

## Select

Cuando haya muchas opciones, utilice un menú desplegable para mostrar y seleccionar las que desee.

### Uso básico

:::demo v-model es el valor de el-option que está seleccionado actualmente.

```
<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>
```

...

## Disabled en el-option

:::demo Establezca el valor de `disabled` en `el-option` como `true` para deshabilitar esta opción.

```
<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>
```

:::

## Disabled en el-select

Desactivar todo el componente.

:::demo Configure disabled de el-select para deshabilitarla.

```
<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>

:::
```

## Select simple limpiable

Puede limpiar un Select con un icono.

:::demo Establezca el atributo `clearable` para `el-select` y aparecerá un icono. Tenga en cuenta que la opción `clearable` es sólo para una selección individual.

```
<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>

:::
```

### Selección múltiple básica

Selección múltiple utiliza tags para mostrar las opciones seleccionadas.

:::demo Configure el atributo `multiple` para `el-select` para habilitar el modo

múltiple. En este caso, el valor del `v-model` será un array de opciones seleccionadas. De forma predeterminada, las opciones seleccionadas se mostrarán como tags. Se pueden contraer a un texto utilizando el atributo `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>
:::

```

### Personalizar template

Puede personalizar templates HTML para las opciones.

:::demo Inserte templates HTML personalizados en el 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>
:::

```

## Agrupando

Mostrar opciones en grupos.

:::demo Utilice `el-option-group` para agrupar las opciones, y su atributo `label` representa el nombre del grupo.

```

<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: 'Popular cities',
        options: [{
          value: 'Shanghai',
          label: 'Shanghai'
        }
      ]
    }
  ]
}

```

```

    }, {
      value: 'Beijing',
      label: 'Beijing'
    }]
  }, {
    label: 'City name',
    options: [{
      value: 'Chengdu',
      label: 'Chengdu'
    }, {
      value: 'Shenzhen',
      label: 'Shenzhen'
    }, {
      value: 'Guangzhou',
      label: 'Guangzhou'
    }, {
      value: 'Dalian',
      label: 'Dalian'
    }]
  }],
  value: ''
}
}
}
</script>
:::

```

### Filtrado de opciones

Puede filtrar opciones como lo desee.

:::demo Añadir `filterable` a `el-select` permite filtrar. Por defecto, Select buscará todas las opciones cuyo atributo `label` contenga el valor del input. Si prefiere otras estrategias de filtrado, puede pasar el `filter-method`. `filter-method` es una función que se llama cuando el valor del input cambia, y su parámetro es el valor del input actual.

```

<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>

:::

```

## Búsqueda remota

Introduzca palabras y datos para buscar desde el servidor.

:::demo Configure el valor de `filterable` y `remote` con `true` para habilitar la búsqueda remota, y debería pasar el método `remote-method`. `remote-method` es una función que se llama cuando el valor del input cambia, y su parámetro es el valor del input actual. Tenga en cuenta que si `el-option` se presenta con la directiva `v-for`, debe agregar el atributo `key` para `el-option`. Su valor tiene que ser unívoco, como el valor de `item.value` en el ejemplo siguiente.

```

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

```

```

    reserve-keyword
    placeholder="Please enter a keyword"
    :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>
:::

```

### Crear nuevos items

Crear y seleccionar nuevos items que no están incluidas en las opciones de selección.

:::demo Al utilizar el atributo `allow-create`, los usuarios pueden crear nuevos elementos escribiendo en el cuadro del input. Tenga en cuenta que para que `allow-create` funcione, `filterable` debe ser `true`.

```

<template>
  <el-select
    v-model="value"
    multiple
    filterable
    allow-create
    placeholder="Choose tags for your article">
    <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 el valor de encuadernación de Select es un objeto, asegúrese de asignar `value-key` como el nombre único de la clave de identidad.

### Select atributos

Atributo	Descripción	Tipo	Valores aceptados	Por defecto
value / v-model	valor enlazado	boolean / string / num- ber	—	—
multiple	si multiple-select esta activo	boolean	—	false
disabled	si Select esta deshabilitado	boolean	—	false
collapse-tags	si se colapsan los tags a un texto cuando <b>multiple</b> es <b>true</b> .	boolean	—	false
value-key	nombre de clave de identidad única para el valor, necesario cuando el valor es un objeto.	string	—	value
size	tamaño del Input	string	large/small/mini	
clearable	si el select puede ser limpiado	boolean	—	false

Atributo	Descripción	Tipo	Valores aceptados	Por defecto
multiple-limit	máximo numero de opciones que el usuario puede seleccionar cuando <b>multiple</b> es <b>true</b> . Sin límite cuando se fija a 0	number	—	0
name	el atributo <b>name</b> del input seleccionado	string	—	—
autocomplete	el atributo <b>autocomplete</b> del input seleccionado	string	—	off
auto-complete	@DEPRECATED en la proxima major versión	string	—	off
placeholder	placeholder	string	—	Select
filterable	si Select es filtrable	boolean	—	false
allow-create	si esta permitido crear nuevos items. Para usar esto, <b>filterable</b> debe ser <b>true</b> .	boolean	—	false
filter-method	método de filtrado personalizado	function	—	—
remote	si las opciones se traerán desde el servidor	boolean	—	false
remote-method	método de búsqueda remota personalizada	function	—	—
loading	si Select está cargando datos del servidor	boolean	—	false
loading-text	texto mostrado durante la carga de datos del servidor, también puedes usar la configuración de slot = "empty"	string	—	Loading
no-match-text	texto mostrado cuando ningún dato coincide con la consulta de filtrado. También puedes usar la configuración de slot = "empty"	string	—	No matching data
no-data-text	texto mostrado cuando no hay opciones	string	—	No data
popper-class	nombre de clase personalizado para el menú desplegable del Select	string	—	—

Atributo	Descripción	Tipo	Valores aceptados	Por defecto
reserve-keyword	cuando <b>multiple</b> y <b>filter</b> es <b>true</b> , si se debe reservar la palabra clave actual después de seleccionar una opción.	boolean	—	false
default-first-option	seleccione la primera opción de coincidencia en la tecla enter. Uso con <b>filterable</b> o <b>remote</b> .	boolean	-	false
popper-append-to-body	si añadir o no el menu popup al body. Si el posicionamiento del popup es incorrecto, puede intentar poner este <b>prop</b> en <b>false</b> .	boolean	-	true
automatic-dropdown	para non-filterable Select, este <b>prop</b> decide si el menú de opciones aparece cuando la entrada está enfocada	boolean	-	false

### Eventos Select

Nombre	Descripción	Parámetros
change	se dispara cuando el valor del select cambia	valor actual del select
visible-change	se dispara cuando el menú desplegable aparece o desaparece	true cuando aparece, y false en otro caso
remove-tag	se dispara cuando un tag es removido en modo múltiple	el valor del tag removido
clear	se dispara cuando el icono se clickea en un Select limpiable	—
blur	se dispara cuando el input pierde el foco	(event: Event)
focus	se dispara cuando el input obtiene el foco	(event: Event)

### Slots Select

Nombre	Descripción
—	lista de los componentes Option

Nombre	Descripción
prefix	contenido prefix de un Select
empty	Lista sin opciones

### Atributos del grupo de opciones

Atributo	Descripción	Tipo	Valores aceptados	Por defecto
label	nombre del grupo	string	—	—
disabled	si se deshabilitan todas las opciones del grupo	boolean	—	false

### Atributos de opciones

Atributo	Descripción	Tipo	Valores aceptados	Por defecto
value	valor de option	string/number/object	—	—
label	etiqueta de option, es igual a <b>value</b> si se omite	string/number	—	—
disabled	si option esta deshabilitado	boolean	—	false

### Métodos

Método	Descripción	Parámetros
focus	Foco en el componente input	-
blur	Quita el focus del componente y oculta el dropdown	-