

实验（实训）八 Javascript 基础

一、实验（实训）目的

- 1、实践 Javascript 基本语法；
- 2、实践 Javascript 函数定义、调用等操作；
- 3、熟悉 Javascript 代码调试。

二、知识要点

- 1、Javascript 基本语法；
- 2、Javascript 函数、对象、数组等使用。

三、主要仪器设备及耗材

- 1、计算机、HBuilder X。

四、实验（实训）内容及步骤

1、使用 JavaScript 计算银行存款收益

假设在中国建设银行、中国工商银行、中国农业银行分别存入 10000 元定期存款，存款年限分别为 4 年、5 年、6 年，年利率分别为 3.75%、3.25%、3%。请问当所有定期存款日期结束时，在哪家银行获得的收益最高？使用 JavaScript 实现算法，用 alert 在页面显示。显示效果图如图 8-1 所示。



图 8-1 计算银行存款收益

Javascript 代码如下：

```
<script>  
var moneyOfBank1 = 10000;           //银行 1 (中国建设银行) 存款 10 000  
var moneyOfBank2 = 10000;           //银行 2 (中国工商银行) 存款 10 000
```

```

var moneyOfBank3 = 10000;           //银行 3 (中国农业银行) 存款 10 000

//通过 for 语句循环 4 次，计算存款日期结束时银行 1 的存款总金额
for (var i=1;i<5;i++) {
    moneyOfBank1 += moneyOfBank1 * 3.75 / 100;
}

//通过 for 语句循环 5 次，计算存款日期结束时银行 2 的存款总金额
for (var j=1;j<6;j++) {
    moneyOfBank2 += moneyOfBank2 * 3.25 / 100;
}

//通过 for 语句循环 6 次，计算存款日期结束时银行 3 的存款总金额
for (var k=1;k<7;k++) {
    moneyOfBank3 += moneyOfBank3 * 3 / 100;
}

var str=" 中国建设银行收益：" +moneyOfBank1+"\n 中国工商银行收益：" +
"+moneyOfBank2+"\n 中国农业银行收益：" +moneyOfBank3;

// 比较各银行最终计算的存款

if (moneyOfBank1 > moneyOfBank2 && moneyOfBank1 > moneyOfBank3) {
    // 如果 moneyOfBank1 值最大，弹出弹窗"中国建设银行收益最高"
    alert(str+"\n\n 中国建设银行收益最高");
} else if (moneyOfBank2 > moneyOfBank1 && moneyOfBank2 > moneyOfBank3){
    // 如果 moneyOfBank2 值最大，弹出弹窗"中国工商银行收益最高"
    alert(str+"\n\n 中国工商银行收益最高");
} else if (moneyOfBank3 > moneyOfBank1 && moneyOfBank3 > moneyOfBank2){
    // 如果 moneyOfBank3 值最大，弹出弹窗"中国农业银行收益最高"
    alert(str+"\n\n 中国农业银行收益最高");
}

</script>

```

2、判断闰年

使用 JavaScript 获取当前年份，用函数计算当前年份是否为闰年。结果显

示如图 8-2 所示。

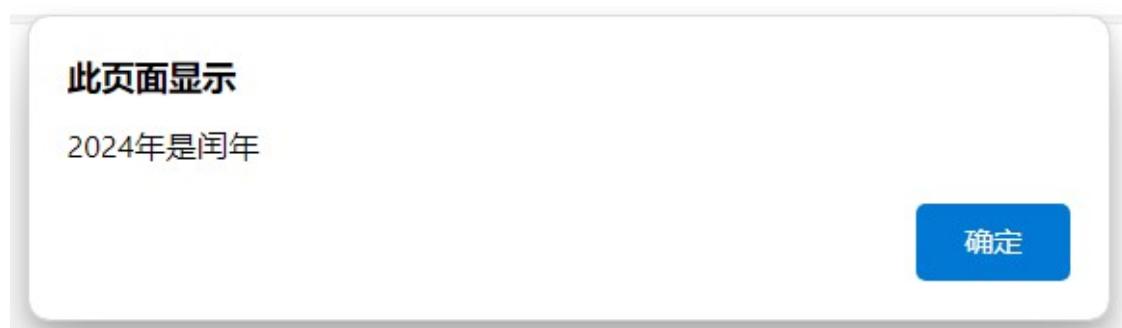


图 8-2 判断闰年

Javascript 代码如下：

```
<script>

// 定义函数

function isLeapYear(year) {
    if ((year % 4 == 0) && (year % 100 != 0) || (year % 400 == 0))
    {
        alert(year + "年是闰年");
    }
    else{
        alert(year + "年不是闰年");
    }
}

// 调用函数

var year = new Date().getFullYear();
isLeapYear(year);

</script>
```

五、注意事项（或操作要点）

- 1、如何在 HTML 中引用 js 文件；
- 2、Javascript 如何调用 window 函数。

六、课后思考

- 1、Javascript 有哪些调试方法？

实验（实训）九 Javascript DOM 和 BOM 操作

一、实验（实训）目的

- 1、熟悉 DOM 和 BOM 模型；
- 2、实践 Javascript 操作 DOM 和 BOM 的方法；
- 3、熟悉 Javascript 调试的方法。

二、知识要点

- 1、DOM 和 BOM 模型；
- 2、Javascript 操作 DOM 和 BOM 的方法。

三、主要仪器设备及耗材

- 1、计算机、HBuilder X。

四、实验（实训）内容及步骤

1、实现线上点餐功能

使用 JavaScript 实现线上点餐功能。功能概述：在输入框中输入菜名，点击确定按钮后，在列表中添加菜品，菜品列表中可以删除不需要的菜品。效果如图 9-1 所示。



图 9-1 在线点餐

1) 为“添加菜品”按钮绑定单击事件

单击事件触发后，判断输入框中的内容是否为空，如果为空则提示用户输入，否则调用 addFood() 函数，传入输入框中的内容，最后将输入框内容清空。代码如下：

```
// 为“添加菜品”按钮绑定单击事件
document.getElementById('add-btn').onclick = function() {
    // 获取输入框元素
    var $input = document.getElementsByTagName('input')[0];
    // 如果未输入任何内容，使用 alert 提示
    if ($input.value.length === 0) {
        alert('请输入想添加的菜品');
        return;
    }
    // 调用 addFood 方法，并传入输入值
    addFood($input.value);
    // 将输入框内容清空
    $input.value = '';
}
```

2) 定义添加菜品的函数 addFood()

在函数内部创建对应的、及<button>节点，并调用 appendTextNode() 函数为对应节点添加文本内容；然后调用 appendClassAttribute() 和 appendClickAttribute() 函数为<button>节点添加 class 属性及 onclick 属性；最后将节点和<button>节点添加到节点，节点添加到列表中。代码如下：

```
// 定义添加菜品函数
function addFood(value) {
    // 创建对应的<li>、<span>节点
    var liNode = document.createElement('li');
    var spanNode = document.createElement('span');
    // 为<span>节点添加文本
    appendTextNode(spanNode, value);
```

```

// 创建<button>节点
var buttonNode = document.createElement('button');

// 为<button>节点添加文本内容、class 属性以及 onclick 属性
appendTextNode(buttonNode, '删除');
appendClassAttribute(buttonNode, 'delete-btn');
appendClickAttribute(buttonNode);

// 将<span>和<button>节点添加到<li>节点
liNode.appendChild(spanNode);
liNode.appendChild(buttonNode);

// 将<li>添加到列表<ul>节点中
document.getElementsByTagName('ul')[0].appendChild(liNode);
}

```

3) 完善其余功能

定义节点操作的函数 appendTextNode()，函数中使用 createText Node() 方法创建一个文本节点，并添加到相应的节点中；appendClassAttribute() 和 appendClickAttribute() 函数中都使用 createAttribute() 方法创建对应的属性节点 class 和 onclick；deleteNode() 函数中通过 parentNode 属性获取当前订单的 li 元素，然后通过 removeChild() 方法将 li 元素从列表中删除。代码如下：

```

// 定义节点操作函数

function appendTextNode(node, value) {
    // 创建文本节点
    var textView = document.createTextNode(value);
    // 将文本节点添加到对应节点
    node.appendChild(textView);
}

// 定义添加 class 属性节点函数

function appendClassAttribute(node, value) {
    // 创建 class 属性
    var attributeView = document.createAttribute('class');

```

```
// 设置属性值
attributeNode.value = value;
// 将 class 属性设置到对应节点
node.setAttributeNode(attributeNode);
}

// 定义添加 onclick 属性节点函数
function appendClickAttribute(node) {
    // 创建 onclick 属性
var attributeNode = document.createAttribute('onclick');
    // 设置属性值
attributeNode.value = 'deleteNode(this);';
    // 将 onclick 属性设置到对应节点
node.setAttributeNode(attributeNode);
}

// 定义删除节点函数，由删除按钮点击触发
function deleteNode(dom) {
    // 获取删除按钮的父节点<li>节点，然后从<ul>节点中删除
const liNode = dom.parentNode;
    document.getElementsByName('ul')[0].removeChild(liNode);
}
```

五、注意事项（或操作要点）

- 1、熟悉 HTML 元素操作和 DOM 节点操作的属性和方法；
- 2、熟悉 BOM 中的 window 对象。

六、课后思考

- 1、总结 DOM 节点操作的方法。

实验（实训）十 Javascript 事件

一、实验（实训）目的

- 1、熟悉事件与事件流，掌握事件绑定方式和事件对象的常用属性和方法；
- 2、实践常见事件类型；
- 3、了解事件优化方式，事件的委托和删除等。

二、知识要点

- 1、如何绑定、删除事件；
- 2、常见的事件类型。

三、主要仪器设备及耗材

- 1、计算机、HBuilder X。

四、实验（实训）内容及步骤

1、制作金企鹅教育主页轮播图

使用 JavaScript 实现金企鹅教育主页轮播图功能。功能概述：

每个图片对应一个数字导航；默认情况下每隔 2 秒自动循环播放图片；鼠标划过整个轮播图区域时停止自动播放；鼠标离开整个轮播图区域时继续播放至下一张；鼠标滑过数字导航时切换到对应图片。效果如图 10-1 所示。



图 10-1 主页轮播图

1) 添加轮播图的内容

在主页 HTML 文件中表示轮播图区块的 `<section id="wrap" class="banner">...</section>` 标签中添加以下代码，设置轮播图中的图片和数

字导航。

```
<ul id="pic" class="banner_p">  
    <li></li>  
    <li></li>  
    <li></li>  
</ul>  
  
<ol id="list" class="banner_b">  
    <li class="on">1</li>  
    <li>2</li>  
    <li>3</li>  
</ol>
```

2) 设置轮播图的样式

在主页 CSS 文件中添加以下代码，使用选择器. banner、. banner . banner_p 等设置样式。

```
.banner { position: relative;  
    width: 1200px;  
    height: 380px;  
    margin: 10px auto 0 auto;  
    box-shadow: 0 0 5px white;  
    overflow: hidden; }  
  
.banner .banner_p{ position: absolute;  
    width: 6000px;  
    height: 380px;  
    z-index: 1; }  
  
.banner .banner_p li{ float: left;  
    width: 1200px;  
    height: 380px; }  
  
.banner .banner_b { position: absolute;  
    bottom: 20px;  
    width: 200px;
```

```

height: 10px;
z-index: 2;
left: 500px; }

.banner .banner_b li{ margin-left: 5px;
display: inline-block;
width: 20px;
height: 20px;
border-radius: 50%;
background-color: #6CA1A0;
text-align: center;
color: white;
cursor: pointer; }

.banner .banner_b li.on{ background-color:#EEF5F7;
color: #6CA1A0; }

```

3) 实现轮播图的动态效果

在 js 文件中添加以下代码, 通过监听页面 load 事件、定义不同的功能函数、绑定不同的鼠标事件、调用不同的功能函数实现轮播图的动态效果, 包括图片自动切换、鼠标划过图片切换停止、鼠标划过数字导航切换至对应图片等。

```
// 监听页面 load 事件, 在页面所有元素加载完成后才触发该事件
window.onload=function(){
```

```
// 获取 HTML 文档中的元素, 并赋值给声明的变量
var wrap=document.getElementById('wrap'),
pic=document.getElementById('pic').getElementsByTagName("li"),
list=document.getElementById('list').getElementsByTagName('li'),
index=0,
timer=null;
```

// 定义图片切换函数

```
function changePic (curIndex) {
for (var i = 0; i < pic.length; ++i) {
```

```
pic[i].style.display = "none";
```

```

        list[i].className = "";
    }

    pic[curIndex].style.display = "block";
    list[curIndex].className = "on";
}

// 定义自动播放函数

function autoPlay () {
    if (++index >= pic.length) {
        index = 0;
    }
    changePic(index);
}

// 设置定时器，按照指定的周期（2 秒）调用自动播放函数 autoPlay

timer = setInterval(autoPlay, 2000);

// 鼠标划过整个轮播图区域时停止自动播放

wrap.onmouseover = function () {
    clearInterval(timer); // 取消 setInterval 设置的定时器
}

// 鼠标离开整个容器时继续播放至下一张

wrap.onmouseout = function () {
    timer = setInterval(autoPlay, 2000);
}

// 遍历所有数字导航实现鼠标划过切换至对应的图片

for (var i = 0; i < list.length; i++) {
    list[i].onmouseover = function () {
        clearInterval(timer);
        index = this.innerText - 1;
        changePic(index);
    }
}

```

}

五、注意事项（或操作要点）

1、HTML 元素产生事件时，事件流的过程；

2、事件绑定和解绑的方法。

六、课后思考

1、总结 DOM 事件流的阶段。