

InputNumber

Input numerical values with a customizable range.

Basic usage

demo Bind a variable to `v-model` in `<el-input-number>` element and you are set.

```
<template>
  <el-input-number v-model="num" @change="handleChange" :min="1" :max="10"></el-
input-number>
</template>
<script>
  export default {
    data() {
      return {
        num: 1
      };
    },
    methods: {
      handleChange(value) {
        console.log(value)
      }
    }
  };
</script>
```

...

Disabled

demo The `disabled` attribute accepts a `boolean`, and if the value is `true`, the component is disabled. If you just need to control the value within a range, you can add `min` attribute to set the minimum value and `max` to set the maximum value. By default, the minimum value is `0`.

```
<template>
  <el-input-number v-model="num" :disabled="true"></el-input-number>
</template>
<script>
  export default {
    data() {
      return {
        num: 1
      }
    }
  };
</script>
```

...

Steps

Allows you to define incremental steps.

:::demo Add `step` attribute to set the step.

```
<template>
  <el-input-number v-model="num" :step="2"></el-input-number>
</template>
<script>
  export default {
    data() {
      return {
        num: 5
      }
    }
  };
</script>
```

...

Step strictly

:::demo The `step-strictly` attribute accepts a `boolean`. If this attribute is `true`, input value can only be multiple of step.

```
<template>
  <el-input-number v-model="num" :step="2" step-strictly></el-input-number>
</template>
<script>
  export default {
    data() {
      return {
        num: 2
      }
    }
  };
</script>
```

...

Precision

:::demo Add `precision` attribute to set the precision of input value.

```
<template>
  <el-input-number v-model="num" :precision="2" :step="0.1" :max="10"></el-input-
number>
</template>
<script>
  export default {
    data() {
      return {
```

```

        num: 1
      }
    }
  };
</script>

```

...

...tip The value of `precision` must be a non negative integer and should not be less than the decimal places of `step`. ...

Size

Use attribute `size` to set additional sizes with `medium`, `small` or `mini`.

...demo

```

<template>
  <el-input-number v-model="num1"></el-input-number>
  <el-input-number size="medium" v-model="num2"></el-input-number>
  <el-input-number size="small" v-model="num3"></el-input-number>
  <el-input-number size="mini" v-model="num4"></el-input-number>
</template>
<script>
  export default {
    data() {
      return {
        num1: 1,
        num2: 1,
        num3: 1,
        num4: 1
      }
    }
  };
</script>

```

...

Controls Position

...demo Set `controls-position` to decide the position of control buttons.

```

<template>
  <el-input-number v-model="num" controls-position="right" @change="handleChange"
  :min="1" :max="10"></el-input-number>
</template>
<script>
  export default {
    data() {
      return {
        num: 1
      };
    }
  };

```

```

    },
    methods: {
      handleChange(value) {
        console.log(value);
      }
    }
  };
</script>

```

...

Attributes

Attribute	Description	Type	Accepted Values	Default
value / v-model	binding value	number	—	0
min	the minimum allowed value	number	—	-Infinity
max	the maximum allowed value	number	—	Infinity
step	incremental step	number	—	1
step-strictly	whether input value can only be multiple of step	boolean	—	false
precision	precision of input value	number	—	—
size	size of the component	string	large/small	—
disabled	whether the component is disabled	boolean	—	false
controls	whether to enable the control buttons	boolean	—	true
controls-position	position of the control buttons	string	right	-
name	same as <code>name</code> in native input	string	—	—
label	label text	string	—	—
placeholder	placeholder in input	string	-	-

Events

Event Name	Description	Parameters
change	triggers when the value changes	currentValue, oldValue
blur	triggers when Input blurs	(event: Event)
focus	triggers when Input focuses	(event: Event)

Methods

Method	Description	Parameters
focus	focus the Input component	-
select	select the text in input element	—