DatePicker

Use Date Picker for date input.

Enter Date

Basic date picker measured by 'day'.

:::demo The measurement is determined by the `type` attribute. You can enable quick options by creating a `picker-options` object with `shortcuts` property. The disabled date is set by `disabledDate` , which is a function.

```
<template>
  <div class="block">
    <span class="demonstration">Default</span>
    <el-date-picker
      v-model="value1"
      type="date"
      placeholder="Pick a day">
    </el-date-picker>
  </div>
  <div class="block">
    <span class="demonstration">Picker with quick options</span>
    <el-date-picker
      v-model="value2"
      type="date"
      placeholder="Pick a day"
      :picker-options="pickerOptions">
    </el-date-picker>
  </div>
</template>

<script>
  export default {
    data() {
      return {
        pickerOptions: {
          disabledDate(time) {
            return time.getTime() > Date.now();
          },
          shortcuts: [{
            text: 'Today',
            onClick(picker) {
              picker.$emit('pick', new Date());
            }
          }, {
            text: 'Yesterday',
            onClick(picker) {
              const date = new Date();
              date.setTime(date.getTime() - 3600 * 1000 * 24);
              picker.$emit('pick', date);
            }
          }, {
            text: 'A week ago',
            onClick(picker) {
              const date = new Date();
              date.setTime(date.getTime() - 3600 * 1000 * 24 * 7);
              picker.$emit('pick', date);
            }
          }]
        },
        value1: '',
        value2: '',
      };
    }
  };
</script>
```

:::

Other measurements

You can choose week, month, year or multiple dates by extending the standard date picker component.

:::demo

```
<div class="container">
  <div class="block">
    <span class="demonstration">Week</span>
    <el-date-picker
      v-model="value1"
      type="week"
      format="Week WW"
      placeholder="Pick a week">
    </el-date-picker>
  </div>
  <div class="block">
    <span class="demonstration">Month</span>
    <el-date-picker
      v-model="value2"
      type="month"
      placeholder="Pick a month">
    </el-date-picker>
  </div>
</div>
<div class="container">
  <div class="block">
    <span class="demonstration">Year</span>
    <el-date-picker
      v-model="value3"
      type="year"
      placeholder="Pick a year">
    </el-date-picker>
  </div>
  <div class="block">
    <span class="demonstration">Dates</span>
    <el-date-picker
      type="dates"
      v-model="value4"
      placeholder="Pick one or more dates">
    </el-date-picker>
  </div>
</div>

<script>
  export default {
    data() {
      return {
        value1: '',
        value2: '',
        value3: '',
        value4: ''
      };
    }
  };
</script>
```

:::

Date Range

Picking a date range is supported.

:::demo When in range mode, the left and right panels are linked by default. If you want the two panels to switch current months independently, you can use the `unlink-panels` attribute.

```
<template>
  <div class="block">
    <span class="demonstration">Default</span>
    <el-date-picker
      v-model="value1"
      type="daterange"
      range-separator="To"
      start-placeholder="Start date"
      end-placeholder="End date">
    </el-date-picker>
  </div>
  <div class="block">
    <span class="demonstration">With quick options</span>
    <el-date-picker
      v-model="value2"
      type="daterange"
      align="right"
      unlink-panels
      range-separator="To"
      start-placeholder="Start date"
      end-placeholder="End date"
      :picker-options="pickerOptions">
    </el-date-picker>
  </div>
</template>

<script>
  export default {
    data() {
      return {
        pickerOptions: {
          shortcuts: [{
            text: 'Last week',
            onClick(picker) {
              const end = new Date();
              const start = new Date();
              start.setTime(start.getTime() - 3600 * 1000 * 24 * 7);
              picker.$emit('pick', [start, end]);
            }
          }, {
            text: 'Last month',
            onClick(picker) {
              const end = new Date();
              const start = new Date();
              start.setTime(start.getTime() - 3600 * 1000 * 24 * 30);
              picker.$emit('pick', [start, end]);
            }
          }, {
            text: 'Last 3 months',
            onClick(picker) {
              const end = new Date();
              const start = new Date();
              start.setTime(start.getTime() - 3600 * 1000 * 24 * 90);
              picker.$emit('pick', [start, end]);
            }
          }]
        },
        value1: '',
        value2: ''
      };
    }
  };
</script>
```

:::

Month Range

Picking a month range is supported.

:::demo When in range mode, the left and right panels are linked by default. If you want the two panels to switch current years independently, you can use the `unlink-panels` attribute.

```
<template>
  <div class="block">
    <span class="demonstration">Default</span>
    <el-date-picker
      v-model="value1"
      type="monthrange"
      range-separator="To"
      start-placeholder="Start month"
      end-placeholder="End month">
    </el-date-picker>
  </div>
  <div class="block">
    <span class="demonstration">With quick options</span>
    <el-date-picker
      v-model="value2"
      type="monthrange"
      align="right"
      unlink-panels
      range-separator="To"
      start-placeholder="Start month"
      end-placeholder="End month"
      :picker-options="pickerOptions">
    </el-date-picker>
  </div>
</template>

<script>
  export default {
    data() {
      return {
        pickerOptions: {
          shortcuts: [{
            text: 'This month',
            onClick(picker) {
              picker.$emit('pick', [new Date(), new Date()]);
            }
          }, {
            text: 'This year',
            onClick(picker) {
              const end = new Date();
              const start = new Date(new Date().getFullYear(), 0);
              picker.$emit('pick', [start, end]);
            }
          }, {
            text: 'Last 6 months',
            onClick(picker) {
              const end = new Date();
              const start = new Date();
              start.setMonth(start.getMonth() - 6);
              picker.$emit('pick', [start, end]);
            }
          }]
        },
        value1: '',
        value2: ''
      };
    }
  };
</script>
```

:::

Default Value

If user hasn't picked a date, shows today's calendar by default. You can use `default-value` to set another date. Its value should be parsable by `new Date()` .

If type is `daterange` , `default-value` sets the left side calendar.

:::demo

```
<template>
  <div class="block">
    <span class="demonstration">date</span>
    <el-date-picker
      v-model="value1"
      type="date"
      placeholder="Pick a date"
      default-value="2010-10-01">
    </el-date-picker>
  </div>
  <div class="block">
    <span class="demonstration">daterange</span>
    <el-date-picker
      v-model="value2"
      type="daterange"
      align="right"
      start-placeholder="Start Date"
      end-placeholder="End Date"
      default-value="2010-10-01">
    </el-date-picker>
  </div>
</template>

<script>
  export default {
    data() {
      return {
        value1: '',
        value2: ''
      };
    }
  };
</script>
```

:::

Date Formats

Use `format` to control displayed text's format in the input box. Use `value-format` to control binding value's format.

By default, the component accepts and emits a `Date` object. Below are supported format strings, using UTC 2017-01-02 03:04:05 as an example:

:::warning Pay attention to capitalization :::

format	meaning	note	example
`yyyy`	year		2017
`M`	month	no leading 0	1
`MM`	month		01
`MMM`	month		Jan
`MMMM`	month		January
`W`	week	only for week picker's `format`; no leading 0	1
`WW`	week	only for week picker's `format`	01
`d`	day	no leading 0	2
`dd`	day		02
`H`	hour	24-hour clock; no leading 0	3
`HH`	hour	24-hour clock	03
`h`	hour	12-hour clock; must be used with `A` or `a`; no leading 0	3
`hh`	hour	12-hour clock; must be used with `A` or `a`	03
`m`	minute	no leading 0	4
`mm`	minute		04
`s`	second	no leading 0	5
`ss`	second		05
`A`	AM/PM	only for `format`, uppercased	AM
`a`	am/pm	only for `format`, lowercased	am
`timestamp`	JS timestamp	only for `value-format`; binding value will be a `number`	1483326245000
`[MM]`	No escape characters	To escape characters, wrap them in square brackets (e.g. [A] [MM])	MM

:::demo

```
<template>
  <div class="block">
    <span class="demonstration">Emits Date object</span>
    <div class="demonstration">Value: {{ value1 }}</div>
    <el-date-picker
      v-model="value1"
      type="date"
      placeholder="Pick a Date"
      format="yyyy/MM/dd">
    </el-date-picker>
  </div>
  <div class="block">
    <span class="demonstration">Use value-format</span>
    <div class="demonstration">Value: {{ value2 }}</div>
    <el-date-picker
      v-model="value2"
      type="date"
      placeholder="Pick a Date"
      format="yyyy/MM/dd"
      value-format="yyyy-MM-dd">
    </el-date-picker>
  </div>
  <div class="block">
    <span class="demonstration">Timestamp</span>
    <div class="demonstration">Value: {{ value3 }}</div>
    <el-date-picker
      v-model="value3"
      type="date"
      placeholder="Pick a Date"
      format="yyyy/MM/dd"
      value-format="timestamp">
    </el-date-picker>
  </div>
</template>

<script>
  export default {
    data() {
      return {
        value1: '',
        value2: '',
        value3: ''
      };
    }
  };
</script>
```

:::

Default time for start date and end date

When picking a date range, you can assign the time part for start date and end date.

:::demo By default, the time part of start date and end date are both `00:00:00`. Setting `default-time` can change their time respectively. It accepts an array of up to two strings with the format of `12:00:00`. The first string sets the time for the start date, and the second for the end date.

```
<template>
  <div class="block">
    <p>Component value: {{ value }}</p>
    <el-date-picker
      v-model="value"
      type="daterange"
      start-placeholder="Start date"
      end-placeholder="End date"
      :default-time="['00:00:00', '23:59:59']">
    </el-date-picker>
  </div>
</template>

<script>
  export default {
    data() {
      return {
        value: ''
      };
    }
  };
</script>
```

:::

Attributes

Attribute	Description	Type	Accepted Values	Default
value / v-model	binding value	date(DatePicker) / array(DateRangePicker)	—	—
readonly	whether DatePicker is read only	boolean	—	false
disabled	whether DatePicker is disabled	boolean	—	false
size	size of Input	string	large/small/mini	—
editable	whether the input is editable	boolean	—	true
clearable	whether to show clear button	boolean	—	true
placeholder	placeholder in non-range mode	string	—	—
start-placeholder	placeholder for the start date in range mode	string	—	—
end-placeholder	placeholder for the end date in range mode	string	—	—
type	type of the picker	string	year/month/date/dates/datetime/week/datetimerange/daterange/monthrange	date
format	format of the displayed value in the input box	string	see date formats	yyyy-MM-dd
align	alignment	left/center/right	left	
popper-class	custom class name for DatePicker's dropdown	string	—	—

picker-options	additional options, check the table below	object	—	{}
range-separator	range separator	string	—	'-'
default-value	optional, default date of the calendar	Date	anything accepted by `new Date()`	—
default-time	optional, the time value to use when selecting date range	string[]	Array with length 2, each item is a string like `12:00:00`. The first item for the start date and then second item for the end date	—
value-format	optional, format of binding value. If not specified, the binding value will be a Date object	string	see date formats	—
name	same as `name` in native input	string	—	—
unlink-panels	unlink two date-panels in range-picker	boolean	—	false
prefix-icon	Custom prefix icon class	string	—	el-icon-date
clear-icon	Custom clear icon class	string	—	el-icon-circle-close
validate-event	whether to trigger form validation	boolean	-	true

Picker Options

Attribute	Description	Type	Accepted Values	Default
shortcuts	a { text, onClick } object array to set shortcut options, check the table below	object[]	—	—
disabledDate	a function determining if a date is disabled	function	—	—

	with that date as its parameter. Should return a Boolean			
cellClassName	set custom className	Function(Date)	—	—
firstDayOfWeek	first day of week	Number	1 to 7	7
onPick	a callback that triggers when the selected date is changed. Only for `daterange` and `datetimerange`.	Function({ maxDate, minDate })	-	-

shortcuts

Attribute	Description	Type	Accepted Values	Default
text	title of the shortcut	string	—	—
onClick	callback function, triggers when the shortcut is clicked, with the `vm` as its parameter. You can change the picker value by emitting the `pick` event. Example: `vm.$emit('pick', new Date())`	function	—	—

Events

Event Name	Description	Parameters
change	triggers when user confirms the value	component's binding value
blur	triggers when Input blurs	component instance
focus	triggers when Input focuses	component instance

Methods

Method	Description	Parameters
focus	focus the Input component	—

Slots

Name	Description
range-separator	custom range separator content