88993

Miscellaneous code I use on my websites

JavaScript ****

Author:

[**Michael D. Newby**](http://www.recursivecreativity.com/)

[www.recursivecreativity.com](http://www.recursivecreativity.com)

[www.excelusersgroup.org](http://www.excelusersgroup.org)

(not active)

Last Updated:

**August 2, 2021**

Table of Contents

[Web Page Footer 4](#_Toc100088113)

[Copyright 4](#_Toc100088114)

[Send E-mail Link 5](#_Toc100088115)

[Meeting Dates 6](#_Toc100088116)

[Weekday.js 6](#_Toc100088117)

[**GetWeekDay** 7](#_Toc100088118)

[**GetNthWeekday** 7](#_Toc100088119)

[Current, Next Meeting 8](#_Toc100088120)

[**WeekdayText** 8](#_Toc100088121)

[**MeetingDate (current month)** 9](#_Toc100088122)

[**MeetingDate (next month)** 10](#_Toc100088123)

[Meetings Pages 11](#_Toc100088124)

[**Main Menu** 11](#_Toc100088125)

[**Meetings Menu** 12](#_Toc100088126)

[**Meetings Menu - Header** 12](#_Toc100088127)

[**Meetings Menu - Years** 13](#_Toc100088128)

[**Meetings Content** 13](#_Toc100088129)

[**MeetingDate** 14](#_Toc100088130)

[**DayOfWeek** 15](#_Toc100088131)

Each heading below is link back to this page, the Table of Contents (TOC).

The HTML comments (<!-- -->) embedded within the JavaScript code ensures the code is ignored by HTML "[validators](https://validator.w3.org/check?uri=http%3A%2F%2Fwww.recursivecreativity.com%2F;accept=text%2Fhtml%2Capplication%2Fxhtml%2Bxml%2Capplication%2Fxml%3Bq%3D0.9%2Cimage%2Fwebp%2C%2A%2F%2A%3Bq%3D0.8;accept-language=en-US%2Cen%3Bq%3D0.5)". I try my best to adhere to “Strict” coding.

At the bottom of each web page on my websites are images that provide links to HTML and CSS validation sites. Each page will be validated if either image is clicked.



Web pages that use most of the JavaScripts in examples mentioned:

<http://www.excelusersgroup.org/default.html>

<http://www.excelusersgroup.org/pages/page-meetings_2021.html>

# [Web Page Footer](#TOC)

## [Copyright](#TOC)

This code first produces the copyright symbol with function that is then followed by "2006 – current year" whereby current year automatically updates from system date of web server at the stroke of midnight each year on January 1.



<h3>&copy;

<script type="text/javascript">

 var CurrentYear=new Date();

 document.write(" 2006-" + CurrentYear.getFullYear() + " ");

</script>

Please note change above from previous version: getFullYear must be used in most recent versions of Google Chrome, Mozilla Firefox and Microsoft Internet Explorer.

An attorney once told me it is good practice to include range of dates and keep it updated. Using the range and updated year protects any content originally posted and covers any content posted thereafter.

The hard-coded 2006 above is first year I had website.

## [Send E-mail Link](#TOC)

This code prevents certain programs from sucking out the e-mail address on web pages and adding the e-mail address to spam lists.



The code uses variables to concatenate the e-mail address used for "contact us" e-mail (mailto:) link.

<script type="text/javascript">

 <!--

 var fname = "Michael";

 var char1 = "@";

 var domain = "RecursiveCreativity";

 var char2 = ".";

 var suf = "com";

 var email = fname + char1 + domain + char2 + suf;

 document.write('<a href="mailto:' + email + '">' + 'Michael D. Newby</a>');

 -->

</script>

**Alternative code to above:**

<script type="text/javascript">

 function WriteEmail(fname, domain, suf, DisplayContent)

 {

 <!--

 document.write('<a href="mailto:' +

 fname + '@' + domain + '.' + suf + '">' + DisplayContent + '</a>');

 -->

 }

</script>

HTML Code calling JavaScript Function:

<script type="text/javascript" src="Scripts/emailpass.js">

WriteEmail("Michael", "@", "RecursiveCreativity", ".", "com", "Michael D. Newby");

</script>

# [Meeting Dates](#TOC)

The following functions are used to automatically generate meeting dates using the following variables:

* Day of week (e.g. Sunday – Saturday = 1 – 7; Thursday = 5; weekday)
* Nth occurrence of weekday in month (e.g. 3rd Thursday)

## [Weekday.js](#TOC)

I use the following code to reference .js file that contains functions for calling the weekday and nth occurrence of weekday in month for meetings. Thus, I do not have to change each web page…unless it changes. If the weekday changes for a single meeting or permanently, I would just need to call respective function for that weekday.

 <script type="text/javascript" src="../files/WeekDay.js"></script>



The following table includes the functions found in above embedded file.

|  |  |
| --- | --- |
| function jsGetWeekDay1(){return '1';} | function jsGetNthWeekday1(){return '1'} |
| function jsGetWeekDay2(){return '2';} | function jsGetNthWeekday2(){return '2'} |
| function jsGetWeekDay3(){return '3';} | function jsGetNthWeekday3(){return '3'} |
| function jsGetWeekDay4(){return '4';} | function jsGetNthWeekday4(){return '4'} |
| function jsGetWeekDay5(){return '5';} | function jsGetNthWeekday5(){return '5'} |
| function jsGetWeekDay6(){return '6';} |  |
| function jsGetWeekDay7(){return '7';} |  |

These functions are used as global variables on web page. They are used by other functions to calculate meeting dates.

### **[GetWeekDay](#TOC)**

This returns the numeric day of week to be used in calculating meeting dates. From example below, 5th day of the week is Thursday.

<script type="text/javascript">

 function GetWeekday()

 {

 var Weekday = **jsGetWeekDay5**();

 return ( Weekday );

 }

</script>

### [**GetNthWeekday**](#TOC)

This returns the which occurrence of weekday in given month, the meeting will take place. From example below, meeting will take on the 3rd occurrence of a given weekday.

<script type="text/javascript">

 function GetNthWeekday()

 {

 var nthWeekday = **jsGetNthWeekday3**();

 return ( nthWeekday );

 }

</script>

## [Current, Next Meeting](#TOC)

The [home page](http://excelusersgroup.org/) will automatically display the details of the meeting for the current month.

### [**WeekdayText**](#TOC)

This function returns the day (text; long as opposed to short, Thur) of the week the meeting for the current month will occur.



<script type="text/javascript">

 function WeekdayText()

 {

 var Weekday = GetWeekday();

var myArray = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday'];

 return ( myArray[Weekday-1] );

 }

</script>

### [**MeetingDate**](#TOC) **(current month)**

This function calculates current meeting date and is used to notify members on home page of meeting for the current month.



<script type="text/javascript">

 function MeetingDate(CurrentMonth)

 {

 var d = new Date();

var CurrentYear = d.getFullYear();

var nthWeekday = GetnthWeekday();

var WeekDay = GetWeekday();

return (nthWeekday-1) \* 7 + 1 + (7 + WeekDay - DayOfWeek((nthWeekday-1) \* 7 + 1,CurrentMonth,CurrentYear)) %7 + ", " + CurrentYear;

 }

</script>

### [**MeetingDate**](#TOC) **(next month)**

This function calculates meeting date in month following current month (next meeting).



<script type="text/javascript">

 function NextMonthText()

 {

 var d = new Date();

var myArray = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December'];

 if (MonthCurrent() + 1 == 12)

 {

 var CurrentMonth = 1;

 }

 else

 {

 var CurrentMonth = d.getMonth() + 1;

 }

 return (myArray[CurrentMonth]);

 }

</script>

## [Meetings Pages](#TOC)

### [**Main Menu**](#TOC)

This script automates selection of Meetings page for current year. For example, in 2021, clicking on Meetings will automatically display the Meetings page for 2021.



<script type="text/javascript">

 <!--

 var x = CurYear();

document.write( '<a class="current" id="meeting\_link" href="http://www.excelusersgroup.org/pages/page-meetings\_' + x + '.html">Meetings</a>');

 -->

</script>

### [**Meetings Menu**](#TOC)

#### [**Meetings Menu - Header**](#TOC)

This calculates the date associated with header on Meetings pages for each year.



<script type="text/javascript">

 <!--

 function Right(str, n)

 {

 if (n <= 0)

 return "";

 else if (n > String(str).length)

 return str;

 else

 {

 var iLen = String(str).length;

 return String(str).substring(iLen, iLen - n);

 }

}

 document.write( Right(MeetingDate(1), 4 ));

 -->

</script>

MEETINGS

#### **[Meetings Menu - Years](#TOC)**

This script generates the 10-year menu found on the Meetings page for each year in given period.



<script type="text/javascript">

 <!--

 var BaseYear = 2020; // This is base year for first 10-year menu 2021 through 2030

 var d=new Date();

 var CurrentYear = d.getYear(); // Gets Current Year

 var myYears=new Array();

 if (CurrentYear - BaseYear > 10)

 {

 var StartYear = CurrentYear – 10; // Will limit to showing 10 years of history

 }

 else

 {

 var StartYear = CurrentYear - (CurrentYear - BaseYear);

 }

 var x=0;

 for (i=StartYear; i<=StartYear+10; i++)

 {

 myYears[x]=i;

 x++;

 }

 myYears.splice(CurrentYear - BaseYear, 1);

 for (i=0; i<myYears.length - 1; i++)

 {

document.write('<a href="http://www.excelusersgroup.org/pages/page-meetings\_' + myYears[i] + '.html">' + myYears[i] + '</a>' + "&nbsp&nbsp&nbsp&nbsp|&nbsp&nbsp&nbsp&nbsp");

 }

document.write('<a href="http://www.excelusersgroup.org/pages/page-meetings\_' + myYears[myYears.length - 1] + '.html">' + myYears[myYears.length - 1] + '</a>');

 -->

</script>

### [**Meetings Content**](#TOC)

#### [**MeetingDate**](#TOC)

This function is used on annual Meetings pages to calculate dates of meetings each month.



<script type="text/javascript">

 function MeetingDate(CurrentMonth)

 {

 var CurrentYear = 2021;

 var nthWeekday = **jsGetNthWeekday3**();

 var WeekDay = **jsGetWeekDay5**();

return (nthWeekday-1) \* 7 + 1 + (7 + WeekDay - DayOfWeek((nthWeekday-1) \* 7 + 1,CurrentMonth,CurrentYear))%7 + ", " + CurrentYear;

 }

</script>

Please note, **highlighted above** are two functions used from the aforementioned [Weekday.js](#WeekdayJs) file.

Usage of above function:

<script type="text/javascript">

 <!--

 document.write( MeetingDate(2) );

 -->

</script>

#### [**DayOfWeek**](#TOC)

This function calculates the weekday or day of week for each meeting for all months in given year.

<script type="text/javascript">

function DayOfWeek(day,month,year)

 {

var a = Math.floor((14 - month) / 12);

var y = year - a;

var m = month + 12 \* a - 2;

var d = (day + y + Math.floor(y / 4) - Math.floor(y / 100) + Math.floor(y / 400) + Math.floor((31 \* m) / 12)) %7;

return d + 1;

}

</script>