Checkbox

A group of options for multiple choices.

Basic usage

Checkbox can be used alone to switch between two states.

:::demo Define `v-model`(bind variable) in `el-checkbox`. The default value is a `Boolean` for single `checkbox`, and it becomes `true` when selected. Content inside the `el-checkbox` tag will become the description following the button of the 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>
```

:::

Disabled State

Disabled state for checkbox.

:::demo Set the `disabled` attribute.

```
<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 group

It is used for multiple checkboxes which are bound in one group, and indicates whether one option is selected by checking if it is checked.

:::demo `checkbox-group` element can manage multiple checkboxes in one group by using `v-model` which is bound as an `Array`. Inside the `el-checkbox` element, `label` is the value of the checkbox. If no content is nested in that tag, `label` will be rendered as the description following the button of the checkbox. `label` also corresponds with the element values in the array. It is selected if the specified value exists in the array, and 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="disabled" disabled></el-checkbox>
    <el-checkbox label="selected and disabled" disabled></el-checkbox>
  </el-checkbox-group>
</template>

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

:::

Indeterminate

The `indeterminate` property can help you to achieve a 'check all' effect.

```
<template>
  <el-checkbox :indeterminate="isIndeterminate" v-model="checkAll" @change="handleCheckAllCh
  <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 items checked

The `min` and `max` properties can help you to limit the number of checked items.

```
<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>
```

Button style

Checkbox with button styles.

:::demo You just need to change `el-checkbox` element into `el-checkbox-button` element. We also provide `size` attribute.

```
<template>
  <div>
    <el-checkbox-group v-model="checkboxGroup1">
      <el-checkbox-button v-for="city in cities" :label="city" :key="city">{{city}}</el-chec
    </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-chec
    </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
    </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-chec
    </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>
```

:::

With borders

:::demo The `border` attribute adds a border to Checkboxes.

```
<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>
```

:::

Checkbox Attributes

Attribute	Description	Type	Options	Default
value / v-model	binding value	string / number / boolean	—	—
label	value of the Checkbox when used inside a `checkbox-group`	string / number / boolean	—	—
true-label	value of the Checkbox if it's checked	string / number	—	—
false-label	value of the Checkbox if it's not checked	string / number	—	—
disabled	whether the Checkbox is disabled	boolean	—	false
border	whether to add a border around Checkbox	boolean	—	false
size	size of the Checkbox, only works when `border` is true	string	medium / small / mini	—
name	native 'name' attribute	string	—	—
checked	if the Checkbox is checked	boolean	—	false

Attribute	Description	Type	Options	Default
indeterminate	same as `indeterminate` in native checkbox	boolean	—	false

Checkbox Events

Event Name	Description	Parameters
change	triggers when the binding value changes	the updated value

Checkbox-group Attributes

Attribute	Description	Type	Options	Default
value / v-model	binding value	array	—	—
size	size of checkbox buttons or bordered checkboxes	string	medium / small / mini	—
disabled	whether the nesting checkboxes are disabled	boolean	—	false
min	minimum number of checkbox checked	number	—	—
max	maximum number of checkbox checked	number	—	—
text-color	font color when button is active	string	—	#ffffff

Attribute	Description	Type	Options	Default
fill	border and background color when button is active	string	—	#409EFF

Checkbox-group Events

Event Name	Description	Parameters
change	triggers when the binding value changes	the updated value

Checkbox-button Attributes

Attribute	Description	Type	Options	Default
label	value of the checkbox when used inside a `checkbox-group`	string / number / boolean	—	—
true-label	value of the checkbox if it's checked	string / number	—	—
false-label	value of the checkbox if it's not checked	string / number	—	—
disabled	whether the checkbox is disabled	boolean	—	false
name	native 'name' attribute	string	—	—

Attribute	Description	Type	Options	Default
checked	if the checkbox is checked	boolean	—	false