

Tag

Used for marking and selection.

Basic usage

:::demo Use the `type` attribute to define Tag's type. In addition, the `color` attribute can be used to set the background color of the Tag.

```
<el-tag>Tag 1</el-tag>
<el-tag type="success">Tag 2</el-tag>
<el-tag type="info">Tag 3</el-tag>
<el-tag type="warning">Tag 4</el-tag>
<el-tag type="danger">Tag 5</el-tag>
```

...

Removable Tag

:::demo `closable` attribute can be used to define a removable tag. It accepts a `Boolean`. By default the removal of Tag has a fading animation. If you don't want to use it, you can set the `disable-transitions` attribute, which accepts a `Boolean`, to `true`. `close` event triggers when Tag is removed.

```
<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: 'Tag 1', type: '' },
          { name: 'Tag 2', type: 'success' },
          { name: 'Tag 3', type: 'info' },
          { name: 'Tag 4', type: 'warning' },
          { name: 'Tag 5', type: 'danger' }
        ]
      };
    }
  }
</script>
```

...

Edit Dynamically

You can use the `close` event to add and remove tag dynamically.

:::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="mini"
  @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: ['Tag 1', 'Tag 2', 'Tag 3'],
      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>

```

...

Sizes

Besides default size, Tag component provides three additional sizes for you to choose among different scenarios.

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

```

<el-tag>Default</el-tag>
<el-tag size="medium">Medium</el-tag>
<el-tag size="small">Small</el-tag>
<el-tag size="mini">Mini</el-tag>

```

...

Theme

Tag provide three different themes: `dark`, `light` and `plain`

:::demo Using `effect` to change, default is `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: 'Tag 1' },
          { type: 'success', label: 'Tag 2' },
          { type: 'info', label: 'Tag 3' },
          { type: 'danger', label: 'Tag 4' },
          { type: 'warning', label: 'Tag 5' }
        ]
      }
    }
  }
</script>
```

...

Attributes

Attribute	Description	Type	Accepted Values	Default
type	component type	string	success/info/warning/danger	—
closable	whether Tag can be removed	boolean	—	false
disable-transitions	whether to disable animations	boolean	—	false
hit	whether Tag has a highlighted border	boolean	—	false
color	background color of the Tag	string	—	—
size	tag size	string	medium / small / mini	—
effect	component theme	string	dark / light / plain	light

Events

--	--	--

Event Name	Description	Parameters
click	triggers when Tag is clicked	—
close	triggers when Tag is removed	—