

Tag 标签

用于标记和选择。

基础用法

::demo 由 `type` 属性来选择tag的类型，也可以通过 `color` 属性来自定义背景色。

```
<el-tag>标签一</el-tag>
<el-tag type="success">标签二</el-tag>
<el-tag type="info">标签三</el-tag>
<el-tag type="warning">标签四</el-tag>
<el-tag type="danger">标签五</el-tag>
```

...

可移除标签

::demo 设置 `closable` 属性可以定义一个标签是否可移除。默认的标志移除时会附带渐变动画，如果不想使用，可以设置 `disable-transitions` 属性，它接受一个 `Boolean`，`true` 为关闭。

```
<el-tag
  v-for="tag in tags"
  :key="tag.name"
  closable
  :type="tag.type">
  {{tag.name}}
</el-tag>

<script>
  export default {
    data() {
      return {
        tags: [
          { name: '标签一', type: '' },
          { name: '标签二', type: 'success' },
          { name: '标签三', type: 'info' },
          { name: '标签四', type: 'warning' },
          { name: '标签五', type: 'danger' }
        ]
      };
    }
  }
</script>
```

...

动态编辑标签

动态编辑标签可以通过点击标签关闭按钮后触发的 `close` 事件来实现

::demo

```

<el-tag
  :key="tag"
  v-for="tag in dynamicTags"
  closable
  :disable-transitions="false"
  @close="handleClose(tag)">
  {{tag}}
</el-tag>
<el-input
  class="input-new-tag"
  v-if="inputVisible"
  v-model="inputValue"
  ref="saveTagInput"
  size="small"
  @keyup.enter.native="handleInputConfirm"
  @blur="handleInputConfirm"
>
</el-input>
<el-button v-else class="button-new-tag" size="small" @click="showInput">+ New
Tag</el-button>

<style>
.el-tag + .el-tag {
  margin-left: 10px;
}
.button-new-tag {
  margin-left: 10px;
  height: 32px;
  line-height: 30px;
  padding-top: 0;
  padding-bottom: 0;
}
.input-new-tag {
  width: 90px;
  margin-left: 10px;
  vertical-align: bottom;
}
</style>

<script>
export default {
  data() {
    return {
      dynamicTags: ['标签一', '标签二', '标签三'],
      inputVisible: false,
      inputValue: ''
    };
  },
  methods: {
    handleClose(tag) {
      this.dynamicTags.splice(this.dynamicTags.indexOf(tag), 1);
    }
  }
}

```

```

    },

    showInput() {
      this.inputVisible = true;
      this.$nextTick(_ => {
        this.$refs.saveTagInput.$refs.input.focus();
      });
    },

    handleInputConfirm() {
      let inputValue = this.inputValue;
      if (inputValue) {
        this.dynamicTags.push(inputValue);
      }
      this.inputVisible = false;
      this.inputValue = '';
    }
  }
}
</script>

```

...

不同尺寸

Tag 组件提供除了默认值以外的三种尺寸，可以在不同场景下选择合适的按钮尺寸。

:::demo 额外的尺寸： `medium`、`small`、`mini`，通过设置 `size` 属性来配置它们。

```

<el-tag closable>默认标签</el-tag>
<el-tag size="medium" closable>中等标签</el-tag>
<el-tag size="small" closable>小型标签</el-tag>
<el-tag size="mini" closable>超小标签</el-tag>

```

...

不同主题

Tag 组件提供了三个不同的主题： `dark`、`light` 和 `plain`

:::demo 通过设置 `effect` 属性来改变主题，默认为 `light`

```

<div class="tag-group">
  <span class="tag-group__title">Dark</span>
  <el-tag
    v-for="item in items"
    :key="item.label"
    :type="item.type"
    effect="dark">
    {{ item.label }}
  </el-tag>
</div>

```

```
<div class="tag-group">
  <span class="tag-group__title">Plain</span>
  <el-tag
    v-for="item in items"
    :key="item.label"
    :type="item.type"
    effect="plain">
    {{ item.label }}
  </el-tag>
</div>

<script>
  export default {
    data() {
      return {
        items: [
          { type: '', label: '标签一' },
          { type: 'success', label: '标签二' },
          { type: 'info', label: '标签三' },
          { type: 'danger', label: '标签四' },
          { type: 'warning', label: '标签五' }
        ]
      }
    }
  }
</script>
```

⋮

Attributes

参数	说明	类型	可选值	默认值
type	类型	string	success/info/warning/danger	—
closable	是否可关闭	boolean	—	false
disable-transitions	是否禁用渐变动画	boolean	—	false
hit	是否有边框描边	boolean	—	false
color	背景色	string	—	—
size	尺寸	string	medium / small / mini	—
effect	主题	string	dark / light / plain	light

Events

事件名称	说明	回调参数
click	点击 Tag 时触发的事件	—
close	关闭 Tag 时触发的事件	—