

Checkbox

Un groupe d'options à choix multiples.

Usage

Checkbox peut être utilisé seul pour switcher entre deux états.

:::demo Définissez `v-model` (variable liée) dans `el-checkbox`. La valeur par défaut est un `Boolean` pour une seule `checkbox` et devient `true` quand elle est sélectionnée. Le contenu à l'intérieur de la balise `el-checkbox` deviendra le label du bouton de la checkbox.

```
<template>
  <!-- `checked` should be true or false -->
  <el-checkbox v-model="checked">Option</el-checkbox>
</template>
<script>
  export default {
    data() {
      return {
        checked: true
      };
    }
  };
</script>
:::
```

Désactivation

La checkbox peut être désactivée.

:::demo Ajouter l'attribut `disabled`.

```
<template>
  <el-checkbox v-model="checked1" disabled>Option</el-checkbox>
  <el-checkbox v-model="checked2" disabled>Option</el-checkbox>
</template>
<script>
  export default {
    data() {
      return {
        checked1: false,
        checked2: true
      };
    }
  };
</script>
```

...

Checkbox groupées

Utile pour grouper des checkbox, indiquant si une option est sélectionnée en vérifiant la case en question.

:::demo L'élément `checkbox-group` peut gérer plusieurs checkbox d'un groupe en utilisant `v-model`, qui sera un `Array`. Dans l'élément `el-checkbox`, `label` est la valeur de la checkbox. S'il n'y a aucun contenu dans la balise, `label` sera également la description de la checkbox. `label` correspond aussi aux valeurs des éléments dans le tableau. Il est sélectionné si la valeur spécifiée existe dans le tableau, et vice-versa.

```
<template>
  <el-checkbox-group v-model="checkList">
    <el-checkbox label="Option A"></el-checkbox>
    <el-checkbox label="Option B"></el-checkbox>
    <el-checkbox label="Option C"></el-checkbox>
    <el-checkbox label="Désactivée" disabled></el-checkbox>
    <el-checkbox label="Sélectionnée et désactivée" disabled></el-checkbox>
  </el-checkbox-group>
</template>
```

```
<script>
  export default {
    data () {
      return {
        checkList: ['selected and disabled', 'Option A']
      };
    }
  };
</script>
```

...

Indéterminée

La propriété `indeterminate` permet de réaliser un effet "Sélectionner tout".

```
<template>
  <el-checkbox :indeterminate="isIndeterminate" v-model="checkAll" @change="handleCheckAllChange">
    <div style="margin: 15px 0;"></div>
    <el-checkbox-group v-model="checkedCities" @change="handleCheckedCitiesChange">
      <el-checkbox v-for="city in cities" :label="city" :key="city">{{city}}</el-checkbox>
    </el-checkbox-group>
  </template>
<script>
```

```

const cityOptions = ['Shanghai', 'Beijing', 'Guangzhou', 'Shenzhen'];
export default {
  data() {
    return {
      checkAll: false,
      checkedCities: ['Shanghai', 'Beijing'],
      cities: cityOptions,
      isIndeterminate: true
    };
  },
  methods: {
    handleCheckAllChange(val) {
      this.checkedCities = val ? cityOptions : [];
      this.isIndeterminate = false;
    },
    handleCheckedCitiesChange(value) {
      let checkedCount = value.length;
      this.checkAll = checkedCount === this.cities.length;
      this.isIndeterminate = checkedCount > 0 && checkedCount < this.cities.length;
    }
  }
};
</script>

```

Minimum / Maximum d'éléments sélectionnés

Les propriétés min et max permettent de limiter la quantité d'éléments sélectionnés.

```

<template>
  <el-checkbox-group
    v-model="checkedCities"
    :min="1"
    :max="2">
    <el-checkbox v-for="city in cities" :label="city" :key="city">{{city}}</el-checkbox>
  </el-checkbox-group>
</template>
<script>
const cityOptions = ['Shanghai', 'Beijing', 'Guangzhou', 'Shenzhen'];
export default {
  data() {
    return {
      checkedCities: ['Shanghai', 'Beijing'],
      cities: cityOptions
    };
  }
}

```

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

Style bouton

Des checkbox avec une apparence de bouton.

:::demo Il vous suffit de changer `el-checkbox` en `el-checkbox-button`. Il existe aussi un attribut `size`.

```
<template>
  <div>
    <el-checkbox-group v-model="checkboxGroup1">
      <el-checkbox-button v-for="city in cities" :label="city" :key="city">{{city}}</el-checkbox-button>
    </el-checkbox-group>
  </div>
  <div style="margin-top: 20px">
    <el-checkbox-group v-model="checkboxGroup2" size="medium">
      <el-checkbox-button v-for="city in cities" :label="city" :key="city">{{city}}</el-checkbox-button>
    </el-checkbox-group>
  </div>
  <div style="margin-top: 20px">
    <el-checkbox-group v-model="checkboxGroup3" size="small">
      <el-checkbox-button v-for="city in cities" :label="city" :disabled="city === 'Beijing'">{{city}}</el-checkbox-button>
    </el-checkbox-group>
  </div>
  <div style="margin-top: 20px">
    <el-checkbox-group v-model="checkboxGroup4" size="mini" disabled>
      <el-checkbox-button v-for="city in cities" :label="city" :key="city">{{city}}</el-checkbox-button>
    </el-checkbox-group>
  </div>
</template>
<script>
  const cityOptions = ['Shanghai', 'Beijing', 'Guangzhou', 'Shenzhen'];

  export default {
    data () {
      return {
        checkboxGroup1: ['Shanghai'],
        checkboxGroup2: ['Shanghai'],
        checkboxGroup3: ['Shanghai'],
        checkboxGroup4: ['Shanghai'],
        cities: cityOptions
      };
    }
  }
</script>
```

...

Bordures

:::demo L'attribut `border` ajoute une bordure aux checkbox.

```
<template>
  <div>
    <el-checkbox v-model="checked1" label="Option1" border></el-checkbox>
    <el-checkbox v-model="checked2" label="Option2" border></el-checkbox>
  </div>
  <div style="margin-top: 20px">
    <el-checkbox v-model="checked3" label="Option1" border size="medium"></el-checkbox>
    <el-checkbox v-model="checked4" label="Option2" border size="medium"></el-checkbox>
  </div>
  <div style="margin-top: 20px">
    <el-checkbox-group v-model="checkboxGroup1" size="small">
      <el-checkbox label="Option1" border></el-checkbox>
      <el-checkbox label="Option2" border disabled></el-checkbox>
    </el-checkbox-group>
  </div>
  <div style="margin-top: 20px">
    <el-checkbox-group v-model="checkboxGroup2" size="mini" disabled>
      <el-checkbox label="Option1" border></el-checkbox>
      <el-checkbox label="Option2" border></el-checkbox>
    </el-checkbox-group>
  </div>
</template>

<script>
  export default {
    data () {
      return {
        checked1: true,
        checked2: false,
        checked3: false,
        checked4: true,
        checkboxGroup1: [],
        checkboxGroup2: []
      };
    }
  }
</script>
...
```

Attributs Checkbox

Attribut	Description	Type	Options	Défaut
value / v-model	La valeur liée.	string / number / boolean	—	—
label	Valeur de la checkbox quand utilisée dans un checkbox-group .	string / number / boolean	—	—
true-label	Valeur de la checkbox si cochée.	string / number	—	—
false-label	Valeur de la checkbox si non cochée.	string / number	—	—
disabled	Désactive la checkbox.	boolean	—	false
border	Ajoute une bordure à la checkbox.	boolean	—	false
size	taille de la checkbox, ne marche que si border est ‘true’	string	medium / small / mini	—
name	Attribut ‘name’ natif.	string	—	—
checked	Si la checkbox est cochée.	boolean	—	false
indeterminate	Identique à indeterminate dans les checkbox natives.	boolean	—	false

Évènements Checkbox

Nom	Description	Paramètres
change	Se déclenche quand la valeur change.	La nouvelle valeur

Attributs Checkbox-group

Attribut	Description	Type	Options	Défaut
value / v-model	La valeur liée.	array	—	—
size	Taille des checkbox-boutons ou des checkbox avec bordure.	string	medium / small / mini	—
disabled	Si les checkbox imbriquées sont désactivées.	boolean	—	false
min	Nombre minimum de checkbox cochées.	number	—	—
max	Nombre maximum de checkbox cochées.	number	—	—
text-color	Couleur de police quand le bouton est actif.	string	—	#ffff
fill	Couleur de la bordure et du fond quand le bouton est actif.	string	—	#409EFF

Évènements **Checkbox-group**

Nom	Description	Paramètres
change	Se déclenche quand la valeur change.	La nouvelle valeur

Attributs **Checkbox-button**

Attribut	Description	Type	Options	Défaut
label	Valeur de la checkbox quand utilisée dans un checkbox-group	string / number / boolean	—	—
true-label	Valeur de la checkbox si cochée.	string / number	—	—
false-label	Valeur de la checkbox si non cochée.	string / number	—	—
disabled	Si la checkbox est désactivée.	boolean	—	false
name	Attribut 'name' natif.	string	—	—
checked	Si la checkbox est cochée.	boolean	—	false