

DatePicker

Utilisez DatePicker pour les champs de dates.

Entrez la date

L'unité de base du DatePicker est le jour.

:::demo L'unité est déterminée par l'attribut `type`. Vous pouvez ajouter des options en créant un objet `picker-options` avec la propriété `shortcuts`. L'état désactivé est contrôlé par `disabledDate` qui est une fonction.

```
<template>
  <div class="block">
    <span class="demonstration">Défaut</span>
    <el-date-picker
      v-model="value1"
      type="date"
      placeholder="Choisissez un jour">
    </el-date-picker>
  </div>
  <div class="block">
    <span class="demonstration">Picker avec raccourcis</span>
    <el-date-picker
      v-model="value2"
      type="date"
      placeholder="Choisissez un jour"
      :picker-options="pickerOptions">
    </el-date-picker>
  </div>
</template>

<script>
export default {
  data() {
    return {
      pickerOptions: {
        disabledDate(time) {
          return time.getTime() > Date.now();
        },
        shortcuts: [{
          text: 'Aujourd\'hui',
          onClick(picker) {
            picker.$emit('pick', new Date());
          }
        }, {
          text: 'Hier',
          onClick(picker) {
            const date = new Date();
            date.setTime(date.getTime() - 3600 * 1000 * 24);
            picker.$emit('pick', date);
          }
        }, {
```

```

        text: 'Il y a une semaine',
        onClick(picker) {
            const date = new Date();
            date.setTime(date.getTime() - 3600 * 1000 * 24 * 7);
            picker.$emit('pick', date);
        }
    ]],
    },
    value1: '',
    value2: '',
  };
}
};
</script>

```

...

Autres unités

Vous pouvez sélectionner une semaine, un mois, une année ou plusieurs dates en ajoutant des options grâce aux attributs.

:::demo

```

<div class="container">
  <div class="block">
    <span class="demonstration">Semaine</span>
    <el-date-picker
      v-model="value1"
      type="week"
      format="Week WW"
      placeholder="Sélectionnez une semaine">
    </el-date-picker>
  </div>
  <div class="block">
    <span class="demonstration">Mois</span>
    <el-date-picker
      v-model="value2"
      type="month"
      placeholder="Sélectionnez un mois">
    </el-date-picker>
  </div>
</div>
<div class="container">
  <div class="block">
    <span class="demonstration">Année</span>
    <el-date-picker
      v-model="value3"
      type="year"
      placeholder="Sélectionnez une année">
    </el-date-picker>
  </div>
</div>

```

```

<div class="block">
  <span class="demonstration">Dates</span>
  <el-date-picker
    type="dates"
    v-model="value4"
    placeholder="Sélectionnez une ou plusieurs dates">
  </el-date-picker>
</div>
</div>

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

```

...

Plage de dates

Vous pouvez sélectionner une plage de dates.

:::demo Lorsque ce mode est activé, les deux panneaux sont liés par défaut. Si vous souhaitez qu'ils changent de mois de manière indépendante, ajoutez l'attribut `unlink-panels`.

```

<template>
  <div class="block">
    <span class="demonstration">Défaut</span>
    <el-date-picker
      v-model="value1"
      type="daterange"
      range-separator="à"
      start-placeholder="Date de début"
      end-placeholder="Date de fin">
    </el-date-picker>
  </div>
  <div class="block">
    <span class="demonstration">Avec des options</span>
    <el-date-picker
      v-model="value2"
      type="daterange"
      align="right"
      unlink-panels
      range-separator="à"
      start-placeholder="Date de début"

```

```

        end-placeholder="Date de fin"
        :picker-options="pickerOptions">
    </el-date-picker>
</div>
</template>

<script>
export default {
  data() {
    return {
      pickerOptions: {
        shortcuts: [{
          text: 'Semaine dernière',
          onClick(picker) {
            const end = new Date();
            const start = new Date();
            start.setTime(start.getTime() - 3600 * 1000 * 24 * 7);
            picker.$emit('pick', [start, end]);
          }
        }, {
          text: 'Mois dernier',
          onClick(picker) {
            const end = new Date();
            const start = new Date();
            start.setTime(start.getTime() - 3600 * 1000 * 24 * 30);
            picker.$emit('pick', [start, end]);
          }
        }, {
          text: 'Trois derniers mois',
          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>

```

⋮

Plage de mois

Vous pouvez sélectionner une plage de mois.

⋮:demo Lorsque ce mode est activé, les panneaux gauche et droit sont liés par défaut. Si vous voulez que les deux panneaux switch indépendamment les années en cours, vous pouvez utiliser l'attribut `unlink-panels`.

```

<template>
  <div class="block">
    <span class="demonstration">Défaut</span>
    <el-date-picker
      v-model="value1"
      type="monthrangle"
      range-separator="à"
      start-placeholder="Mois de début"
      end-placeholder="Mois de fin">
    </el-date-picker>
  </div>
  <div class="block">
    <span class="demonstration">Avec options</span>
    <el-date-picker
      v-model="value2"
      type="monthrangle"
      align="right"
      unlink-panels
      range-separator="à"
      start-placeholder="Mois de début"
      end-placeholder="Mois de fin"
      :picker-options="pickerOptions">
    </el-date-picker>
  </div>
</template>

<script>
export default {
  data() {
    return {
      pickerOptions: {
        shortcuts: [{
          text: 'Ce mois',
          onClick(picker) {
            picker.$emit('pick', [new Date(), new Date()]);
          }
        }, {
          text: 'Cette année',
          onClick(picker) {
            const end = new Date();
            const start = new Date(new Date().getFullYear(), 0);
            picker.$emit('pick', [start, end]);
          }
        }, {
          text: 'Les derniers 6 mois',
          onClick(picker) {
            const end = new Date();
            const start = new Date();
            start.setMonth(start.getMonth() - 6);
            picker.$emit('pick', [start, end]);
          }
        }
      ]
    }
  }
}

```

```

        ]]
      },
      value1: '',
      value2: ''
    };
  }
};
</script>

```

...

Valeur par défaut

Si l'utilisateur n'a pas sélectionné de date, vous pouvez montrer la date d'aujourd'hui par défaut. Utilisez `default-value` pour montrer une autre date. Sa valeur doit être parsable par `new Date()`.

Si le type est `daterange`, `default-value` configure la panneau de gauche.

...demo

```

<template>
  <div class="block">
    <span class="demonstration">Date</span>
    <el-date-picker
      v-model="value1"
      type="date"
      placeholder="Sélectionnez une date"
      default-value="2010-10-01">
    </el-date-picker>
  </div>
  <div class="block">
    <span class="demonstration">Plage de dates</span>
    <el-date-picker
      v-model="value2"
      type="daterange"
      align="right"
      start-placeholder="Date de début"
      end-placeholder="Date de fin"
      default-value="2010-10-01">
    </el-date-picker>
  </div>
</template>

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

```

```
};  
</script>
```

...

Formats de date

Utilisez `format` pour contrôler l'affichage de la date dans le champ. Utilisez `value-format` pour contrôler le format de la variable stockée.

Par défaut, le composant accepte et émet un objet `Date`. Les formats supportés se trouvent ci-dessous, avec la date UTC 2017-01-02 03:04:05 comme exemple:

...warning Attention à la capitalisation ! ...

Format	Signification	Note	Exemple
----- ----- ----- ----- -----	yyyy Année	2017	M Mois pas de 0 au début 1
MM Mois	01	MMM Mois	Jan
MMMM Mois	Janvier	W Semaine uniquement pour les format des types 'semaine' ; pas de 0 au début 1	WW Semaine uniquement pour les format des types 'semaine' 01
d Jour pas de 0 au début 2	dd Jour	02	H Heure Horloge 24 heures; pas de 0 au début 3
HH Heure Horloge 24 heures	03	h Heure Horloge 12 heures ; doit être utilisé avec A ou a ; pas de 0 au début 3	hh Heure Horloge 12 heures ; doit être utilisé avec A ou a 03
m Minute pas de 0 au début 4	mm Minute	04	s Seconde pas de 0 au début 5
ss Seconde	05	A AM/PM uniquement pour format , majuscules AM	a am/pm uniquement pour format , minuscules am
timestamp timestamp JS uniquement pour value-format ; la variable stockée sera un number 1483326245000	[MM] Pas de caractère d'échappement Pour échapper des caractères, placez-les entre crochets (ex: [A] [MM]) MM		

...demo

```
<template>  
  <div class="block">  
    <span class="demonstration">Émet un objet Date</span>  
    <div class="demonstration">Value: {{ value1 }}</div>  
    <el-date-picker  
      v-model="value1"  
      type="date"  
      placeholder="Sélectionnez une date"  
      format="yyyy/MM/dd">  
    </el-date-picker>  
  </div>  
  <div class="block">  
    <span class="demonstration">Utilise value-format</span>  
    <div class="demonstration">Value: {{ value2 }}</div>  
    <el-date-picker  
      v-model="value2"  
      type="date"  
      placeholder="Sélectionnez une date"  
      format="yyyy/MM/dd"  
      value-format="yyyy-MM-dd">  
    </el-date-picker>  
  </div>  
</div>
```

```

<span class="demonstration">Timestamp</span>
<div class="demonstration">Value: {{ value3 }}</div>
<el-date-picker
  v-model="value3"
  type="date"
  placeholder="Sélectionnez une date"
  format="yyyy/MM/dd"
  value-format="timestamp">
</el-date-picker>
</div>
</template>

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

```

...

Horaire par défaut pour les dates de fin et de début

Lorsque vous choisissez une plage de dates, vous pouvez assigner l'horaire de départ et de fin pour chacune des dates.

:::demo Par défaut, l'horaire pour les dates de début et de fin est 00:00:00 . Configurez default-time pour changer l'horaire. Cet attribut prends un tableau de un ou deux horaires au format 12:00:00 . Le premier est pour la date de début, le deuxième pour la date de fin.

```

<template>
  <div class="block">
    <p>Valeur: {{ value }}</p>
    <el-date-picker
      v-model="value"
      type="daterange"
      start-placeholder="Date de début"
      end-placeholder="Date de fin"
      :default-time="['00:00:00', '23:59:59']">
    </el-date-picker>
  </div>
</template>

<script>
  export default {
    data() {
      return {

```



```

        value: ''
    };
}
};
</script>

```

⋮

Attributs

Attribut	Description	Type	Valeurs acceptées	Défaut
value / v-model	La valeur liée.	date(DatePicker) / array(DateRangePicker)	—	—
readonly	Si le DatePicker est en lecture seule.	boolean	—	false
disabled	Si le DatePicker est désactivé.	boolean	—	false
size	Taille du champs.	string	large/small/mini	—
editable	Si le champ est éditable.	boolean	—	true
clearable	Si la croix de reset doit apparaître.	boolean	—	true
placeholder	Le placeholder en mode normal.	string	—	—
start-placeholder	Le placeholder pour la date de début en mode plage de dates.	string	—	—
end-placeholder	Le placeholder pour la date de fin en mode plage de dates.	string	—	—
type	Type du picker.	string	year/month/date/dates/datetime/week/datetimerange/daterange/monthrange	date
format	Format d'affichage dans le champ.	string	Voir formats de date .	YYYY-MM-dd
align	Alignement.	left/center/right	left	

popper-class	Nom de classe pour le menu déroulant du DatePicker.	string	—	—
picker-options	Options additionnelles, voir la table ci-dessous.	object	—	{}
range-separator	Séparateur de plage de dates.	string	—	'_'
default-value	Date par défaut du calendrier, optionnelle.	Date	Tout ce qui est accepté par <code>new Date()</code>	—
default-time	Horaire par défaut quand on sélectionne une plage dates, optionnel.	string[]	Tableau de taille 2, chaque valeur est au format <code>12:00:00</code> . La première pour la date de début, la deuxième pour la date de fin.	—
value-format	Format de la variable stockée, optionnel. Si non spécifié, la valeur sera un objet Date.	string	Voir formats de date	—
name	Identique au <code>name</code> de l'input natif	string	—	—
unlink-panels	Rend indépendants les deux panneaux de plage de dates.	boolean	—	false
prefix-icon	Icône de préfixe.	string	—	el-icon-date
clear-icon	Icône de reset.	string	—	el-icon-circle-close
validate-event	Si la validation doit être déclenchée.	boolean	-	true

Options du Picker

Attribut	Description	Type	Valeurs acceptées	Défaut
shortcuts	Un tableau d'objets { text, onClick } pour configurer les raccourcis, voir table ci-dessous.	object[]	—	—
disabledDate	Une fonction qui détermine si une date est désactivée ou pas, avec cette date en paramètre. Doit retourner un booléen.	function	—	—
cellClassName	set custom className	Function(Date)	—	—
firstDayOfWeek	Premier jour du mois.	Number	1 to 7	7
onPick	Callback se déclenchant quand la date sélectionnée change. Uniquement pour <code>daterange</code> et <code>datetimerange</code> .	Function({ maxDate, minDate })	-	-

Raccourcis

Attribut	Description	Type	Valeurs acceptées	Défaut
text	Titre du raccourci.	string	—	—
onClick	Callback se déclenchant quand le raccourci est cliqué, avec <code>vm</code> en paramètre. Vous pouvez changer la valeur du picker en émettant l'évènement <code>pick</code> . Exemple: <code>vm.\$emit('pick', new Date())</code>	function	—	—

Évènements

Nom	Description	Paramètres
change	Se déclenche quand l'utilisateur confirme la valeur	component's binding value
blur	Se déclenche quand le champ perd le focus.	component instance
focus	Se déclenche quand le champ a le focus.	component instance

Méthodes

Méthode	Description	Paramètres
focus	Focus sur le champ.	—

Slots

Nom	Description
range-separator	Séparateur d'intervalle personnalisé.