

Tag

Les tags sont utiles pour marquer certaines données afin d'ajouter des informations et de les retrouver plus facilement.

Usage

:::demo Utilisez l'attribut `type` pour définir le type de tag. De plus, l'attribut `color` détermine la couleur de fond.

```
<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>
```

:::

Supprimer des tags

:::demo L'attribut `closable` détermine si un tag est supprimable grâce à un `Boolean`. Par défaut la suppression bénéficie d'une animation de fading. Utilisez `disable-transitions` si vous ne souhaitez pas d'animations en utilisant un `Boolean` à `true`. L'évènement `close` se déclenche quand un tag est supprimé.

```
<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>
```

...

Édition dynamique

Vous pouvez utiliser l'évènement `close` pour ajouter et supprimer des tags dynamiquement.

```
<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">+ Nouveau Tag</el-b

<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>

```

Tailles

En plus de la taille par défaut, Tag fournit d'autres tailles pour vos composants.

:::demo Utilisez `size` pour choisir une autre taille parmi `medium`, `small` ou `mini`.

```

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

```

:::

Theme

Les balises utilisent trois thèmes différents: `dark`, `light` et `plain`

:::demo Utilisez **effect** pour changer. La valeur par défaut est **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>

:::
```

Attributs

Attribut	Description	Type	Valeurs acceptées	Défaut
type	Type du tag.	string	success/info/warning/danger	—

Attribut	Description	Type	Valeurs acceptées	Défaut
closable	Si le tag peut être supprimé.	boolean	—	false
disable-transitions	Si les animations sont désactivées.	boolean	—	false
hit	Si le tag à une bordure mise en valeur.	boolean	—	false
color	Couleur de fond du tag.	string	—	—
size	Taille du tag.	string	medium / small / mini	—
effect	component theme	string	dark / light / plain	light

Évènements

Nom	Description	Paramètres
click	Se déclenche quand le tag est cliqué.	—
close	Se déclenche quand le tag est supprimé.	—