

Progress

Progress is used to show the progress of current operation, and inform the user the current status.

Linear progress bar

:::demo Use `percentage` attribute to set the percentage. It's **required** and must be between 0-100. You can custom text format by setting `format`.

```
<el-progress :percentage="50"></el-progress>
<el-progress :percentage="100" :format="format"></el-progress>
<el-progress :percentage="100" status="success"></el-progress>
<el-progress :percentage="100" status="warning"></el-progress>
<el-progress :percentage="50" status="exception"></el-progress>

<script>
  export default {
    methods: {
      format(percentage) {
        return percentage === 100 ? 'Full' : `${percentage}%`;
      }
    }
  };
</script>

:::
```

Internal percentage

In this case the percentage takes no additional space.

:::demo `stroke-width` attribute decides the width of progress bar, and use `text-inside` attribute to put description inside the progress bar.

```
<el-progress :text-inside="true" :stroke-width="26" :percentage="70"></el-progress>
<el-progress :text-inside="true" :stroke-width="24" :percentage="100" status="success"></el-progress>
<el-progress :text-inside="true" :stroke-width="22" :percentage="80" status="warning"></el-progress>
<el-progress :text-inside="true" :stroke-width="20" :percentage="50" status="exception"></el-progress>

:::
```

Custom color

You can use `color` attr to set the progress bar color. it accepts color string, function, or array.

```
<el-progress :percentage="percentage" :color="customColor"></el-progress>

<el-progress :percentage="percentage" :color="customColorMethod"></el-progress>
```

```

<el-progress :percentage="percentage" :color="customColors"></el-progress>
<div>
  <el-button-group>
    <el-button icon="el-icon-minus" @click="decrease"></el-button>
    <el-button icon="el-icon-plus" @click="increase"></el-button>
  </el-button-group>
</div>

<script>
export default {
  data() {
    return {
      percentage: 20,
      customColor: '#409eff',
      customColors: [
        {color: '#f56c6c', percentage: 20},
        {color: '#e6a23c', percentage: 40},
        {color: '#5cb87a', percentage: 60},
        {color: '#1989fa', percentage: 80},
        {color: '#6f7ad3', percentage: 100}
      ]
    };
  },
  methods: {
    customColorMethod(percentage) {
      if (percentage < 30) {
        return '#909399';
      } else if (percentage < 70) {
        return '#e6a23c';
      } else {
        return '#67c23a';
      }
    },
    increase() {
      this.percentage += 10;
      if (this.percentage > 100) {
        this.percentage = 100;
      }
    },
    decrease() {
      this.percentage -= 10;
      if (this.percentage < 0) {
        this.percentage = 0;
      }
    }
  }
}

```

```

    }
  }
</script>

```

Circular progress bar

:::demo You can specify `type` attribute to `circle` to use circular progress bar, and use `width` attribute to change the size of circle.

```

<el-progress type="circle" :percentage="0"></el-progress>
<el-progress type="circle" :percentage="25"></el-progress>
<el-progress type="circle" :percentage="100" status="success"></el-progress>
<el-progress type="circle" :percentage="70" status="warning"></el-progress>
<el-progress type="circle" :percentage="50" status="exception"></el-progress>
:::

```

Dashboard progress bar

You also can specify `type` attribute to `dashboard` to use dashboard progress bar.

```

<el-progress type="dashboard" :percentage="percentage" :color="colors"></el-progress>
<div>
  <el-button-group>
    <el-button icon="el-icon-minus" @click="decrease"></el-button>
    <el-button icon="el-icon-plus" @click="increase"></el-button>
  </el-button-group>
</div>

<script>
  export default {
    data() {
      return {
        percentage: 10,
        colors: [
          {color: '#f56c6c', percentage: 20},
          {color: '#e6a23c', percentage: 40},
          {color: '#5cb87a', percentage: 60},
          {color: '#1989fa', percentage: 80},
          {color: '#6f7ad3', percentage: 100}
        ]
      }
    },
    methods: {
      increase() {
        this.percentage += 10;
        if (this.percentage > 100) {

```

```

        this.percentage = 100;
    },
    decrease() {
        this.percentage -= 10;
        if (this.percentage < 0) {
            this.percentage = 0;
        }
    }
}
}
</script>

```

Attributes

Attribute	Description	Type	Accepted Values	Default
percentage	percentage, required	number	0-100	0
type	the type of progress bar	string	line/circle/dashboard	line
stroke-width	the width of progress bar	number	—	6
text-inside	whether to place the percentage inside progress bar, only works when type is 'line'	boolean	—	false
status	the current status of progress bar	string	success/exception/warning	
color	background color of progress bar. Overrides status prop	string/function/array		”
width	the canvas width of circle progress bar	number	—	126
show-text	whether to show percentage	boolean	—	true

Attribute	Description	Type	Accepted Values	Default
stroke-linecap	circle/dashboard type shape at the end path	string	butt/round/square	round
format	custom text format	function(percentage)		—