

## Select

Utile lorsqu'il faut sélectionner des options parmi un large choix, affiché grâce à un menu déroulant.

### Usage

:::demo v-model est la valeur du el-option sélectionné.

```
<template>
  <el-select v-model="value" placeholder="Select">
    <el-option
      v-for="item in options"
      :key="item.value"
      :label="item.label"
      :value="item.value">
    </el-option>
  </el-select>
</template>
```

```
<script>
  export default {
    data() {
      return {
        options: [{
          value: 'Option1',
          label: 'Option1'
        }, {
          value: 'Option2',
          label: 'Option2'
        }, {
          value: 'Option3',
          label: 'Option3'
        }, {
          value: 'Option4',
          label: 'Option4'
        }, {
          value: 'Option5',
          label: 'Option5'
        }
      ],
      value: ''
    }
  }
</script>
```

...

## Option désactivée

:::demo Mettez l'attribut `disabled` dans `el-option` à `true` pour désactiver cette option.

```
<template>
  <el-select v-model="value" placeholder="Select">
    <el-option
      v-for="item in options"
      :key="item.value"
      :label="item.label"
      :value="item.value"
      :disabled="item.disabled">
    </el-option>
  </el-select>
</template>
```

```
<script>
export default {
  data() {
    return {
      options: [{
        value: 'Option1',
        label: 'Option1'
      }, {
        value: 'Option2',
        label: 'Option2',
        disabled: true
      }, {
        value: 'Option3',
        label: 'Option3'
      }, {
        value: 'Option4',
        label: 'Option4'
      }, {
        value: 'Option5',
        label: 'Option5'
      }
    ],
    value: ''
  }
}
</script>
```

:::

## Sélecteur désactivé

Vous pouvez désactiver le composant lui-même.

:::demo Ajoutez `disabled` à `el-select` pour le désactiver.

```
<template>
  <el-select v-model="value" disabled placeholder="Select">
    <el-option
      v-for="item in options"
      :key="item.value"
      :label="item.label"
      :value="item.value">
    </el-option>
  </el-select>
</template>

<script>
export default {
  data() {
    return {
      options: [{
        value: 'Option1',
        label: 'Option1'
      }, {
        value: 'Option2',
        label: 'Option2'
      }, {
        value: 'Option3',
        label: 'Option3'
      }, {
        value: 'Option4',
        label: 'Option4'
      }, {
        value: 'Option5',
        label: 'Option5'
      }],
      value: ''
    }
  }
}
</script>

:::
```

## Effacer la sélection

Vous pouvez ajouter un bouton pour effacer la sélection.

:::demo Ajoutez l'attribut `clearable` à `el-select` et l'icône de fermeture s'affichera après une sélection. Notez que `clearable` ne marche qu'avec les sélecteurs à choix unique.

```
<template>
  <el-select v-model="value" clearable placeholder="Select">
    <el-option
      v-for="item in options"
      :key="item.value"
      :label="item.label"
      :value="item.value">
    </el-option>
  </el-select>
</template>

<script>
export default {
  data() {
    return {
      options: [{
        value: 'Option1',
        label: 'Option1'
      }, {
        value: 'Option2',
        label: 'Option2'
      }, {
        value: 'Option3',
        label: 'Option3'
      }, {
        value: 'Option4',
        label: 'Option4'
      }, {
        value: 'Option5',
        label: 'Option5'
      }],
      value: ''
    }
  }
}
</script>

:::
```

### Sélecteur multiple

Les sélecteurs multiples utilisent des tags pour afficher les différentes options choisies.

:::demo Ajoutez `multiple` à `el-select` pour le changer en sélecteur multiple. La valeur de `v-model` devient un tableau contenant toutes les options. Par défaut les différents choix sont affichés sous forme de tags. Vous pouvez réduire leur nombre en utilisant l'attribut `collapse-tags`.

```
<template>
  <el-select v-model="value1" multiple placeholder="Select">
    <el-option
      v-for="item in options"
      :key="item.value"
      :label="item.label"
      :value="item.value">
    </el-option>
  </el-select>

  <el-select
    v-model="value2"
    multiple
    collapse-tags
    style="margin-left: 20px;"
    placeholder="Select">
    <el-option
      v-for="item in options"
      :key="item.value"
      :label="item.label"
      :value="item.value">
    </el-option>
  </el-select>
</template>

<script>
export default {
  data() {
    return {
      options: [{
        value: 'Option1',
        label: 'Option1'
      }, {
        value: 'Option2',
        label: 'Option2'
      }, {
        value: 'Option3',
        label: 'Option3'
      }, {
        value: 'Option4',
        label: 'Option4'
      }
    ]
  }
}
```

```

    }, {
      value: 'Option5',
      label: 'Option5'
    }],
    value1: [],
    value2: []
  }
}
</script>
:::

```

### Template personnalisé

Vous pouvez définir un template HTML pour l'affichage des options.

:::demo Insérez votre template dans le slot de `el-option`.

```

<template>
  <el-select v-model="value" placeholder="Select">
    <el-option
      v-for="item in cities"
      :key="item.value"
      :label="item.label"
      :value="item.value">
      <span style="float: left">{{ item.label }}</span>
      <span style="float: right; color: #8492a6; font-size: 13px">{{ item.value }}</span>
    </el-option>
  </el-select>
</template>

<script>
export default {
  data() {
    return {
      cities: [{
        value: 'Beijing',
        label: 'Beijing'
      }, {
        value: 'Shanghai',
        label: 'Shanghai'
      }, {
        value: 'Nanjing',
        label: 'Nanjing'
      }, {
        value: 'Chengdu',

```

```

        label: 'Chengdu'
      }, {
        value: 'Shenzhen',
        label: 'Shenzhen'
      }, {
        value: 'Guangzhou',
        label: 'Guangzhou'
      }
    ],
    value: ''
  }
}
</script>
:::

```

### Grouper les options

Vous pouvez définir des groupes pour les options du menu.

:::demo Utilisez `el-option-group` pour grouper les options. L'attribut `label` définit le nom du groupe.

```

<template>
  <el-select v-model="value" placeholder="Select">
    <el-option-group
      v-for="group in options"
      :key="group.label"
      :label="group.label">
      <el-option
        v-for="item in group.options"
        :key="item.value"
        :label="item.label"
        :value="item.value">
      </el-option>
    </el-option-group>
  </el-select>
</template>

<script>
export default {
  data() {
    return {
      options: [{
        label: 'Villes célèbres',
        options: [{
          value: 'Shanghai',

```

```

        label: 'Shanghai'
      }, {
        value: 'Beijing',
        label: 'Beijing'
      }]
    }, {
      label: 'Nom de ville',
      options: [{
        value: 'Chengdu',
        label: 'Chengdu'
      }, {
        value: 'Shenzhen',
        label: 'Shenzhen'
      }, {
        value: 'Guangzhou',
        label: 'Guangzhou'
      }, {
        value: 'Dalian',
        label: 'Dalian'
      }]
    }],
    value: ''
  }
}
</script>
:::

```

### Filtrage des options

Vous pouvez ajouter un mode de filtrage pour trouver les options désirées plus rapidement.

:::demo Ajoutez `filterable` à `el-select` pour activer le filtrage. Par défaut, Select cherchera les options dont le `label` contient la valeur du filtre. Si vous préférez une autre stratégie de filtrage, utilisez `filter-method`. C'est une `Function` qui est appelée quand la valeur change, avec pour paramètre la valeur courante.

```

<template>
  <el-select v-model="value" filterable placeholder="Select">
    <el-option
      v-for="item in options"
      :key="item.value"
      :label="item.label"
      :value="item.value">

```



```

    </el-option>
  </el-select>
</template>

<script>
export default {
  data() {
    return {
      options: [{
        value: 'Option1',
        label: 'Option1'
      }, {
        value: 'Option2',
        label: 'Option2'
      }, {
        value: 'Option3',
        label: 'Option3'
      }, {
        value: 'Option4',
        label: 'Option4'
      }, {
        value: 'Option5',
        label: 'Option5'
      }],
      value: ''
    }
  }
}
</script>
:::

```

## Recherche à distance

Vous pouvez aller chercher les options sur le serveur de manière dynamique.

:::demo Ajoutez **filterable** et **remote** pour activer la recherche distante, ainsi que **remote-method**. Cette dernière est une **Function** qui est appelée lorsque la valeur change, avec pour paramètre la valeur courante. Notes que si **el-option** est rendu dans une directive **v-for**, vous devriez ajouter l'attribut **key** aux **el-option**. Cette valeur doit être unique, comme **item.value** dans l'exemple suivant.

```

<template>
  <el-select
    v-model="value"
    multiple
    filterable

```

```

remote
reserve-keyword
placeholder="Entrez un mot-clé"
:remote-method="remoteMethod"
:loading="loading">
<el-option
  v-for="item in options"
  :key="item.value"
  :label="item.label"
  :value="item.value">
</el-option>
</el-select>
</template>

<script>
export default {
  data() {
    return {
      options: [],
      value: [],
      list: [],
      loading: false,
      states: ["Alabama", "Alaska", "Arizona",
        "Arkansas", "California", "Colorado",
        "Connecticut", "Delaware", "Florida",
        "Georgia", "Hawaii", "Idaho", "Illinois",
        "Indiana", "Iowa", "Kansas", "Kentucky",
        "Louisiana", "Maine", "Maryland",
        "Massachusetts", "Michigan", "Minnesota",
        "Mississippi", "Missouri", "Montana",
        "Nebraska", "Nevada", "New Hampshire",
        "New Jersey", "New Mexico", "New York",
        "North Carolina", "North Dakota", "Ohio",
        "Oklahoma", "Oregon", "Pennsylvania",
        "Rhode Island", "South Carolina",
        "South Dakota", "Tennessee", "Texas",
        "Utah", "Vermont", "Virginia",
        "Washington", "West Virginia", "Wisconsin",
        "Wyoming"]
    }
  },
  mounted() {
    this.list = this.states.map(item => {
      return { value: `value:${item}`, label: `label:${item}` };
    });
  },

```

```

    methods: {
      remoteMethod(query) {
        if (query !== '') {
          this.loading = true;
          setTimeout(() => {
            this.loading = false;
            this.options = this.list.filter(item => {
              return item.label.toLowerCase()
                .indexOf(query.toLowerCase()) > -1;
            });
          }, 200);
        } else {
          this.options = [];
        }
      }
    }
  }
</script>
:::

```

### Créer des options

Vous pouvez entrer des choix dans le champ de sélection qui ne sont pas incluses dans le menu.

:::demo En utilisant `allow-create`, peuvent créer de nouveaux choix en les entrant dans le champ d'input. Cette option ne marche que si `filterable` est activé. Cette exemple montre aussi `default-first-option`, qui permet de sélectionner la première option en pressant Entrée sans avoir à utiliser la souris ou le clavier.

```

<template>
  <el-select
    v-model="value"
    multiple
    filterable
    allow-create
    default-first-option
    placeholder="Choisissez les tags de vos articles">
    <el-option
      v-for="item in options"
      :key="item.value"
      :label="item.label"
      :value="item.value">
    </el-option>
  </el-select>

```

```

</template>

<script>
  export default {
    data() {
      return {
        options: [{
          value: 'HTML',
          label: 'HTML'
        }, {
          value: 'CSS',
          label: 'CSS'
        }, {
          value: 'JavaScript',
          label: 'JavaScript'
        }],
        value: []
      }
    }
  }
</script>

...

```

Si la valeur de Select est un objet, assurez-vous d'utiliser **value-key** comme identifiant unique.

### Attributs de Select

Attribut	Description	Type	Valeurs acceptées	Défaut
value / v-model	La valeur liée.	boolean / string / number	—	—
multiple	Si la sélection multiple est activée.	boolean	—	false
disabled	Si le sélecteur est désactivé.	boolean	—	false
value-key	Identifiant unique pour la valeur, requis quand la valeur est un objet.	string	—	value

Attribut	Description	Type	Valeurs acceptées	Défaut
size	Taille de l'Input.	string	large/small/mini	—
clearable	Si la sélection est effaçable.	boolean	—	false
collapse-tags	Si les tags peuvent être réduits, dans le cas d'une sélection multiple.	boolean	—	false
multiple-limit	Nombre maximum d'options multiples sélectionnable. Pas de limites quand est à 0.	number	—	0
name	L'attribut name natif du sélecteur.	string	—	—
autocomplete	L'attribut autocomplete natif du sélecteur.	string	—	off
auto-complete	@DEPRECATED dans la prochaine version majeure.	string	—	off
placeholder	Le placeholder du champ.	string	—	Select
filterable	Si les options sont filtrables.	boolean	—	false

Attribut	Description	Type	Valeurs acceptées	Défaut
allow-create	Si l'utilisateur peut créer des options. Dans ce cas <b>filterable</b> doit être activé.	boolean	—	false
filter-method	Méthode de filtrage personnalisée.	function	—	—
remote	Si les options sont chargées dynamiquement depuis le serveur.	boolean	—	false
remote-method	Méthode pour la recherche distante.	function	—	—
loading	Si le sélecteur est en train de charger des options.	boolean	—	false
loading-text	Texte à afficher pendant le chargement.	string	—	Loading
no-match-text	Texte à afficher quand le filtrage ne retourne aucune option. Vous pouvez aussi utiliser le slot <b>empty</b> .	string	—	No matching data

Attribut	Description	Type	Valeurs acceptées	Défaut
no-data-text	Texte à afficher quand il n'y a aucune option. Vous pouvez aussi utiliser le slot <b>empty</b> .	string	—	No data
popper-class	Classe du menu déroulant.	string	—	—
reserve-keyword	Quand <b>multiple</b> et <b>filter</b> sont activés, s'il faut réserver le mot-clé courant après la sélection d'une option.	boolean	—	false
default-first-option	Sélectionne la première option avec Entrée. Utilisable avec <b>filterable</b> ou <b>remote</b>	boolean	-	false
popper-append-to-body	Si le menu déroulant doit être ajouté au body. Si le positionnement du menu est incorrect, essayez de mettre cette option à <b>false</b> .	boolean	-	true

Attribut	Description	Type	Valeurs acceptées	Défaut
automatic-dropdown	Pour les sélecteurs non filtrables, détermine si le menu apparaît au focus du champ.	boolean	-	false

### Évènements de Select

Nom	Description	Paramètre
change	Se déclenche quand la valeur change.	current selected value
visible-change	Se déclenche quand le menu apparaît ou disparaît.	true quand il apparaît, sinon false .
remove-tag	Se déclenche quand un tag est retiré.	La valeur du tag retiré.
clear	Se déclenche quand le champ est effacé grâce au bouton.	—
blur	Se déclenche quand le champ perd le focus.	(event: Event)
focus	Se déclenche gagne le focus.	(event: Event)

### Slots de Select

Nom	Description
—	Liste de options.
prefix	Contenu du préfixe du sélecteur.
empty	Contenu lorsqu'il n'y a aucune option.

### Attributs du groupe d'options



Attribut	Description	Type	Valeurs acceptées	Défaut
label	Nom du group.	string	—	—
disabled	Si les options du groupe sont désactivées.	boolean	—	false

### Attributs des options

Attribut	Description	Type	Valeurs acceptées	Défaut
value	Valeur de l'option.	string/number/object		—
label	Label de l'option, identique à <b>value</b> si omis.	string/number		—
disabled	Si l'option est désactivée.	boolean	—	false

### Méthodes

Méthode	Description	Paramètres
focus	Focus sur l'input.	-
blur	Enlève le focus de l'input et cache le menu.	-