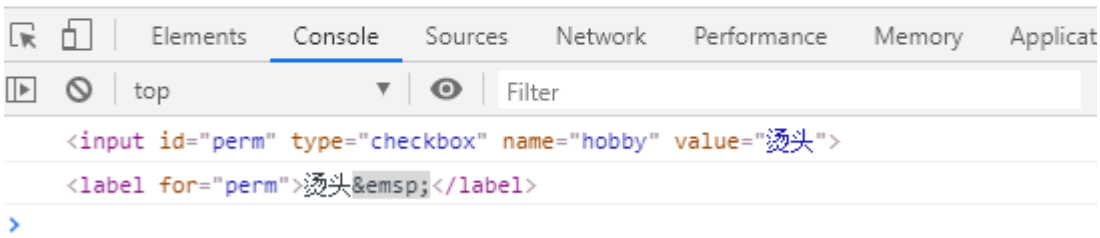

```
<!DOCTYPE html>
<html lang="zh">
<head>
    <meta charset="UTF-8">
    <title>dom操作样式</title>
</head>
<body>
<!--
dom操作样式
    1. 设置一个css样式
        语法: element.style.驼峰格式样式属性名    获取或者修改一个css样式
    2. 批量设置css样式
```

```
        语法: element.style.cssText 获取后者修改 标签的style属性的文本值
    3. 通过class设置样式
        语法: element.className 获取或者修改标签的class属性的文本值
    4. 切换class样式
        语法: element.classList es6特别提供的操作元素class的接口
            element.classList常用方法有四个:
            add()  添加一个class样式
            remove() 移除一个class样式
            contains() 判断是否包含某一个样式
            toggle() 切换一个class样式 有则删除,无则添加


-->
<style>
    #p1{ background-color: red;}
    .mp{color:green}
    .mpp{background-color: lightgray;}
</style>
<p id="p1">1. 设置一个css样式</p>
<p id="p2" >2. 批量设置css样式</p>
<p id="p3" class="mpp">3. 通过class设置样式</p>
<p id="p4">4. 切换class样式</p>

<script >
    let p1 = document.querySelector("#p1");//获取段落标签
    let p2 = document.querySelector("#p2");//获取段落标签
    let p3 = document.querySelector("#p3");//获取段落标签
    let p4 = document.querySelector("#p4");//获取段落标签

    // 1. 设置一个css样式
    p1.style.color = "white";//白色字体
    p1.style.backgroundColor = "black";//黑色背景
    console.log(p1.style.backgroundColor);//获取一个样式名
    p1.style.display = "none";//隐藏不显示
    p1.style.display = "block";//显示

    // 2. 批量设置css样式
    p2.style.cssText = "border:1px solid gold;color:green;font-family:'楷体';";//批量修改样式

    // 3. 通过class设置样式
    console.log(p3.className);//mpp
    p3.className =p3.className+ " mp";//修改class的文本值

    // 4. 切换class样式
    let classList = p4.classList;//p4的class样式集合
    classList.add("mpp");//添加一个class样式
    console.log(classList.contains("mp"));//false
    classList.add("mp");//添加一个class样式
    classList.remove("mp");//移除一个class样式

    classList.toggle("mp");//切换,无则添加
    classList.toggle("mp");//切换,有则删除
```

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

3.6dom操作元素





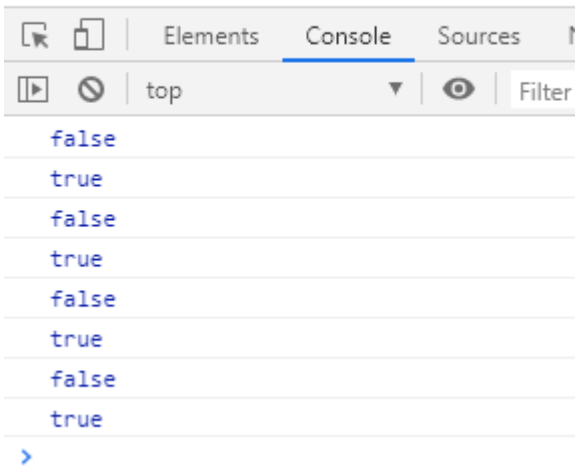

```
<!DOCTYPE html>
<html lang="zh">
<head>
    <meta charset="UTF-8">
    <title>dom操作元素</title>
</head>
<body>
<!--
dom操作元素
    1. 后面添加喝酒(掌握)
        innerHTML 获取或者设置标签的html内容
    2. 后面添加烫头(了解)
        document.createElement(标签名称) 创建一个标签对象
        parentNode.appendChild(newNode) 给父标签添加一个子标签
    3.移除元素(了解)
```

```
        outerHTML

-->

<div id="container">
    <input id="smoke" type="checkbox" name="hobby" value="抽烟">
    <label for="smoke">抽烟 </label>

<!--    <input id="drink" type="checkbox" name="hobby" value="喝酒">-->
<!--    <label for="drink">喝酒 </label>-->

<!--    <input id="perm" type="checkbox" name="hobby" value="烫头">-->
<!--    <label for="perm">烫头 </label>-->

</div>
<script >
    let container = document.querySelector("#container");//获取div标签对象
    //1. 后面添加喝酒(掌握) innerHTML
    container.innerHTML += '<input id="drink" type="checkbox" name="hobby" value="喝酒">' +
        '<label for="drink">喝酒 </label>';//添加喝酒复选框
    //2. 后面添加烫头(了解)
    let input = document.createElement("input");//创建一个input标签
    // console.log(input);//<input/>
    input.id = "perm";//添加一个id属性
    input.type="checkbox";//添加一个type属性
    input.name = "hobby";//添加一个name属性
    input.value = "烫头";//添加一个value属性
    console.log(input);//<input id="perm" type="checkbox" name="hobby" value="烫头">
    container.appendChild(input);//向container中追加input元素

    let label = document.createElement("label");//创建一个label标签对象
    // console.log(label);//<label></label>
    label.setAttribute("for","perm");//添加一个for属性
    label.innerHTML = "烫头 ";//添加html内容
    console.log(label);//<label for="perm">烫头 </label>
    container.appendChild(label);//向container中追加label标签对象

    //3.移除元素(了解) outerHTML
    // container.outerHTML = "";//删除包含自身的所有html内容



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

第四章 正则表达式

> 正则表达式提供了强大的字符串模式匹配功能.

4.1-正则表达式




```
<!DOCTYPE html>
<html lang="zh">
<head>
    <meta charset="UTF-8">
    <title>正则表达式</title>
</head>
<body>
<!--
正则表达式
    1. 理解正则表达式语法
        语法:
        1. /pattern/attributes (推荐使用)
        2. new RegExp(pattern, attributes); (不推荐)正则表达式中的特殊符号需要转义
        pattern 正则表达式
        attributes 正则表达式修饰符
                    i 不区分大小写 ,匹配字符串时不区分大小写
                    g 全局匹配 ,匹配到一个之后不停止,匹配字符串的所有内容
                    m 多行匹配 ,匹配字符串的所有行的内容
        常用正则表达式对象的方法
            test() 语法 RegExpObject.test(string)
            如果字符串 string 中含有与 RegExpObject 匹配的文本, 则返回 true, 否则返回 false。
    2. 会使用如下正则表达式匹配字符串

验证邮编
/\d{6}/
校验是否全由8位数字组成
/^[0-9]{8}$/
中文名称
/^[\u4E00-\u9FA5]{2,4}$/
是否带有小数
/^\d+\.\d+$/
验证身份证号
```

```
/\d{17}[\d|X]|\d{15}/
校验电话码格式
/^((0\d{2,3}-\d{7,8})|(1[3584]\d{9}))$/
验证网址
/http(s)?:[//]{2}([\w-]+\.)+[\w-]+([/]{1}[\w- ./?%&=]*)?/
验证EMail
/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/
-->
<script >
    //1. 理解正则表达式语法
    let reg1 = new RegExp("\\d{3}");//需要转义符,不推荐使用.创建一个正则表达式对象,匹配有三个数字
    console.log(reg1.test("abc"));//false
    console.log(reg1.test("123"));//true

    let reg2 = /\d{3}/;//不需要转义符,推荐使用.创建一个正则表达式对象,匹配有三个数字
    console.log(reg2.test("abc"));//false
    console.log(reg2.test("123"));//true



    //正则表达式修饰符(了解)
    let regi = /[amn]/i;//不区分大小写匹配amn
    let resi = "ABC".match(regi);//从"ABC"中匹配regi模式字符串
    console.log(resi);
    let regg = /\d/g;//全局查找数字
    let resg = "1 plus 2 equals 3".match(regg);
    console.log(resg);
    let regm = /^\d/m;//多行匹配开头的数字
    let resm = "abc1 plus 2 equals 3\n6abcmnk".match(regm);
    console.log(resm);


    //2. 会使用如下正则表达式匹配字符串
    let email = /\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/;//创建一个正则表达式对象,匹配邮箱
格式
    console.log(email.test("123abc"));//false
    console.log(email.test("123abc@qq.com"));//true

    let phone = /^((0\d{2,3}-\d{7,8})|(1[3584]\d{9}))$/;//创建一个正则表达式,匹配电话号码
    console.log(phone.test("123456"));//false
    console.log(phone.test("18312312312"));//true



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

第五章 综合案例

5.1-案例-表单校验


```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <title>案例-表单校验</title>
        <style type="text/css">
                    .regist_bg {
                        width: 100%;
                        height: 600px;
                        padding-top: 40px;
                        background-image: url(../img/bg.jpg);
                    }
                    .regist {
                        border: 7px inset #ccc;
                        width: 700px;
                        padding: 40px 0;
                        padding-left: 80px;
                        background-color: #fff;
                        margin-left: 25%;
                        border-radius: 10px;
                    }
                    input[type="submit"] {
                        background-color: aliceblue;
                        width: 100px;
                        height: 35px;
                        color: red;
                        cursor: pointer;
                        border-radius: 5px;
                    }
                    .warn{
                        color:red;
```

```
                        font-size: 12px;
                        display: none;
                    }
        </style>
<!--
表单校验
    1. 两次密码输入一致
    2. 邮箱格式正确
    3. 手机号格式正确
    4. 提交表单时校验表单项是否合法.
总结:
    form表单的 onsubmit 事件 表单提交之前触发,用法实例:
        onsubmit="return 函数()"  函数返回true则表单正常提交,函数返回false则阻止表单提交

-->
    </head>
    <body>
        <div class="regist_bg">
            <div class="regist">
                <form action="#" onsubmit="return judge()">
                    <table width="700px" height="350px">
                        <tr>
                            <td colspan="3">
                                <font color="#3164af">会员注册</font> USER REGISTER
                            </td>
                        </tr>
                        <tr>
                            <td align="right">用户名</td>
                            <td colspan="2"><input id="loginnameId" type="text"
name="loginname" size="50" /><span id="loginnamewarn"   class="warn">用户名不能为空</span>
</td>
                        </tr>
                        <tr>
                            <td align="right">密码</td>
                            <td colspan="2"><input id="loginpwdId" type="password"
name="loginpwd" size="50" /> </td>
                        </tr>
                        <tr>
                            <td align="right">确认密码</td>
                            <td colspan="2"><input id="reloginpwdId" type="password"
name="reloginpwd" size="50" /><span id="pwdwarn" class="warn">密码不一致</span> </td>
                        </tr>
                        <tr>
                            <td align="right">Email</td>
                            <td colspan="2"><input id="emailId" type="text" name="email"
size="50" /> <span id="emailwarn"  class="warn">邮箱格式有误</span> </td>
                        </tr>
                        <tr>
                            <td align="right">姓名</td>
                            <td colspan="2"><input name="text" name="username" size="50" />
</td>
                        </tr>
                        <tr>
```

```
                            <td align="right">性别</td>
                            <td colspan="2">
                                <input type="radio" name="gender" value="男"
checked="checked" />男
                                <input type="radio" name="gender" value="女" />女
                            </td>
                        </tr>
                        <tr>
                            <td align="right">电话号码</td>
                            <td colspan="2"><input  id="phone"  type="text" name="phone"
size="50" /> <span id="phonewarn"  class="warn">手机格式有误</span> </td>
                        </tr>
                        <tr>
                            <td align="right">所在地</td>
                            <td colspan="3">
                                <select id="provinceId" onchange="selectCity(this.value)"
style="width:150px">
                                    <option value="">----请-选-择-省----</option>
                                    <option value="0">北京</option>
                                    <option value="1">辽宁省</option>
                                    <option value="2">江苏省</option>
                                </select>
                                <select id="cityId" style="width:150px">
                                    <option>----请-选-择-市----</option>
                                </select>
                            </td>
                        </tr>
                        <tr>
                            <td width="80" align="right">验证码</td>
                            <td width="100"><input type="text" name="verifyCode" /> </td>
                            <td><img src="../img/checkMa.png" /> </td>
                        </tr>
                        <tr>
                            <td></td>
                            <td colspan="2">
                                <input id="rebtn" type="submit" value="注册" />
                            </td>
                        </tr>
                    </table>
                </form>
            </div>
        </div>
<script >


    let loginpwdId = document.querySelector("#loginpwdId");//获取密码框
    let reloginpwdId = document.querySelector("#reloginpwdId");//获取确认密码框
    let pwdwarn = document.querySelector("#pwdwarn");//密码警告信息



    let emailId = document.querySelector("#emailId");//邮件输入框
    let emailwarn = document.querySelector("#emailwarn");//邮件警告信息
```

```
    // 1. 两次密码输入一致
    function pwd(){//校验确认密码
        let boo = loginpwdId.value == reloginpwdId.value;//判断密码是否一致
        if(boo){//合法,隐藏警告
            pwdwarn.style.display = "none";//隐藏
        }else{//不合法,显示警告
            pwdwarn.style.display = "inline";//显示
        }
        return boo;//返回密码是否合法
    }

    reloginpwdId.onchange  = pwd;//当确认密码值改变时调用pwd方法执行

    // 2. 邮箱格式正确
    function mail(){//校验邮件格式
        let email = /\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/;//创建一个正则表达式对象,匹
配邮箱格式
        let boo = email.test(emailId.value);//校验邮件格式
        if(boo){//合法,隐藏警告
            emailwarn.style.display = "none";//隐藏
        }else{//不合法,显示警告
            emailwarn.style.display = "inline";//显示
        }
        return boo;//返回邮件是否合法
    }

    emailId.onchange = mail;//当邮件输入框的值改变时调用mail方法执行

    // 3. 手机号格式正确
    let phone = /^((0\d{2,3}-\d{7,8})|(1[3584]\d{9}))$/;//创建一个正则表达式,匹配电话号码

    //4. 提交表单时校验表单项是否合法.
    function judge(){//表单提交校验
        return pwd() && mail();//所有表单项都合法才返回true
    }


</script>

    </body>
</html>
```

5.2-案例-商品全选

```
<!DOCTYPE html>
<html lang="zh">
<head>
    <meta charset="UTF-8">
    <title>案例-商品全选</title>
</head>
<body>
<!--
商品全选
    1. 全选 点击全选按钮,所有复选框都被选中
    2. 反选 点击反选按钮,所有复选框状态取反
-->
<button id="btn1">1. 全选</button>
<button id="btn2">2. 反选</button>
<br/>
<input type="checkbox">电脑<input type="checkbox">手机<input type="checkbox">汽车<input
type="checkbox">别墅<input type="checkbox" checked="checked">笔记本
<script >

    let btn1 = document.querySelector("#btn1");//全选
    let btn2 = document.querySelector("#btn2");//反选

    let boxes = document.querySelectorAll('input[type="checkbox"]');//获取所以复选框



    btn1.onclick = function(){//全选
        for(let b of boxes){
            b.checked = true;//全选
        }
    }
    btn2.onclick = function(){//反选
        for(let b of boxes){
            b.checked = !b.checked;//反选
        }
    }


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

5.3-案例-省市级联


```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <title>案例-省市级联</title>
        <style type="text/css">
                    .regist_bg {
                        width: 100%;
                        height: 600px;
                        padding-top: 40px;
                        background-image: url(../img/bg.jpg);
                    }
                    .regist {
                        border: 7px inset #ccc;
                        width: 600px;
                        padding: 40px 0;
                        padding-left: 80px;
                        background-color: #fff;
                        margin-left: 25%;
                        border-radius: 10px;
                    }
                    input[type="submit"] {
                        background-color: aliceblue;
                        width: 100px;
                        height: 35px;
                        color: red;
                        cursor: pointer;
```

```
                        border-radius: 5px;
                    }
        </style>

    </head>
    <body>
        <div class="regist_bg">
            <div class="regist">
                <form action="#">
                    <table width="600" height="350px">
                        <tr>
                            <td colspan="3">
                                <font color="#3164af">会员注册</font> USER REGISTER
                            </td>
                        </tr>
                        <tr>
                            <td align="right">用户名</td>
                            <td colspan="2"><input id="loginnameId" type="text"
name="loginname" size="60" /> </td>
                        </tr>
                        <tr>
                            <td align="right">密码</td>
                            <td colspan="2"><input id="loginpwdId" type="password"
name="loginpwd" size="60" /> </td>
                        </tr>
                        <tr>
                            <td align="right">确认密码</td>
                            <td colspan="2"><input id="reloginpwdId" type="password"
name="reloginpwd" size="60" /> </td>
                        </tr>
                        <tr>
                            <td align="right">Email</td>
                            <td colspan="2"><input id="emailId" type="text" name="email"
size="60" /> </td>
                        </tr>
                        <tr>
                            <td align="right">姓名</td>
                            <td colspan="2"><input name="text" name="username" size="60" />
</td>
                        </tr>
                        <tr>
                            <td align="right">性别</td>
                            <td colspan="2">
                                <input type="radio" name="gender" value="男"
checked="checked" />男
                                <input type="radio" name="gender" value="女" />女
                            </td>
                        </tr>
                        <tr>
                            <td align="right">电话号码</td>
                            <td colspan="2"><input type="text" name="phone" size="60" />
</td>
                        </tr>
```

```
                        <tr>
                            <td align="right">所在地</td>
                            <td colspan="3">
                                <select id="provinceId" style="width:150px">
                                    <option value="">----请-选-择-省----</option>

                                </select>
                                <select id="cityId" style="width:150px">
                                    <option value="">----请-选-择-市----</option>
                                </select>
                            </td>
                        </tr>
                        <tr>
                            <td width="80" align="right">验证码</td>
                            <td width="100"><input type="text" name="verifyCode" /> </td>
                            <td><img src="../img/checkMa.png" /> </td>
                        </tr>
                        <tr>
                            <td></td>
                            <td colspan="2">
                                <input type="submit" value="注册" />
                            </td>
                        </tr>
                    </table>
                </form>
            </div>
        </div>
<!--
省市级联
    1. 页面加载完成后自动装载省数据
    2. 当选中省时,装载该省的市数据
-->
        <script type="text/javascript">
            //准备省市数据
            let provinceData = ["北京","河北","辽宁"];
            //准备省对应的市数据
            let cityData = {
                "北京":["顺义区","昌平区","朝阳区"],
                "河北":["保定","石家庄","廊坊"],
                "辽宁":["沈阳","铁岭","抚顺"]
            };


            let provinceId = document.querySelector("#provinceId");//获取省下拉列表
            let cityId = document.querySelector("#cityId");//获取市下拉列表
            //页面加载完成时填充省选项
            window.onload  = function(){//页面加载完成时
                for(let p of provinceData){//迭代省选项
                    provinceId.innerHTML += "<option value='"+p+"'>"+p+"</option>";//追加省
选项

                }
            }
            //当省选项值改变时
```

```
            provinceId.onchange = function(){//省选项值改变时
                //每天填充市数据时先初始化一次
                cityId.innerHTML = '<option value="">----请-选-择-市----</option>';
                console.log(this.value);//选中的省
                let citys = cityData[this.value];//当前省对应的市数据
                //把市数据填充到市选项中
                for(let c of citys){
                    cityId.innerHTML += "<option value='"+c+"'>"+c+"</option>";//追加市选项
                }
            }

        </script>

    </body>
</html>
```

5.4-案例-隔行变色

全/反选	分类ID	分类名称	分类描述	操作
	1	手机数码	手机数码类商品	修改|删除
	2	电脑办公	电脑办公类商品	修改|删除
	3	鞋靴箱包	鞋靴箱包类商品	修改|删除
	4	家居饰品	家居饰品类商品	修改|删除
	5	牛奶制品	牛奶制品类商品	修改|删除
	6	大豆制品	大豆制品类商品	修改|删除
	7	海参制品	海参制品类商品	修改|删除
	8	羊绒制品	羊绒制品类商品	修改|删除
	9	海洋产品	海洋产品类商品	修改|删除
	10	奢侈用品	奢侈用品类商品	修改|删除
	4	家居饰品	家居饰品类商品	修改|删除
	4	家居饰品	家居饰品类商品	修改|删除
	4	家居饰品	家居饰品类商品	修改|删除
	10	奢侈用品	奢侈用品类商品	修改|删除
	4	家居饰品	家居饰品类商品	修改|删除
	4	家居饰品	家居饰品类商品	修改|删除
	4	家居饰品	家居饰品类商品	修改|删除

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8" />
        <title>案例-隔行变色</title>
```

```
    </head>
    <body>
        <table id="tab1" border="1" width="800" align="center">
            <tr >
                <th width="100px"><input type="checkbox" >全/<input type="checkbox" >反选
</th>
                <th>分类ID</th>
                <th>分类名称</th>
                <th>分类描述</th>
                <th>操作</th>
            </tr>
            <tr>
                <td><input type="checkbox" class="checkbox"></td>
                <td>1</td>
                <td>手机数码</td>
                <td>手机数码类商品</td>
                <td><a href="">修改</a>|<a href="">删除</a></td>
            </tr>
            <tr>
                <td><input type="checkbox" class="checkbox"></td>
                <td>2</td>
                <td>电脑办公</td>
                <td>电脑办公类商品</td>
                <td><a href="">修改</a>|<a href="">删除</a></td>
            </tr>
            <tr>
                <td><input type="checkbox" class="checkbox"></td>
                <td>3</td>
                <td>鞋靴箱包</td>
                <td>鞋靴箱包类商品</td>
                <td><a href="">修改</a>|<a href="">删除</a></td>
            </tr>
            <tr>
                <td><input type="checkbox" class="checkbox"></td>
                <td>4</td>
                <td>家居饰品</td>
                <td>家居饰品类商品</td>
                <td><a href="">修改</a>|<a href="">删除</a></td>
            </tr>
            <tr>
                <td><input type="checkbox" class="checkbox"></td>
                <td>5</td>
                <td>牛奶制品</td>
                <td>牛奶制品类商品</td>
                <td><a href="">修改</a>|<a href="">删除</a></td>
            </tr>
            <tr>
                <td><input type="checkbox" class="checkbox"></td>
                <td>6</td>
                <td>大豆制品</td>
                <td>大豆制品类商品</td>
                <td><a href="">修改</a>|<a href="">删除</a></td>
            </tr>
```

```html
        <tr>
            <td><input type="checkbox" class="checkbox"></td>
            <td>7</td>
            <td>海参制品</td>
            <td>海参制品类商品</td>
            <td><a href="">修改</a>|<a href="">删除</a></td>
        </tr>
        <tr>
            <td><input type="checkbox" class="checkbox"></td>
            <td>8</td>
            <td>羊绒制品</td>
            <td>羊绒制品类商品</td>
            <td><a href="">修改</a>|<a href="">删除</a></td>
        </tr>
        <tr>
            <td><input type="checkbox" class="checkbox"></td>
            <td>9</td>
            <td>海洋产品</td>
            <td>海洋产品类商品</td>
            <td><a href="">修改</a>|<a href="">删除</a></td>
        </tr>
        <tr>
            <td><input type="checkbox" class="checkbox"></td>
            <td>10</td>
            <td>奢侈用品</td>
            <td>奢侈用品类商品</td>
            <td><a href="">修改</a>|<a href="">删除</a></td>
        </tr>
        <tr>
            <td><input type="checkbox" class="checkbox"></td>
            <td>4</td>
            <td>家居饰品</td>
            <td>家居饰品类商品</td>
            <td><a href="">修改</a>|<a href="">删除</a></td>
        </tr>
        <tr>
            <td><input type="checkbox" class="checkbox"></td>
            <td>4</td>
            <td>家居饰品</td>
            <td>家居饰品类商品</td>
            <td><a href="">修改</a>|<a href="">删除</a></td>
        </tr>
        <tr>
            <td><input type="checkbox" class="checkbox"></td>
            <td>4</td>
            <td>家居饰品</td>
            <td>家居饰品类商品</td>
            <td><a href="">修改</a>|<a href="">删除</a></td>
        </tr>
        <tr>
            <td><input type="checkbox" class="checkbox"></td>
            <td>10</td>
            <td>奢侈用品</td>
```

```
                <td>奢侈用品类商品</td>
                <td><a href="">修改</a>|<a href="">删除</a></td>
            </tr>
            <tr>
                <td><input type="checkbox" class="checkbox"></td>
                <td>4</td>
                <td>家居饰品</td>
                <td>家居饰品类商品</td>
                <td><a href="">修改</a>|<a href="">删除</a></td>
            </tr>
            <tr>
                <td><input type="checkbox" class="checkbox"></td>
                <td>4</td>
                <td>家居饰品</td>
                <td>家居饰品类商品</td>
                <td><a href="">修改</a>|<a href="">删除</a></td>
            </tr>
            <tr>
                <td><input type="checkbox" class="checkbox"></td>
                <td>4</td>
                <td>家居饰品</td>
                <td>家居饰品类商品</td>
                <td><a href="">修改</a>|<a href="">删除</a></td>
            </tr>
        </table>
<!--
隔行变色
    1. 表格奇偶行颜色不同
    2. 鼠标移入颜色高亮
-->
    <script >
        let trs = document.querySelectorAll("table tr");//获取所有的行
        for(let i=0;i<trs.length;i++){
            if(i%2 == 0){//索引是偶数行
                trs[i].style.backgroundColor ="#efefef";//浅灰色
            }else{//索引是奇数行
                trs[i].style.backgroundColor ="#ccc";//灰色
            }
            let oldColor = trs[i].style.backgroundColor;//原来的颜色
            trs[i].onmouseover = function(){//鼠标移入的时候
                this.style.backgroundColor = "pink";//粉色
            }
            trs[i].onmouseout = function(){//鼠标移出的时候
                this.style.backgroundColor = oldColor;//回复原色
            }

        }


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