**序号： 13**

云南大学软件学院

实 验 报 告

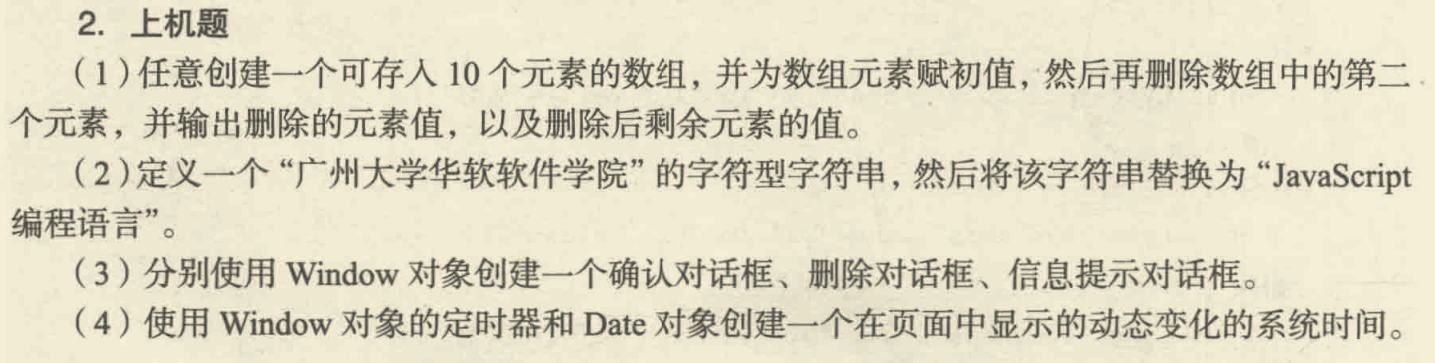
**课程名称：** WEB前端设计与开发  **实验名称：** 实验9

**学 号：** 20221060155  **姓 名：** 金泽均

**实验内容**

将下列的每个任务完成并生成相应的网页：

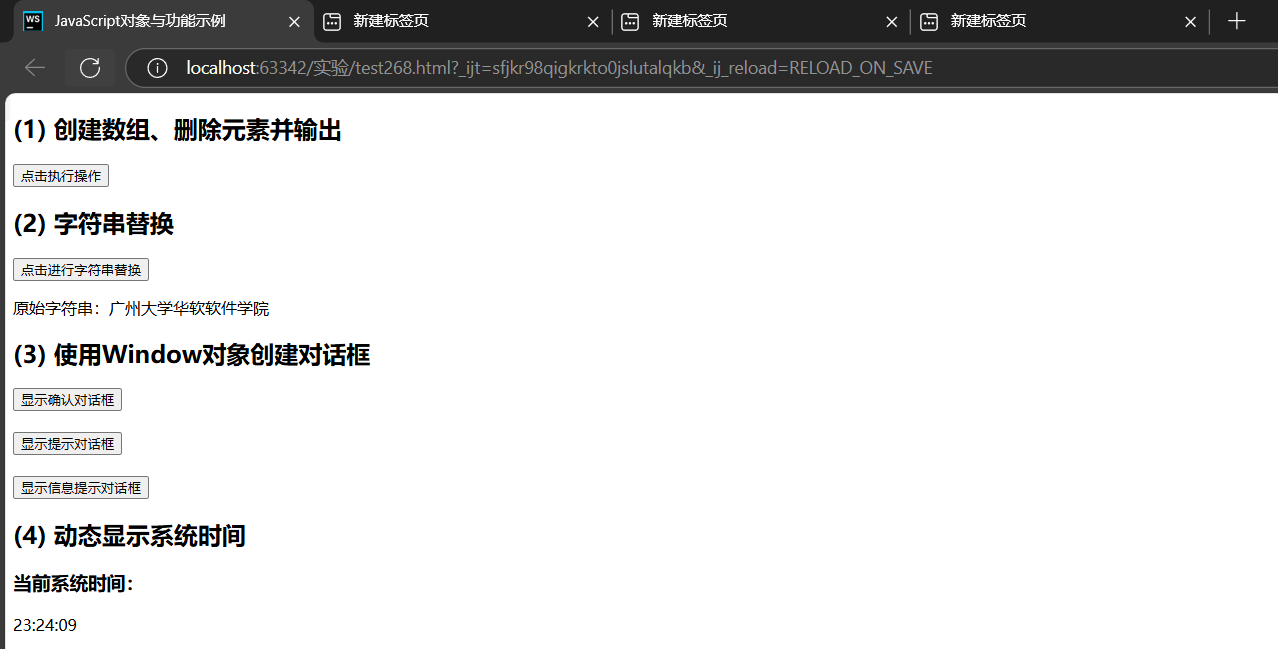
1. P268，2.上机题 （共 4 个小题）



**代码实现**

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>JavaScript对象与功能示例</title>  
</head>  
<body>  
  
<h2>(1) 创建数组、删除元素并输出</h2>  
<button onclick="handleArray()">点击执行操作</button>  
<p id="arrayOutput"></p>  
  
<h2>(2) 字符串替换</h2>  
<button onclick="replaceString()">点击进行字符串替换</button>  
<p id="stringOutput">原始字符串：广州大学华软软件学院</p>  
  
<h2>(3) 使用Window对象创建对话框</h2>  
<!-- 分开的按钮 -->  
<button onclick="showConfirmDialog()">显示确认对话框</button><br><br>  
<button onclick="showPromptDialog()">显示提示对话框</button><br><br>  
<button onclick="showAlertDialog()">显示信息提示对话框</button>  
  
<h2>(4) 动态显示系统时间</h2>  
<h3>当前系统时间：</h3>  
<p id="currentTime"></p>  
  
<script>  
 // (1) 创建数组、删除元素并输出  
 function handleArray() {  
 let initialArray = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];  
 let arrayAfterDelete = initialArray.slice();  
 let deletedElement = arrayAfterDelete.splice(1, 1)[0];  
  
 *document*.getElementById("arrayOutput").innerHTML =  
 `初始数组：${initialArray}<br>删除的元素值：${deletedElement}<br>删除后剩余元素的值：${arrayAfterDelete}`;  
 }  
  
 // (2) 字符串替换  
 function replaceString() {  
 let originalString = "广州大学华软软件学院";  
 let replacedString = originalString.replace(originalString, "JavaScript编程语言");  
 *document*.getElementById("stringOutput").innerText = `替换后的字符串：${replacedString}`;  
 }  
  
 // (3) 使用Window对象创建不同对话框  
 function showConfirmDialog() {  
 let confirmResult = *window*.confirm("这是一个确认对话框。点击确定或取消。");  
 if (confirmResult) *console*.log("用户点击了确定。");  
 else *console*.log("用户点击了取消。");  
 }  
  
 function showPromptDialog() {  
 let promptResult = *window*.prompt("这是一个提示对话框。请输入一些内容：", "");  
 if (promptResult !== null) {  
 *console*.log("用户输入的内容：", promptResult);  
 } else {  
 *console*.log("用户取消了输入。");  
 }  
 }  
  
 function showAlertDialog() {  
 *window*.alert("这是一个信息提示对话框。");  
 }  
  
 // (4) 动态显示系统时间  
 function displayTime() {  
 let date = new *Date*();  
 let timeString = date.toLocaleTimeString();  
 *document*.getElementById("currentTime").innerText = timeString;  
 }  
  
 setInterval(displayTime, 1000);  
 displayTime(); // 页面加载时立即显示当前时间  
</script>  
  
</body>  
</html>

**运行截图**



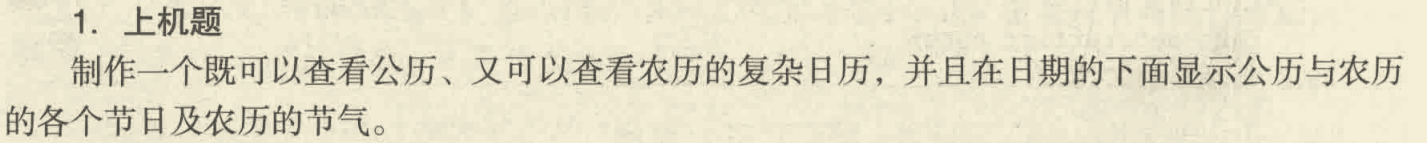
图形用户界面, 应用程序

AI 生成的内容可能不正确。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

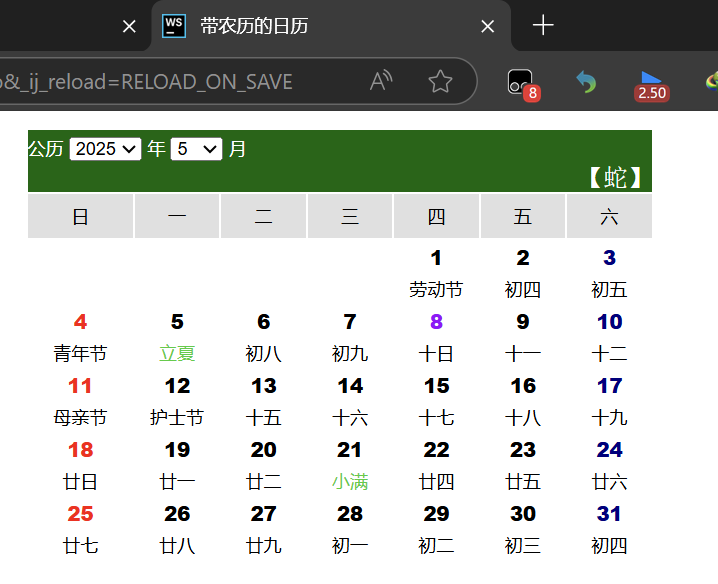
1. P288，1.上机题



**代码实现**

<!DOCTYPE html>  
<html>  
<head>  
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  
 <title>带农历的日历</title>  
  
<body onload=initial()>  
<center>  
 <form name=CLD>  
 <table>  
 <tr>  
 <td align=middle>  
 <table border=1 cellpadding="0" cellspacing="0" bordercolordark="#FFFFFF" bordercolor="#ffffff"  
 bordercolorlight="#EEEEEE">  
 <tr bgcolor="#006600">  
 <td colspan=7>  
 <font color=#ffffff style="FONT-SIZE: 9pt">公历  
 <select name=SY onchange=changeCld() style="FONT-SIZE: 9pt">  
 <script language="JavaScript">  
 for (i = 1900; i < 2050; i++) *document*.write('<option>' + i);  
 </script>  
 </select> 年 <select name=SM onchange=changeCld() style="FONT-SIZE: 9pt">  
 <script language="JavaScript">  
 for (i = 1; i < 13; i++) *document*.write('<option>' + i);  
 </script>  
 </select> 月 </font>  
 <font color=#ffffff face=宋体 id=GZ style="FONT-SIZE: 12pt"></font><BR>  
 </td>  
 </tr>  
 <tr align=middle bgColor=#e0e0e0>  
 <td width=54 style="font-size:9pt; padding:5pt;">日</td>  
 <td width=54 style="font-size:9pt ">一</td>  
 <td width=54 style="font-size:9pt ">二</td>  
 <td width=54 style="font-size:9pt ">三</td>  
 <td width=54 style="font-size:9pt ">四</td>  
 <td width=54 style="font-size:9pt ">五</td>  
 <td width=54 style="font-size:9pt ">六</td>  
 </tr>  
 <script type="text/javascript">  
 var *gNum*;  
 for (i = 0; i < 6; i++) {  
 *document*.write('<tr align=center>');  
 for (j = 0; j < 7; j++) {  
 *gNum* = i \* 7 + j;  
 *document*.write('<td id="GD' + *gNum* + '"><font id="SD' + *gNum* + '" size=2 face="Arial Black"');  
 if (j == 0) {  
 *document*.write('color=red');  
 }  
 if (j == 6) {  
 *document*.write('color=#000080');  
 }  
 *document*.write(' title=""></font><br><font id="LD' + *gNum* + '" size=2 style="font-size:9pt"> </font></td>');  
 }  
 *document*.write('</tr>');  
 }  
 </script>  
 </table>  
 </td>  
 </tr>  
 </table>  
 </form>  
</center>  
  
<script type="text/javascript">  
 var *lunarInfo* = new *Array*(  
 0x04bd8, 0x04ae0, 0x0a570, 0x054d5, 0x0d260, 0x0d950, 0x16554, 0x056a0, 0x09ad0, 0x055d2,  
 0x04ae0, 0x0a5b6, 0x0a4d0, 0x0d250, 0x1d255, 0x0b540, 0x0d6a0, 0x0ada2, 0x095b0, 0x14977,  
 0x04970, 0x0a4b0, 0x0b4b5, 0x06a50, 0x06d40, 0x1ab54, 0x02b60, 0x09570, 0x052f2, 0x04970,  
 0x06566, 0x0d4a0, 0x0ea50, 0x06e95, 0x05ad0, 0x02b60, 0x186e3, 0x092e0, 0x1c8d7, 0x0c950,  
 0x0d4a0, 0x1d8a6, 0x0b550, 0x056a0, 0x1a5b4, 0x025d0, 0x092d0, 0x0d2b2, 0x0a950, 0x0b557,  
 0x06ca0, 0x0b550, 0x15355, 0x04da0, 0x0a5d0, 0x14573, 0x052d0, 0x0a9a8, 0x0e950, 0x06aa0,  
 0x0aea6, 0x0ab50, 0x04b60, 0x0aae4, 0x0a570, 0x05260, 0x0f263, 0x0d950, 0x05b57, 0x056a0,  
 0x096d0, 0x04dd5, 0x04ad0, 0x0a4d0, 0x0d4d4, 0x0d250, 0x0d558, 0x0b540, 0x0b5a0, 0x195a6,  
 0x095b0, 0x049b0, 0x0a974, 0x0a4b0, 0x0b27a, 0x06a50, 0x06d40, 0x0af46, 0x0ab60, 0x09570,  
 0x04af5, 0x04970, 0x064b0, 0x074a3, 0x0ea50, 0x06b58, 0x055c0, 0x0ab60, 0x096d5, 0x092e0,  
 0x0c960, 0x0d954, 0x0d4a0, 0x0da50, 0x07552, 0x056a0, 0x0abb7, 0x025d0, 0x092d0, 0x0cab5,  
 0x0a950, 0x0b4a0, 0x0baa4, 0x0ad50, 0x055d9, 0x04ba0, 0x0a5b0, 0x15176, 0x052b0, 0x0a930,  
 0x07954, 0x06aa0, 0x0ad50, 0x05b52, 0x04b60, 0x0a6e6, 0x0a4e0, 0x0d260, 0x0ea65, 0x0d530,  
 0x05aa0, 0x076a3, 0x096d0, 0x04bd7, 0x04ad0, 0x0a4d0, 0x1d0b6, 0x0d250, 0x0d520, 0x0dd45,  
 0x0b5a0, 0x056d0, 0x055b2, 0x049b0, 0x0a577, 0x0a4b0, 0x0aa50, 0x1b255, 0x06d20, 0x0ada0)  
  
 var *solarMonth* = new *Array*(31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31);  
 var *Animals* = new *Array*("鼠", "牛", "虎", "兔", "龙", "蛇", "马", "羊", "猴", "鸡", "狗", "猪");  
 var *solarTerm* = new *Array*("小寒", "大寒", "立春", "雨水", "惊蛰", "春分", "清明", "谷雨", "立夏", "小满", "芒种", "夏至", "小暑", "大暑", "立秋", "处暑", "白露", "秋分", "寒露", "霜降", "立冬", "小雪", "大雪", "冬至");  
 var *sTermInfo* = new *Array*(0, 21208, 42467, 63836, 85337, 107014, 128867, 150921, 173149, 195551, 218072, 240693, 263343, 285989, 308563, 331033, 353350, 375494, 397447, 419210, 440795, 462224, 483532, 504758);  
 var *nStr1* = new *Array*('日', '一', '二', '三', '四', '五', '六', '七', '八', '九', '十');  
 var *nStr2* = new *Array*('初', '十', '廿', '卅');  
 //公历节日  
 var *sFtv* = new *Array*(  
 "0101 元旦",  
 "0214 情人节",  
 "0308 妇女节",  
 "0312 植树节",  
 "0315 消费者权益日",  
 "0401 愚人节",  
 "0501 劳动节",  
 "0504 青年节",  
 "0512 护士节",  
 "0601 儿童节",  
 "0701 建党节",  
 "0801 建军节",  
 "0910 教师节",  
 "0928 孔子诞辰",  
 "1001 国庆节",  
 "1006 老人节",  
 "1024 联合国日",  
 "1224 平安夜",  
 "1225 圣诞节")  
 //农历节日  
 var *lFtv* = new *Array*(  
 "0101 春节",  
 "0115 元宵节",  
 "0505 端午节",  
 "0707 七夕情人节",  
 "0715 中元节",  
 "0815 中秋节",  
 "0909 重阳节",  
 "1208 腊八节",  
 "1224 小年")  
 //返回农历y年的总天数  
 function lYearDays(y) {  
 var i, sum = 348;  
 for (i = 0x8000; i > 0x8; i >>= 1)sum += (*lunarInfo*[y - 1900] & i) ? 1 : 0;  
 return (sum + leapDays(y));  
 }  
 //返回农历y年闰月的天数  
 function leapDays(y) {  
 if (leapMonth(y)) return ((*lunarInfo*[y - 1900] & 0x10000) ? 30 : 29);  
 else return (0);  
 }  
 //判断y年的农历中那个月是闰月,不是闰月返回0  
 function leapMonth(y) {  
 return (*lunarInfo*[y - 1900] & 0xf);  
 }  
 //返回农历y年m月的总天数  
 function monthDays(y, m) {  
 return ((*lunarInfo*[y - 1900] & (0x10000 >> m)) ? 30 : 29);  
 }  
 //算出当前月第一天的农历日期和当前农历日期下一个月农历的第一天日期  
 function Dianaday(objDate) {  
 var i, leap = 0, temp = 0;  
 var baseDate = new *Date*(1900, 0, 31);  
 var offset = (objDate - baseDate) / 86400000;  
 this.dayCyl = offset + 40;  
 this.monCyl = 14;  
 for (i = 1900; i < 2050 && offset > 0; i++) {  
 temp = lYearDays(i)  
 offset -= temp;  
 this.monCyl += 12;  
 }  
 if (offset < 0) {  
 offset += temp;  
 i--;  
 this.monCyl -= 12;  
 }  
 this.year = i;  
 this.yearCyl = i - 1864;  
 leap = leapMonth(i); //闰哪个月  
 this.isLeap = false;  
 for (i = 1; i < 13 && offset > 0; i++) {  
 if (leap > 0 && i == (leap + 1) && this.isLeap == false) { //闰月  
 --i; this.isLeap = true; temp = leapDays(this.year);  
 }  
 else {  
 temp = monthDays(this.year, i);  
 }  
 if (this.isLeap == true && i == (leap + 1)) this.isLeap = false; //解除闰月  
 offset -= temp;  
 if (this.isLeap == false) this.monCyl++;  
 }  
 if (offset == 0 && leap > 0 && i == leap + 1)  
 if (this.isLeap) { this.isLeap = false; }  
 else { this.isLeap = true; --i; --this.monCyl; }  
 if (offset < 0) { offset += temp; --i; --this.monCyl; }  
 this.month = i;  
 this.day = offset + 1;  
 }  
 //返回公历y年m+1月的天数  
 function solarDays(y, m) {  
 if (m == 1)  
 return (((y % 4 == 0) && (y % 100 != 0) || (y % 400 == 0)) ? 29 : 28);  
 else  
 return (*solarMonth*[m]);  
 }  
 //记录公历和农历某天的日期  
 function calElement(sYear, sMonth, sDay, week, lYear, lMonth, lDay, isLeap) {  
 this.isToday = false;  
 //公历  
 this.sYear = sYear;  
 this.sMonth = sMonth;  
 this.sDay = sDay;  
 this.week = week;  
 //农历  
 this.lYear = lYear;  
 this.lMonth = lMonth;  
 this.lDay = lDay;  
 this.isLeap = isLeap;  
 //节日记录  
 this.lunarFestival = ''; //农历节日  
 this.solarFestival = ''; //公历节日  
 this.solarTerms = ''; //节气  
 }  
 //返回某年的第n个节气为几日(从0小寒起算)  
 function sTerm(y, n) {  
 var offDate = new *Date*((31556925974.7 \* (y - 1900) + *sTermInfo*[n] \* 60000) + Date.UTC(1900, 0, 6, 2, 5));  
 return (offDate.getUTCDate())  
 }  
 //保存y年m+1月的相关信息  
 var *fat* = mat = 9;  
 var *eve* = 0;  
 function calendar(y, m) {  
 *fat* = mat = 0;  
 var sDObj, lDObj, lY, lM, lD = 1, lL, lX = 0, tmp1, tmp2;  
 var lDPOS = new *Array*(3);  
 var n = 0;  
 var firstLM = 0;  
 sDObj = new *Date*(y, m, 1); //当月第一天的日期  
 this.length = solarDays(y, m); //公历当月天数  
 this.firstWeek = sDObj.getDay(); //公历当月1日星期几  
 if ((m + 1) == 5) { *fat* = sDObj.getDay() }  
 if ((m + 1) == 6) { mat = sDObj.getDay() }  
 for (var i = 0; i < this.length; i++) {  
 if (lD > lX) {  
 sDObj = new *Date*(y, m, i + 1); //当月第一天的日期  
 lDObj = new Dianaday(sDObj); //农历  
 lY = lDObj.year; //农历年  
 lM = lDObj.month; //农历月  
 lD = lDObj.day; //农历日  
 lL = lDObj.isLeap; //农历是否闰月  
 lX = lL ? leapDays(lY) : monthDays(lY, lM); //农历当月最后一天  
 if (lM == 12) { *eve* = lX }  
 if (n == 0) firstLM = lM;  
 lDPOS[n++] = i - lD + 1;  
 }  
 this[i] = new calElement(y, m + 1, i + 1, *nStr1*[(i + this.firstWeek) % 7], lY, lM, lD++, lL);  
 if ((i + this.firstWeek) % 7 == 0) {  
 this[i].color = 'red'; //周日颜色  
 }  
 }  
 //节气  
 tmp1 = sTerm(y, m \* 2) - 1;  
 tmp2 = sTerm(y, m \* 2 + 1) - 1;  
 this[tmp1].solarTerms = *solarTerm*[m \* 2];  
 this[tmp2].solarTerms = *solarTerm*[m \* 2 + 1];  
 if ((this.firstWeek + 12) % 7 == 5) {//黑色星期五  
 this[12].solarFestival += '黑色星期五';  
 }  
 if (y == *tY* && m == *tM*) {  
 this[*tD* - 1].isToday = true; //今日  
 }  
 }  
 //用中文显示农历的日期  
 function cDay(d) {  
 var s;  
 switch (d) {  
 case 10:  
 s = '初十';  
 break;  
 case 20:  
 s = '二十';  
 break;  
 case 30:  
 s = '三十';  
 break;  
 default:  
 s = *nStr2*[*Math*.floor(d / 10)];  
 //s += nStr1[d % 10];  
 s += *nStr1*[parseInt(d % 10)];  
 break;  
 }  
 return (s);  
 }  
 //在表格中显示公历和农历的日期,以及相关节日  
 var *cld*;  
 function drawCld(SY, SM) {  
 var TF = true;  
 var p1 = p2 = "";  
 var i, sD, s, size;  
 *cld* = new calendar(SY, SM);  
 GZ.innerHTML = '                       【' + *Animals*[(SY - 4) % 12] + '】'; //生肖  
 for (i = 0; i < 42; i++) {  
 sObj = eval('SD' + i);  
 lObj = eval('LD' + i);  
 sObj.className = '';  
 sD = i - *cld*.firstWeek;  
 if (sD > -1 && sD < *cld*.length) { //日期内  
 sObj.innerHTML = sD + 1;  
 if (*cld*[sD].isToday) {  
 sObj.style.color = '#9900FF'; //今日颜色  
 } else {  
 sObj.style.color = '';  
 }  
 if (*cld*[sD].lDay == 1) { //显示农历月  
 lObj.innerHTML = '<b>' + (*cld*[sD].isLeap ? '闰' : '') + *cld*[sD].lMonth + '月' + (monthDays(*cld*[sD].lYear, *cld*[sD].lMonth) == 29 ? '小' : '大') + '</b>';  
 }  
 else { //显示农历日  
 lObj.innerHTML = cDay(*cld*[sD].lDay);  
 }  
 var Slfw = Ssfw = null;  
 s = *cld*[sD].solarFestival;  
 for (var ipp = 0; ipp < *lFtv*.length; ipp++) { //农历节日  
 if (parseInt(*lFtv*[ipp].substr(0, 2)) === parseInt(*cld*[sD].lMonth)) {  
 if (parseInt(*lFtv*[ipp].substr(2, 4)) === parseInt(*cld*[sD].lDay)) {  
 lObj.innerHTML = *lFtv*[ipp].substr(5);  
 Slfw = *lFtv*[ipp].substr(5);  
 }  
 }  
 if (12 == (*cld*[sD].lMonth)) { //判断是否为除夕  
 if (*eve* == (*cld*[sD].lDay)) { lObj.innerHTML = "除夕"; Slfw = "除夕"; }  
 }  
 }  
 for (var ipp = 0; ipp < *sFtv*.length; ipp++) { //公历节日  
 if (parseInt(*sFtv*[ipp].substr(0, 2)) == (SM + 1)) {  
 if (parseInt(*sFtv*[ipp].substr(2, 4)) == (sD + 1)) {  
 lObj.innerHTML = *sFtv*[ipp].substr(5);  
 Ssfw = *sFtv*[ipp].substr(5);  
 }  
 }  
 }  
 if ((SM + 1) == 5) { //母亲节  
 if (*fat* == 0) {  
 if ((sD + 1) == 7) { Ssfw = "母亲节"; lObj.innerHTML = "母亲节" }  
 }  
 else if (*fat* < 9) {  
 if ((sD + 1) == ((7 - *fat*) + 8)) { Ssfw = "母亲节"; lObj.innerHTML = "母亲节" }  
 }  
 }  
 if ((SM + 1) == 6) { //父亲节  
 if (mat == 0) {  
 if ((sD + 1) == 14) { Ssfw = "父亲节"; lObj.innerHTML = "父亲节" }  
 }  
 else if (mat < 9) {  
 if ((sD + 1) == ((7 - mat) + 15)) { Ssfw = "父亲节"; lObj.innerHTML = "父亲节" }  
 }  
 }  
 if (s.length <= 0) { //设置节气的颜色  
 s = *cld*[sD].solarTerms;  
 if (s.length > 0) s = s.fontcolor('limegreen');  
 }  
 if (s.length > 0) { lObj.innerHTML = s; Slfw = s; } //节气  
 if ((Slfw != null) && (Ssfw != null)) {  
 lObj.innerHTML = Slfw + "/" + Ssfw;  
 }  
 }  
 else { //非日期  
 sObj.innerHTML = '';  
 lObj.innerHTML = '';  
 }  
 }  
 }  
 //在下拉列表中选择年月时,调用自定义函数drawCld(),显示公历和农历的相关信息  
 function changeCld() {  
 var y, m;  
 y = CLD.SY.selectedIndex + 1900;  
 m = CLD.SM.selectedIndex;  
 drawCld(y, m);  
 }  
 //用自定义变量保存当前系统中的年月日  
 var *Today* = new *Date*();  
 var *tY* = *Today*.getFullYear();  
 var *tM* = *Today*.getMonth();  
 var *tD* = *Today*.getDate();  
 //打开页时,在下拉列表中显示当前年月,并调用自定义函数drawCld(),显示公历和农历的相关信息  
 function initial() {  
 CLD.SY.selectedIndex = *tY* - 1900;  
 CLD.SM.selectedIndex = *tM*;  
 drawCld(*tY*, *tM*);  
 }  
</script>  
</body>  
  
</html>

**运行截图**



图形用户界面

AI 生成的内容可能不正确。

指导教师签名： 李震雄