

Mel's Calendar Utility

History:

The first version of the Calendar Utility was written in 1965 to calculate the number of days between two dates on a programmable calculator.

The next version was written in C language in 1987 and was expanded to display the calendars as well as calculate days between dates. It was a perpetual calendar of sorts and accounted for the UK adjustment in 1752 when September 3 thru 14 were skipped to more closely align with the solar year.

This present version was written in 2009 to experiment with using html strictly for content, external css for all formatting and presentation, and javascript to create dynamic content. It also includes an experiment with rounded corners and transparency (opacity). The CSS rules for opacity have been tested with browsers Avant 11.7, Chrome 2.0, IE 7, IE 8, Maxthon 1.6.3, Maxthon 2.5.2, Firefox 3.0.7, Netscape 7.1, Opera 9.52 and Safari 3.2.2. All worked satisfactorily even though opacity rules are not part of the current W3C CSS(2) standard.

Instructions:

Enter one or two dates in the format mm/dd/yyyy or only one year (yyyy).

This utility will display a calendar for each date or, if only one year is supplied, the full year calendar will be displayed. If two dates are supplied, it also reports the days between the dates.

If no dates are supplied, it will display the calendar for the current system date.

The results are displayed in a separate window. The generated content is formatted HTML.

The source code may be saved in some browsers (IE, FF). After it is saved, it may be edited, but please do not alter or remove the copyright notice.

Files:

calendar.html	(Calendar Utility input form)
calendar.css	(style rules for Calendar Utility and generated calendar pages)
calendar.js	(javascript generates html calendar content)
Images/lake.jpg	(background-image)
images/upper.gif	(upper image with rounded corners)
images/middle.gif	(expandable middle image for the container)
images/lower.gif	(lower image with rounded corners)
documentation.pdf	(this file)
calendar.exe	(Downloadable archive containing this entire application)

calendar.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=us-ascii">
```

```
<title>Calendar Utility (updated 8/1/2009 12:32am)</title>
```

```
<link rel="stylesheet" href="calendar.css" type="text/css">
```

```
<script type="text/javascript" src="calendar.js"></script>
```

```
</head>
```

```
<body>
```

```
<div id="wrapper">
```

```
<div id="mainarea">
```

```
<div class="container transparent">
```

```
<div class="upper transparent">
```

```
<h1>Calendar Utility</h1>
```

```
</div>
```

```
<div class="middle transparent">
```

```
<div class="textbox">
```

Enter one or two dates in the format mm/dd/yyyy or only one year (yyyy). This utility will display a calendar for each date or If only one year is supplied, the full year calendar will be displayed. If two dates are supplied, it also reports the days between the dates. If no dates are supplied, it will display the calendar for the current system date.

```
</div><br><br>
```

```
<div class="normal">
```

```
<form method='get' name='form' action='calendar.html' onsubmit='calendar(this);' id="form">
```

```
Enter one or two dates (mm/dd/yyyy) or (yyyy)<br>
```

```
<input type='text' name='arg' size='20'> <button type="button" name="go" onclick='calendar(this);'>Click to
```

```
Submit</button>
```

```
</form>
```

```
<p>
```

<documentation.pdf>
[Download Calendar Utility](calendar.exe)

</p>

</div>

</div>

<div class='lower copyright transparent'>

© Calendar Utility 2009 Mel Knogle

</div>

</div><!-- End of container -->

</div><!-- End of mainarea -->

</div>

</body>

</html>

calendar.css

```
/* Calendar Utility Style Sheet (updated 8/1/2009 12:32am) */
```

```
/* Tag Rules */
```

```
body {  
  text-align: center; margin: auto;font-family: Verdana, Arial, Helvetica, sans-serif;  
  background: url(images/photo.jpg) no-repeat top left;  
}  
table, tr, th, td {border: 1px solid #890000;}  
h1 {font-size: 1.2em; font-weight: 900; line-height: 0.8em; margin-top: 4px;}  
strong {background-color: transparent; color: #890000; font-weight: 900;}  
button { font-weight: bold; margin-top: 6px;}
```

```
/* Class Rules */
```

```
.form, .calendar, .summary, .links, .single, .double, .quad {text-align: center; margin: auto auto;}  
.calendar {float: left; margin: 2px; font-size: 1em; line-height: 1em; background-color: #c0c0c0; color: #000000; table-layout: fixed; width:  
11.25em; height: 10.625em; text-align: center; border: 1px solid #890000; border-collapse: collapse;}  
.header { background-color: #88c000; color: #000000;}  
.hilite { font-size: 0.8em; background-color: #c0c0c0; color: #000000;}  
.today {background-color: #88c000; color: #000000;}  
.skip {background-color: #6e6b69; color: #ffffff;}  
.summary {font-size: 0.9em; padding-top: 4px;}  
.copyright {font-size: 0.8em; font-weight: normal;}  
.single { width: 11.375em; height: 11.25em;}  
.double { width: 23.375em; height: 11.25em;}  
.quad { width: 46.75em; height: 32.5em;}  
.button { font-family: inherit; border: 2px outset #fafafa; background-color: #c0c0c0; color: #000000; text-decoration: none; padding: 2px;}  
.upper { background: url(images/upper.gif) no-repeat top left; height: 31px;}  
.middle { background: url(images/middle.gif) repeat-y center left; margin-top: -8px;}  
.lower { background: url(images/lower.gif) no-repeat bottom left; margin-top: -8px; height: 31px;}  
.container { width: 100%; min-height: 220px; color: inherit; font-weight: 900;}
```

```
.textbox { clear: both; text-align: justify; padding-left: 10px; padding-right: 20px;}
.emphasize { font-weight: bolder; font-size: large; color: #890000;}
.normal { font-size: 1em; background-color: inherit; color: #000000;}

.transparent { width: 100%; padding: 4px;
  /* The following opacity rules work in most major browsers but are NOT part of the current W3c standard. */
  filter:alpha(opacity=90);          /* opacity setting for IE types          (0 thru 100%) */
  -moz-opacity: 0.90;              /* opacity setting for Moz and ns      (0 thru 1.0) */
  opacity: 0.90;                   /* opacity setting for Opera and Safari (0 thru 1.0) */
}

/* ID Rules */
#wrapper { text-align: center; margin: auto auto; width: 966px; min-height: 548px;}
#mainarea { text-align: center; margin: auto auto; width:760px; min-height:240px;}
```

calendar.js

```
/* Calendar Utility (c) 2009 Mel Knoyle  calendar.js (do not remove this notice) (updated 8/1/2009 12:32am)
   (This utility works with the Gregorian Calendar (Oct 1582 - 4000))
```

```
/*global ActiveXObject, form, window, document, alert */
```

```
function leapadj(yyyy) { // leapyear test
  if (((yyyy % 4) === 0) && ((yyyy % 100) !== 0) || ((yyyy % 400) === 0)) {
    return 1;
  } else {
    return 0;
  }
}

function daysInMonth(month, year) {
  var leap = leapadj(year);
  if (month === 1) {
    return (leap + 28);
  }
  if (month === 3 || month === 5 || month === 8 || month === 10) {
    return (30);
  } else {
    return (31);
  }
}

function monthTable(cal, mday, month, year, startdate, weekday, maxdays, full) {
  var monthname = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December'];
  var nextday = 1; // days in the month start with 1
  var inmonth = 0; // days prior to first = false (0) 1st and after is true (1)
  var i = 0; // counter
  cal += ' <table class="calendar">\n';
  cal += ' <tr class="header"><th colspan = "7">' + monthname[month] + ' ' + year + '</th></tr>';
  cal += ' c a l + = ' \ n
```

< t r

```

class="hilite"><th>Su</th><th>Mo</th><th>Tu</th><th>We</th><th>Th</th><th>Fr</th><th>Sa</th></tr>';
while (nextday <= maxdays) {
  for (i = 0; (i < 7); i++) {
    if (i === 0) {
      cal += '\n          <tr>';
    }
    inmonth += (i === weekday);

    if ((full === 0) && (nextday === mday) && inmonth) {
      cal += '<td class="today">';          // highlight the target day of the month
    } else {
      if ((year === 1752) && (month === 8) && ((nextday > 2) && (nextday < 14))) {
        cal += '<td class="skip">';          // this day was skipped in Sep 1752
      } else {
        if ((year === 1582) && (month === 9) && ((nextday > 4) && (nextday < 15))) {
          cal += '<td class="skip">';          // this day was skipped in Sep 1582
        } else {
          if ((year === 4000) && (month === 1) && (nextday === 29)) {
            cal += '<td class="skip">';          // this day may be skipped in Feb 4000
          } else {
            cal += '<td>';
          }
        }
      }
    }
  }
}

if (((i < weekday) && (nextday === 1)) || (nextday > maxdays)) {
  cal += ' ';          // before the 1st or after the last day of the month
} else {
  cal += ' ' + nextday + ' ';
  nextday += 1;
}

```

```

        cal += '<\td>';
        if (i === 6) {
            cal += '<\tr>';
        }
    }
}
cal += '\n          <\table>\n';
return cal;
}
function calendar(form) {
    var monthname = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December'];
    var cmd = "";           // user input command string
    var cal = "";          // contains calendar in html format
    var container = "";    // contains css class for calendar area
    var i = 0;             // counter
    var mm = 0;            // month number
    var m = 0;             // months between dates
    var days = 0;          // days between dates
    var yr = 0;            // years between dates
    var daysleft = 0;      // days remaining after calculating years and months between
    var d1 = 0;            // day of the month (first date)
    var m1 = 0;            // month number (0 thru 11) first date
    var y1 = 0;            // year (4 digits) first date
    var myDate = new Date(); // default date = system date
    var nextDate;          // second date identified in the command argument
    var ms = 86400000;     // milliseconds in one day
    var dnoun = "";        // day or days
    var str = document.form.arg.value; // user supplied argument (1 or 2 dates formatted mm/dd/yyyy)
    var dstring = str.replace(/^\s+/g, ""); // trim all leading spaces
    dstring = str.replace(\s+/g, " "); // replace multiple spaces between dates with single space
    dstring = str.replace(\|-/g, " "); // replace forward slashes and dashes with spaces
    cmd = dstring.split(" "); // separate date string elements

```

```

var mday = 0; // the day of the month.
var month = 0; // the month number (0 thru 11).
var year = 0; // the year (4 digits).
var startdate = 0; // ms for day 1 of month
var weekday = 0; // month starts on weekday (0 thru 6)
var leap = 0; // leap year is true (1) or false (0)
var maxdays = 0; // number of days in month
var full = 0; // full year calendar true (1) or false (0)
for (i = 0; i < 6; i++) {
    cmd[i] = parseInt(cmd[i], 10); // convert input string values to numeric
}
if (cmd[2] && cmd[5]) {
    container = "double"; // css double class reserves space for 2 months
} else {
    if (str.length === 4) {
        container = "quad"; // css quad class reserves space for 4 months
        full = 1;
    } else {
        container = "single"; // css single class reserves space for 1 month
    }
}
if (cmd[0] && (str.length > 4)) { // Validate the first date.
    if ((cmd[0] > 0) && (cmd[0] <= 12) && (cmd[1] <= 31) && (cmd[2] > 999)) { // if year, month and day are valid
        myDate = new Date(cmd[2], (cmd[0] - 1), cmd[1]); // myDate is new date object
    } else {
        alert(cmd[0] + "/" + cmd[1] + "/" + cmd[2] + " = wrong format or invalid date.");
        return;
    }
}
if (cmd[3]) { // Validate the second date, if given.
    if ((cmd[3] > 0) && (cmd[3] <= 12) && (cmd[4] <= 31) && (cmd[5] > 999)) { // if year, month and day are valid
        nextDate = new Date(cmd[5], (cmd[3] - 1), cmd[4]); // nextDate is new date object
    }
}

```

```

    } else {
        alert(cmd[3] + "/" + cmd[4] + "/" + cmd[5] + " = wrong format or invalid date.");
        return;
    }
}
if (nextDate) {
    days = Math.abs(Math.round((nextDate - myDate) / ms));
    if ((Math.round(myDate / ms) < -141437) && (Math.round(nextDate / ms) > -141428)) {
        days = days - 10;
    }
    if ((Math.round(myDate / ms) < -79377) && (Math.round(nextDate / ms) > -79367)) {
        days = days - 11;
    }
}

// display calendar information in a new window
var calpage = window.open("", 'calendar', 'width=900, height=640, top=28, left=0, resizable=1, scrollbars=1, menubar = 1, directories = 1');
calpage.document.open('text/html', '_self');
// create html for calendar display
calpage.document.writeln('<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">');
calpage.document.writeln('<html>');
calpage.document.writeln(' <head>');
calpage.document.writeln(' <title>Calendar</title>');
calpage.document.writeln(' <meta http-equiv="Content-Type" content="text/html; charset=us-ascii">');
calpage.document.writeln(' <link rel="stylesheet" href="calendar.css" type="text/css">');
calpage.document.writeln(' </head>');
calpage.document.writeln(' <body>');
calpage.document.writeln(' <div id="wrapper">');
calpage.document.writeln(' <div id="mainarea">');
calpage.document.writeln(' <div class="upper transparent">');
calpage.document.writeln(' <h1>Calendar Utility</h1>');
calpage.document.writeln(' </div><!-- End of upper transparent -->');

```

```

calpage.document.writeln('                <div class="middle transparent">');
calpage.document.writeln('                <div class="" + container + "">');
if (str.length === 4) {
    // display full year calendar
    for (i = 0; i < 12; i++) {
        myDate = new Date(str, i, 1);           // convert str to date object
        mday = myDate.getDate();               // the day of the month
        month = myDate.getMonth();            // the month number (0 thru 11).
        year = myDate.getFullYear();          // the year (4 digits).
        startdate = new Date(year, month, 1);  // ms for day 1 of month
        weekday = startdate.getDay();         // month starts on weekday (0 thru 6)
        leap = leapadj(year);                 // leap year is true (1) or false (0)
        maxdays = daysInMonth(month, year);  // number of days in month
        cal = monthTable(cal, mday, month, year, startdate, weekday, maxdays, full);
    }
} else {
    // display calendar for 1st month entered
    mday = myDate.getDate();
    month = myDate.getMonth();
    year = myDate.getFullYear();
    startdate = new Date(year, month, 1);
    weekday = startdate.getDay();
    leap = leapadj(year);
    maxdays = daysInMonth(month, year);
    cal = monthTable(cal, mday, month, year, startdate, weekday, maxdays, full);
    if (nextDate) {
        // prepare calendar for second month
        d1 = mday;
        // save old day of the month
        mday = nextDate.getDate();           // Get the day of the month.
        m1 = month;
        month = nextDate.getMonth();        // Get the month number (0 thru 11).
        y1 = year;
        year = nextDate.getFullYear();      // Get the year (4 digits).
        startdate = new Date(year, month, 1);
        weekday = startdate.getDay();
    }
}

```

```

    leap = leapadj(year);
    maxdays = daysInMonth(month, year);
    cal = monthTable(cal, mday, month, year, startdate, weekday, maxdays, full);
}
}
calpage.document.write(cal);
calpage.document.writeln("          </div><!-- End of single double or quad -->");
if (nextDate) {
    calpage.document.writeln("          <div class='summary'>");
    calpage.document.write("          There ");
    if (days === 1) {
        calpage.document.write("is ");
        dnoun = "day";
    } else {
        calpage.document.write("are ");
        dnoun = "days";
    }
    calpage.document.writeln(days + " " + dnoun + " between " + monthname[m1] + " " + d1 + ", " + y1 + " and " + monthname[cmd[3] - 1]
+ " " + cmd[4] + ", " + cmd[5] + ".");
    yr = cmd[5] - cmd[2];           // calculate years between (2nd year - 1st year)
    if (cmd[0] > cmd[3]) {         // if 1st month > 2nd month, last year is less than 12 months
        yr = yr - 1;
        m = 12 - cmd[0] + cmd[3];   // calculate months between (12 - 1st month + 2nd month)
    } else {
        m = cmd[3] - cmd[0];        // if 1st month < 2nd month (month = 2nd month - 1st month)
    }
    // adjust months and years
    if (cmd[4] < cmd[1]) {         // if 2nd monthdate < 1st monthdate, last month is less than a full month
        m = m - 1;
        if (cmd[3] === cmd[0]) {   // if 2nd month = 1st month, last year is less than 12 months
            yr = yr - 1;
            m = 11;

```

```

    }
}
daysleft = cmd[4] - cmd[1];           // calculate days between (2nd monthdate - 1st monthdate)
if (daysleft < 0) {
    mm = cmd[3] - 1;                   // mm = 1 month prior to 2nd month note cmd[3] contains (1 thru 12)
    if (mm === 0) {                   // december of previous year
        mm = 12;                     // mm = december
        leap = leapadj(cmd[5] - 1);   // if prior year is a leap year, leap is true (1). else leap is false (0)
    } else {
        leap = leapadj(cmd[5]);       // if 2nd year is a leap year.
    }
    maxdays = daysInMonth(mm - 1, year); // mm - 1 (months 0 thru 11)
    daysleft = maxdays + daysleft;
}
// display years, months and days between dates
if ((days > 1) && ((m > 0) || (yr > 0))) {
    calpage.document.write("<br>" + days + " " + dnoun + " = ");
    if (yr) {
        if (yr === 1) {
            calpage.document.write(yr + " year ");
        } else {
            calpage.document.write(yr + " years ");
        }
    }
}
if (m) {
    if (m === 1) {
        calpage.document.write(m + " month ");
    } else {
        calpage.document.write(m + " months ");
    }
}
if (daysleft) {

```

```

    if (daysleft === 1) {
        calpage.document.write(" and " + daysleft + " day");
    } else {
        calpage.document.write(" and " + daysleft + " days");
    }
}
}
calpage.document.writeln("\n        </div>");
}
if (((str.length === 4) && (str === "1582")) || (cmd[2] === 1582) || ((cmd[5] && (cmd[5] === 1582)))) {
    calpage.document.writeln("        <div class='textbox'>");
    calpage.document.writeln("            <br><strong>Note: </strong>");
    calpage.document.writeln("            In October 1582, Pope Gregory ordered 10 days <span class='skip'>( 5 thru 14 )</span> be omitted
to more closely align with the vernal equinox.");
    calpage.document.writeln("        </div>");
}
if (((str.length === 4) && (str === "1752")) || (cmd[2] === 1752) || ((cmd[5] && (cmd[5] === 1752)))) {
    calpage.document.writeln("        <div class='textbox'>");
    calpage.document.writeln("            <br><strong>Note:</strong>");
    calpage.document.writeln("            11 days <span class='skip'>( 3 thru 13 )</span> were omitted in September 1752 when the UK
converted to the Gregorian calendar.");
    calpage.document.writeln("        </div>");
}
if ((cmd[2] >= 4000) || (cmd[5] >= 4000) || (str >= 4000)) {
    calpage.document.writeln("        <div class='textbox'>");
    calpage.document.writeln("            <br><strong>Note:</strong>");
    calpage.document.writeln("            There will be a calendar error of approximately one day by the year 4000.<br>Feb 29, 4000, may
be skipped.");
    calpage.document.writeln("        </div>");
}
calpage.document.writeln("        </div><!-- End of middle transparent -->");
calpage.document.writeln("        <div class='lower copyright transparent'>");

```

```
calpage.document.writeln("          <br>&copy; Calendar Utility 2009 Mel Knoyle");
calpage.document.writeln("        </div><!-- End of lower copyright transparent");
calpage.document.writeln("      </div><!-- End of mainarea -->");
calpage.document.writeln("    </div><!-- End of wrapper -->");
calpage.document.writeln("  </body>");
calpage.document.writeln("</html>");
calpage.document.close();
}
```