

Click to verify



In this article, you will learn how to select multiple dates in jQuery DatePicker. There are so many libraries are available to implement the datetime" title="DateTime">DatePicker but here we will show you how to implement the date picker" title="Picker">picker for multiple date selection. Checkout more articles on JavaScript /jQuery array-in-half-using-javascript" title="Divide an array in half using JavaScript">Divide an array in half using JavaScript image-from-input-type-file-using-jquery" title="How to preview selected image from input type file using jQuery">How to preview selected image from input type file using jQuery popup-using-jquery" title="Create a Modal Popup using jQuery">Create a Modal Popup using jQuery Select or Deselect all Checkboxes using jQuery bootstrap-colorpicker-example-using-jquery" title="Bootstrap Colorpicker Example using jQuery">Bootstrap Colorpicker Example using jQuery Demo Application Output - Select multiple Dates in jQuery DatePicker - Clue Mediator Steps to implement datepicker for multiple date selection 1. Add jQuery library for datepicker Let's add the following libraries to implement the datepicker. 2. Import datepicker CSS To design the picker, use the following CSS links. body { font-family: Arial, Helvetica, sans-serif; } input { width: 300px; padding: 7px; } .ui-state-highlight { border: 0 !important; } .ui-state-highlight a { background: #363636 !important; color: #fff !important; } Select multiple Dates in jQuery DatePicker - Clue Mediator \$(document).ready(function () { \$('#datepicker').multiDatesPicker(); }); Open the HTML page in browser and check the output. I hope you find this article helpful. Thank you for reading. Happy Coding.!! Share — copy and redistribute the material in any medium or format for any purpose, even commercially. Adapt — remix, transform, and build upon the material for any purpose, even commercially. The licensor cannot revoke these freedoms as long as you follow the license terms. Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use. ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits. You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation. No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material. A separate datepicker instance is created for each targeted control: \$(selector).datepicker(). When applied to a div or span the datepicker is rendered inline. Otherwise it appears as a popup on demand. Each instance may have different settings, e.g. \$(selector).datepicker({minDate: new Date()}); Alternately, you can specify the settings per control by adding a data-datepicker attribute, e.g. The datepicker functionality can only be applied once. Affected controls are flagged with the is-datepicker class and are not re-processed if targeted again. Use the option command if you want to change settings after the initial setup. Below are the settings that may be applied to each datepicker instance. All are optional. A note on Date - the JavaScript Date constructor expects the year, month, and day as parameters. However, the month ranges from 0 to 11. To make explicit what date is intended (does a month of 3 mean March or April?) I specify the month from 1 to 12 and manually subtract the 1. Thus the following denotes 25 December, 2014. \$(selector).datepicker({minDate: new Date(2014, 12-1, 25)}); alignment [altField | altFormat | autoSize | calculateWeek | changeMonth | commands | commandsAsDateFormat | constrainInput | defaultDate | fixedWeeks | maxDate | minDate | monthsOffset | monthsToJump | monthsToShow | multiSelect | multiSeparator | onChangeMonthYear | onClose | onDate | onSelect | onShow | pickerClass | popupContainer | rangeSelect | rangeSeparator | selectDefaultDate | selectOtherMonths | shortYearCutoff | showAnim | showOnFocus | showOptions | showOtherMonths | showSpeed | showTrigger | useMouseWheel | yearRange NameTypeDefaultComments pickerClassstring Any extra CSS class(es) to add to this datepicker instance. By specifying a unique class you can individually target datepicker instances with special styling.\$(selector).datepicker({pickerClass: 'my-picker'}); Since 4.0.0. showOnFocusbooleantrue When true a popup datepicker appears for an input field when it gains focus. When false you should specify a showTrigger instead. Since 4.0.0 - previously incorporated into showOn. showTriggerstring or element or jQuerynull The element(s) that will trigger a popup datepicker to appear when they are clicked. You can provide either the element itself, a jQuery collection containing the element, a string selector for the desired element, or a string version of the element. The trigger is cloned and placed after the input field (taking right-to-left languages into account).\$(selector).datepicker({showTrigger: ''}); \$(selector).datepicker({showTrigger: '#myicon'}); Since 4.0.0 - previously you used showOn, buttonText, buttonImage, and buttonImageOnly. showAnimstring'show' The name of the animation to use when a popup datepicker appears and disappears. The value can be one of the standard animations - 'show', 'fadeIn', or 'slideDown' - or any of the jQuery UI effects (provided you have included the appropriate plugin). Set this to '' for no animation.\$(selector).datepicker({showAnim: 'fadeIn'}); \$(selector).datepicker({showAnim: 'clip'}); \$(selector).datepicker({showAnim: ''}); // Immediate Since 3.7.3 - '' for no animation. showOptionsobject{} For jQuery UI animations you can specify any additional options with this setting. For example, the clip animation can be set to run horizontally instead of vertically.\$(selector).datepicker({showAnim: 'clip', showOptions: {direction: 'horizontal'}}); showSpeedstring or number'normal' The speed at which the animation runs. Use one of the standard speeds - 'slow', 'normal', or 'fast' - or specify the duration in milliseconds.\$(selector).datepicker({showSpeed: 'fast'}); \$(selector).datepicker({showSpeed: 1500}); Since 4.0.0 - previously it was duration, string or element or jQuerynull The container for the popup datepicker, allowing you to control where in the DOM the datepicker is appended. You can provide either the element itself, a Query collection containing the element, or a string selector for the desired element. It defaults to the document body.\$(selector).datepicker({popupContainer: '#mydiv'}); Since 4.0.0. alignmentstring'bottom' Control the alignment of a popup datepicker with respect to its input field. Use one of the following values: 'bottom' for below or 'top' for above, both of which are left- or right- aligned depending on the localisation preference, or 'topLeft', 'topRight', 'bottomLeft', or 'bottomRight'. The first two options will reposition the datepicker if it does not fit in the requested space. Since 3.7.0. fixedWeeksbooleanfalse Set to true to always have six weeks shown, or false to only show as many weeks as are needed. This setting only applies to a datepicker for a single month as all multi-month datepickers are always fixed. Since 4.0.0. calculateWeekfunction\$.datepicker. iso8601Week A function to calculate the week of the year for a given date. The date (Date) is passed as a parameter and the function returns a number indicating the corresponding week of the year. If set to null the built-in ISO 8601 calculation is used. You should use the weekOfYearRenderer to actually display the value.\$(selector).datepicker({calculateWeek: myWeek, renderer: \$.datepicker.weekOfYearRenderer}); function myWeek(date) { return Math.floor((\$.datepicker.dayOfYear(date) - 1) / 7) + 1; } monthsToShownumber or number[2]1 The number of months to show in the datepicker. It may be expressed as a single number of columns, or as an array of rows and columns.\$(selector).datepicker({monthsToShow: 3}); \$(selector).datepicker({monthsToShow: [2, 3]}); Since 4.0.0 - previously it was numberOfMonths. Since 4.0.1 - mark first and last months in each row with classes first and last. monthsOffsetnumber or function0 When showing multiple months, this setting indicates at which position the current month is shown, starting from zero. If specified as a function, it accepts a date as its parameter and returns the corresponding offset. The example below shows three months with the current one in the middle.\$(selector).datepicker({ monthsToShow: 3, monthsOffset: 1}); Since 4.0.0 - previously it was showCurrentAtPos. Since 4.0.6 - can be a function. monthsToJumpnumber12 The number of months to move when the previous or next month commands are invoked. Since 4.0.0 - previously it was stepMonths. useMouseWheelbooleantrue If the MouseWheel plugin is available and this setting is true then you can use the mouse wheel to step through the months or years (with the Ctrl key). If this setting is false then the mouse wheel has no effect within the datepicker, even if the plugin is available. Since 4.0.3. changeMonthbooleantrue Set to true to allow the month and year to be changed via a drop-down selection on the first month shown in the datepicker. Set to false to only allow changes via the various previous and next commands. Since 4.0.0 - previously it only affected the month drop-down with changeYear controlling the year drop-down. yearRangestring'c-10:c+10' Specify the range of years shown in the year drop-down. The setting contains the start and end of the range separated by a colon (:). Each limit may be an absolute year ('1980'), an offset from today ('-10' or '+10'), or an offset from the currently selected date ('c-5' or 'c+5'). Place the maximum value first to have the list appear in descending order. Set to 'any' to allow direct input of the year without selection from a drop-down list. The default is to show 10 years before and after the currently selected date. Note that this setting does not restrict the dates that may be selected. You should use the minDate and maxDate settings to impose limits on the dates that may be selected.\$(selector).datepicker({yearRange: '1980:2000'}); \$(selector).datepicker({yearRange: '1960-18'}); \$(selector).datepicker({yearRange: 'c+10:c-10'}); // Descending \$(selector).datepicker({yearRange: 'any'}); Since 3.7.3 - relative to today's year and combinations. Since 4.0.0 - 'any'. Since 4.0.3 - descending ranges. shortYearCutoffstring or number'+10' When a two-digit year format is used (see dateFormat), this value helps determine the century for a given date. If expressed as a number (0 to 99) it is the year beyond which the century should be the last one instead of the current one. If expressed as a string, it is converted to a number and added to the current year before making the comparison above. If set to -1 the year is always in the 1900s. Since 3.5.2 - disable with -1. showOtherMonthsbooleanfalse Set to true to show the days in other months that appear in the partial weeks before or after the current month. selectOtherMonthsbooleanfalse Set to true to allow the days in other months that appear in the partial weeks before or after the current month to be selected. This setting only applies if showOtherMonths is true.\$(selector).datepicker({ showOtherMonths: true, selectOtherMonths: true}); Since 3.5.0. defaultDate Date or number or stringnull Specify the date to show if no other date has been selected. This may be specified as an actual date (Date), as a date string in the current dateFormat, as a number of days relative to today, or as a string of offsets and periods relative to today. For the last use 'y' for years, 'm' for months, 'w' for weeks, or 'd' for days. Multiple offsets may be combined in the one string. Set to null for a default date of today.\$(selector).datepicker({ defaultDate: new Date(2013, 1-1, 26)}); \$(selector).datepicker({defaultDate: '01/26/2013'}); \$(selector).datepicker({defaultDate: +7}); \$(selector).datepicker({defaultDate: '+1m -1d'}); Since 3.7.0 - added date string. selectDefaultDatebooleanfalse Set to true to automatically select the defaultDate when no other date has been selected.\$(selector).datepicker({ defaultDate: '01/26/2013', selectDefaultDate: true}); Since 4.0.0 - previously it was showDefault. minDate Date or number or stringnull Specify the minimum date allowed to be selected. This may be specified as an actual date (Date), as a date string in the current dateFormat, as a number of days relative to today, or as a string of offsets and periods relative to today. For the last use 'y' for years, 'm' for months, 'w' for weeks, or 'd' for days. Multiple offsets may be combined in the one string. Set to null for no minimum.\$(selector).datepicker({ minDate: new Date(2012, 12-1, 25)}); \$(selector).datepicker({minDate: '12/25/2012'}); \$(selector).datepicker({minDate: -10}); \$(selector).datepicker({minDate: '-1m -15d'}); Since 3.7.0 - added date string. maxDate Date or number or stringnull Specify the maximum date allowed to be selected. This may be specified as an actual date (Date), as a date string in the current dateFormat, as a number of days relative to today, or as a string of offsets and periods relative to today. For the last use 'y' for years, 'm' for months, 'w' for weeks, or 'd' for days. Multiple offsets may be combined in the one string. Set to null for no maximum.\$(selector).datepicker({ maxDate: new Date(2013, 12-1, 25)}); \$(selector).datepicker({maxDate: '12/25/2013'}); \$(selector).datepicker({maxDate: +10}); \$(selector).datepicker({maxDate: '+1m +15d'}); Since 3.7.0 - added date string. autoSizebooleanfalse Set to true to resize the input field based on the maximum length of a date in the current dateFormat. Set to false to not change the field length. Since 3.7.1. rangeSelectbooleanfalse Set to true to allow the selection of a date range in the datepicker. The first selected date is the start of the range and the second selected date is the end of the range. A popup datepicker closes automatically on selection of the range end. Set to false to select a single date. rangeSeparatorstring'-' Specify the separator shown between the two dates in a date range. multiSelectnumber0 Specify the maximum number of individual separate dates that may be selected in the datepicker. Dates may be de-selected by clicked on them a second time. A popup datepicker closes automatically on selection of the maximum number allowed. Set to 0 to select a single date. Since 3.6.0. multiSeparatorstring',' Specify the separator shown between the dates selected in a multiple date datepicker. Since 3.6.0. onDatefunctionnull Specify a callback function to provide additional details about individual dates shown in the datepicker. The function is called for each date displayed and receives the date as a parameter (Date, with time portion set to 12 noon), along with a Boolean indicating whether the date is in the main month being shown (as opposed to being an extra day before or after the main month), while this refers to the associated control. It returns an object with the following attributes (all optional): selectable (boolean) true if the date is selectable, false if not; dateClass (string) any CSS class(es) to apply to this date; title (string) a tooltip for this date; content (string) content for this date to replace the basic day number.\$(selector).datepicker({ onDate: function(date, current) { return 'current ? { }': {content: date.getDate() + ' '+ \$.datepicker.dayOfYear(date) + ', dateClass: 'showDoY'}; } }); Since 4.0.0 - previously beforeShowDay provided similar functionality. Since 4.0.4 - time portion is 12 noon. onShowfunctionnull Specify a callback function to provide additional functionality to a datepicker. The function is called just before the datepicker is shown and receives the completed datepicker division as a jQuery object and the current instance settings as parameters, while this refers to the associated control. It should update the datepicker division as necessary. See the various examples in the datepick.ext module. If you need multiple onShow callbacks, use the multipleEvents function and pass the relevant handlers to it.\$(selector).datepicker({ onShow: function(picker, inst) { picker.find('td:even').addClass('alternate-dates'); } }); Since 4.0.0. onChangeMonthYearfunctionnull Specify a callback function to be notified of changes to the month and year shown in a datepicker. The function is called when the month/year changes and receives the year and month as parameters, while this refers to the associated control. If you need multiple onChangeMonthYear callbacks, use the multipleEvents function and pass the relevant handlers to it.\$(selector).datepicker({ onChangeMonthYear: function(year, month) { \$('#monthYear').text(month + '/' + year); } }); onSelectfunctionnull Specify a callback function to be notified of date selection in a datepicker. The function is called when each date is selected and receives the currently selected dates (Date[]) as the parameter, while this refers to the associated control. The array is empty if no dates have been selected. All dates have their time portion set to 12 noon. Note that when the start of a range is selected the dates array contains two entries, with both being that starting date. If you need multiple onSelect callbacks, use the multipleEvents function and pass the relevant handlers to it.\$(selector).datepicker({ onSelect: function(dates) { \$('#fcount').text('Selected' + dates.length + ' date(s)'; var minDate = dates[0]; for (var i = 1; i < dates.length; i++) { if (dates[i].getTime() < minDate.getTime()) { minDate = dates[i]; } } \$('#minDate').text(\$.datepicker.formatDate(minDate)); } }); Since 3.5.2 - added Date type parameter. Since 4.0.0 - dates parameter only. Since 4.0.4 - time portion is 12 noon. onClosefunctionnull Specify a callback function to be notified of a popup datepicker closing. The function is called when the datepicker is closed (by any means) and receives the currently selected dates (Date[]) as the parameter, while this refers to the associated control. The array is empty if no dates have been selected. All dates have their time portion set to 12 noon. Note that range selections always have two entries, with both being the starting date if no end date has been selected. If you need multiple onClose callbacks, use the multipleEvents function and pass the relevant handlers to it.\$(selector).datepicker({ onClose: function(dates) { var selected = ''; for (var i = 0; i < dates.length; i++) { selected += ' ' + \$.datepicker.formatDate(dates[i]); } alert('Selected dates are: ' + selected.substring(1)); } }); Since 3.5.2 - added Date type parameter. Since 4.0.0 - dates parameter only. Since 4.0.4 - time portion is 12 noon. altFieldstring or element or jQuerynull Specify another field to be updated in sync with the datepicker selections. This and the following setting allow you to automatically show selected dates in one format for the user, while maintaining a second field with a more useful format for further processing. The alternate field may be specified as either the element itself, a jQuery collection containing the element, or a string selector for the desired element.\$(selector).datepicker({dateFormat: 'DD, MM d, yyyy', altField: '#isoDate', altFormat: 'yyyy-mm-dd'}); altFormatstringnull Use in conjunction with the altField setting to automatically maintain two different views of the selected date(s). See the dateFormat setting for the list of possible values. constrainInputbooleantrue Set to true to only allow the entry of characters specified by the dateFormat, or false to allow any characters. commandsAsDateFormatbooleanfalse Set to true to apply the dateFormat function to all command display text. This allows you to label commands with the dates to which they refer, for example the previous and next links could show the month to which they move. Remember to quote (') any command text you do not want substituted.\$(selector).datepicker({commandsAsDateFormat: true, prevText: '< M', todayText: 'M y', nextText: 'M >'}); Since 4.0.0 - previously it was navigationAsDateFormat. commandsobject\$.datepicker. commands See the Commands tab for more details. Since 4.0.0. The datepicker contains a series of commands that operate upon it, allowing these to be customised or new commands added. Each command consists of an object with the attributes shown below, and is added to the commands setting. The standard commands are listed below. Since 4.0.0. NameTypeComments textstring The name of the localisation setting that provides the text value for this command. statusstring The name of the localisation setting that provides the tooltip (status) value for this command. keystrokeobject The keystroke that triggers this command. The object's attributes are: keyCode (number) the numeric code for the keystroke; ctrlKey (boolean, optional) true if the Ctrl key must be used in conjunction with the above keyCode; altKey (boolean, optional) true if the Alt key must be used in conjunction with the above keyCode; shiftKey (boolean, optional) true if the Shift key must be used in conjunction with the above keyCode. enabledfunction The function that indicates whether the command can be invoked at present. It takes the current instance settings as a parameter, while this is the associated target control. It returns true if the command may be executed or false if it may not. dateFormat The function that provides a target date for this command. For example, the target date for the previous command is the first day of the previous month. It takes the current instance settings as a parameter, while this is the associated target control. It returns a Date if an appropriate date can be determined or null if not. actionfunction The function that implements the action for this command. It takes the current instance settings as a parameter, while this is the associated target control. Standard Commands The standard commands defined by the datepicker are shown below and are available as \$.datepicker.commands. These may be positioned in the datepicker by including the {link:name} or {button:name} substitution points in the renderer values, where name is the command name below. Commands can be invoked on a datepicker with the performAction command. These settings comprise the regional settings that may be localised by a language package. They can be overridden for individual instances: \$(selector).datepicker({prevText: