

Select 选择器

当选项过多时，使用下拉菜单展示并选择内容。

基础用法

适用广泛的基础单选 `::demo` `v-model` 的值为当前被选中的 `el-option` 的 `value` 属性值

```
<template>
  <el-select v-model="value" placeholder="请选择">
    <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: '选项1',
          label: '黄金糕'
        }, {
          value: '选项2',
          label: '双皮奶'
        }, {
          value: '选项3',
          label: '蚵仔煎'
        }, {
          value: '选项4',
          label: '龙须面'
        }, {
          value: '选项5',
          label: '北京烤鸭'
        }],
        value: ''
      }
    }
  }
</script>
```

...

有禁用选项

`::demo` 在 `el-option` 中，设定 `disabled` 值为 `true`，即可禁用该选项

```

<template>
  <el-select v-model="value" placeholder="请选择">
    <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: '选项1',
          label: '黄金糕'
        }, {
          value: '选项2',
          label: '双皮奶',
          disabled: true
        }, {
          value: '选项3',
          label: '蚵仔煎'
        }, {
          value: '选项4',
          label: '龙须面'
        }, {
          value: '选项5',
          label: '北京烤鸭'
        }
      ],
      value: ''
    }
  }
}
</script>

```

...

禁用状态

选择器不可用状态

::demo 为 el-select 设置 disabled 属性，则整个选择器不可用

```

<template>
  <el-select v-model="value" disabled placeholder="请选择">
    <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: '选项1',
        label: '黄金糕'
      }, {
        value: '选项2',
        label: '双皮奶'
      }, {
        value: '选项3',
        label: '蚵仔煎'
      }, {
        value: '选项4',
        label: '龙须面'
      }, {
        value: '选项5',
        label: '北京烤鸭'
      }],
      value: ''
    }
  }
}
</script>

```

...

可清空单选

包含清空按钮，可将选择器清空为初始状态

::demo 为 `el-select` 设置 `clearable` 属性，则可将选择器清空。需要注意的是，`clearable` 属性仅适用于单选。

```

<template>
  <el-select v-model="value" clearable placeholder="请选择">
    <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: '选项1',
          label: '黄金糕'
        }, {
          value: '选项2',
          label: '双皮奶'
        }, {
          value: '选项3',
          label: '蚵仔煎'
        }, {
          value: '选项4',
          label: '龙须面'
        }, {
          value: '选项5',
          label: '北京烤鸭'
        }],
        value: ''
      }
    }
  }
</script>

```

:::

基础多选

适用性较广的基础多选，用 Tag 展示已选项

:::demo 为 `el-select` 设置 `multiple` 属性即可启用多选，此时 `v-model` 的值为当前选中值所组成的数组。默认情况下选中值会以 Tag 的形式展现，你也可以设置 `collapse-tags` 属性将它们合并为一段文字。

```

<template>
  <el-select v-model="value1" multiple placeholder="请选择">
    <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="请选择">

```

```

    <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: '选项1',
          label: '黄金糕'
        }, {
          value: '选项2',
          label: '双皮奶'
        }, {
          value: '选项3',
          label: '蚵仔煎'
        }, {
          value: '选项4',
          label: '龙须面'
        }, {
          value: '选项5',
          label: '北京烤鸭'
        }],
        value1: [],
        value2: []
      }
    }
  }
</script>

```

...

自定义模板

可以自定义备选项

::demo 将自定义的 HTML 模板插入 `el-option` 的 slot 中即可。

```

<template>
  <el-select v-model="value" placeholder="请选择">
    <el-option
      v-for="item in cities"
      :key="item.value"
      :label="item.label"
      :value="item.value">
      <span style="float: left">{{ item.label }}</span>
    </el-option>
  </el-select>
</template>

```

```

        <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: '北京'
                }, {
                    value: 'Shanghai',
                    label: '上海'
                }, {
                    value: 'Nanjing',
                    label: '南京'
                }, {
                    value: 'Chengdu',
                    label: '成都'
                }, {
                    value: 'Shenzhen',
                    label: '深圳'
                }, {
                    value: 'Guangzhou',
                    label: '广州'
                }
            ],
            value: ''
        }
    }
</script>

```

⋮

分组

备选项进行分组展示

⋮demo 使用 `el-option-group` 对备选项进行分组，它的 `label` 属性为分组名

```

<template>
    <el-select v-model="value" placeholder="请选择">
        <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: '热门城市',
        options: [{
          value: 'Shanghai',
          label: '上海'
        }, {
          value: 'Beijing',
          label: '北京'
        }]
      }, {
        label: '城市名',
        options: [{
          value: 'Chengdu',
          label: '成都'
        }, {
          value: 'Shenzhen',
          label: '深圳'
        }, {
          value: 'Guangzhou',
          label: '广州'
        }, {
          value: 'Dalian',
          label: '大连'
        }]
      }],
      value: ''
    }
  }
}
</script>

```

...

可搜索

可以利用搜索功能快速查找选项

...demo 为 `el-select` 添加 `filterable` 属性即可启用搜索功能。默认情况下，Select 会找出所有 `label` 属性包含输入值的选项。如果希望使用其他的搜索逻辑，可以通过传入一个 `filter-method` 来实现。 `filter-method` 为一个 `Function`，它会在输入值发生变化时调用，参数为当前输入值。

```

<template>
  <el-select v-model="value" filterable placeholder="请选择">
    <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: '选项1',
          label: '黄金糕'
        }, {
          value: '选项2',
          label: '双皮奶'
        }, {
          value: '选项3',
          label: '蚵仔煎'
        }, {
          value: '选项4',
          label: '龙须面'
        }, {
          value: '选项5',
          label: '北京烤鸭'
        }],
        value: ''
      }
    }
  }
</script>

```

...

远程搜索

从服务器搜索数据，输入关键字进行查找。demo 为了启用远程搜索，需要将 `filterable` 和 `remote` 设置为 `true`，同时传入一个 `remote-method`。`remote-method` 为一个 `Function`，它会在输入值发生变化时调用，参数为当前输入值。需要注意的是，如果 `el-option` 是通过 `v-for` 指令渲染出来的，此时需要为 `el-option` 添加 `key` 属性，且其值需具有唯一性，比如此例中的 `item.value`。

```

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

```



```

remote
reserve-keyword
placeholder="请输入关键词"
: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>

```

⋮

创建条目

可以创建并选中选项中不存在的条目 ⋮demo 使用 `allow-create` 属性即可通过在输入框中输入文字来创建新的条目。注意此时 `filterable` 必须为真。本例还使用了 `default-first-option` 属性，在该属性打开的情况下，按下回车就可以选中当前选项列表中的第一个选项，无需使用鼠标或键盘方向键进行定位。

```

<template>
  <el-select
    v-model="value"
    multiple
    filterable
    allow-create
    default-first-option
    placeholder="请选择文章标签">
    <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>
```

...

...tip 如果 Select 的绑定值为对象类型，请务必指定 `value-key` 作为它的唯一性标识。 ...

Select Attributes

参数	说明	类型	可选值	默认值
value / v-model	绑定值	boolean / string / number	—	—
multiple	是否多选	boolean	—	false
disabled	是否禁用	boolean	—	false
value-key	作为 value 唯一标识的键名，绑定值为对象类型时必须填	string	—	value
size	输入框尺寸	string	medium/small/mini	—
clearable	是否可以清空选项	boolean	—	false
collapse-tags	多选时是否将选中值按文字的形式展示	boolean	—	false
multiple-limit	多选时用户最多可以选择的项目数，为 0 则不限制	number	—	0
name	select input 的 name 属性	string	—	—
autocomplete	select input 的 autocomplete 属性	string	—	off
auto-complete	下个主版本弃用	string	—	off
placeholder	占位符	string	—	请选择
filterable	是否可搜索	boolean	—	false
allow-create	是否允许用户创建新条目，需配合 filterable 使用	boolean	—	false
filter-method	自定义搜索方法	function	—	—
remote	是否为远程搜索	boolean	—	false
remote-method	远程搜索方法	function	—	—

loading	是否正在从远程获取数据	boolean	—	false
loading-text	远程加载时显示的文字	string	—	加载中
no-match-text	搜索条件无匹配时显示的文字，也可以使用slot="empty"设置	string	—	无匹配数据
no-data-text	选项为空时显示的文字，也可以使用slot="empty"设置	string	—	无数据
popper-class	Select 下拉框的类名	string	—	—
reserve-keyword	多选且可搜索时，是否在选中一个选项后保留当前的搜索关键词	boolean	—	false
default-first-option	在输入框按下回车，选择第一个匹配项。需配合 filterable 或 remote 使用	boolean	-	false
popper-append-to-body	是否将弹出框插入至 body 元素。在弹出框的定位出现问题时，可将该属性设置为 false	boolean	-	true
automatic-dropdown	对于不可搜索的 Select，是否在输入框获得焦点后自动弹出选项菜单	boolean	-	false

Select Events

事件名称	说明	回调参数
change	选中值发生变化时触发	目前的选中值
visible-change	下拉框出现/隐藏时触发	出现则为 true，隐藏则为 false
remove-tag	多选模式下移除tag时触发	移除的tag值
clear	可清空的单选模式下用户点击清空按钮时触发	—
blur	当 input 失去焦点时触发	(event: Event)
focus	当 input 获得焦点时触发	(event: Event)

Select Slots

name	说明
—	Option 组件列表
prefix	Select 组件头部内容
empty	无选项时的列表

Option Group Attributes

参数	说明	类型	可选值	默认值

label	分组的组名	string	—	—
disabled	是否将该分组下所有选项置为禁用	boolean	—	false

Option Attributes

参数	说明	类型	可选值	默认值
value	选项的值	string/number/object	—	—
label	选项的标签，若不设置则默认与 value 相同	string/number	—	—
disabled	是否禁用该选项	boolean	—	false

Methods

方法名	说明	参数
focus	使 input 获取焦点	-
blur	使 input 失去焦点，并隐藏下拉框	-