

以下是tdesign(vue3)的文档

安装

使用 npm 安装

推荐使用 npm 方式进行开发

```
npm i tdesign-vue-next
```

通过 浏览器引入 安装

目前可以通过 unpkg.com/tdesign-vue-next 获取到最新版本的资源，在页面上引入 js 和 css 文件即可开始使用。

```
<script src="https://unpkg.com/vue@next"></script>
<link rel="stylesheet" href="https://unpkg.com/tdesign-vue-next/dist/tdesign.min.css" />
<script src="https://unpkg.com/tdesign-vue-next/dist/tdesign.min.js"></script>
```

```
...
```

```
<script>
  Vue.createApp({}).use(TDesign).mount('#app');
</script>
```

npm package 中提供了多种构建产物，可以阅读 [这里](#) 了解不同目录下产物的差别。

使用

基础使用

```
...
```

```
import { createApp } from 'vue';
import TDesign from 'tdesign-vue-next';
import App from './app.vue';
```

// 引入组件库全局样式资源

```
import 'tdesign-vue-next/es/style/index.css';
```

```
const app = createApp(App);
```

```
app.use(TDesign);
```

npm package 中提供了多种构建产物，可以阅读 [这里](#) 了解不同目录下产物的差别。

按需引入使用

对产物大小有要求的场景，可以按需引入组件，借助 Webpack 或 Rollup 等支持 tree-shaking 特性的构建工具达到按需引入的使用效果。

```
import { createApp } from 'vue';
import { Button as TButton } from 'tdesign-vue-next';
import App from './app.vue';
```

// 引入组件库全局样式资源

```
import 'tdesign-vue-next/es/style/index.css';
```

```
const app = createApp(App);
```

```
app.use(TButton);
```

```
...
```

通过插件按需引用

除此之外，也可以使用 unplugin-vue-components 和 unplugin-auto-import 来实现自动导入：

```
npm install -D unplugin-vue-components unplugin-auto-import
```

然后在 Webpack 或 Vite 对应的配置文件添加上述插件。

Vite

```
...
```

```
import AutoImport from 'unplugin-auto-import/vite';
import Components from 'unplugin-vue-components/vite';
import { TDesignResolver } from 'unplugin-vue-components/resolvers';
export default {
  plugins: [
    // ...
    AutoImport({
      resolvers: [TDesignResolver({
        library: 'vue-next'
      })],
    }),
  ],
},
```

```

    Components({
      resolvers: [TDesignResolver({
        library: 'vue-next',
      })],
    }),
  ],
};
Webpack
const AutoImport = require('unplugin-auto-import/webpack');
const Components = require('unplugin-vue-components/webpack');
const { TDesignResolver } = require('unplugin-vue-components/resolvers');
module.exports = {
  // ...
  plugins: [
    AutoImport({
      resolvers: [TDesignResolver({
        library: 'vue-next',
      })],
    }),
    Components({
      resolvers: [TDesignResolver({
        library: 'vue-next',
      })],
    }),
  ],
};
};

```

TDesignResolver 支持的配置，可以点击此链接。

编辑器提示

安装注册 TDesign 之后，在开发项目时，可以配合插件在VSCode等主流编辑器中达到提示组件名及API的效果。

推荐安装 volar，并在项目的 tsconfig.json 的 includes 属性中增加node_modules/tdesign-vue-next/global.d.ts，即可实现该效果。

全局特性配置

Pagination 分页

```

<template>
  <t-config-provider :global-config="globalConfig">
    <div class="tdesign-demo-item--locale-provider-base">
      <t-pagination v-model="current" :total="36" show-jumper :max-page-btn="5" />
    </div>
  </t-config-provider>
</template>

<script setup>
import { ref } from 'vue';

const current = ref(1);
const globalConfig = {
  pagination: {
    itemsPerPage: '{size} / page',
    jumpTo: 'jump to',
    page: '',
    total: 'Total {total} items',
  },
};
</script>

```

```

[componentName:"button",apiData:[{docTitle:"Button Props说明",tableRows:
[attribute:"block",description:"Boolean",type:"false",default:"是否为块级元素"},
{attribute:"content",description:"String / Slot / Function",type:"-","default:"按钮内容。TS 类型:
string | TNode。通用类型定义"},{attribute:"default",description:"String / Slot / Function",type:"-
",default:"按钮内容。TS 类型: string | TNode。通用类型定义"},
{attribute:"disabled",description:"Boolean",type:"false",default:"禁用状态"},
{attribute:"ghost",description:"Boolean",type:"false",default:"是否为幽灵按钮（镂空按钮）"},

```

{attribute:"href",description:"String",type:"-","default:"跳转地址。href 存在时, 按钮标签默认使用 <a> 渲染; 如果指定了 tag 则使用指定的标签渲染"}, {attribute:"icon",description:"Slot / Function",type:"-","default:"按钮内部图标, 可完全自定义。TS 类型: TNode。通用类型定义"}, {attribute:"loading",description:"Boolean",type:"false",default:"是否显示为加载状态"}, {attribute:"shape",description:"String",type:"rectangle",default:"按钮形状, 有 4 种: 长方形、正方形、圆角长方形、圆形。可选项: rectangle/square/round/circle"}, {attribute:"size",description:"String",type:"medium",default:"组件尺寸。可选项: small/medium/large。TS 类型: SizeEnum。通用类型定义"}, {attribute:"suffix",description:"Slot / Function",type:"-","default:"右侧内容, 可用于定义右侧图标。TS 类型: TNode。通用类型定义"}, {attribute:"tag",description:"String",type:"-","default:"渲染按钮的 HTML 标签, 默认使用标签 <button> 渲染, 可以自定义为 <a> <div> 等。透传全部 HTML 属性, 如: href/target/data-* 等。⚠ 禁用按钮 <button disabled>无法显示 Popup 浮层信息, 可通过修改 tag=div 解决这个问题。可选项: button/a/div"}, {attribute:"theme",description:"String",type:"-","default:"组件风格, 依次为默认色、品牌色、危险色、警告色、成功色。可选项: default/primary/danger/warning/success"}, {attribute:"type",description:"String",type:"button",default:"按钮类型。可选项: submit/reset/button"}, {attribute:"variant",description:"String",type:"base",default:"按钮形式, 基础、线框、虚线、文字。可选项: base/outline/dashed/text"}, {attribute:"onClick",description:"Function",type:"","default:"TS 类型: (e: MouseEvent) => void 点击时触发"}], {docTitle:"Button Events说明", tableRows:[{attribute:"click",description:"(e: MouseEvent)", type:"点击时触发"}]}], {componentName:"icon", apiData:[{docTitle:"IconSVG Props说明", tableRows:[{attribute:"loadDefaultIcons",description:"Boolean", type:"true", default:"是否加载组件库内置图标"}, {attribute:"name",description:"String", type:"-","default:"必需。图标名称"}, {attribute:"size",description:"String", type:"undefined", default:"图标尺寸, 支持 'small', 'medium', 'large', '35px', '3em' 等"}, {attribute:"style",description:"String", type:"-","default:'HTML 原生属性。可用于设置图标颜色, 如: style='color: red'"}, {attribute:"url",description:"String / Array", type:"-","default:"图标地址, 地址内容参考组件内部默认加载图标。TS 类型: string | Array<string>"}, {attribute:"onClick",description:"Function", type:"","default:"TS 类型: (context: { e: MouseEvent }) => void 点击时触发"}]}], {docTitle:"IconSVG Events说明", tableRows:[{attribute:"click",description:"(context: { e: MouseEvent })", type:"点击时触发"}]}, {docTitle:"Iconfont Props说明", tableRows:[{attribute:"loadDefaultIcons",description:"Boolean", type:"true", default:"是否加载组件库内置图标"}, {attribute:"name",description:"String", type:"-","default:"必需。图标名称"}, {attribute:"size",description:"String", type:"undefined", default:"图标尺寸, 支持 'small', 'medium', 'large', '35px', '3em' 等"}, {attribute:"style",description:"String", type:"-","default:'HTML 原生属性。可用于设置图标颜色, 如: style='color: red'"}, {attribute:"tag",description:"String", type:"i", default:"图标 DOM 元素, 可选值: i/span/div/..."}, {attribute:"url",description:"String / Array", type:"-","default:"图标地址, 地址内容参考组件内部默认加载图标。也可以在 index.html 中引入图标地址。TS 类型: string | Array<string>"}, {attribute:"onClick",description:"Function", type:"","default:"TS 类型: (context: { e: MouseEvent }) => void 点击时触发"}]}], {docTitle:"Iconfont Events说明", tableRows:[{attribute:"click",description:"(context: { e: MouseEvent })", type:"点击时触发"}]}], {componentName:"link", apiData:[{docTitle:"Link Props说明", tableRows:[{attribute:"content",description:"String / Slot / Function", type:"-","default:"链接内容。TS 类型: string | TNode。通用类型定义"}, {attribute:"default",description:"String / Slot / Function", type:"-","default:"链接内容, 同 content。TS 类型: string | TNode。通用类型定义"}, {attribute:"disabled",description:"Boolean", type:"-","default:"禁用链接"}, {attribute:"hover",description:"String", type:"underline", default:"链接悬浮态样式, 有 文本颜色变化、添加下划线等 2 种方法。可选项: color/underline"}, {attribute:"href",description:"String", type:"-","default:"跳转链接"}, {attribute:"prefixIcon",description:"Slot / Function", type:"-","default:"前置图标。TS 类型: TNode。通用类型定义"}, {attribute:"size",description:"String", type:"medium", default:"尺寸。可选项: small/medium/large。TS 类型: SizeEnum。通用类型定义"}, {attribute:"suffixIcon",description:"Slot / Function", type:"-","default:"后置图标。TS 类型: TNode。通用类型定义"}, {attribute:"target",description:"String", type:"-","default:"跳转方式, 如: 当前页面打开、新页面打开等, 同 HTML 属性 target 含义相同"}, {attribute:"theme",description:"String", type:"default", default:"组件风格, 依次为默认色、品牌色、危险色、警告色、成功色。可选项: default/primary/danger/warning/success"}, {attribute:"underline",description:"Boolean", type:"-","default:"普通状态是否显示链接下划线"}, {attribute:"onClick",description:"Function", type:"","default:"TS 类型: (e: MouseEvent) => void 点击事件, 禁用状态不会触发点击事件"}]}], {docTitle:"Link Events说明", tableRows:[{attribute:"click",description:"(e: MouseEvent)", type:"点击事件, 禁用状态不会触发点击事件"}]}], {componentName:"divider", apiData:[{docTitle:"Divider Props说明", tableRows:[{attribute:"align",description:"String", type:"center", default:"文本位置 (仅在水平分割线有效)。可选项: left/right/center"}, {attribute:"content",description:"String / Slot / Function", type:"-","default:"子元素。TS 类型: string | TNode。通用类型定义"}, {attribute:"dashed",description:"Boolean", type:"false", default:"是否虚线 (仅在水平分割线有效)"}, {attribute:"default",description:"String / Slot / Function", type:"-","default:"子元素, 同 content。TS 类型: string | TNode。通用类型定义"}, {attribute:"layout",description:"String", type:"horizontal", default:"分隔线类型有两种: 水平和垂直。可选项: horizontal/vertical"}, {attribute:"theme",description:"String", type:"-","default:"已废弃。请更为

使用 layout。分隔线类型有两种：水平和垂直。可选项：horizontal/vertical"]]]],
 {componentName:"grid",apiData:[{docTitle:"Row Props说明",tableRows:
 [{attribute:"align",description:"String",type:"top",default:"纵向对齐方式，CSS 属性 align-items 值。
 其中 top 和 start 等效；middle 和 center 等效；bottom 和 end 等效。可选项：
 start/end/center/stretch/baseline/top/middle/bottom"}],{attribute:"gutter",description:"Number /
 Object / Array",type:"0",default:"栅格间隔，示例：{ xs: 8, sm: 16, md: 24}。当数据类型为 Number 和
 Object 时，用于指定横向间隔。当数据类型为数组时，第一个参数为横向间隔，第二个参数为纵向间隔， [水平
 间隔，垂直间隔]。TS 类型：number | GutterObject | Array<GutterObject | number> interface
 GutterObject { xs: number; sm: number; md: number }。详细类型定义"},
 {attribute:"justify",description:"String",type:"start",default:"flex 布局下的水平排列方式。可选项：
 start/end/center/space-around/space-between"}],
 {attribute:"tag",description:"String",type:"div",default:"自定义元素标签"}]]],{docTitle:"Col Props说明",
 tableRows:[{attribute:"flex",description:"String / Number",type:"-","default":"flex 布局填充。CSS
 属性 flex 值。示例：2 / 3 / '100px' / 'auto' / '1 1 200px'"},{attribute:"lg",description:"Number /
 Object",type:"-","default":"≥1200px 响应式栅格，可为栅格数或一个包含其他属性的对象（小尺寸电脑）。TS
 类型：number | BaseColProps"},{attribute:"md",description:"Number / Object",type:"-","
 default":"≥992px 响应式栅格，可为栅格数或一个包含其他属性的对象（超小尺寸电脑）。TS 类型：number |
 BaseColProps"},{attribute:"offset",description:"Number",type:"0",default:"栅格左侧的间隔格数，间隔内
 不可以有栅格"},{attribute:"order",description:"Number",type:"0",default:"栅格顺序，flex 布局模式下有
 效"},{attribute:"pull",description:"Number",type:"0",default:"栅格向左移动格数"},
 {attribute:"push",description:"Number",type:"0",default:"栅格向右移动格数"},
 {attribute:"sm",description:"Number / Object",type:"-","default":"≥768px 响应式栅格，可为栅格数或一个
 包含其他属性的对象（平板）。TS 类型：number | BaseColProps"},
 {attribute:"span",description:"Number",type:"-","default:"栅格占位格数，为 0 时相当于 display:
 none"},{attribute:"tag",description:"String",type:"div",default:"自定义元素标签"},
 {attribute:"xl",description:"Number / Object",type:"-","default":"≥1400px 响应式栅格，可为栅格数或一
 个包含其他属性的对象（中尺寸电脑）。TS 类型：number | BaseColProps"},
 {attribute:"xs",description:"Number / Object",type:"-","default":"<768px 响应式栅格，可为栅格数或一个
 包含其他属性的对象（手机）。TS 类型：number | BaseColProps interface BaseColProps { offset: number;
 order: number; pull: number; push: number; span: number }。详细类型定义"},
 {attribute:"xxl",description:"Number / Object",type:"-","default":"≥1880px 响应式栅格，可为栅格数或一
 个包含其他属性的对象（大尺寸电脑）。TS 类型：number | BaseColProps"}]]]],
 {componentName:"layout",apiData:[{docTitle:"Layout Props说明",tableRows:
 [{attribute:"direction",description:"String",type:"-","default:"【开发中】布局方向。可选项：
 vertical/horizontal"}]]],{docTitle:"Header Props说明",tableRows:
 [{attribute:"height",description:"String",type:"-","default:"顶栏高度。样式表（class）中定义的默认高
 度为：64px"}]]],{docTitle:"Aside Props说明",tableRows:
 [{attribute:"width",description:"String",type:"-","default:"侧边栏宽度。样式表（class）中定义的默认宽
 度为：232px"}]]],{docTitle:"Footer Props说明",tableRows:
 [{attribute:"height",description:"String",type:"-","default:"底栏高度。样式表（class）中定义的默认高
 度为：24px"}]]]],{componentName:"space",apiData:[{docTitle:"Space Props说明",tableRows:
 [{attribute:"align",description:"String",type:"-","default:"对齐方式。可选项：
 start/end/center/baseline"},{attribute:"breakLine",description:"Boolean",type:"false",default:"是否
 自动换行，仅在 horizontal 时有效"},
 {attribute:"direction",description:"String",type:"horizontal",default:"间距方向。可选项：
 vertical/horizontal"},{attribute:"separator",description:"String / Slot / Function",type:"-
 ","default:"分隔符。TS 类型：string | TNode。通用类型定义"},{attribute:"size",description:"String /
 Number / Array",type:"medium",default:"间距大小。TS 类型：SpaceSize | SpaceSize[] type SpaceSize =
 number | string | SizeEnum。通用类型定义。详细类型定义"}]]]],{componentName:"affix",apiData:
 [{docTitle:"Affix Props说明",tableRows:[{attribute:"container",description:"String /
 Function",type:"() => () => window",default:"指定滚动的容器。数据类型为 String 时，会被当作选择器
 处理，进行节点查询。示例：'body' 或 () => document.body。TS 类型：ScrollContainer"},
 {attribute:"content（暂未实现）",description:"String / Slot / Function",type:"-","default:"内容。TS
 类型：string | TNode。通用类型定义"},{attribute:"default",description:"String / Slot /
 Function",type:"-","default:"内容，同 content。TS 类型：string | TNode。通用类型定义"},
 {attribute:"offsetBottom",description:"Number",type:"0",default:"距离容器顶部达到指定距离后触发固
 定"},{attribute:"offsetTop",description:"Number",type:"0",default:"距离容器底部达到指定距离后触发固
 定"},{attribute:"zIndex",description:"Number",type:"-","default:"固钉定位层级，样式默认为 500"},
 {attribute:"onFixedChange",description:"Function",type:"","default:"TS 类型：(affixed: boolean,
 context: { top: number }) => void固定状态发生变化时触发"}]]],{docTitle:"Affix Events说明",tableRows:
 [{attribute:"fixed-change",description:"(affixed: boolean, context: { top: number })",type:"固定状态
 发生变化时触发"}]]]],{componentName:"anchor",apiData:[{docTitle:"Anchor Props说明",tableRows:
 [{attribute:"affixProps",description:"Object",type:"-","default:"透传 Affix 组件属性，即让 Anchor 组
 件支持所有 Affix 组件特性。TS 类型：AffixProps, Affix API Documents。详细类型定义"},
 {attribute:"bounds",description:"Number",type:"5",default:"锚点区域边界"},
 {attribute:"container",description:"String / Function",type:"() => () => window",default:"指定滚动
 的容器。数据类型为 String 时，会被当作选择器处理，进行节点查询。示例：'body' 或 () =>
 document.body。TS 类型：ScrollContainer"},{attribute:"cursor",description:"Slot / Function",type:"-
 ","default:"用于自定义选中项左侧游标。TS 类型：TNode。通用类型定义"},

```

{attribute:"size",description:"String",type:"medium",default:"组件尺寸， small(120px)，
medium(200px)， large(320px)。可选项： small/medium/large。 TS 类型： SizeEnum。通用类型定义"},
{attribute:"targetOffset",description:"Number",type:"0",default:"锚点滚动偏移量"},
{attribute:"onChange",description:"Function",type:"",default:"TS 类型： (currentLink: string,
prevLink: string) => void锚点改变时触发"},
{attribute:"onClick",description:"Function",type:"",default:"TS 类型： (link: { href: string; title:
string; e: MouseEvent }) => void锚点被点击时触发"}], {docTitle:"Anchor Events说明",tableRows:
[[{attribute:"change",description:"(currentLink: string, prevLink: string)",type:"锚点改变时触发"},
{attribute:"click",description:"(link: { href: string; title: string; e: MouseEvent })",type:"锚点被
点击时触发"}], {docTitle:"AnchorItem Props说明",tableRows:
[[{attribute:"href",description:"String",type:"-","default:"必需。锚点链接， 如果是 hash 模式需要加上当前
path"}, {attribute:"target",description:"String",type:"_self",default:"锚点文本。可选项：
_self/_blank/_parent/_top"}, {attribute:"title",description:"String / Slot /
Function",type:"""",default:"锚点文本。TS 类型： string | TNode。通用类型定义"}]],
{docTitle:"AnchorTarget Props说明",tableRows:[{attribute:"id",description:"String",type:"-
",default:"必需。目标内容 id"}, {attribute:"tag",description:"String",type:"div",default:"渲染的标
签"}]], {componentName:"back-top",apiData:[{docTitle:"BackTop Props说明",tableRows:
[[{attribute:"container",description:"String / Function",type:"body",default:"监听滚动的容器。数据
类型为 String 时， 会被当作选择器处理， 进行节点查询。示例： 'body' 或 () => document.body。 TS 类型：
AttachNode。通用类型定义"}, {attribute:"content",description:"String / Slot / Function",type:"-
",default:"回到顶部内容。TS 类型： string | TNode。通用类型定义"},
{attribute:"default",description:"String / Slot / Function",type:"-","default:"回到顶部内容， 同
content。 TS 类型： string | TNode。通用类型定义"},
{attribute:"duration",description:"Number",type:"200",default:"回到顶部的耗时单位： 毫秒"},
{attribute:"offset",description:"Array",type:"['24px', '80px']",default:"回到顶部相对右下角的位置偏
移， 示例： [10, 20] 或 ['10em', '8rem']。 TS 类型： Array<string | number>"},
{attribute:"shape",description:"String",type:"square",default:"回到顶部的形状。可选项：
circle/square。 TS 类型： BackTopShapeEnum type BackTopShapeEnum = 'circle' | 'square'。详细类型定
义"}, {attribute:"size",description:"String",type:"medium",default:"组件尺寸。可选项： medium/small"},
{attribute:"target",description:"String / Function",type:"body",default:"指定回到该对象。数据类型
为 String 时， 会被当作选择器处理， 进行节点查询。示例： 'body' 或 () => document.body。 TS 类型：
AttachNode。通用类型定义"}, {attribute:"theme",description:"String",type:"light",default:"组件主题风
格， 浅色、主色、深色。可选项： light/primary/dark"}, {attribute:"visibleHeight",description:"String /
Number",type:"200px",default:"滚动高度达到此参数值才出现"},
{attribute:"onClick",description:"Function",type:"",default:"TS 类型： (context: { e: MouseEvent })
=> void点击回到顶部时触发"}], {docTitle:"BackTop Events说明",tableRows:
[[{attribute:"click",description:"(context: { e: MouseEvent })",type:"点击回到顶部时触发"}]]],
{componentName:"breadcrumb",apiData:[{docTitle:"Breadcrumb Props说明",tableRows:
[[{attribute:"maxItemWidth",description:"String",type:"undefined",default:"单项最大宽度， 超出后会以省
略号形式呈现"}, {attribute:"options",description:"Array",type:"-","default:"面包屑项， 功能同
BreadcrumbItem。 TS 类型： Array<TdBreadcrumbItemProps>"}, {attribute:"separator",description:"String /
Slot / Function",type:"-","default:"自定义分隔符。TS 类型： string | TNode。通用类型定义"},
{attribute:"theme",description:"String",type:"light",default:"组件风格。可选项： light"}]],
{docTitle:"BreadcrumbItem Props说明",tableRows:[{attribute:"content",description:"String / Slot /
Function",type:"-","default:"子元素。TS 类型： string | TNode。通用类型定义"},
{attribute:"default",description:"String / Slot / Function",type:"-","default:"子元素， 同 content。 TS
类型： string | TNode。通用类型定义"}, {attribute:"disabled",description:"Boolean",type:"-
",default:"是否禁用当前项点击"}, {attribute:"href",description:"String",type:"-","default:"跳转链接"},
{attribute:"icon",description:"Slot / Function",type:"-","default:"面板屑项内的前置图标。 TS 类型：
TNode。通用类型定义"}, {attribute:"maxWidth",description:"String",type:"undefined",default:"最大宽
度， 超出后会以省略号形式呈现。优先级高于 Breadcrumb 中的 maxItemWidth"},
{attribute:"replace",description:"Boolean",type:"false",default:"路由跳转是否采用覆盖的方式（覆盖后
将没有浏览器历史记录）"}, {attribute:"router",description:"Object",type:"-","default:"路由对象。如果项
目存在 Router， 则默认使用 Router。 TS 类型： any"},
{attribute:"target",description:"String",type:"_self",default:"链接或路由跳转方式。可选项：
_blank/_self/_parent/_top"}, {attribute:"to",description:"String / Object",type:"-","default:"路由跳转
目标， 当且仅当 Router 存在时， 该 API 有效。 TS 类型： string | Route interface Route { path?: string;
name?: string; hash?: string; query?: RouteData; params?: RouteData } type RouteData = { [key:
string]: string | string[] }。详细类型定义"}]]], {componentName:"dropdown",apiData:
[[{docTitle:"Dropdown Props说明",tableRows:
[[{attribute:"direction",description:"String",type:"right",default:"多层级操作时， 子层级展开方向。可
选项： left/right"}, {attribute:"disabled",description:"Boolean",type:"false",default:"是否禁用组件"},
{attribute:"hideAfterItemClick",description:"Boolean",type:"true",default:"点击选项后是否自动隐藏弹
窗"}, {attribute:"maxColumnWidth",description:"String / Number",type:"100",default:"选项最大宽度， 内
容超出时， 显示为省略号。值为字符串时， 值就是最大宽度； 值为数字时， 单位： px"},
{attribute:"maxHeight",description:"Number",type:"300",default:"弹窗最大高度， 单位： px 。统一控制每
一列的高度"}, {attribute:"minColumnWidth",description:"String / Number",type:"10",default:"选项最小宽
度。值为字符串时， 值就是最小宽度； 值为数字时， 单位： px"},
{attribute:"options",description:"Array",type:"[]",default:"下拉操作项。 TS 类型：

```



```

Array<DropdownOption> type DropdownOption = { children?: Array<TdDropdownItemProps> } &
TdDropdownItemProps & Record<string, any>。详细类型定义",
{attribute:"placement",description:"String",type:"bottom-left",default:"弹窗定位方式，可选值同 Popup
组件。可选项：top/left/right/bottom/top-left/top-right/bottom-left/bottom-right/left-top/left-
bottom/right-top/right-bottom"},{attribute:"popupProps",description:"Object",type:"-","default:"透传
Popup 组件属性，方便更加自由地控制。比如使用 popupProps.overlayStyle 设置浮层样式。TS 类型：
PopupProps, Popup API Documents。详细类型定义"},
{attribute:"trigger",description:"String",type:"hover",default:"触发下拉显示的方式。可选项：
hover/click/focus/context-menu"},{attribute:"onClick",description:"Function",type:"",default:"TS 类
型：(dropdownItem: DropdownOption, context: { e: MouseEvent }) => void下拉操作项点击时触发"}]],
{docTitle:"Dropdown Events说明",tableRows:[{attribute:"click",description:"(dropdownItem:
DropdownOption, context: { e: MouseEvent })",type:"下拉操作项点击时触发"}]],{docTitle:"DropdownItem
Props说明",tableRows:[{attribute:"active",description:"Boolean",type:"false",default:"是否高亮当前操
作项"},{attribute:"content",description:"String / Slot / Function",type:"""",default:"下拉操作项内
容。TS 类型：string | TNode。通用类型定义"},
{attribute:"disabled",description:"Boolean",type:"false",default:"是否禁用操作项"},
{attribute:"divider",description:"Boolean",type:"false",default:"是否显示操作项之间的分隔线（分隔线
默认在下方）"},{attribute:"prefixIcon",description:"Slot / Function",type:"-","default:"组件前置图
标。TS 类型：TNode。通用类型定义"},
{attribute:"theme",description:"String",type:"default",default:"下拉菜单选项主题。可选项：
default/success/warning/error。TS 类型：DropdownItemTheme type DropdownItemTheme = 'default' |
'success' | 'warning' | 'error'。详细类型定义"},{attribute:"value",description:"String / Number /
Object",type:"-","default:"下拉操作项唯一标识。TS 类型：string | number | { [key: string]: any }"},
{attribute:"onClick",description:"Function",type:"",default:"TS 类型：(dropdownItem: DropdownOption,
context: { e: MouseEvent }) => void点击时触发"}]],{docTitle:"DropdownItem Events说明",tableRows:
[{{attribute:"click",description:"(dropdownItem: DropdownOption, context: { e: MouseEvent
})",type:"点击时触发"}]]}],{componentName:"menu",apiData:[{docTitle:"Menu Props说明",tableRows:
[{{attribute:"collapsed",description:"Boolean",type:"false",default:"是否收起菜单"},
{attribute:"expandMutex",description:"Boolean",type:"false",default:"同级别互斥展开"},
{attribute:"expandType",description:"String",type:"normal",default:"二级菜单展开方式，平铺展开和浮层
展开。可选项：normal/popup"},{attribute:"expanded",description:"Array",type:"[]",default:"子菜单展开
的导航集合。支持语法糖 v-model:expanded。TS 类型：Array<MenuValue>"},
{attribute:"defaultExpanded",description:"Array",type:"[]",default:"子菜单展开的导航集合。非受控属
性。TS 类型：Array<MenuValue>"},{attribute:"logo",description:"Slot / Function",type:"-","default:"站
点 LOGO。TS 类型：TNode。通用类型定义"},{attribute:"operations",description:"Slot /
Function",type:"-","default:"导航操作区域。TS 类型：TNode。通用类型定义"},
{attribute:"theme",description:"String",type:"light",default:"菜单风格，有亮色模式和暗色模式两种。当
theme = global 时，模式随整个组件库；当 theme = system 时，模式跟随系统。⚠ global/system 正在开发
中，暂勿使用。可选项：light/dark/global/system"},{attribute:"value",description:"String /
Number",type:"-","default:"激活菜单项。支持语法糖 v-model 或 v-model:value。TS 类型：MenuValue type
MenuValue = string | number。详细类型定义"},{attribute:"defaultValue",description:"String /
Number",type:"-","default:"激活菜单项。非受控属性。TS 类型：MenuValue type MenuValue = string |
number。详细类型定义"},{attribute:"width",description:"String / Number /
Array",type:"""232px""",default:"菜单宽度。值类型为数组时，分别表示菜单展开和折叠的宽度。[ 展开时的宽
度, 折叠时的宽度 ], 示例：['200px', '80px']。TS 类型：string | number | Array<string | number>"},
{attribute:"onChange",description:"Function",type:"",default:"TS 类型：(value: MenuValue) => void激
活菜单项发生变化时触发"},{attribute:"onExpand",description:"Function",type:"",default:"TS 类型：
(value: Array<MenuValue>) => void展开的菜单项发生变化时触发"}]],{docTitle:"Menu Events说
明",tableRows:[{attribute:"change",description:"(value: MenuValue)",type:"激活菜单项发生变化时触
发"},{attribute:"expand",description:"(value: Array<MenuValue>)",type:"展开的菜单项发生变化时触
发"}]],{docTitle:"HeadMenu Props说明",tableRows:
[{{attribute:"expandType",description:"String",type:"normal",default:"二级菜单展开方式，平铺展开和浮
层展开。可选项：normal/popup"},{attribute:"expanded",description:"Array",type:"[]",default:"展开的子
菜单集合。支持语法糖 v-model:expanded。TS 类型：Array<MenuValue>"},
{attribute:"defaultExpanded",description:"Array",type:"[]",default:"展开的子菜单集合。非受控属性。TS
类型：Array<MenuValue>"},{attribute:"logo",description:"Slot / Function",type:"-","default:"站点
LOGO。TS 类型：TNode。通用类型定义"},{attribute:"operations",description:"Slot / Function",type:"-
",default:"导航操作区域。TS 类型：TNode。通用类型定义"},
{attribute:"theme",description:"String",type:"light",default:"可选项：light/dark"},
{attribute:"value",description:"String / Number",type:"-","default:"激活菜单项。支持语法糖 v-model 或
v-model:value。TS 类型：MenuValue type MenuValue = string | number。详细类型定义"},
{attribute:"defaultValue",description:"String / Number",type:"-","default:"激活菜单项。非受控属性。TS
类型：MenuValue type MenuValue = string | number。详细类型定义"},
{attribute:"onChange",description:"Function",type:"",default:"TS 类型：(value: MenuValue) => void激
活菜单项发生变化时触发"},{attribute:"onExpand",description:"Function",type:"",default:"TS 类型：
(value: Array<MenuValue>) => void展开的菜单项发生变化时触发"}]],{docTitle:"HeadMenu Events说
明",tableRows:[{attribute:"change",description:"(value: MenuValue)",type:"激活菜单项发生变化时触
发"},{attribute:"expand",description:"(value: Array<MenuValue>)",type:"展开的菜单项发生变化时触
发"}]],{docTitle:"Submenu Props说明",tableRows:[{attribute:"content",description:"String / Slot /

```

Function", type: "-", default: "菜单项内容。TS 类型: string | TNode。通用类型定义"},
 {attribute: "default", description: "String / Slot / Function", type: "-", default: "菜单项内容, 同 content。TS 类型: string | TNode。通用类型定义"}, {attribute: "disabled", description: "Boolean", type: "-", default: "是否禁用菜单项展开/收起/跳转等功能"}, {attribute: "icon", description: "Slot / Function", type: "-", default: "菜单项图标。TS 类型: TNode。通用类型定义"},
 {attribute: "title", description: "String / Slot / Function", type: "-", default: "二级菜单内容。TS 类型: string | TNode。通用类型定义"}, {attribute: "value", description: "String / Number", type: "-", default: "菜单项唯一标识。TS 类型: MenuValue"}], {docTitle: "MenuItem Props说明", tableRows: [{attribute: "content", description: "String / Slot / Function", type: "-", default: "菜单项内容。TS 类型: string | TNode。通用类型定义"}, {attribute: "default", description: "String / Slot / Function", type: "-", default: "菜单项内容, 同 content。TS 类型: string | TNode。通用类型定义"}, {attribute: "disabled", description: "Boolean", type: "-", default: "是否禁用菜单项展开/收起/跳转等功能"}, {attribute: "href", description: "String", type: "-", default: "跳转链接"}, {attribute: "icon", description: "Slot / Function", type: "-", default: "图标。TS 类型: TNode。通用类型定义"}, {attribute: "replace", description: "Boolean", type: "false", default: "路由跳转是否采用覆盖的方式 (覆盖后将没有浏览器历史记录)"}, {attribute: "router", description: "Object", type: "-", default: "路由对象。如果项目存在 Router, 则默认使用 Router。TS 类型: Record<string, any>"}, {attribute: "target", description: "String", type: "-", default: "链接或路由跳转方式。可选项: _blank/_self/_parent/_top"}, {attribute: "to", description: "String / Object", type: "-", default: "路由跳转目标, 当且仅当 Router 存在时, 该 API 有效。TS 类型: MenuRoute interface MenuRoute { path?: string; name?: string; hash?: string; query?: MenuQueryData; params?: MenuQueryData } type MenuQueryData = { [key: string]: string | string[] }。详细类型定义"}, {attribute: "value", description: "String / Number", type: "-", default: "菜单项唯一标识。TS 类型: MenuValue"}, {attribute: "onClick", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent }) => void 点击时触发"}], {docTitle: "MenuItem Events说明", tableRows: [{attribute: "click", description: "(context: { e: MouseEvent })", type: "点击时触发"}], {docTitle: "MenuGroup Props说明", tableRows: [{attribute: "title", description: "String / Slot / Function", type: "-", default: "菜单组标题。TS 类型: string | TNode。通用类型定义"}], {componentName: "pagination", apiData: [{docTitle: "Pagination Props说明", tableRows: [{attribute: "current", description: "Number", type: "1", default: "当前页。支持语法糖 v-model 或 v-model:current"}, {attribute: "defaultCurrent", description: "Number", type: "1", default: "当前页。非受控属性"}, {attribute: "disabled", description: "Boolean", type: "-", default: "是否禁用分页组件"}, {attribute: "foldedMaxPageBtn", description: "Number", type: "5", default: "折叠时最多显示页码按钮数"}, {attribute: "maxPageBtn", description: "Number", type: "10", default: "最多显示页码按钮数"}, {attribute: "pageEllipsisMode", description: "String", type: "mid", default: "页码数量超出时, 前后省略模式, mid表示中间省略, both-ends 表示两端省略。可选项: mid/both-ends"}, {attribute: "pageSize", description: "Number", type: "10", default: "每一页的数据量。支持语法糖 v-model:pageSize"}, {attribute: "defaultPageSize", description: "Number", type: "10", default: "每一页的数据量。非受控属性"}, {attribute: "pageSizeOptions", description: "Array", type: "[5, 10, 20, 50]", default: "分页大小控制器, 值为 [] 则不显示。TS 类型: Array<number | { label: string; value: number }>"}, {attribute: "selectProps", description: "Object", type: "-", default: "透传全部属性到 Select 组件, 也可使用 selectProps.popupProps 透传全部 Popup 组件。TS 类型: SelectProps, Select API Documents。详细类型定义"}, {attribute: "showFirstAndLastPageBtn", description: "Boolean", type: "false", default: "是否显示跳转首页尾页页码控制器"}, {attribute: "showJumper", description: "Boolean", type: "false", default: "是否显示跳转页码控制器"}, {attribute: "showPageNumber", description: "Boolean", type: "true", default: "是否显示页码控制器"}, {attribute: "showPageSize", description: "Boolean", type: "true", default: "是否显示分页数量控制器"}, {attribute: "showPreviousAndNextBtn", description: "Boolean", type: "true", default: "是否显示跳转前后页页码控制器"}, {attribute: "size", description: "String", type: "medium", default: "分页组件尺寸。可选项: small/medium"}, {attribute: "theme", description: "String", type: "default", default: "分页组件风格。可选项: default/simple"}, {attribute: "total", description: "Number", type: "0", default: "数据总条数"}, {attribute: "totalContent", description: "Boolean / Slot / Function", type: "true", default: "用于自定义总条数呈现内容。默认显示总条数, 值为 false 则不显示。TS 类型: boolean | TNode。通用类型定义"}, {attribute: "onChange", description: "Function", type: "", default: "TS 类型: (pageInfo: PageInfo) => void 当前页或分页大小发生变化时触发。详细类型定义。interface PageInfo { current: number; previous: number; pageSize: number }"}, {attribute: "onCurrentChange", description: "Function", type: "", default: "TS 类型: (current: number, pageInfo: PageInfo) => void 当前页发生变化时触发"}, {attribute: "onPageSizeChange", description: "Function", type: "", default: "TS 类型: (pageSize: number, pageInfo: PageInfo) => void 分页大小发生变化时触发"}], {docTitle: "Pagination Events说明", tableRows: [{attribute: "change", description: "(pageInfo: PageInfo)", type: "当前页或分页大小发生变化时触发。详细类型定义。interface PageInfo { current: number; previous: number; pageSize: number }"}, {attribute: "current-change", description: "(current: number, pageInfo: PageInfo)", type: "当前页发生变化时触发"}, {attribute: "page-size-change", description: "(pageSize: number, pageInfo: PageInfo)", type: "分页大小发生变化时触发"}], {docTitle: "PaginationMini Props说明", tableRows: [{attribute: "disabled", description: "Boolean / Object", type: "-", default: "按钮禁用配置。TS 类型: boolean | JumperDisabledConfig type JumperDisabledConfig = { prev?: boolean; current?: boolean; next?: boolean }。详细类型定义"}, {attribute: "layout", description: "String", type: "horizontal", default: "按钮方向。可选项: horizontal/vertical"}, {attribute: "showCurrent", description: "Boolean", type: "true", default: "是否展示当前按钮。TS 类型: boolean"}, {attribute: "size", description: "String", type: "medium", default: "按钮尺寸。可选项: small/medium/large。TS 类型: SizeEnum。通用类型定义"},

```

{attribute:"tips",description:"Object",type:"-","default":"提示文案配置，值为 true 显示默认文案；值为 false 不显示提示文案；值类型为对象则单独配置文案内容。TS 类型：boolean | JumperTipsConfig type JumperTipsConfig = { prev?: string; current?: string; next?: string; }。详细类型定义"},
{attribute:"variant",description:"String",type:"text",default:"按钮形式。可选项：text/outline"},
{attribute:"onChange",description:"Function",type:"",default:"TS 类型：(context: {e: MouseEvent, trigger: JumperTrigger}) => void按钮点击事件回调。详细类型定义。type JumperTrigger = 'prev' | 'current' | 'next'"}], {docTitle:"PaginationMini Events说明",tableRows:
[[{attribute:"change",description:"(context: {e: MouseEvent, trigger: JumperTrigger})",type:"按钮点击事件回调。详细类型定义。type JumperTrigger = 'prev' | 'current' | 'next'"}]]],
{componentName:"steps",apiData:[{docTitle:"Steps Props说明",tableRows:
[[{attribute:"current",description:"String / Number",type:"-","default":"当前步骤，即整个步骤条进度。默认根据步骤下标判断步骤的完成状态，当前步骤为进行中，当前步骤之前的步骤为已完成，当前步骤之后的步骤为未开始。如果每个步骤没有设置 value，current 值为步骤长度则表示所有步骤已完成。如果每个步骤设置了自定义 value，则 current = 'FINISH' 表示所有状态完成。支持语法糖 v-model 或 v-model:current"},
{attribute:"defaultCurrent",description:"String / Number",type:"-","default":"当前步骤，即整个步骤条进度。默认根据步骤下标判断步骤的完成状态，当前步骤为进行中，当前步骤之前的步骤为已完成，当前步骤之后的步骤为未开始。如果每个步骤没有设置 value，current 值为步骤长度则表示所有步骤已完成。如果每个步骤设置了自定义 value，则 current = 'FINISH' 表示所有状态完成。非受控属性"},
{attribute:"layout",description:"String",type:"horizontal",default:"步骤条方向，有两种：横向和纵向。可选项：horizontal/vertical"},
{attribute:"options",description:"Array",type:"-","default":"步骤条数据列表（作用和 StepItem 效果一样）。TS 类型：Array<TdStepItemProps>"},
{attribute:"readonly",description:"Boolean",type:"false",default:"只读状态"},
{attribute:"separator",description:"String",type:"line",default:"步骤条分割符。可选项：line/dashed/arrow"},
{attribute:"sequence",description:"String",type:"positive",default:"步骤条顺序。可选项：positive/reverse"},
{attribute:"theme",description:"String",type:"default",default:"步骤条风格。可选项：default/dot"},
{attribute:"onChange",description:"Function",type:"",default:"TS 类型：(current: string | number, previous: string | number, context?: { e?: MouseEvent }) => void当前步骤发生变化时触发"}]], {docTitle:"Steps Events说明",tableRows:[{attribute:"change",description:"(current: string | number, previous: string | number, context?: { e?: MouseEvent })",type:"当前步骤发生变化时触发"}]], {docTitle:"StepItem Props说明",tableRows:
[[{attribute:"content",description:"String / Slot / Function",type:"","",default:"步骤描述。TS 类型：string | TNode。通用类型定义"},
{attribute:"default",description:"String / Slot / Function",type:"-","default":"步骤描述，同 content。TS 类型：string | TNode。通用类型定义"},
{attribute:"extra",description:"String / Slot / Function",type:"-","default":"显示在步骤描述下方的额外内容，比如：操作项。TS 类型：string | TNode。通用类型定义"},
{attribute:"icon",description:"Boolean / Slot / Function",type:"true",default:"图标，默认显示内置图标，也可以自定义图标，值为 false 则不显示图标。优先级大于 status 定义的图标。TS 类型：boolean | TNode。通用类型定义"},
{attribute:"status",description:"String",type:"default",default:"当前步骤的状态：默认状态（未开始）、进行中状态、完成状态、错误状态。可选项：default/process/finish/error。TS 类型：StepStatus type StepStatus = 'default' | 'process' | 'finish' | 'error'。详细类型定义"},
{attribute:"title",description:"String / Slot / Function",type:"","",default:"标题。TS 类型：string | TNode。通用类型定义"},
{attribute:"value",description:"String / Number",type:"-","default":"当前步骤标识"}]]], {componentName:"sticky-tool",apiData:[{docTitle:"StickyTool Props说明",tableRows:
[[{attribute:"list",description:"Array",type:"[]",default:"列表。TS 类型：Array<TdStickyItemProps>"},
{attribute:"offset",description:"Array",type:"-","default":"相对于 placement 的偏移量，示例：[-10, 20] 或 ['10em', '8rem']。TS 类型：Array<string | number>"},
{attribute:"placement",description:"String",type:"right-bottom",default:"固定位置。可选项：right-top/right-center/right-bottom/left-top/left-center/left-bottom"},
{attribute:"popupProps",description:"Object",type:"-","default":"透传 Popup 组件全部特性，优先级低于 StickyItem.popupProps。TS 类型：PopupProps, Popup API Documents。详细类型定义"},
{attribute:"shape",description:"String",type:"square",default:"侧边栏菜单形状，有 2 种：方形、圆形。可选项：square/round"},
{attribute:"type",description:"String",type:"normal",default:"侧边栏菜单类型，有 2 种：常规型和紧凑型。可选项：normal/compact"},
{attribute:"width",description:"String / Number",type:"-","default":"宽度"},
{attribute:"onClick",description:"Function",type:"",default:"TS 类型：(context: { e: MouseEvent; item: TdStickyItemProps }) => void点击某一项时触发"},
{attribute:"onHover",description:"Function",type:"",default:"TS 类型：(context: { e: MouseEvent; item: TdStickyItemProps }) => void悬浮到某一项时触发"}]], {docTitle:"StickyTool Events说明",tableRows:[{attribute:"click",description:"(context: { e: MouseEvent; item: TdStickyItemProps })",type:"点击某一项时触发"},
{attribute:"hover",description:"(context: { e: MouseEvent; item: TdStickyItemProps })",type:"悬浮到某一项时触发"}]], {docTitle:"StickyItem Props说明",tableRows:
[[{attribute:"icon",description:"Slot / Function",type:"-","default":"图标。TS 类型：TNode。通用类型定义"},
{attribute:"label",description:"String / Slot / Function",type:"-","default":"名称。TS 类型：string | TNode。通用类型定义"},
{attribute:"popup",description:"String / Slot / Function",type:"-","default":"浮层内容。TS 类型：string | TNode。通用类型定义"},
{attribute:"popupProps",description:"Object",type:"-","default":"透传浮层组件全部属性。TS 类型：PopupProps, Popup API Documents。详细类型定义"},
{attribute:"trigger",description:"String",type:"hover",default:"触发浮层显示的方式。可选项：hover/click"}]]], {componentName:"tabs",apiData:[{docTitle:"Tabs Props说明",tableRows:
[[{attribute:"action",description:"String / Slot / Function",type:"-","default":"【开发中】选项卡右侧的

```


操作区域。TS 类型: string | TNode。通用类型定义"},
 {attribute:"addable",description:"Boolean",type:"false",default:"选项卡是否可增加"},
 {attribute:"disabled",description:"Boolean",type:"false",default:"是否禁用选项卡"},
 {attribute:"dragSort",description:"Boolean",type:"false",default:"是否开启拖拽调整顺序"},
 {attribute:"list",description:"Array",type:"-",default:"选项卡列表。TS 类型: Array<TdTabPanelProps>"}, {attribute:"placement",description:"String",type:"top",default:"选项卡位置。可选项: left/top/bottom/right"}, {attribute:"size",description:"String",type:"medium",default:"组件尺寸。可选项: medium/large"}, {attribute:"theme",description:"String",type:"normal",default:"选项卡风格, 包含 默认风格 和 卡片风格两种。可选项: normal/card"}, {attribute:"value",description:"String / Number",type:"-",default:"激活的选项卡值。支持语法糖 v-model 或 v-model:value。TS 类型: TabValue type TabValue = string | number。详细类型定义"}, {attribute:"defaultValue",description:"String / Number",type:"-",default:"激活的选项卡值。非受控属性。TS 类型: TabValue type TabValue = string | number。详细类型定义"}, {attribute:"onAdd",description:"Function",type:"",default:"TS 类型: (context: { e: MouseEvent }) => void添加选项卡时触发"},
 {attribute:"onChange",description:"Function",type:"",default:"TS 类型: (value: TabValue, label: string) => void激活的选项卡发生变化时触发"},
 {attribute:"onDragSort",description:"Function",type:"",default:"TS 类型: (context: TabsDragSortContext) => void拖拽排序时触发。详细类型定义。interface TabsDragSortContext { currentIndex: number; current: TabValue; targetIndex: number; target: TabValue }"},
 {attribute:"onRemove",description:"Function",type:"",default:"TS 类型: (options: { value: TabValue; index: number; e: MouseEvent }) => void删除选项卡时触发"}], {docTitle:"Tabs Events说明",tableRows: [{attribute:"add",description:"(context: { e: MouseEvent })",type:"添加选项卡时触发"}, {attribute:"change",description:"(value: TabValue, label: string)",type:"激活的选项卡发生变化时触发"}, {attribute:"drag-sort",description:"(context: TabsDragSortContext)",type:"拖拽排序时触发。详细类型定义。interface TabsDragSortContext { currentIndex: number; current: TabValue; targetIndex: number; target: TabValue }"}, {attribute:"remove",description:"(options: { value: TabValue; index: number; e: MouseEvent })",type:"删除选项卡时触发"}], {docTitle:"TabPanel Props说明",tableRows: [{attribute:"default",description:"Slot / Function",type:"-",default:"用于自定义选项卡导航, 同 panel。TS 类型: TNode。通用类型定义"}, {attribute:"destroyOnHide",description:"Boolean",type:"true",default:"选项卡内容隐藏时是否销毁"}, {attribute:"disabled",description:"Boolean",type:"false",default:"是否禁用当前选项卡"}, {attribute:"draggable",description:"Boolean",type:"true",default:"选项卡组件开启允许拖动排序时, 当前选项卡是否允许拖动"}, {attribute:"label",description:"String / Slot / Function",type:"-",default:"选项卡名称, 可自定义选项卡导航内容。TS 类型: string | TNode。通用类型定义"}, {attribute:"panel",description:"String / Slot / Function",type:"-",default:"用于自定义选项卡面板内容。TS 类型: string | TNode。通用类型定义"}, {attribute:"removable",description:"Boolean",type:"false",default:"当前选项卡是否允许移除"}, {attribute:"value",description:"String / Number",type:"-",default:"选项卡的值, 唯一标识。TS 类型: TabValue, Tabs API Documents。详细类型定义"}, {attribute:"onRemove",description:"Function",type:"",default:"TS 类型: (options: { value: TabValue; e: MouseEvent }) => void点击删除按钮时触发"}], {docTitle:"TabPanel Events说明",tableRows: [{attribute:"remove",description:"(options: { value: TabValue; e: MouseEvent })",type:"点击删除按钮时触发"}], {componentName:"auto-complete",apiData:[{docTitle:"AutoComplete Props说明",tableRows: [{attribute:"autofocus",description:"Boolean",type:"-",default:"自动获取焦点"}, {attribute:"clearable",description:"Boolean",type:"-",default:"是否允许清空"}, {attribute:"default",description:"String / Slot / Function",type:"-",default:"触发显示联想词下拉框的元素, 同 triggerElement。TS 类型: string | TNode。通用类型定义"}, {attribute:"disabled",description:"Boolean",type:"-",default:"是否禁用"}, {attribute:"filter",description:"Function",type:"-",default:"自定义过滤规则, 用于对现有数据进行搜索过滤, 判断是否过滤某一项数据。参数 filterWords 表示搜索词, option表示单个选项内容, 返回值为 true 保留该选项, 返回值为 false 则隐藏该选项。使用该方法时无需设置 filterable。TS 类型: (filterWords: string, option: T) => boolean | Promise<boolean>"}, {attribute:"filterable",description:"Boolean",type:"true",default:"是否根据输入内容过滤联想词。默认过滤规则不区分大小写, 全文本任意位置匹配。如果默认搜索规则不符合业务需求, 可以更为使用 filter 自定义过滤规则。部分场景下输入关键词和下拉联想词完全不同, 此时可以设置为 false"}, {attribute:"highlightKeyword",description:"Boolean",type:"true",default:"是否高亮联想词中和输入值的相同部分"}, {attribute:"inputProps",description:"Object",type:"-",default:"透传 Input 组件全部特性。TS 类型: InputProps, Input API Documents。详细类型定义"}, {attribute:"options",description:"Array",type:"-",default:"下拉联想词列表。示例一: ['联想词一', '联想词二']。示例二: { label: () => <div>联想词元素</div>, text: '用于搜索的纯联想词' }。TS 类型: Array<T>"}, {attribute:"panelBottomContent",description:"String / Slot / Function",type:"-",default:"面板内的底部内容。TS 类型: string | TNode。通用类型定义"}, {attribute:"panelTopContent",description:"String / Slot / Function",type:"-",default:"面板内的顶部内容。TS 类型: string | TNode。通用类型定义"}, {attribute:"placeholder",description:"String",type:"undefined",default:"输入框为空时的占位提示。组件本身默认值为 undefined, 但全局配置存在默认值, 不同语言全局默认值不同"}, {attribute:"popupProps",description:"Object",type:"-",default:"透传 Popup 组件全部特性。TS 类型: PopupProps, Popup API Documents。详细类型定义"}, {attribute:"readonly",description:"Boolean",type:"-",default:"是否只读"}, {attribute:"size",description:"String",type:"medium",default:"组件尺寸。可选

项: small/medium/large。TS 类型: SizeEnum。通用类型定义"},
 {attribute:"status",description:"String",type:"default",default:"输入框状态。可选项: default/success/warning/error"}, {attribute:"textareaProps",description:"Object",type:"-",default:"透传 Textarea 组件全部特性。TS 类型: TextareaProps, Textarea API Documents。详细类型定义"},
 {attribute:"tips",description:"String / Slot / Function",type:"-",default:"输入框下方提示文本, 会根据不同的 status 呈现不同的样式。TS 类型: string | TNode。通用类型定义"},
 {attribute:"triggerElement",description:"String / Slot / Function",type:"-",default:"触发显示联想词下拉框的元素, 默认为 Input 组件, 可以使用 trigger 自定义为 Textarea 组件或其他组件。TS 类型: string | TNode。通用类型定义"}, {attribute:"value",description:"String",type:"-",default:"输入框的值, 即当前指定的联想词。支持语法糖 v-model 或 v-model:value"},
 {attribute:"defaultValue",description:"String",type:"-",default:"输入框的值, 即当前指定的联想词。非受控属性"}, {attribute:"onBlur",description:"Function",type:"",default:"TS 类型: (context: { e: FocusEvent; value: string }) => void失去焦点时触发"},
 {attribute:"onChange",description:"Function",type:"",default:"TS 类型: (value: string, context?: { e?: InputEvent | MouseEvent | CompositionEvent | KeyboardEvent }) => void输入框值发生变化时触发"},
 {attribute:"onClear",description:"Function",type:"",default:"TS 类型: (context: { e: MouseEvent }) => void清空按钮点击时触发"}, {attribute:"onCompositionend",description:"Function",type:"",default:"TS 类型: (context: { e: CompositionEvent; value: string }) => void中文输入结束时触发"},
 {attribute:"onCompositionstart",description:"Function",type:"",default:"TS 类型: (context: { e: CompositionEvent; value: string }) => void中文输入开始时触发"},
 {attribute:"onEnter",description:"Function",type:"",default:"TS 类型: (context: { e: KeyboardEvent; value: string }) => void回车键按下时触发"},
 {attribute:"onFocus",description:"Function",type:"",default:"TS 类型: (context: { e: FocusEvent; value: string }) => void获得焦点时触发"},
 {attribute:"onSelect",description:"Function",type:"",default:"TS 类型: (value: string, context: { e: MouseEvent | KeyboardEvent }) => void选中联想词时触发"}], {docTitle:"AutoComplete Events说明",tableRows:[{attribute:"blur",description:"(context: { e: FocusEvent; value: string })",type:"失去焦点时触发"}, {attribute:"change",description:"(value: string, context?: { e?: InputEvent | MouseEvent | CompositionEvent | KeyboardEvent })",type:"输入框值发生变化时触发"},
 {attribute:"clear",description:"(context: { e: MouseEvent })",type:"清空按钮点击时触发"},
 {attribute:"compositionend",description:"(context: { e: CompositionEvent; value: string })",type:"中文输入结束时触发"}, {attribute:"compositionstart",description:"(context: { e: CompositionEvent; value: string })",type:"中文输入开始时触发"}, {attribute:"enter",description:"(context: { e: KeyboardEvent; value: string })",type:"回车键按下时触发"}, {attribute:"focus",description:"(context: { e: FocusEvent; value: string })",type:"获得焦点时触发"}, {attribute:"select",description:"(value: string, context: { e: MouseEvent | KeyboardEvent })",type:"选中联想词时触发"}]]},
 {componentName:"cascader",apiData:[{docTitle:"Cascader Props说明",tableRows:[{attribute:"checkProps",description:"Object",type:"-",default:"参考 checkbox 组件 API。TS 类型: CheckboxProps, Checkbox API Documents。详细类型定义"},
 {attribute:"checkStrictly",description:"Boolean",type:"false",default:"父子节点选中状态不再关联, 可各自选中或取消"}, {attribute:"clearable",description:"Boolean",type:"false",default:"是否支持清空选项"}, {attribute:"collapsedItems",description:"Slot / Function",type:"-",default:"多选情况下, 用于设置折叠项内容, 默认为 +N。如果需要悬浮就显示其他内容, 可以使用 collapsedItems 自定义。TS 类型: TNode< { value: CascaderOption[]; collapsedSelectedItems: CascaderOption[]; count: number }>。通用类型定义"}, {attribute:"disabled",description:"Boolean",type:"false",default:"是否禁用组件"},
 {attribute:"empty",description:"String / Slot / Function",type:"-",default:"无匹配选项时的内容, 默认全局配置为 '暂无数据'。TS 类型: string | TNode。通用类型定义"},
 {attribute:"filter",description:"Function",type:"-",default:"自定义过滤方法, 用于对现有数据进行搜索过滤, 判断是否过滤某一项数据。TS 类型: (filterWords: string, node: TreeNodeModel) => boolean | Promise<boolean>"}, {attribute:"filterable",description:"Boolean",type:"false",default:"是否可搜索"},
 {attribute:"inputProps",description:"Object",type:"-",default:"透传 Input 输入框组件的全部属性。TS 类型: InputProps, Input API Documents。详细类型定义"}, {attribute:"keys",description:"Object",type:"-",default:"用来定义 value / label / children 在 options 中对应的字段别名。TS 类型: CascaderKeysType interface CascaderKeysType { value?: string; label?: string; children?: string | boolean }。通用类型定义。详细类型定义"}, {attribute:"lazy",description:"Boolean",type:"true",default:"延迟加载 children 为 true 的子节点, 即使 expandAll 被设置为 true, 也同样延迟加载"},
 {attribute:"load",description:"Function",type:"-",default:"加载子树数据的方法 (仅当节点 children 为 true 时生效)。TS 类型: (node: TreeNodeModel<CascaderOption>) => Promise<Array<CascaderOption>>"},
 {attribute:"loading",description:"Boolean",type:"false",default:"是否为加载状态"},
 {attribute:"loadingText",description:"String / Slot / Function",type:"-",default:"远程加载时显示的文字, 支持自定义。如加上超链接。TS 类型: string | TNode。通用类型定义"},
 {attribute:"max",description:"Number",type:"0",default:"用于控制多选数量, 值为 0 则不限制"},
 {attribute:"minCollapsedNum",description:"Number",type:"0",default:"最小折叠数量, 用于多选情况下折叠选中项, 超出该数值的选中项折叠。值为 0 则表示不折叠"},
 {attribute:"multiple",description:"Boolean",type:"false",default:"是否允许多选"},
 {attribute:"options",description:"Array",type:"[]",default:"可选项数据源。TS 类型: Array<CascaderOption>"}, {attribute:"placeholder",description:"String",type:"undefined",default:"占位符"}, {attribute:"popupProps",description:"Object",type:"-",default:"参考 popup 组件 API。TS 类型: PopupProps, Popup API Documents。详细类型定义"},

```

{attribute:"popupVisible",description:"Boolean",type:"-",default:"是否显示下拉框"},
{attribute:"readonly",description:"Boolean",type:"false",default:"只读状态, 值为真会隐藏输入框, 且无法打开下拉框"}, {attribute:"selectInputProps",description:"Object",type:"-",default:"透传 SelectInput 筛选器输入框组件的全部属性。TS 类型: SelectInputProps, SelectInput API Documents。详细类型定义"},
{attribute:"showAllLevels",description:"Boolean",type:"true",default:"选中值使用完整路径, 输入框在单选时也显示完整路径"}, {attribute:"size",description:"String",type:"medium",default:"组件尺寸。可选项: large/medium/small。TS 类型: SizeEnum。通用类型定义"},
{attribute:"status",description:"String",type:"-",default:"输入框状态。可选项: default/success/warning/error"}, {attribute:"tagInputProps",description:"Object",type:"-",default:"透传 TagInput 标签输入框组件的全部属性。TS 类型: TagInputProps, TagInput API Documents。详细类型定义"}, {attribute:"tagProps",description:"Object",type:"-",default:"透传 Tag 标签组件全部属性。TS 类型: TagProps, Tag API Documents。详细类型定义"}, {attribute:"tips",description:"String / Slot / Function",type:"-",default:"输入框下方提示文本, 会根据不同的 status 呈现不同的样式。TS 类型: string | TNode。通用类型定义"}, {attribute:"trigger",description:"String",type:"click",default:"展开下一层级的方式。可选项: click/hover"}, {attribute:"value",description:"String / Number / Array",type:"[]",default:"选中项的值。支持语法糖 v-model 或 v-model:value。TS 类型: CascaderValue<CascaderOption> type CascaderValue<T extends TreeOptionData = TreeOptionData> = string | number | T | Array<CascaderValue<T>>。通用类型定义。详细类型定义"}, {attribute:"defaultValue",description:"String / Number / Array",type:"[]",default:"选中项的值。非受控属性。TS 类型: CascaderValue<CascaderOption> type CascaderValue<T extends TreeOptionData = TreeOptionData> = string | number | T | Array<CascaderValue<T>>。通用类型定义。详细类型定义"},
{attribute:"valueMode",description:"String",type:"onlyLeaf",default:"选中值模式。all 表示父节点和子节点全部会出现在选中值里面; parentFirst 表示当子节点全部选中时, 仅父节点在选中值里面; onlyLeaf 表示无论什么情况, 选中值仅呈现叶子节点。可选项: onlyLeaf/parentFirst/all"},
{attribute:"valueType",description:"String",type:"single",default:"用于控制选中值的类型。single 表示输入输出值为 叶子结点值, full 表示输入输出值为全路径。可选项: single/full"},
{attribute:"onBlur",description:"Function",type:"",default:"TS 类型: (context: { value: CascaderValue<CascaderOption>; e: FocusEvent }) => void当输入框失去焦点时触发"},
{attribute:"onChange",description:"Function",type:"",default:"TS 类型: (value: CascaderValue<CascaderOption>, context: CascaderChangeContext<CascaderOption>) => void选中值发生变化时触发。TreeNodeModel 从树组件中导出。context.node 表示触发事件的节点, context.source 表示触发事件的来源。详细类型定义。interface CascaderChangeContext<CascaderOption> { node?: TreeNodeModel<CascaderOption>; source: CascaderChangeSource }import { TreeNodeModel } from '@Tree' type CascaderChangeSource = 'invalid-value' | 'check' | 'clear' | 'uncheck'"},
{attribute:"onFocus",description:"Function",type:"",default:"TS 类型: (context: { value: CascaderValue<CascaderOption>; e: FocusEvent }) => void获得焦点时触发"},
{attribute:"onPopupVisibleChange",description:"Function",type:"",default:"TS 类型: (visible: boolean, context: PopupVisibleChangeContext) => void下拉框显示或隐藏时触发。详细类型定义。import { PopupVisibleChangeContext } from '@Popup'"},
{attribute:"onRemove",description:"Function",type:"",default:"TS 类型: (context: RemoveContext<CascaderOption>) => void多选模式下, 选中数据被移除时触发。详细类型定义。interface RemoveContext<T> { value: CascaderValue<T>; node: TreeNodeModel<T> }"}], {docTitle:"Cascader Events 说明",tableRows:[{attribute:"blur",description:"(context: { value: CascaderValue<CascaderOption>; e: FocusEvent })",type:"当输入框失去焦点时触发"}, {attribute:"change",description:"(value: CascaderValue<CascaderOption>, context: CascaderChangeContext<CascaderOption>)",type:"选中值发生变化时触发。TreeNodeModel 从树组件中导出。context.node 表示触发事件的节点, context.source 表示触发事件的来源。详细类型定义。interface CascaderChangeContext<CascaderOption> { node?: TreeNodeModel<CascaderOption>; source: CascaderChangeSource }import { TreeNodeModel } from '@Tree' type CascaderChangeSource = 'invalid-value' | 'check' | 'clear' | 'uncheck'"}, {attribute:"focus",description:"(context: { value: CascaderValue<CascaderOption>; e: FocusEvent })",type:"获得焦点时触发"}, {attribute:"popup-visible-change",description:"(visible: boolean, context: PopupVisibleChangeContext)",type:"下拉框显示或隐藏时触发。详细类型定义。import { PopupVisibleChangeContext } from '@Popup'"}, {attribute:"remove",description:"(context: RemoveContext<CascaderOption>)",type:"多选模式下, 选中数据被移除时触发。详细类型定义。interface RemoveContext<T> { value: CascaderValue<T>; node: TreeNodeModel<T> }"}]]}], {componentName:"checkbox",apiData:[{docTitle:"Checkbox Props说明",tableRows:[{attribute:"checkAll",description:"Boolean",type:"false",default:"用于标识是否为「全选选项」。单独使用无效, 需在 CheckboxGroup 中使用"}, {attribute:"checked",description:"Boolean",type:"false",default:"是否选中。支持语法糖 v-model 或 v-model:checked"}, {attribute:"defaultChecked",description:"Boolean",type:"false",default:"是否选中。非受控属性"}, {attribute:"default",description:"String / Slot / Function",type:"-",default:"多选框内容, 同 label。TS 类型: string | TNode。通用类型定义"}, {attribute:"disabled",description:"Boolean",type:"undefined",default:"是否禁用组件"}, {attribute:"indeterminate",description:"Boolean",type:"false",default:"是否为半选"}, {attribute:"label",description:"String / Slot / Function",type:"-",default:"主文案。TS 类型: string | TNode。通用类型定义"}, {attribute:"name",description:"String",type:"-",default:"HTML 元素原生属性"}, {attribute:"readonly",description:"Boolean",type:"false",default:"只读状态"}, {attribute:"value",description:"String / Number",type:"-",default:"多选框的值。TS 类型: string | number"}, {attribute:"onChange",description:"Function",type:"",default:"TS 类型: (checked: boolean,

```

```

context: { e: Event }) => void值变化时触发"}]], {docTitle:"Checkbox Events说明", tableRows:
[[{attribute:"change",description:"(checked: boolean, context: { e: Event })",type:"值变化时触发"}]],
{docTitle:"CheckboxGroup Props说明", tableRows:[{attribute:"disabled",description:"Boolean",type:"-
",default:"是否禁用组件"}, {attribute:"max",description:"Number",type:"undefined",default:"支持最多选
中的数量"}, {attribute:"name",description:"String",type:"-",default:"统一设置内部复选框 HTML 属性"},
{attribute:"options",description:"Array",type:"[]",default:"以配置形式设置子元素。示例1: ['北京',
'上海'], 示例2: [{ label: '全选', checkAll: true }, { label: '上海', value: 'shanghai' } ]。checkAll
值为 true 表示当前选项为「全选选项」。TS 类型: Array<CheckboxOption> type CheckboxOption = string |
number | CheckboxOptionObj interface CheckboxOptionObj { label?: string | TNode; value?: string |
number; disabled?: boolean; name?: string; checkAll?: true }。通用类型定义。详细类型定义"},
{attribute:"value",description:"Array",type:"[]",default:"选中值。支持语法糖 v-model 或 v-
model:value。TS 类型: CheckboxGroupValue type CheckboxGroupValue = Array<string | number>。详细类型
定义"}, {attribute:"defaultValue",description:"Array",type:"[]",default:"选中值。非受控属性。TS 类
型: CheckboxGroupValue type CheckboxGroupValue = Array<string | number>。详细类型定义"},
{attribute:"onChange",description:"Function",type:"",default:"TS 类型: (value: CheckboxGroupValue,
context: CheckboxGroupChangeContext) => void值变化时触发。context.current 表示当前变化的数据项, 如果
是全选则为空; context.type 表示引起选中数据变化的是选中或是取消选中, context.option 表示当前变化的数
据项。详细类型定义。interface CheckboxGroupChangeContext { e: Event; current: string | number;
option: CheckboxOption | TdCheckboxProps; type: 'check' | 'unchecked' }"}]], {docTitle:"CheckboxGroup
Events说明", tableRows:[{attribute:"change",description:"(value: CheckboxGroupValue, context:
CheckboxGroupChangeContext)",type:"值变化时触发。context.current 表示当前变化的数据项, 如果是全选则
为空; context.type 表示引起选中数据变化的是选中或是取消选中, context.option 表示当前变化的数据项。详
细类型定义。interface CheckboxGroupChangeContext { e: Event; current: string | number; option:
CheckboxOption | TdCheckboxProps; type: 'check' | 'unchecked' }"}]]], {componentName:"color-
picker", apiData:[{docTitle:"ColorPicker Props说明", tableRows:
[[{attribute:"clearable",description:"Boolean",type:"false",default:"是否可清空"},
{attribute:"closeBtn",description:"String / Boolean / Slot / Function",type:"true",default:"关闭按
钮, 值为 true 显示默认关闭按钮; 值为 false 或 undefined 则不显示关闭按钮; 值类型为函数, 则表示自定义
关闭按钮。TS 类型: string | boolean | TNode。通用类型定义"},
{attribute:"colorModes",description:"Array",type:"()=> ['monochrome', 'linear-
gradient']",default:"颜色模式选择。同时支持单色和渐变两种模式, 可仅使用单色或者渐变其中一种模式, 也
可以同时使用。monochrome 表示单色, linear-gradient 表示渐变色。TS 类型: Array<'monochrome' |
'linear-gradient'>"}, {attribute:"disabled",description:"Boolean",type:"-","default:"是否禁用组件"},
{attribute:"enableAlpha",description:"Boolean",type:"false",default:"是否开启透明通道"},
{attribute:"format",description:"String",type:"RGB",default:"格式化色值。enableAlpha 为真时,
RGBA/HSIA/HSV 等值有效。可选项: RGB/RGBA/HSL/HSIA/HSB/HSV/HSV/HSV/HEX/CMYK/CSS"},
{attribute:"inputProps",description:"Object",type:"-","default:"透传 Input 输入框组件全部属性。TS 类
型: InputProps, Input API Documents。详细类型定义"},
{attribute:"multiple",description:"Boolean",type:"false",default:"【开发中】是否允许选中多个颜色"},
{attribute:"popupProps",description:"Object",type:"-","default:"透传 Popup 组件全部属性, 如 placement
overlayStyle overlayClassName trigger等。TS 类型: PopupProps, Popup API Documents。详细类型定义"},
{attribute:"recentColors",description:"Array",type:"[]",default:"最近使用的颜色。值为 [] 表示以组件
内部的“最近使用颜色”为准, 值长度大于 0 则以该值为准显示“最近使用颜色”。值为 null 则完全不显示
“最近使用颜色”。支持语法糖 v-model:recentColors。TS 类型: boolean | Array<string>"},
{attribute:"defaultRecentColors",description:"Array",type:"[]",default:"最近使用的颜色。值为 [] 表示
以组件内部的“最近使用颜色”为准, 值长度大于 0 则以该值为准显示“最近使用颜色”。值为 null 则完全不
显示“最近使用颜色”。非受控属性。TS 类型: boolean | Array<string>"},
{attribute:"selectInputProps",description:"Object",type:"-","default:"透传 SelectInputProps 筛选器输
入框组件全部属性。TS 类型: SelectInputProps, SelectInput API Documents。详细类型定义"},
{attribute:"showPrimaryColorPreview",description:"Boolean",type:"true",default:"是否展示颜色选择条右
侧的颜色预览区域"}, {attribute:"swatchColors",description:"Array",type:"-","default:"系统预设的颜色样
例, 值为 null 或 [] 则不显示系统色, 值为 undefined 会显示组件内置的系统默认色。TS 类型:
Array<string>"}, {attribute:"value",description:"String",type:"-","default:"色值。支持语法糖 v-model
或 v-model:value"}, {attribute:"defaultValue",description:"String",type:"-","default:"色值。非受控属
性"}, {attribute:"onChange",description:"Function",type:"",default:"TS 类型: (value: string, context:
{ color: ColorObject; trigger: ColorPickerChangeTrigger }) => void选中的色值发生变化时触发, 第一个参
数 value 表示新色值, context.color 表示当前调色板控制器的色值, context.trigger 表示触发颜色变化的来
源。详细类型定义。type ColorPickerChangeTrigger = 'palette-saturation-brightness' | 'palette-
saturation' | 'palette-brightness' | 'palette-hue-bar' | 'palette-alpha-bar' | 'input' | 'preset' |
'recent'"}, {attribute:"onPaletteBarChange",description:"Function",type:"",default:"TS 类型:
(context: { color: ColorObject }) => void调色板控制器的值变化时触发, context.color 指调色板控制器的
值。详细类型定义。interface ColorObject { alpha: number; css: string; hex: string; hex8: string;
hsl: string; hsla: string; hsv: string; hsva: string; rgb: string; rgba: string; saturation: number;
value: number; isGradient: boolean; linearGradient?: string; }"},
{attribute:"onRecentColorsChange",description:"Function",type:"",default:"TS 类型: (value:
Array<string>) => void最近使用颜色发生变化时触发"}]], {docTitle:"ColorPicker Events说明", tableRows:
[[{attribute:"change",description:"(value: string, context: { color: ColorObject; trigger:
ColorPickerChangeTrigger })",type:"选中的色值发生变化时触发, 第一个参数 value 表示新色值,
context.color 表示当前调色板控制器的色值, context.trigger 表示触发颜色变化的来源。详细类型定义。type

```

ColorPickerChangeTrigger = 'palette-saturation-brightness' | 'palette-saturation' | 'palette-brightness' | 'palette-hue-bar' | 'palette-alpha-bar' | 'input' | 'preset' | 'recent'"}, {attribute: "palette-bar-change", description: "(context: { color: ColorObject })", type: "调色板控制器的值变化时触发, context.color 指调色板控制器的值。详细类型定义。interface ColorObject { alpha: number; css: string; hex: string; hex8: string; hsl: string; hsla: string; hsv: string; hsva: string; rgb: string; rgba: string; saturation: number; value: number; isGradient: boolean; linearGradient?: string; }"}, {attribute: "recent-colors-change", description: "(value: Array<string>)", type: "最近使用颜色发生变化时触发"}]]], {componentName: "date-picker", apiData: [{docTitle: "DatePicker Props说明", tableRows: [{attribute: "allowInput", description: "Boolean", type: "false", default: "是否允许输入日期"}, {attribute: "clearable", description: "Boolean", type: "false", default: "是否显示清除按钮"}, {attribute: "defaultTime", description: "String", type: "'00:00:00'", default: "时间选择器默认值, 当value/defaultValue 未设置值时有效"}, {attribute: "disableDate", description: "Object / Array / Function", type: "-", default: "禁用日期, 示例: ['A', 'B'] 表示日期 A 和日期 B 会被禁用。{ from: 'A', to: 'B' } 表示在 A 到 B 之间的日期会被禁用。{ before: 'A', after: 'B' } 表示在 A 之前和在 B 之后的日期都会被禁用。其中 A = '2021-01-01', B = '2021-02-01'。值类型为 Function 则表示返回值为 true 的日期会被禁用。TS 类型: DisableDate type DisableDate = Array<DateValue> | DisableDateObj | ((date: DateValue) => boolean) interface DisableDateObj { from?: string; to?: string; before?: string; after?: string }。详细类型定义"}, {attribute: "disabled", description: "Boolean", type: "-", default: "是否禁用组件"}, {attribute: "enableTimePicker", description: "Boolean", type: "false", default: "是否显示时间选择"}, {attribute: "firstDayOfWeek", description: "Number", type: "7", default: "第一天从星期几开始。可选项: 1/2/3/4/5/6/7"}, {attribute: "format", description: "String", type: "'YYYY-MM-DD'", default: "仅用于格式化日期显示的格式, 不影响日期值。注意和 valueType 的区别, valueType会直接决定日期值 value 的格式。全局配置默认为: 'YYYY-MM-DD', 详细文档"}, {attribute: "inputProps", description: "Object", type: "-", default: "透传给输入框 (Input) 组件的参数。TS 类型: InputProps, Input API Documents。详细类型定义"}, {attribute: "mode", description: "String", type: "date", default: "选择器模式。可选项: year/quarter/month/week/date"}, {attribute: "placeholder", description: "String / Array", type: "undefined", default: "占位符。TS 类型: string"}, {attribute: "popupProps", description: "Object", type: "-", default: "透传给 popup 组件的参数。TS 类型: PopupProps, Popup API Documents。详细类型定义"}, {attribute: "prefixIcon", description: "Slot / Function", type: "-", default: "用于自定义组件前置图标。TS 类型: TNode。通用类型定义"}, {attribute: "presets", description: "Object", type: "-", default: "预设快捷日期选择, 示例: { '元旦': '2021-01-01', '昨天': dayjs().subtract(1, 'day').format('YYYY-MM-DD') }, '特定日期': () => ['2021-02-01'] }。TS 类型: PresetDate interface PresetDate { [name: string]: DateValue | (() => DateValue) }。详细类型定义"}, {attribute: "presetsPlacement", description: "String", type: "bottom", default: "预设面板展示区域 (包含确定按钮)。可选项: left/top/right/bottom"}, {attribute: "size", description: "String", type: "medium", default: "输入框尺寸。可选项: small/medium/large。TS 类型: SizeEnum。通用类型定义"}, {attribute: "status", description: "String", type: "default", default: "输入框状态。可选项: default/success/warning/error"}, {attribute: "suffixIcon", description: "Slot / Function", type: "-", default: "用于自定义组件后置图标。TS 类型: TNode。通用类型定义"}, {attribute: "timePickerProps", description: "Object", type: "-", default: "透传 TimePicker 组件属性。TS 类型: TimePickerProps, TimePicker API Documents。详细类型定义"}, {attribute: "tips", description: "String / Slot / Function", type: "-", default: "输入框下方提示文本, 会根据不同的 status 呈现不同的样式。TS 类型: string | TNode。通用类型定义"}, {attribute: "value", description: "String / Number / Array / Date", type: "'", default: "选中值。支持语法糖 v-model 或 v-model:value。TS 类型: DateValue type DateValue = string | number | Date。详细类型定义"}, {attribute: "defaultValue", description: "String / Number / Array / Date", type: "'", default: "选中值。非受控属性。TS 类型: DateValue type DateValue = string | number | Date。详细类型定义"}, {attribute: "valueType", description: "String", type: "-", default: "用于格式化日期的值, 仅支持部分格式, 时间戳、日期等。⚠️ YYYYMMDD 这种格式不支持, 请勿使用, 如果希望支持可以给 dayjs 提个 PR。注意和 format 的区别, format 仅用于处理日期在页面中呈现的格式。ValueTypeEnum 即将废弃, 请更为使用 DatePickerValueType。TS 类型: DatePickerValueType type DatePickerValueType = 'time-stamp' | 'Date' | 'YYYY' | 'YYYY-MM' | 'YYYY-MM-DD' | 'YYYY-MM-DD HH' | 'YYYY-MM-DD HH:mm' | 'YYYY-MM-DD HH:mm:ss' | 'YYYY-MM-DD HH:mm:ss:SSS' type ValueTypeEnum = DatePickerValueType。详细类型定义"}, {attribute: "onBlur", description: "Function", type: "", default: "TS 类型: (context: { value: DateValue; e: FocusEvent }) => void当输入框失去焦点时触发"}, {attribute: "onChange", description: "Function", type: "", default: "TS 类型: (value: DateValue, context: { dayjsValue?: Dayjs, trigger?: DatePickerTriggerSource }) => void选中值发生变化时触发。详细类型定义。import { Dayjs } from 'dayjs' type DatePickerTriggerSource = 'confirm' | 'pick' | 'enter' | 'preset' | 'clear'"}, {attribute: "onFocus", description: "Function", type: "", default: "TS 类型: (context: { value: DateValue; e: FocusEvent }) => void输入框获得焦点时触发"}, {attribute: "onPick", description: "Function", type: "", default: "TS 类型: (value: DateValue) => void面板选中值后触发"}]]], {docTitle: "DatePicker Events说明", tableRows: [{attribute: "blur", description: "(context: { value: DateValue; e: FocusEvent })", type: "当输入框失去焦点时触发"}, {attribute: "change", description: "(value: DateValue, context: { dayjsValue?: Dayjs, trigger?: DatePickerTriggerSource })", type: "选中值发生变化时触发。详细类型定义。import { Dayjs } from 'dayjs' type DatePickerTriggerSource = 'confirm' | 'pick' | 'enter' | 'preset' | 'clear'"}, {attribute: "focus", description: "(context: { value: DateValue; e: FocusEvent })", type: "输入框获得焦点时触发"}, {attribute: "pick", description: "(value: DateValue)", type: "面板选中值后触发"}]]], {docTitle: "DateRangePicker Props说明", tableRows:


```
[{attribute:"allowInput",description:"Boolean",type:"false",default:"是否允许输入日期"},
{attribute:"clearable",description:"Boolean",type:"false",default:"是否显示清楚按钮"},
{attribute:"defaultTime",description:"Array",type:"['00:00:00', '23:59:59']",default:"时间选择器默认值, 当 value/defaultValue 未设置值时有效。TS 类型: string[]"},
{attribute:"disableDate",description:"Object / Array / Function",type:"-",default:"禁用日期, 示例: ['A', 'B'] 表示日期 A 和日期 B 会被禁用。{ from: 'A', to: 'B' } 表示在 A 到 B 之间的日期会被禁用。{ before: 'A', after: 'B' } 表示在 A 之前和在 B 之后的日期都会被禁用。其中 A = '2021-01-01', B = '2021-02-01'。值类型为 Function 则表示返回值为 true 的日期会被禁用。TS 类型: DisableRangeDate type DisableRangeDate = Array<DateValue> | DisableDateObj | ((context: { date: DateRangeValue; partial: DateRangePickerPartial }) => boolean) interface DisableDateObj { from?: string; to?: string; before?: string; after?: string } type DateRangePickerPartial = 'start' | 'end'。详细类型定义"},
{attribute:"disabled",description:"Boolean",type:"-",default:"是否禁用组件"},
{attribute:"enableTimePicker",description:"Boolean",type:"false",default:"是否显示时间选择"},
{attribute:"firstDayOfWeek",description:"Number",type:"-",default:"第一天从星期几开始。可选项: 1/2/3/4/5/6/7"},
{attribute:"format",description:"String",type:"-",default:"用于格式化日期, 详细文档"},
{attribute:"mode",description:"String",type:"date",default:"选择器模式。可选项: year/quarter/month/week/date"},
{attribute:"panelPreselection",description:"Boolean",type:"true",default:"在开始日期选中之前, 面板是否显示预选状态, 即是否高亮预选日期"},
{attribute:"placeholder",description:"String / Array",type:"-",default:"占位符, 值为数组表示可分别为开始日期和结束日期设置占位符。TS 类型: string | Array<string>"},
{attribute:"popupProps",description:"Object",type:"-",default:"透传给 popup 组件的参数。TS 类型: PopupProps, Popup API Documents。详细类型定义"},
{attribute:"prefixIcon",description:"Slot / Function",type:"-",default:"组件前置图标。TS 类型: TNode。通用类型定义"},
{attribute:"presets",description:"Object",type:"-",default:"预设快捷日期选择, 示例: { '特定日期范围': ['2021-01-01', '2022-01-01'], '本月': [dayjs().startOf('month'), dayjs().endOf('month')] }。TS 类型: PresetRange interface PresetRange { [range: string]: DateRange | (() => DateRange) } type DateRange = [DateValue, DateValue]。详细类型定义"},
{attribute:"presetsPlacement",description:"String",type:"bottom",default:"预设面板展示区域 (包含确定按钮)。可选项: left/top/right/bottom"},
{attribute:"rangeInputProps",description:"Object",type:"-",default:"透传给范围输入框 RangeInput 组件的参数。TS 类型: RangeInputProps, RangeInput API Documents。详细类型定义"},
{attribute:"separator",description:"String",type:"-",default:"日期分隔符, 支持全局配置, 默认为 '-'"},
{attribute:"size",description:"String",type:"medium",default:"输入框尺寸。可选项: small/medium/large。TS 类型: SizeEnum。通用类型定义"},
{attribute:"status",description:"String",type:"default",default:"输入框状态。可选项: default/success/warning/error"},
{attribute:"suffixIcon",description:"Slot / Function",type:"-",default:"组件后置图标。TS 类型: TNode。通用类型定义"},
{attribute:"timePickerProps",description:"Object",type:"-",default:"透传 TimePicker 组件属性。TS 类型: TimePickerProps, TimePicker API Documents。详细类型定义"},
{attribute:"tips",description:"String / Slot / Function",type:"-",default:"输入框下方提示文本, 会根据不同的 status 呈现不同的样式。TS 类型: string | TNode。通用类型定义"},
{attribute:"value",description:"Array",type:"[]",default:"选中值。支持语法糖 v-model 或 v-model:value。TS 类型: DateRangeValue type DateRangeValue = Array<DateValue>。详细类型定义"},
{attribute:"defaultValue",description:"Array",type:"[]",default:"选中值。非受控属性。TS 类型: DateRangeValue type DateRangeValue = Array<DateValue>。详细类型定义"},
{attribute:"valueType",description:"String",type:"-",default:"用于格式化日期的值, 仅支持部分格式, 时间戳、日期等。⚠️ YYYYMMDD 这种格式不支持, 请勿使用, 如果希望支持可以给 dayjs 提个 PR。注意和 format 的区别, format 仅用于处理日期在页面中呈现的格式。可选项: 'time-stamp'"},
{attribute:"onBlur",description:"Function",type:"",default:"TS 类型: (context: { value: DateRangeValue; partial: DateRangePickerPartial; e: FocusEvent }) => void当输入框失去焦点时触发"},
{attribute:"onChange",description:"Function",type:"",default:"TS 类型: (value: DateRangeValue, context: { dayjsValue?: Dayjs[], trigger?: DatePickerTriggerSource }) => void选中值发生变化时触发。详细类型定义。import { Dayjs } from 'dayjs'"},
{attribute:"onFocus",description:"Function",type:"",default:"TS 类型: (context: { value: DateRangeValue; partial: DateRangePickerPartial; e: FocusEvent }) => void输入框获得焦点时触发"},
{attribute:"onInput",description:"Function",type:"",default:"TS 类型: (context: { input: string; value: DateRangeValue; partial: DateRangePickerPartial; e: InputEvent }) => void输入框数据发生变化时触发, 参数 input 表示输入内容, value 表示组件当前有效值"},
{attribute:"onPick",description:"Function",type:"",default:"TS 类型: (value: DateValue, context: PickContext) => void选中日期时触发, 可能是开始日期, 也可能是结束日期, 第二个参数可以区分是开始日期或是结束日期。详细类型定义。interface PickContext { e: MouseEvent; partial: DateRangePickerPartial }"}], {docTitle:"DateRangePicker Events说明",tableRows:[{attribute:"blur",description:"(context: { value: DateRangeValue; partial: DateRangePickerPartial; e: FocusEvent })",type:"当输入框失去焦点时触发"},
{attribute:"change",description:"(value: DateRangeValue, context: { dayjsValue?: Dayjs[], trigger?: DatePickerTriggerSource })",type:"选中值发生变化时触发。详细类型定义。import { Dayjs } from 'dayjs'"},
{attribute:"focus",description:"(context: { value: DateRangeValue; partial: DateRangePickerPartial; e: FocusEvent })",type:"输入框获得焦点时触发"},
{attribute:"input",description:"(context: { input: string; value: DateRangeValue; partial: DateRangePickerPartial; e: InputEvent })",type:"输入框数据发生变化时触发, 参数 input 表示输入内容, value 表示组件当前有效值"},
{attribute:"pick",description:"(value: DateValue, context: PickContext)",type:"选中日期时触发, 可能是开始日期, 也可能是结束日期, 第二个参数可以区分是开始日期或
```


是结束日期。详细类型定义。interface PickContext { e: MouseEvent; partial: DateRangePickerPartial } } } }], { docTitle: "DatePickerPanel Props说明", tableRows: [{ attribute: "defaultTime", description: "String", type: "'00:00:00'", default: "时间选择器默认值, 当 value/defaultValue 未设置值时有效", { attribute: "Pick<DatePickerProps, 'value' | 'defaultValue' | 'disableDate' | 'enableTimePicker' | 'firstDayOfWeek' | 'format' | 'mode' | 'presets' | 'presetsPlacement' | 'timePickerProps'>", description: "-", type: "-", default: "继承 Pick<DatePickerProps, 'value' | 'defaultValue' | 'disableDate' | 'enableTimePicker' | 'firstDayOfWeek' | 'format' | 'mode' | 'presets' | 'presetsPlacement' | 'timePickerProps'> 中的全部属性", { attribute: "onCellClick", description: "Function", type: "", default: "TS 类型: (context: { date: Date, e: MouseEvent }) => void 点击日期单元格时触发", { attribute: "onChange", description: "Function", type: "", default: "TS 类型: (value: DateValue, context: { dayjsValue?: Dayjs, e?: MouseEvent, trigger?: DatePickerTriggerSource }) => void 选中值发生变化时触发。参数 context.trigger 表示触发当前事件的来源, 不同的模式触发来源也会不同", { attribute: "onConfirm", description: "Function", type: "", default: "TS 类型: (context: { date: Date, e: MouseEvent }) => void 如果存在“确认”按钮, 则点击“确认”按钮时触发", { attribute: "onMonthChange", description: "Function", type: "", default: "TS 类型: (context: { month: number, date: Date, e?: MouseEvent, trigger: DatePickerMonthChangeTrigger }) => void 月份切换发生变化时触发。详细类型定义。type DatePickerMonthChangeTrigger = 'month-select' | 'month-arrow-next' | 'month-arrow-previous' | 'today'", { attribute: "onPanelClick", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent }) => void 点击面板时触发", { attribute: "onPresetClick", description: "Function", type: "", default: "TS 类型: (context: { preset: PresetDate, e: MouseEvent }) => void 如果存在“确认”按钮, 则点击“确认”按钮时触发", { attribute: "onTimeChange", description: "Function", type: "", default: "TS 类型: (context: { time: string, date: Date, trigger: DatePickerTimeChangeTrigger, e?: MouseEvent }) => void 时间切换发生变化时触发。详细类型定义。type DatePickerTimeChangeTrigger = 'time-hour' | 'time-minute' | 'time-second'", { attribute: "onYearChange", description: "Function", type: "", default: "TS 类型: (context: { year: number, date: Date, trigger: DatePickerYearChangeTrigger, e?: MouseEvent }) => void 年份切换发生变化时触发。详细类型定义。type DatePickerYearChangeTrigger = 'year-select' | 'year-arrow-next' | 'year-arrow-previous' | 'today'"]] }, { docTitle: "DatePickerPanel Events说明", tableRows: [{ attribute: "cell-click", description: "(context: { date: Date, e: MouseEvent })", type: "点击日期单元格时触发", { attribute: "change", description: "(value: DateValue, context: { dayjsValue?: Dayjs, e?: MouseEvent, trigger?: DatePickerTriggerSource })", type: "选中值发生变化时触发。参数 context.trigger 表示触发当前事件的来源, 不同的模式触发来源也会不同", { attribute: "confirm", description: "(context: { date: Date, e: MouseEvent })", type: "如果存在“确认”按钮, 则点击“确认”按钮时触发", { attribute: "month-change", description: "(context: { month: number, date: Date, e?: MouseEvent, trigger: DatePickerMonthChangeTrigger })", type: "月份切换发生变化时触发。详细类型定义。type DatePickerMonthChangeTrigger = 'month-select' | 'month-arrow-next' | 'month-arrow-previous' | 'today'", { attribute: "panel-click", description: "(context: { e: MouseEvent })", type: "点击面板时触发", { attribute: "preset-click", description: "(context: { preset: PresetDate, e: MouseEvent })", type: "如果存在“确认”按钮, 则点击“确认”按钮时触发", { attribute: "time-change", description: "(context: { time: string, date: Date, trigger: DatePickerTimeChangeTrigger, e?: MouseEvent })", type: "时间切换发生变化时触发。详细类型定义。type DatePickerTimeChangeTrigger = 'time-hour' | 'time-minute' | 'time-second'", { attribute: "year-change", description: "(context: { year: number, date: Date, trigger: DatePickerYearChangeTrigger, e?: MouseEvent })", type: "年份切换发生变化时触发。详细类型定义。type DatePickerYearChangeTrigger = 'year-select' | 'year-arrow-next' | 'year-arrow-previous' | 'today'"]] }, { docTitle: "DateRangePickerPanel Props说明", tableRows: [{ attribute: "defaultTime", description: "Array", type: "['00:00:00', '23:59:59']", default: "时间选择器默认值, 当 value/defaultValue 未设置值时有效。TS 类型: string[]", { attribute: "Pick<DateRangePickerProps, 'value' | 'defaultValue' | 'disableDate' | 'enableTimePicker' | 'firstDayOfWeek' | 'format' | 'mode' | 'presets' | 'presetsPlacement' | 'panelPreselection' | 'timePickerProps'>", description: "-", type: "-", default: "继承 Pick<DateRangePickerProps, 'value' | 'defaultValue' | 'disableDate' | 'enableTimePicker' | 'firstDayOfWeek' | 'format' | 'mode' | 'presets' | 'presetsPlacement' | 'panelPreselection' | 'timePickerProps'> 中的全部属性", { attribute: "onCellClick", description: "Function", type: "", default: "TS 类型: (context: { date: Date[], partial: DateRangePickerPartial, e: MouseEvent }) => void 点击日期单元格时触发", { attribute: "onChange", description: "Function", type: "", default: "TS 类型: (value: DateRangeValue, context: { dayjsValue?: Dayjs[], partial: DateRangePickerPartial, e?: MouseEvent, trigger?: DatePickerTriggerSource }) => void 选中值发生变化时触发。参数 context.trigger 表示触发当前事件的来源, 不同的模式触发来源也会不同", { attribute: "onConfirm", description: "Function", type: "", default: "TS 类型: (context: { date: Date[], e: MouseEvent }) => void 如果存在“确认”按钮, 则点击“确认”按钮时触发", { attribute: "onMonthChange", description: "Function", type: "", default: "TS 类型: (context: { month: number, date: Date[], partial: DateRangePickerPartial, e?: MouseEvent, trigger: DatePickerMonthChangeTrigger }) => void 月份切换发生变化时触发", { attribute: "onPanelClick", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent }) => void 点击面板时触发", { attribute: "onPresetClick", description: "Function", type: "", default: "TS 类型: (context: { preset: PresetDate, e: MouseEvent }) => void 如果存在“确认”按钮, 则点击“确认”按钮时触发", { attribute: "onTimeChange", description: "Function", type: "", default: "TS 类型: (context: { time: string, date: Date[], partial: DateRangePickerPartial, trigger: DatePickerTimeChangeTrigger, e?: MouseEvent }) => void 时间切换发生变化时触发",

```

{attribute:"onYearChange",description:"Function",type:"",default:"TS 类型: (context: { year: number,
date: Date[], partial: DateRangePickerPartial, trigger: DatePickerYearChangeTrigger, e?: MouseEvent
}) => void年份切换发生变化时触发"}], {docTitle:"DateRangePickerPanel Events说明",tableRows:
[ {attribute:"cell-click",description:"(context: { date: Date[], partial: DateRangePickerPartial, e:
MouseEvent })",type:"点击日期单元格时触发"}, {attribute:"change",description:"(value: DateRangeValue,
context: { dayjsValue?: Dayjs[], partial: DateRangePickerPartial, e?: MouseEvent, trigger?:
DatePickerTriggerSource })",type:"选中值发生变化时触发。参数 context.trigger 表示触发当前事件的来源,
不同的模式触发来源也会不同"}, {attribute:"confirm",description:"(context: { date: Date[], e:
MouseEvent })",type:"如果存在“确认”按钮,则点击“确认”按钮时触发"}, {attribute:"month-
change",description:"(context: { month: number, date: Date[], partial: DateRangePickerPartial, e?:
MouseEvent, trigger: DatePickerMonthChangeTrigger })",type:"月份切换发生变化时触发"},
{attribute:"panel-click",description:"(context: { e: MouseEvent })",type:"点击面板时触发"},
{attribute:"preset-click",description:"(context: { preset: PresetDate, e: MouseEvent })",type:"如果
存在“确认”按钮,则点击“确认”按钮时触发"}, {attribute:"time-change",description:"(context: { time:
string, date: Date[], partial: DateRangePickerPartial, trigger: DatePickerTimeChangeTrigger, e?:
MouseEvent })",type:"时间切换发生变化时触发"}, {attribute:"year-change",description:"(context: {
year: number, date: Date[], partial: DateRangePickerPartial, trigger: DatePickerYearChangeTrigger,
e?: MouseEvent })",type:"年份切换发生变化时触发"}]]], {componentName:"form",apiData:[{docTitle:"Form
Props说明",tableRows:[{attribute:"colon",description:"Boolean",type:"false",default:"是否在表单标签
字段右侧显示冒号"}, {attribute:"data",description:"Object",type:"{}",default:"表单数据。TS 类型:
FormData"}, {attribute:"disabled",description:"Boolean",type:"undefined",default:"是否禁用整个表单"},
{attribute:"errorMessage",description:"Object",type:"-",default:"表单错误信息配置, 示例: { idcard:
'请输入正确的身份证号码', max: '字符长度不能超过 ${max}' }。TS 类型: FormErrorMessage"},
{attribute:"formControlledComponents",description:"Array",type:"-",default:"允许表单统一控制禁用状态的
自定义组件名称列表。默认会有组件库的全部输入类组件: TInput、TInputNumber、TCascader、TSelect、
TOption、TSwitch、TCheckbox、TCheckboxGroup、TRadio、TRadioGroup、TTreeSelect、TDatePicker、
TTimePicker、TUpload、TTransfer、TSlider。对于自定义组件, 组件内部需要包含可以控制表单禁用状态的变量
formDisabled。示例: ['CustomUpload', 'CustomInput']。TS 类型: Array<string>"},
{attribute:"labelAlign",description:"String",type:"right",default:"表单字段标签对齐方式: 左对齐、右
对齐、顶部对齐。可选项: left/right/top"}, {attribute:"labelWidth",description:"String /
Number",type:"'100px'",default:"可以整体设置label标签宽度, 默认为100px"},
{attribute:"layout",description:"String",type:"vertical",default:"表单布局, 有两种方式: 纵向布局 和
行内布局。可选项: vertical/inline"},
{attribute:"preventSubmitDefault",description:"Boolean",type:"true",default:"是否阻止表单提交默认事件
(表单提交默认事件会刷新页面), 设置为 true 可以避免刷新"},
{attribute:"requiredMark",description:"Boolean",type:"true",default:"是否显示必填符号(*), 默认显示"},
{attribute:"resetType",description:"String",type:"empty",default:"重置表单的方式, 值为 empty 表示
重置表单为空, 值为 initial 表示重置表单数据为初始值。可选项: empty/initial"},
{attribute:"rules",description:"Object",type:"-",default:"表单字段校验规则。TS 类型:
FormRules<FormData> type FormRules<T extends Data> = { [field in keyof T]?: Array<FormRule> }。详细
类型定义"}, {attribute:"scrollToFirstError",description:"String",type:"-",default:"表单校验不通过时,
是否自动滚动到第一个校验不通过的字段, 平滑滚动或是瞬间直达。值为空则表示不滚动。可选项:
''/smooth/auto"}, {attribute:"showErrorMessage",description:"Boolean",type:"true",default:"校验不
通过时, 是否显示错误提示信息, 统一控制全部表单项。如果希望控制单个表单项, 请给 FormItem 设置该属
性"}, {attribute:"statusIcon",description:"Boolean / Slot / Function",type:"undefined",default:"校验
状态图标, 值为 true 显示默认图标, 默认图标有 成功、失败、警告 等, 不同的状态图标不同。statusIcon 值
为 false, 不显示图标。statusIcon 值类型为渲染函数, 则可以自定义右侧状态图标。TS 类型: boolean |
TNode<TdFormItemProps>。通用类型定义"},
{attribute:"submitWithWarningMessage",description:"Boolean",type:"false",default:"【讨论中】当校验结
果只有告警信息时, 是否触发 submit 提交事件"},
{attribute:"onReset",description:"Function",type:"",default:"TS 类型: (context: { e?: FormResetEvent
}) => void表单重置时触发"}, {attribute:"onSubmit",description:"Function",type:"",default:"TS 类型:
(context: SubmitContext<FormData>) => void表单提交时触发。其中 context.validateResult 表示校验结果,
context.firstError 表示校验不通过的第一个规则提醒。context.validateResult 值为 true 表示校验通过; 如
果校验不通过, context.validateResult 值为校验结果列表。【注意】⚠️ 默认情况, 输入框按下 Enter 键会自
动触发提交事件, 如果希望禁用这个默认行为, 可以给输入框添加 enter 事件, 并在事件中设置
e.preventDefault()。详细类型定义。interface SubmitContext<T extends Data = Data> { e?:
FormSubmitEvent; validateResult: FormValidateResult<T>; firstError?: string }type
FormValidateResult<T> = boolean | ValidateResultObj<T>type ValidateResultObj<T> = { [key in keyof
T]: boolean | ValidateResultList }type ValidateResultList = Array<AllValidateResult>type
AllValidateResult = CustomValidateObj | ValidateResultTypeinterface ValidateResultType extends
FormRule { result: boolean }type ValidateResult<T> = { [key in keyof T]: boolean | ErrorList }type
ErrorList = Array<FormRule>"}, {attribute:"onValidate",description:"Function",type:"",default:"TS 类
型: (result: ValidateResultContext<FormData>) => void校验结束后触发, result 值为 true 表示校验通过;
如果校验不通过, result 值为校验结果列表。详细类型定义。type ValidateResultContext<T extends Data> =
Omit<SubmitContext<T>, 'e'>"}]], {docTitle:"Form Events说明",tableRows:
[ {attribute:"reset",description:"(context: { e?: FormResetEvent })",type:"表单重置时触发"},
{attribute:"submit",description:"(context: SubmitContext<FormData>)",type:"表单提交时触发。其中
context.validateResult 表示校验结果, context.firstError 表示校验不通过的第一个规则提醒。"}]]],

```

17/59

入正确的邮箱地址' }。TS 类型: boolean | IsEmailOptions import { IsEmailOptions } from 'validator/es/lib/isEmail'。详细类型定义", {attribute: "enum", description: "Array", type: "-", default: "内置校验方法, 校验值是否属于枚举值中的值。示例: { enum: ['primary', 'info', 'warning'], message: '值只能是 primary/info/warning 中的一种' }。TS 类型: Array<string>"}, {attribute: "idcard", description: "Boolean", type: "-", default: "内置校验方法, 校验值是否为身份证号码, 组件校验正则则为 /^(\\d{18}|\\d{15}|\\d{17}|x)\$/i, 示例: { idcard: true, message: '请输入正确的身份证号码' }"}, {attribute: "len", description: "Number / Boolean", type: "-", default: "内置校验方法, 校验值固定长度, 如: len: 10 表示值的字符长度只能等于 10, 中文表示 2 个字符, 英文为 1 个字符。示例: { len: 10, message: '内容长度不对' }。如果希望字母和中文都是同样的长度, 示例: { validator: (val) => val.length === 10, message: '内容文本长度只能是 10 个字' }"}, {attribute: "max", description: "Number / Boolean", type: "-", default: "内置校验方法, 校验值最大长度, 如: max: 100 表示值最多不能超过 100 个字符, 中文表示 2 个字符, 英文为 1 个字符。示例: { max: 10, message: '内容超出' }。如果希望字母和中文都是同样的长度, 示例: { validator: (val) => val.length <= 10, message: '内容文本长度不能超过 10 个字' }"}, {attribute: "message", description: "String", type: "-", default: "校验未通过时呈现的错误信息, 值为空则不显示"}, {attribute: "min", description: "Number / Boolean", type: "-", default: "内置校验方法, 校验值最小长度, 如: min: 10 表示值最多不能少于 10 个字符, 中文表示 2 个字符, 英文为 1 个字符。示例: { min: 10, message: '内容长度不够' }。如果希望字母和中文都是同样的长度, 示例: { validator: (val) => val.length >= 10, message: '内容文本长度至少为 10 个字' }。如果数据类型数字 (Number), 则自动变为数字大小的比对"}, {attribute: "number", description: "Boolean", type: "-", default: "内置校验方法, 校验值是否为数字 (1.2、1e5 都算数字), 示例: { number: true, message: '请输入数字' }"}, {attribute: "pattern", description: "Object", type: "-", default: "内置校验方法, 校验值是否符合正则表达式匹配结果, 示例: { pattern: /@qq.com/, message: '请输入 QQ 邮箱' }。TS 类型: RegExp"}, {attribute: "required", description: "Boolean", type: "-", default: "内置校验方法, 校验值是否已经填写。该值为 true, 默认显示必填标记, 可通过设置 requiredMark: false 隐藏必填标记"}, {attribute: "telnumber", description: "Boolean", type: "-", default: "内置校验方法, 校验值是否为手机号码, 校验正则则为 /^1[3-9]\\d{9}\$/, 示例: { telnumber: true, message: '请输入正确的手机号码' }"}, {attribute: "trigger", description: "String", type: "change", default: "校验触发方式。可选项: change/blur"}, {attribute: "type", description: "String", type: "error", default: "校验未通过时呈现的错误信息类型, 有 告警信息提示 和 错误信息提示 等两种。可选项: error/warning"}, {attribute: "url", description: "Boolean / Object", type: "-", default: "内置校验方法, 校验值是否为网络链接地址, 参数文档, 示例: { url: { protocols: ['http', 'https', 'ftp'] }, message: '请输入正确的 Url 地址' }。TS 类型: boolean | IsURLOptions import { IsURLOptions } from 'validator/es/lib/isURL'。详细类型定义"}, {attribute: "validator", description: "Function", type: "-", default: "自定义校验规则, 示例: { validator: (val) => val.length > 0, message: '请输入内容' }。TS 类型: CustomValidator type CustomValidator = (val: ValueType) => CustomValidateResolveType | Promise<CustomValidateResolveType> type CustomValidateResolveType = boolean | CustomValidateObj interface CustomValidateObj { result: boolean; message: string; type?: 'error' | 'warning' | 'success' } type ValueType = any。详细类型定义"}, {attribute: "whitespace", description: "Boolean", type: "-", default: "内置校验方法, 校验值是否为空格。示例: { whitespace: true, message: '值不能为空' }"}], {docTitle: "FormErrorMessage说明", tableRows: [{attribute: "boolean", description: "String", type: "-", default: "布尔类型校验不通过时的表单项显示文案, 全局配置默认是: '\$ {name} 数据类型必须是布尔类型'"}, {attribute: "date", description: "String", type: "-", default: "日期校验规则不通过时的表单项显示文案, 全局配置默认是: '请输入正确的\${name}'"}, {attribute: "enum", description: "String", type: "-", default: "枚举值校验规则不通过时的表单项显示文案, 全局配置默认是: '\${name} 只能是\${validate}等"}, {attribute: "idcard", description: "String", type: "-", default: "身份证号码校验不通过时的表单项显示文案, 全局配置默认是: '请输入正确的\${name}'"}, {attribute: "len", description: "String", type: "-", default: "值长度校验不通过时的表单项显示文案, 全局配置默认是: '\$ {name} 字符长度必须是 \${validate}'"}, {attribute: "max", description: "String", type: "-", default: "值的长度太长或值本身太大时, 校验不通过的表单项显示文案, 全局配置默认是: '\$ {name} 字符长度不能超过 \${validate} 个字符, 一个中文等于两个字符"}, {attribute: "min", description: "String", type: "-", default: "值的长度太短或值本身太小时, 校验不通过的表单项显示文案, 全局配置默认是: '\$ {name} 字符长度不能少于 \${validate} 个字符, 一个中文等于两个字符"}, {attribute: "number", description: "String", type: "-", default: "数字类型校验不通过时的表单项显示文案, 全局配置默认是: '\$ {name} 必须是数字'"}, {attribute: "pattern", description: "String", type: "-", default: "正则表达式校验不通过时的表单项显示文案, 全局配置默认是: '请输入正确的\${name}'"}, {attribute: "required", description: "String", type: "-", default: "没有填写必填项时的表单项显示文案, 全局配置默认是: '\$ {name} 必填'"}, {attribute: "telnumber", description: "String", type: "-", default: "手机号码校验不通过时的表单项显示文案, 全局配置默认是: '请输入正确的\${name}'"}, {attribute: "url", description: "String", type: "-", default: "链接校验规则不通过时的表单项显示文案, 全局配置默认是: '请输入正确的\${name}'"}, {attribute: "validator", description: "String", type: "-", default: "自定义校验规则校验不通过时的表单项显示文案, 全局配置默认是: '\$ {name} 不符合要求'"}]]], {componentName: "input", apiData: [{docTitle: "Input Props说明", tableRows: [{attribute: "align", description: "String", type: "left", default: "文本内容位置, 居左/居中/居右。可选项: left/center/right"}, {attribute: "allowInputOverMax", description: "Boolean", type: "false", default: "超出 maxlength 或 maxcharacter 之后是否允许继续输入"}, {attribute: "autoWidth", description: "Boolean", type: "false", default: "宽度随内容自适应"}, {attribute: "autocomplete", description: "String", type: "undefined", default: "是否开启自动填充功能, HTML5 原生属性, 点击查看详情"}, {attribute: "autofocus", description: "Boolean", type: "false", default: "自动聚焦"}, {attribute: "clearable", description: "Boolean", type: "false", default: "是否可清空"}]}]

{attribute:"disabled",description:"Boolean",type:"-",default:"是否禁用输入框"},
 {attribute:"format",description:"Function",type:"-",default:"指定输入框展示值的格式。TS 类型: InputFormatType type InputFormatType = (value: InputValue) => string。详细类型定义"},
 {attribute:"inputClass",description:"String / Object / Array",type:"-",default:"t-input 同级类名, 示例: 'name1 name2 name3' 或 ['name1', 'name2'] 或 [{ 'name1': true }]。TS 类型: ClassName。通用类型定义"},
 {attribute:"label",description:"String / Slot / Function",type:"-",default:"左侧文本。TS 类型: string | TNode。通用类型定义"},
 {attribute:"maxcharacter",description:"Number",type:"-",default:"用户最多可以输入的字符个数, 一个中文汉字表示两个字符长度。maxcharacter 和 maxlength 二选一使用"},
 {attribute:"maxlength",description:"Number",type:"-",default:"用户最多可以输入的文本长度, 一个中文等于一个计数长度。值为空, 则表示不限制输入长度。maxcharacter 和 maxlength 二选一使用"},
 {attribute:"name",description:"String",type:"-",default:"名称"},
 {attribute:"placeholder",description:"String",type:"undefined",default:"占位符"},
 {attribute:"prefixIcon",description:"Slot / Function",type:"-",default:"组件前置图标。TS 类型: TNode。通用类型定义"},
 {attribute:"readonly",description:"Boolean",type:"false",default:"只读状态"},
 {attribute:"showClearIconOnEmpty",description:"Boolean",type:"false",default:"输入框内容为空时, 悬浮状态是否显示清空按钮, 默认不显示"},
 {attribute:"showLimitNumber",description:"Boolean",type:"false",default:"是否在输入框右侧显示字数统计"},
 {attribute:"size",description:"String",type:"medium",default:"输入框尺寸。可选项: small/medium/large。TS 类型: SizeEnum。通用类型定义"},
 {attribute:"status",description:"String",type:"undefined",default:"输入框状态。默认情况会由组件内部根据实际情况呈现, 如果文本过长引起的状态变化。可选项: default/success/warning/error"},
 {attribute:"suffix",description:"String / Slot / Function",type:"-",default:"后置图标前的后置内容。TS 类型: string | TNode。通用类型定义"},
 {attribute:"suffixIcon",description:"Slot / Function",type:"-",default:"组件后置图标。TS 类型: TNode。通用类型定义"},
 {attribute:"tips",description:"String / Slot / Function",type:"-",default:"输入框下方提示文本, 会根据不同的 status 呈现不同的样式。TS 类型: string | TNode。通用类型定义"},
 {attribute:"type",description:"String",type:"text",default:"输入框类型。可选项: text/number/url/tel/password/search/submit/hidden"},
 {attribute:"value",description:"String / Number",type:"'",default:"输入框的值。支持语法糖 v-model 或 v-model:value。TS 类型: InputValue type InputValue = string。详细类型定义"},
 {attribute:"defaultValue",description:"String / Number",type:"'",default:"输入框的值。非受控属性。TS 类型: InputValue type InputValue = string。详细类型定义"},
 {attribute:"onBlur",description:"Function",type:"'",default:"TS 类型: (value: InputValue, context: { e: FocusEvent }) => void失去焦点时触发"},
 {attribute:"onChange",description:"Function",type:"'",default:"TS 类型: (value: InputValue, context?: { e?: InputEvent | MouseEvent | CompositionEvent; trigger: 'input' | 'initial' | 'clear' }) => void输入框值发生变化时触发。trigger=initial 表示传入的数据不符合预期, 组件自动处理后触发 change 告知父组件。如: 初始值长度超过 maxlength 限制"},
 {attribute:"onClear",description:"Function",type:"'",default:"TS 类型: (context: { e: MouseEvent }) => void清空按钮点击时触发"},
 {attribute:"onClick",description:"Function",type:"'",default:"TS 类型: (context: { e: MouseEvent }) => void点击组件时触发"},
 {attribute:"onCompositionend",description:"Function",type:"'",default:"TS 类型: (value: InputValue, context: { e: CompositionEvent }) => void中文输入结束时触发"},
 {attribute:"onCompositionstart",description:"Function",type:"'",default:"TS 类型: (value: InputValue, context: { e: CompositionEvent }) => void中文输入开始时触发"},
 {attribute:"onEnter",description:"Function",type:"'",default:"TS 类型: (value: InputValue, context: { e: KeyboardEvent }) => void回车键按下时触发"},
 {attribute:"onFocus",description:"Function",type:"'",default:"TS 类型: (value: InputValue, context: { e: FocusEvent }) => void获得焦点时触发"},
 {attribute:"onKeydown",description:"Function",type:"'",default:"TS 类型: (value: InputValue, context: { e: KeyboardEvent }) => void键盘按下时触发"},
 {attribute:"onKeyPress",description:"Function",type:"'",default:"TS 类型: (value: InputValue, context: { e: KeyboardEvent }) => void按下字符键时触发 (keydown -> keypress -> keyup) "},
 {attribute:"onKeyUp",description:"Function",type:"'",default:"TS 类型: (value: InputValue, context: { e: KeyboardEvent }) => void释放键盘时触发"},
 {attribute:"onMouseenter",description:"Function",type:"'",default:"TS 类型: (context: { e: MouseEvent }) => void进入输入框时触发"},
 {attribute:"onMouseleave",description:"Function",type:"'",default:"TS 类型: (context: { e: MouseEvent }) => void离开输入框时触发"},
 {attribute:"onPaste",description:"Function",type:"'",default:"TS 类型: (context: { e: ClipboardEvent; pasteValue: string }) => void粘贴事件, pasteValue 表示粘贴板的内容"},
 {attribute:"onValidate",description:"Function",type:"'",default:"TS 类型: (context: { error?: 'exceed-maximum' | 'below-minimum' }) => void字数超出限制时触发"},
 {attribute:"onWheel",description:"Function",type:"'",default:"TS 类型: (context: { e: WheelEvent }) => void输入框中滚动鼠标时触发"}], {docTitle:"Input Events说明",tableRows:
 [{attribute:"blur",description:"(value: InputValue, context: { e: FocusEvent })",type:"失去焦点时触发"},
 {attribute:"change",description:"(value: InputValue, context?: { e?: InputEvent | MouseEvent | CompositionEvent; trigger: 'input' | 'initial' | 'clear' })",type:"输入框值发生变化时触发。trigger=initial 表示传入的数据不符合预期, 组件自动处理后触发 change 告知父组件。如: 初始值长度超过 maxlength 限制"},
 {attribute:"clear",description:"(context: { e: MouseEvent })",type:"清空按钮点击时触发"},
 {attribute:"click",description:"(context: { e: MouseEvent })",type:"点击组件时触发"},

```

{attribute:"compositionend",description:"(value: InputValue, context: { e: CompositionEvent
})",type:"中文输入结束时触发"},{attribute:"compositionstart",description:"(value: InputValue,
context: { e: CompositionEvent })",type:"中文输入开始时触发"},{attribute:"enter",description:"
(value: InputValue, context: { e: KeyboardEvent })",type:"回车键按下时触发"},
{attribute:"focus",description:"(value: InputValue, context: { e: FocusEvent })",type:"获得焦点时触
发"},{attribute:"keydown",description:"(value: InputValue, context: { e: KeyboardEvent })",type:"键
盘按下时触发"},{attribute:"keypress",description:"(value: InputValue, context: { e: KeyboardEvent
})",type:"按下字符键时触发 (keydown -> keypress -> keyup) "},{attribute:"keyup",description:"(value:
InputValue, context: { e: KeyboardEvent })",type:"释放键盘时触发"},
{attribute:"mouseenter",description:"(context: { e: MouseEvent })",type:"进入输入框时触发"},
{attribute:"mouseleave",description:"(context: { e: MouseEvent })",type:"离开输入框时触发"},
{attribute:"paste",description:"(context: { e: ClipboardEvent; pasteValue: string })",type:"粘贴事
件, pasteValue 表示粘贴板的内容"},{attribute:"validate",description:"(context: { error?: 'exceed-
maximum' | 'below-minimum' })",type:"字数超出限时触发"},{attribute:"wheel",description:"(context:
{ e: WheelEvent })",type:"输入框中滚动鼠标时触发"}]],{docTitle:"InputGroup Props说明",tableRows:
[[{attribute:"separate",description:"Boolean",type:"-","default:"多个输入框之间是否需要间隔"}]]],
{componentName:"input-adornment",apiData:[{docTitle:"InputAdornment Props说明",tableRows:
[[{attribute:"append",description:"String / Slot / Function",type:"-","default:"后缀装饰。TS 类型:
string | TNode。通用类型定义"},{attribute:"prepend",description:"String / Slot / Function",type:"-
","default:"前缀装饰。TS 类型: string | TNode。通用类型定义"}]]}],{componentName:"input-
number",apiData:[{docTitle:"InputNumber Props说明",tableRows:
[[{attribute:"align",description:"String",type:"-","default:"文本内容位置, 居左/居中/居右。可选项:
left/center/right"},{attribute:"allowInputOverLimit",description:"Boolean",type:"true",default:"是否
允许输入超过 max min 范围外的数字。为保障用户体验, 仅在失去焦点时进行数字范围矫正。默认允许超出, 数
字超出范围时, 输入框变红提醒"},{attribute:"autoWidth",description:"Boolean",type:"false",default:"宽
度随内容自适应"},{attribute:"decimalPlaces",description:"Number",type:"undefined",default:"小数位
数"},{attribute:"disabled",description:"Boolean",type:"-","default:"禁用组件"},
{attribute:"format",description:"Function",type:"-","default:"格式化输入框展示值。第二个事件参数
context.fixedNumber 表示处理过小数位数 decimalPlaces 的数字。TS 类型: (value: InputNumberValue,
context?: { fixedNumber?: InputNumberValue }) => InputNumberValue"},
{attribute:"inputProps",description:"Object",type:"-","default:"透传 Input 输入框组件全部属性。TS 类
型: InputProps, Input API Documents。详细类型定义"},{attribute:"label",description:"String / Slot /
Function",type:"-","default:"左侧文本。TS 类型: string | TNode。通用类型定义"},
{attribute:"largeNumber",description:"Boolean",type:"false",default:"是否作为大数使用。JS 支持的最大
数字位数是 16 位, 超过 16 位的数字需作为字符串大数处理。此时, 数据类型必须保持为字符串, 否则会丢失数
据"},{attribute:"max",description:"String / Number",type:"Infinity",default:"最大值。如果是大数, 请
传入字符串。TS 类型: InputNumberValue"},{attribute:"min",description:"String / Number",type:"-
Infinity",default:"最小值。如果是大数, 请传入字符串。TS 类型: InputNumberValue"},
{attribute:"placeholder",description:"String",type:"undefined",default:"占位符"},
{attribute:"readonly",description:"Boolean",type:"false",default:"只读状态"},
{attribute:"size",description:"String",type:"medium",default:"组件尺寸。可选项:
small/medium/large"},{attribute:"status",description:"String",type:"-","default:"文本框状态。可选项:
default/success/warning/error"},{attribute:"step",description:"String / Number",type:"1",default:"数
值改变步数, 可以是小数。如果是大数, 请保证数据类型为字符串。TS 类型: InputNumberValue"},
{attribute:"suffix",description:"String / Slot / Function",type:"-","default:"后置内容。TS 类型:
string | TNode。通用类型定义"},{attribute:"theme",description:"String",type:"row",default:"按钮布
局。可选项: column/row/normal"},{attribute:"tips",description:"String / Slot / Function",type:"-
","default:"输入框下方提示文本, 会根据不同的 status 呈现不同的样式。TS 类型: string | TNode。通用类型
定义"},{attribute:"value",description:"String / Number",type:"-","default:"数字输入框的值。当值为 ''
时, 输入框显示为空。支持语法糖 v-model 或 v-model:value。TS 类型: T type InputNumberValue = number |
string。详细类型定义"},{attribute:"defaultValue",description:"String / Number",type:"-","default:"数
字输入框的值。当值为 '' 时, 输入框显示为空。非受控属性。TS 类型: T type InputNumberValue = number |
string。详细类型定义"},{attribute:"onBlur",description:"Function",type:"",default:"TS 类型: (value:
InputNumberValue, context: { e: FocusEvent }) => void失去焦点时触发"},
{attribute:"onChange",description:"Function",type:"",default:"TS 类型: (value: T, context:
ChangeContext) => void值变化时触发。详细类型定义。interface ChangeContext { type: ChangeSource; e:
InputEvent | MouseEvent | FocusEvent | KeyboardEvent }type ChangeSource = 'add' | 'reduce' | 'input'
| 'blur' | 'enter' | ''"},{attribute:"onEnter",description:"Function",type:"",default:"TS 类型:
(value: InputNumberValue, context: { e: KeyboardEvent }) => void回车键按下时触发"},
{attribute:"onFocus",description:"Function",type:"",default:"TS 类型: (value: InputNumberValue,
context: { e: FocusEvent }) => void获取焦点时触发"},
{attribute:"onKeyDown",description:"Function",type:"",default:"TS 类型: (value: InputNumberValue,
context: { e: KeyboardEvent }) => void键盘按下时触发"},
{attribute:"onKeyPress",description:"Function",type:"",default:"TS 类型: (value: InputNumberValue,
context: { e: KeyboardEvent }) => void按下字符键时触发 (keydown -> keypress -> keyup) "},
{attribute:"onKeyUp",description:"Function",type:"",default:"TS 类型: (value: InputNumberValue,
context: { e: KeyboardEvent }) => void释放键盘时触发"},
{attribute:"onValidate",description:"Function",type:"",default:"TS 类型: (context: { error?:
'exceed-maximum' | 'below-minimum' }) => void最大值或最小值校验结束后触发, exceed-maximum 表示超出最

```


大值, below-minimum 表示小于最小值"}], {docTitle: "InputNumber Events说明", tableRows: [[{attribute: "blur", description: "(value: InputNumberValue, context: { e: FocusEvent })", type: "失去焦点时触发"}, {attribute: "change", description: "(value: T, context: ChangeContext)", type: "值变化时触发。详细类型定义。interface ChangeContext { type: ChangeSource; e: InputEvent | MouseEvent | FocusEvent | KeyboardEvent } type ChangeSource = 'add' | 'reduce' | 'input' | 'blur' | 'enter' | ''", {attribute: "enter", description: "(value: InputNumberValue, context: { e: KeyboardEvent })", type: "回车键按下时触发"}, {attribute: "focus", description: "(value: InputNumberValue, context: { e: FocusEvent })", type: "获取焦点时触发"}, {attribute: "keydown", description: "(value: InputNumberValue, context: { e: KeyboardEvent })", type: "键盘按下时触发"}, {attribute: "keypress", description: "(value: InputNumberValue, context: { e: KeyboardEvent })", type: "按下字符键时触发 (keydown -> keypress -> keyup)"}, {attribute: "keyup", description: "(value: InputNumberValue, context: { e: KeyboardEvent })", type: "释放键盘时触发"}, {attribute: "validate", description: "(context: { error?: 'exceed-maximum' | 'below-minimum' })", type: "最大值或最小值校验结束后触发, exceed-maximum 表示超出最大值, below-minimum 表示小于最小值"}]]], {componentName: "radio", apiData: [{docTitle: "Radio Props说明", tableRows: [{attribute: "allowUncheck", description: "Boolean", type: "false", default: "是否允许取消选中"}, {attribute: "checked", description: "Boolean", type: "false", default: "是否选中。支持语法糖 v-model 或 v-model:checked"}, {attribute: "defaultChecked", description: "Boolean", type: "false", default: "是否选中。非受控属性"}, {attribute: "default", description: "String / Slot / Function", type: "-", default: "单选按钮内容, 同 label。TS 类型: string | TNode。通用类型定义"}, {attribute: "disabled", description: "Boolean", type: "undefined", default: "是否为禁用态。如果存在父组件 RadioGroup, 默认值由 RadioGroup.disabled 控制。Radio.disabled 优先级高于 RadioGroup.disabled"}, {attribute: "label", description: "String / Slot / Function", type: "-", default: "主文案。TS 类型: string | TNode。通用类型定义"}, {attribute: "name", description: "String", type: "-", default: "HTML 元素原生属性"}, {attribute: "value", description: "String / Number / Boolean", type: "undefined", default: "单选按钮的值。TS 类型: string | number | boolean"}, {attribute: "onChange", description: "Function", type: "", default: "TS 类型: (checked: boolean, context: { e: Event }) => void选中状态变化时触发"}, {attribute: "onClick", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent }) => void点击时出发, 一般用于外层阻止冒泡场景"}]], {docTitle: "Radio Events说明", tableRows: [{attribute: "change", description: "(checked: boolean, context: { e: Event })", type: "选中状态变化时触发"}, {attribute: "click", description: "(context: { e: MouseEvent })", type: "点击时出发, 一般用于外层阻止冒泡场景"}]], {docTitle: "RadioGroup Props说明", tableRows: [{attribute: "allowUncheck", description: "Boolean", type: "false", default: "是否允许取消选中"}, {attribute: "disabled", description: "Boolean", type: "-", default: "是否禁用全部子单选框。默认为 false。RadioGroup.disabled 优先级低于 Radio.disabled"}, {attribute: "name", description: "String", type: "-", default: "HTML 元素原生属性"}, {attribute: "options", description: "Array", type: "-", default: "单选组件按钮形式。RadioOption 数据类型为 string 或 number 时, 表示 label 和 value 值相同。TS 类型: Array<RadioOption> type RadioOption = string | number | RadioOptionObj interface RadioOptionObj { label?: string | TNode; value?: string | number | boolean; disabled?: boolean }。通用类型定义。详细类型定义"}, {attribute: "size", description: "String", type: "medium", default: "组件尺寸【讨论中】。可选项: small/medium/large。TS 类型: SizeEnum。通用类型定义"}, {attribute: "value", description: "String / Number / Boolean", type: "-", default: "选中的值。支持语法糖 v-model 或 v-model:value。TS 类型: T type RadioValue = string | number | boolean。详细类型定义"}, {attribute: "defaultValue", description: "String / Number / Boolean", type: "-", default: "选中的值。非受控属性。TS 类型: T type RadioValue = string | number | boolean。详细类型定义"}, {attribute: "variant", description: "String", type: "outline", default: "单选组件按钮形式。可选项: outline/primary-filled/default-filled"}, {attribute: "onChange", description: "Function", type: "", default: "TS 类型: (value: T, context: { e: Event }) => void选中值发生变化时触发"}]], {docTitle: "RadioGroup Events说明", tableRows: [{attribute: "change", description: "(value: T, context: { e: Event })", type: "选中值发生变化时触发"}]]], {componentName: "range-input", apiData: [{docTitle: "RangeInput Props说明", tableRows: [{attribute: "activeIndex", description: "Number", type: "-", default: "输入框高亮状态序号"}, {attribute: "clearable", description: "Boolean", type: "false", default: "是否可清空"}, {attribute: "disabled", description: "Boolean", type: "-", default: "是否禁用范围输入框"}, {attribute: "format", description: "Array / Function", type: "-", default: "指定输入框展示值的格式。TS 类型: InputFormatType | Array<InputFormatType>"}, {attribute: "inputProps", description: "Object / Array", type: "-", default: "透传 Input 输入框组件全部属性, 数组第一项表示第一个输入框属性, 第二项表示第二个输入框属性。示例: [{ label: 'A', name: 'A-name' }, { label: 'B', name: 'B-name' }]。TS 类型: InputProps | Array<InputProps>, Input API Documents。详细类型定义"}, {attribute: "label", description: "String / Slot / Function", type: "-", default: "左侧内容。TS 类型: string | TNode。通用类型定义"}, {attribute: "placeholder", description: "String / Array", type: "-", default: "占位符, 示例: '请输入' 或者 ['开始日期', '结束日期']。TS 类型: string | Array<string>"}, {attribute: "prefixIcon", description: "Slot / Function", type: "-", default: "组件前置图标。TS 类型: TNode。通用类型定义"}, {attribute: "readonly", description: "Boolean", type: "false", default: "只读状态"}, {attribute: "separator", description: "String / Slot / Function", type: "-", default: "范围分隔符。TS 类型: string | TNode。通用类型定义"}, {attribute: "showClearIconOnEmpty", description: "Boolean", type: "false", default: "输入框内容为空时, 悬浮状态是否显示清空按钮, 默认不显示"}, {attribute: "size", description: "String", type: "medium", default: "输入框尺寸。可选项: small/medium/large"}]]],

```

{attribute:"status",description:"String",type:"default",default:"输入框状态。可选项：
default/success/warning/error"},{attribute:"suffix",description:"String / Slot / Function",type:"-
",default:"后置图标前的后置内容。TS 类型：string | TNode。通用类型定义"},
{attribute:"suffixIcon",description:"Slot / Function",type:"-",default:"组件后置图标。TS 类型：
TNode。通用类型定义"},{attribute:"tips",description:"String / Slot / Function",type:"-",default:"输
入框下方提示文本，会根据不同的 status 呈现不同的样式。TS 类型：string | TNode。通用类型定义"},
{attribute:"value",description:"Array",type:"[]",default:"范围输入框的值。支持语法糖 v-model 或 v-
model:value。TS 类型：RangeInputValue type RangeInputValue = Array<InputValue>。详细类型定义"},
{attribute:"defaultValue",description:"Array",type:"[]",default:"范围输入框的值。非受控属性。TS 类
型：RangeInputValue type RangeInputValue = Array<InputValue>。详细类型定义"},
{attribute:"onBlur",description:"Function",type:"",default:"TS 类型：(value: RangeInputValue,
context?: { e?: FocusEvent; position?: RangeInputPosition }) => void范围输入框失去焦点时触发"},
{attribute:"onChange",description:"Function",type:"",default:"TS 类型：(value: RangeInputValue,
context?: { e?: InputEvent | MouseEvent | CompositionEvent; position?: RangeInputPosition; trigger?:
'input' | 'initial' | 'clear' }) => void范围输入框值发生变化时触发。详细类型定义。type
RangeInputPosition = 'first' | 'second' | 'all'"},
{attribute:"onClear",description:"Function",type:"",default:"TS 类型：(context: { e: MouseEvent })
=> void清空按钮点击时触发"},{attribute:"onClick",description:"Function",type:"",default:"TS 类型：
(context?: { e?: MouseEvent; position?: RangeInputPosition }) => void范围输入框点击时触发"},
{attribute:"onEnter",description:"Function",type:"",default:"TS 类型：(value: RangeInputValue,
context?: { e?: InputEvent | MouseEvent; position?: RangeInputPosition }) => void回车键按下时触发"},
{attribute:"onFocus",description:"Function",type:"",default:"TS 类型：(value: RangeInputValue,
context?: { e?: FocusEvent; position?: RangeInputPosition }) => void范围输入框获得焦点时触发"},
{attribute:"onMouseenter",description:"Function",type:"",default:"TS 类型：(context: { e: MouseEvent
}) => void进入输入框时触发"},{attribute:"onMouseleave",description:"Function",type:"",default:"TS 类
型：(context: { e: MouseEvent }) => void离开输入框时触发"}]],{docTitle:"RangeInput Events说
明",tableRows:[{attribute:"blur",description:"(value: RangeInputValue, context?: { e?: FocusEvent;
position?: RangeInputPosition })",type:"范围输入框失去焦点时触发"},{attribute:"change",description:"
(value: RangeInputValue, context?: { e?: InputEvent | MouseEvent | CompositionEvent; position?:
RangeInputPosition; trigger?: 'input' | 'initial' | 'clear' })",type:"范围输入框值发生变化时触发。详
细类型定义。type RangeInputPosition = 'first' | 'second' | 'all'"},{attribute:"clear",description:"
(context: { e: MouseEvent })",type:"清空按钮点击时触发"},{attribute:"click",description:"(context?:
{ e?: MouseEvent; position?: RangeInputPosition })",type:"范围输入框点击时触发"},
{attribute:"enter",description:"(value: RangeInputValue, context?: { e?: InputEvent | MouseEvent;
position?: RangeInputPosition })",type:"回车键按下时触发"},{attribute:"focus",description:"(value:
RangeInputValue, context?: { e?: FocusEvent; position?: RangeInputPosition })",type:"范围输入框获得
焦点时触发"},{attribute:"mouseenter",description:"(context: { e: MouseEvent })",type:"进入输入框时触
发"},{attribute:"mouseleave",description:"(context: { e: MouseEvent })",type:"离开输入框时触发"}]],
{docTitle:"RangeInputInstanceFunctions 组件实例方法说明",tableRows:[{attribute:"blur",description:"
(options?: { position?: RangeInputPosition })",type:"-",default:"使其中一个输入框失去焦点"},
{attribute:"focus",description:"(options?: { position?: RangeInputPosition })",type:"-",default:"使其
中一个输入框获得焦点"},{attribute:"select",description:"(options?: { position?:
RangeInputPosition })",type:"-",default:"使其中一个输入框选中内容"}]],{docTitle:"RangeInputPopup
Props说明",tableRows:[{attribute:"autoWidth",description:"Boolean",type:"false",default:"宽度随内容
自适应"},{attribute:"disabled",description:"Boolean",type:"-",default:"是否禁用范围输入框，值为数组
表示可分别控制某一个输入框是否禁用"},{attribute:"inputValue",description:"Array",type:"-
",default:"输入框的值。支持语法糖 v-model:inputValue。TS 类型：RangeInputValue"},
{attribute:"defaultInputValue",description:"Array",type:"-",default:"输入框的值。非受控属性。TS 类
型：RangeInputValue"},{attribute:"panel",description:"String / Slot / Function",type:"-",default:"下
拉框内容，可完全自定义。TS 类型：string | TNode。通用类型定义"},
{attribute:"popupProps",description:"Object",type:"-",default:"透传 Popup 浮层组件全部属性。TS 类
型：PopupProps, Popup API Documents。详细类型定义"},
{attribute:"popupVisible",description:"Boolean",type:"-",default:"是否显示下拉框"},
{attribute:"rangeInputProps",description:"Object",type:"-",default:"透传 RangeInput 组件全部属性。TS
类型：RangeInputProps, RangeInput API Documents。详细类型定义"},
{attribute:"readonly",description:"Boolean",type:"false",default:"只读状态，值为真会隐藏输入框，且无
法打开下拉框"},{attribute:"status",description:"String",type:"default",default:"输入框状态。可选项：
default/success/warning/error"},{attribute:"tips",description:"String / Slot / Function",type:"-
",default:"输入框下方提示文本，会根据不同的 status 呈现不同的样式。TS 类型：string | TNode。通用类型
定义"},{attribute:"onInputChange",description:"Function",type:"",default:"TS 类型：(value:
RangeInputValue, context?: RangeInputValueChangeContext) => void输入框值发生变化时触发，
context.trigger 表示触发输入框值变化的来源：文本输入触发、清除按钮触发等。详细类型定义。type
RangeInputValueChangeContext = { e?: InputEvent | MouseEvent; trigger?: 'input' | 'clear',
position?: RangeInputPosition }"},
{attribute:"onPopupVisibleChange",description:"Function",type:"",default:"TS 类型：(visible:
boolean, context: PopupVisibleChangeContext) => void下拉框显示或隐藏时触发。详细类型定义。import {
PopupVisibleChangeContext } from '@Popup'"}]],{docTitle:"RangeInputPopup Events说明",tableRows:
[{attribute:"input-change",description:"(value: RangeInputValue, context?:
RangeInputValueChangeContext)",type:"输入框值发生变化时触发，context.trigger 表示触发输入框值变化的

```

来源：文本输入触发、清除按钮触发等。详细类型定义。type RangeInputValueChangeContext = { e?: InputEvent | MouseEvent; trigger?: 'input' | 'clear', position?: RangeInputPosition }}, {attribute: "popup-visible-change", description: "(visible: boolean, context: PopupVisibleChangeContext)", type: "下拉框显示或隐藏时触发。详细类型定义。import { PopupVisibleChangeContext } from '@Popup'"}]]], {componentName: "select", apiData: [{docTitle: "Select Props说明", tableRows: [{attribute: "autoWidth", description: "Boolean", type: "false", default: "宽度随内容自适应"}, {attribute: "autofocus", description: "Boolean", type: "false", default: "自动聚焦"}, {attribute: "borderless", description: "Boolean", type: "false", default: "无边框模式"}, {attribute: "clearable", description: "Boolean", type: "false", default: "是否可以清空选项"}, {attribute: "collapsedItems", description: "Slot / Function", type: "-", default: "多选情况下，用于设置折叠项内容，默认为 +N。如果需要悬浮就显示其他内容，可以使用 collapsedItems 自定义。value 表示当前存在的所有标签，collapsedTags 表示折叠的标签，泛型 T 继承 SelectOption，表示选项数据；count 表示折叠的数量。TS 类型：TNode<{ value: T[]; collapsedSelectedItems: T[]; count: number }>。通用类型定义"}, {attribute: "creatable", description: "Boolean", type: "false", default: "是否允许用户创建新条目，需配合 filterable 使用"}, {attribute: "disabled", description: "Boolean", type: "-", default: "是否禁用组件"}, {attribute: "empty", description: "String / Slot / Function", type: "-", default: "当下拉列表为空时显示的内容。TS 类型：string | TNode。通用类型定义"}, {attribute: "filter", description: "Function", type: "-", default: "自定义搜索规则，用于对现有数据进行搜索，判断是否过滤某一项数据。参数 filterWords 表示搜索词，option 表示单个选项内容，返回值为 true 保留该选项，返回值为 false 则隐藏该选项。使用该方法时无需设置 filterable。TS 类型：(filterWords: string, option: T) => boolean | Promise<boolean>"}, {attribute: "filterable", description: "Boolean", type: "false", default: "是否可搜索，默认搜索规则不区分大小写，全文本任意位置匹配。如果默认搜索规则不符合业务需求，可以更为使用 filter 自定义过滤规则"}, {attribute: "inputProps", description: "Object", type: "-", default: "透传 Input 输入框组件的全部属性。TS 类型：InputProps, Input API Documents。详细类型定义"}, {attribute: "inputValue", description: "String / Number", type: "-", default: "输入框的值。支持语法糖 v-model:inputValue。TS 类型：InputValue, Input API Documents。详细类型定义"}, {attribute: "defaultInputValue", description: "String / Number", type: "-", default: "输入框的值。非受控属性。TS 类型：InputValue, Input API Documents。详细类型定义"}, {attribute: "keys", description: "Object", type: "-", default: "用来定义 value / label 在 options 中对应的字段别名。TS 类型：SelectKeysType interface SelectKeysType { value?: string; label?: string }。通用类型定义。详细类型定义"}, {attribute: "label", description: "String / Slot / Function", type: "-", default: "左侧文本。TS 类型：string | TNode。通用类型定义"}, {attribute: "loading", description: "Boolean", type: "false", default: "是否为加载状态"}, {attribute: "loadingText", description: "String / Slot / Function", type: "-", default: "远程加载时显示的文字，支持自定义。如加上超链接。TS 类型：string | TNode。通用类型定义"}, {attribute: "max", description: "Number", type: "0", default: "用于控制多选数量，值为 0 则不限制"}, {attribute: "minCollapsedNum", description: "Number", type: "0", default: "最小折叠数量，用于多选情况下折叠选中项，超出该数值的选中项折叠。值为 0 则表示不折叠"}, {attribute: "multiple", description: "Boolean", type: "false", default: "是否允许多选"}, {attribute: "options", description: "Array", type: "-", default: "数据化配置选项内容。TS 类型：Array<T>"}, {attribute: "panelBottomContent", description: "String / Slot / Function", type: "-", default: "面板内的底部内容。TS 类型：string | TNode。通用类型定义"}, {attribute: "panelTopContent", description: "String / Slot / Function", type: "-", default: "面板内的顶部内容。TS 类型：string | TNode。通用类型定义"}, {attribute: "placeholder", description: "String", type: "undefined", default: "占位符"}, {attribute: "popupProps", description: "Object", type: "-", default: "透传给 popup 组件的全部属性。TS 类型：PopupProps, Popup API Documents。详细类型定义"}, {attribute: "popupVisible", description: "Boolean", type: "-", default: "是否显示下拉框。支持语法糖 v-model:popupVisible"}, {attribute: "defaultPopupVisible", description: "Boolean", type: "-", default: "是否显示下拉框。非受控属性"}, {attribute: "prefixIcon", description: "Slot / Function", type: "-", default: "组件前置图标。TS 类型：TNode。通用类型定义"}, {attribute: "readonly", description: "Boolean", type: "false", default: "只读状态，值为真会隐藏输入框，且无法打开下拉框"}, {attribute: "reserveKeyword", description: "Boolean", type: "false", default: "多选且可搜索时，是否在选中一个选项后保留当前的搜索关键词"}, {attribute: "scroll", description: "Object", type: "-", default: "懒加载和虚拟滚动。为保证组件收益最大化，当数据量小于阈值 scroll.threshold 时，无论虚拟滚动的配置是否存在，组件内部都不会开启虚拟滚动，scroll.threshold 默认为 100。TS 类型：InfinityScroll。通用类型定义"}, {attribute: "selectInputProps", description: "Object", type: "-", default: "透传 SelectInput 筛选器输入框组件的全部属性。TS 类型：SelectInputProps, SelectInput API Documents。详细类型定义"}, {attribute: "showArrow", description: "Boolean", type: "true", default: "是否显示右侧箭头，默认显示"}, {attribute: "size", description: "String", type: "medium", default: "组件尺寸。可选项：small/medium/large。TS 类型：SizeEnum。通用类型定义"}, {attribute: "status", description: "String", type: "default", default: "输入框状态。可选项：default/success/warning/error"}, {attribute: "suffix", description: "String / Slot / Function", type: "-", default: "后置图标前的后置内容。TS 类型：string | TNode。通用类型定义"}, {attribute: "suffixIcon", description: "Slot / Function", type: "-", default: "组件后置图标。TS 类型：TNode。通用类型定义"}, {attribute: "tagInputProps", description: "Object", type: "-", default: "透传 TagInput 标签输入框组件的全部属性。TS 类型：TagInputProps, TagInput API Documents。详细类型定义"}, {attribute: "tagProps", description: "Object", type: "-", default: "透传 Tag 标签组件全部属性。TS 类型：TagProps, Tag API Documents。详细类型定义"}, {attribute: "tips", description: "String / Slot / Function", type: "-", default: "输入框下方提示文本，会根据不同的 status 呈现不同的样式。TS 类型：string | TNode。通用类型定义"}, {attribute: "value", description: "String / Number / Boolean / Object /

Array", type: "-", default: "选中值。支持语法糖 v-model 或 v-model:value。TS 类型: SelectValue type
 SelectValue<T extends SelectOption = SelectOption> = string | number | boolean | T |
 Array<SelectValue<T>>。详细类型定义", {attribute: "defaultValue", description: "String / Number /
 Boolean / Object / Array", type: "-", default: "选中值。非受控属性。TS 类型: SelectValue type
 SelectValue<T extends SelectOption = SelectOption> = string | number | boolean | T |
 Array<SelectValue<T>>。详细类型定义", {attribute: "valueDisplay", description: "String / Slot /
 Function", type: "-", default: "自定义选中项呈现的内容。TS 类型: string | TNode<{ value: SelectValue;
 onClose: (index: number) => void; displayValue?: SelectValue }>。通用类型定义",
 {attribute: "valueType", description: "String", type: "value", default: "用于控制选中值的类型。假设数据选项
 为: [{ label: '姓名', value: 'name' }], value 表示值仅返回数据选项中的 value, object 表示值返回全部
 数据。。可选项: value/object", {attribute: "onBlur", description: "Function", type: "", default: "TS 类型:
 (context: { value: SelectValue; e: FocusEvent | KeyboardEvent }) => void输入框失去焦点时触发",
 {attribute: "onChange", description: "Function", type: "", default: "TS 类型: (value: SelectValue, context:
 { option?: T, selectedOptions: T[], trigger: SelectValueChangeTrigger; e?: MouseEvent |
 KeyboardEvent }) => void选中值变化时触发。context.trigger 表示触发变化的来源;
 context.selectedOptions 表示选中值的完整对象, 数组长度一定和 value 相同; context.option 表示当前操作
 的选项, 不一定存在。详细类型定义。type SelectValueChangeTrigger = 'clear' | 'tag-remove' |
 'backspace' | 'check' | 'unchecked'", {attribute: "onClear", description: "Function", type: "", default: "TS
 类型: (context: { e: MouseEvent }) => void点击清除按钮时触发",
 {attribute: "onCreate", description: "Function", type: "", default: "TS 类型: (value: string | number) =>
 void当选择新创建的条目时触发", {attribute: "onEnter", description: "Function", type: "", default: "TS 类
 型: (context: { inputValue: string; e: KeyboardEvent; value: SelectValue }) => void回车键按下时触
 发。inputValue 表示输入框的值, value 表示选中值",
 {attribute: "onFocus", description: "Function", type: "", default: "TS 类型: (context: { value:
 SelectValue; e: FocusEvent | KeyboardEvent }) => void输入框获得焦点时触发",
 {attribute: "onInputChange", description: "Function", type: "", default: "TS 类型: (value: InputValue,
 context?: SelectInputValueChangeContext) => void输入框值发生变化时触发, context.trigger 表示触发输入
 框值变化的来源: 文本输入触发、清除按钮触发、失去焦点等。详细类型定义。import {
 SelectInputValueChangeContext } from '@SelectInput'",
 {attribute: "onPopupVisibleChange", description: "Function", type: "", default: "TS 类型: (visible:
 boolean, context: PopupVisibleChangeContext) => void下拉框显示或隐藏时触发。详细类型定义。import {
 PopupVisibleChangeContext } from '@Popup'",
 {attribute: "onRemove", description: "Function", type: "", default: "TS 类型: (options:
 SelectRemoveContext<T>) => void多选模式下, 选中数据被移除时触发。详细类型定义。interface
 SelectRemoveContext<T> { value: string | number; data: T; e: MouseEvent | KeyboardEvent }",
 {attribute: "onSearch", description: "Function", type: "", default: "TS 类型: (filterWords: string,
 context: { e: KeyboardEvent }) => void输入值变化时, 触发搜索事件。主要用于远程搜索新数据"}],
 {docTitle: "Select Events说明", tableRows: [{attribute: "blur", description: "(context: { value:
 SelectValue; e: FocusEvent | KeyboardEvent })", type: "输入框失去焦点时触发",
 {attribute: "change", description: "(value: SelectValue, context: { option?: T, selectedOptions: T[],
 trigger: SelectValueChangeTrigger; e?: MouseEvent | KeyboardEvent })", type: "选中值变化时触发。
 context.trigger 表示触发变化的来源; context.selectedOptions 表示选中值的完整对象, 数组长度一定和
 value 相同; context.option 表示当前操作的选项, 不一定存在。详细类型定义。type
 SelectValueChangeTrigger = 'clear' | 'tag-remove' | 'backspace' | 'check' | 'unchecked'",
 {attribute: "clear", description: "(context: { e: MouseEvent })", type: "点击清除按钮时触发",
 {attribute: "create", description: "(value: string | number)", type: "当选择新创建的条目时触发",
 {attribute: "enter", description: "(context: { inputValue: string; e: KeyboardEvent; value: SelectValue
 })", type: "回车键按下时触发。inputValue 表示输入框的值, value 表示选中值",
 {attribute: "focus", description: "(context: { value: SelectValue; e: FocusEvent | KeyboardEvent
 })", type: "输入框获得焦点时触发", {attribute: "input-change", description: "(value: InputValue,
 context?: SelectInputValueChangeContext)", type: "输入框值发生变化时触发, context.trigger 表示触发输入
 框值变化的来源: 文本输入触发、清除按钮触发、失去焦点等。详细类型定义。import {
 SelectInputValueChangeContext } from '@SelectInput'", {attribute: "popup-visible-
 change", description: "(visible: boolean, context: PopupVisibleChangeContext)", type: "下拉框显示或隐藏
 时触发。详细类型定义。import { PopupVisibleChangeContext } from '@Popup'",
 {attribute: "remove", description: "(options: SelectRemoveContext<T>)", type: "多选模式下, 选中数据被移除
 时触发。详细类型定义。interface SelectRemoveContext<T> { value: string | number; data: T; e:
 MouseEvent | KeyboardEvent }", {attribute: "search", description: "(filterWords: string, context: { e:
 KeyboardEvent })", type: "输入值变化时, 触发搜索事件。主要用于远程搜索新数据"}], {docTitle: "Option
 Props说明", tableRows: [{attribute: "checkAll", description: "Boolean", type: "false", default: "当前选项是否
 为全选, 全选可以在顶部, 也可以在底部。点击当前选项会选中禁用态除外的全部选项, 即使是分组选择器也会选
 中全部选项", {attribute: "content", description: "String / Slot / Function", type: "-", default: "用于定义
 复杂的选项内容。TS 类型: string | TNode。通用类型定义", {attribute: "default", description: "String /
 Slot / Function", type: "-", default: "用于定义复杂的选项内容。同 content。TS 类型: string | TNode。通用
 类型定义", {attribute: "disabled", description: "Boolean", type: "false", default: "是否禁用该选项",
 {attribute: "label", description: "String", type: "-", default: "选项名称",
 {attribute: "title", description: "String", type: "-", default: "选项标题, 在选项过长时hover选项展示",
 {attribute: "value", description: "String / Number", type: "-", default: "选项值"}], {docTitle: "OptionGroup
 Props说明", tableRows: [{attribute: "divider", description: "Boolean", type: "true", default: "是否显示分隔

线"}, {attribute: "label", description: "String", type: "-", default: "分组别名"}], {docTitle: "TScroll说明", tableRows: [{attribute: "bufferSize", description: "Number", type: "20", default: "表示除可视区域外, 额外渲染的行数, 避免快速滚动过程中, 新出现的内容来不及渲染从而出现空白"}, {attribute: "isFixedRowHeight", description: "Boolean", type: "false", default: "表示每行内容是否同一个固定高度, 仅在 scroll.type 为 virtual 时有效, 该属性设置为 true 时, 可用于简化虚拟滚动内部计算逻辑, 提升性能, 此时则需要明确指定 scroll.rowHeight 属性的值"}, {attribute: "rowHeight", description: "Number", type: "-", default: "行高, 不会给<tr>元素添加样式高度, 仅作为滚动时的行高参考。一般情况不需要设置该属性。如果设置, 可尽量将该属性设置为每行平均高度, 从而使得滚动过程更加平滑"}, {attribute: "threshold", description: "Number", type: "100", default: "启动虚拟滚动的阈值。为保证组件收益最大化, 当数据量小于阈值 scroll.threshold 时, 无论虚拟滚动的配置是否存在, 组件内部都不会开启虚拟滚动"}, {attribute: "type", description: "String", type: "-", default: "必需。滚动加载类型, 有两种: 懒加载和虚拟滚动。值为 lazy, 表示滚动时会进行懒加载, 非可视区域内的内容将不会默认渲染, 直到该内容可见时, 才会进行渲染, 并且已渲染的内容滚动到不可见时, 不会被销毁; 值为virtual时, 表示会进行虚拟滚动, 无论滚动条滚动到哪个位置, 同一时刻, 仅渲染该可视区域内的内容, 当需要展示的数据量较大时, 建议开启该特性。可选项: lazy/virtual"}]]], {componentName: "select-input", apiData: [{docTitle: "SelectInput Props说明", tableRows: [{attribute: "allowInput", description: "Boolean", type: "false", default: "是否允许输入"}, {attribute: "autoWidth", description: "Boolean", type: "false", default: "宽度随内容自适应"}, {attribute: "autofocus", description: "Boolean", type: "false", default: "自动聚焦"}, {attribute: "borderless", description: "Boolean", type: "false", default: "无边框模式"}, {attribute: "clearable", description: "Boolean", type: "false", default: "是否可清空"}, {attribute: "collapsedItems", description: "Slot / Function", type: "-", default: "标签过多的情况下, 折叠项内容, 默认为 +N。如果需要悬浮就显示其他内容, 可以使用 collapsedItems 自定义。value 表示所有标签值, collapsedTags 表示折叠标签值, count 表示折叠的数量。TS 类型: TNode<{ value: SelectInputValue; collapsedTags: SelectInputValue; count: number }>。通用类型定义"}, {attribute: "disabled", description: "Boolean", type: "-", default: "是否禁用"}, {attribute: "inputProps", description: "Object", type: "-", default: "透传 Input 输入框组件全部属性。TS 类型: InputProps, Input API Documents。详细类型定义"}, {attribute: "inputValue", description: "String / Number", type: "-", default: "输入框的值。支持语法糖 v-model:inputValue。TS 类型: InputValue, Input API Documents。详细类型定义"}, {attribute: "defaultInputValue", description: "String / Number", type: "-", default: "输入框的值。非受控属性。TS 类型: InputValue, Input API Documents。详细类型定义"}, {attribute: "keys", description: "Object", type: "-", default: "定义字段别名, 示例: { label: 'text', value: 'id', children: 'list' }。TS 类型: SelectInputKeys interface SelectInputKeys { label?: string; value?: string; children?: string }。详细类型定义"}, {attribute: "label", description: "String / Slot / Function", type: "-", default: "左侧文本。TS 类型: string | TNode。通用类型定义"}, {attribute: "loading", description: "Boolean", type: "false", default: "是否处于加载状态"}, {attribute: "minCollapsedNum", description: "Number", type: "0", default: "最小折叠数量, 用于标签数量过多的情况下折叠选中项, 超出该数值的选中项折叠。值为 0 则表示不折叠"}, {attribute: "multiple", description: "Boolean", type: "false", default: "是否为多选模式, 默认为单选"}, {attribute: "panel", description: "String / Slot / Function", type: "-", default: "下拉框内容, 可完全自定义。TS 类型: string | TNode。通用类型定义"}, {attribute: "placeholder", description: "String", type: "-", default: "占位符"}, {attribute: "popupProps", description: "Object", type: "-", default: "透传 Popup 浮层组件全部属性。TS 类型: PopupProps, Popup API Documents。详细类型定义"}, {attribute: "popupVisible", description: "Boolean", type: "-", default: "是否显示下拉框。支持语法糖 v-model:popupVisible"}, {attribute: "defaultPopupVisible", description: "Boolean", type: "-", default: "是否显示下拉框。非受控属性"}, {attribute: "readonly", description: "Boolean", type: "false", default: "只读状态, 值为真会隐藏输入框, 且无法打开下拉框"}, {attribute: "status", description: "String", type: "default", default: "输入框状态。可选项: default/success/warning/error"}, {attribute: "suffix", description: "String / Slot / Function", type: "-", default: "后置图标前的后置内容。TS 类型: string | TNode。通用类型定义"}, {attribute: "suffixIcon", description: "Slot / Function", type: "-", default: "组件后置图标。TS 类型: TNode。通用类型定义"}, {attribute: "tag", description: "String / Slot / Function", type: "-", default: "多选场景下, 自定义选中标签的内部内容。注意和 valueDisplay 区分, valueDisplay 是用来定义全部标签内容, 而非某一个标签。TS 类型: string | TNode<{ value: string | number }>。通用类型定义"}, {attribute: "tagInputProps", description: "Object", type: "-", default: "透传 TagInput 组件全部属性。TS 类型: TagInputProps, TagInput API Documents。详细类型定义"}, {attribute: "tagProps", description: "Object", type: "-", default: "透传 Tag 标签组件全部属性。TS 类型: TagProps, Tag API Documents。详细类型定义"}, {attribute: "tips", description: "String / Slot / Function", type: "-", default: "输入框下方提示文本, 会根据不同的 status 呈现不同的样式。TS 类型: string | TNode。通用类型定义"}, {attribute: "value", description: "String / Number / Boolean / Object / Array / Date", type: "-", default: "全部标签值。值为数组表示多个标签, 值为非数组表示单个数值。TS 类型: SelectInputValue type SelectInputValue = string | number | boolean | Date | Object | Array<any> | Array<SelectInputValue>。详细类型定义"}, {attribute: "valueDisplay", description: "String / Slot / Function", type: "-", default: "自定义值呈现的全部内容, 参数为所有标签的值。TS 类型: string | TNode<{ value: TagInputValue; onClose: (index: number, item?: any) => void }>。通用类型定义"}, {attribute: "onBlur", description: "Function", type: "", default: "TS 类型: (value: SelectInputValue, context: SelectInputBlurContext) => void失去焦点时触发, context.inputValue 表示输入框的值; context.tagInputValue 表示标签输入框的值。详细类型定义。type SelectInputBlurContext = PopupVisibleChangeContext & { inputValue: string; tagInputValue?: TagInputValue; }"}, {attribute: "onClear", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent })

```
=> void清空按钮点击时触发", {attribute: "onEnter", description: "Function", type: "", default: "TS 类型: (value: SelectInputValue, context: { e: KeyboardEvent; inputValue: InputValue; tagInputValue?: TagInputValue }) => void按钮按下 Enter 时触发",
{attribute: "onFocus", description: "Function", type: "", default: "TS 类型: (value: SelectInputValue, context: SelectInputFocusContext) => void聚焦时触发。详细类型定义。interface SelectInputFocusContext { inputValue: InputValue; tagInputValue?: TagInputValue; e: FocusEvent }"},
{attribute: "onInputChange", description: "Function", type: "", default: "TS 类型: (value: InputValue, context?: SelectInputValueChangeContext) => void输入框值发生变化时触发, context.trigger 表示触发输入框值变化的来源: 文本输入触发、清除按钮触发等。详细类型定义。interface SelectInputValueChangeContext { e?: Event | InputEvent | MouseEvent | FocusEvent | KeyboardEvent | CompositionEvent; trigger: 'input' | 'clear' | 'blur' | 'focus' | 'initial' | 'change' }"},
{attribute: "onMouseenter", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent }) => void进入输入框时触发", {attribute: "onMouseleave", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent }) => void离开输入框时触发",
{attribute: "onPaste", description: "Function", type: "", default: "TS 类型: (context: { e: ClipboardEvent; pasteValue: string }) => void粘贴事件, pasteValue 表示粘贴板的内容",
{attribute: "onPopupVisibleChange", description: "Function", type: "", default: "TS 类型: (visible: boolean, context: PopupVisibleChangeContext) => void下拉框显示或隐藏时触发。详细类型定义。import { PopupVisibleChangeContext } from '@Popup'",
{attribute: "onTagChange", description: "Function", type: "", default: "TS 类型: (value: TagInputValue, context: SelectInputChangeContext) => void值变化时触发, 参数 context.trigger 表示数据变化的触发来源; context.index 指当前变化项的下标; context.item 指当前变化项; context.e 表示事件参数。详细类型定义。type SelectInputChangeContext = TagInputChangeContext"}]], {docTitle: "SelectInput Events说明", tableRows: [{attribute: "blur", description: "(value: SelectInputValue, context: SelectInputBlurContext)", type: "失去焦点时触发, context.inputValue 表示输入框的值; context.tagInputValue 表示标签输入框的值。详细类型定义。type SelectInputBlurContext = PopupVisibleChangeContext & { inputValue: string; tagInputValue?: TagInputValue; }"},
{attribute: "clear", description: "(context: { e: MouseEvent })", type: "清空按钮点击时触发",
{attribute: "enter", description: "(value: SelectInputValue, context: { e: KeyboardEvent; inputValue: InputValue; tagInputValue?: TagInputValue })", type: "按钮按下 Enter 时触发",
{attribute: "focus", description: "(value: SelectInputValue, context: SelectInputFocusContext)", type: "聚焦时触发。详细类型定义。interface SelectInputFocusContext { inputValue: InputValue; tagInputValue?: TagInputValue; e: FocusEvent }"}, {attribute: "input-change", description: "(value: InputValue, context?: SelectInputValueChangeContext)", type: "输入框值发生变化时触发, context.trigger 表示触发输入框值变化的来源: 文本输入触发、清除按钮触发等。详细类型定义。interface SelectInputValueChangeContext { e?: Event | InputEvent | MouseEvent | FocusEvent | KeyboardEvent | CompositionEvent; trigger: 'input' | 'clear' | 'blur' | 'focus' | 'initial' | 'change' }"}, {attribute: "mouseenter", description: "(context: { e: MouseEvent })", type: "进入输入框时触发", {attribute: "mouseleave", description: "(context: { e: MouseEvent })", type: "离开输入框时触发",
{attribute: "paste", description: "(context: { e: ClipboardEvent; pasteValue: string })", type: "粘贴事件, pasteValue 表示粘贴板的内容", {attribute: "popup-visible-change", description: "(visible: boolean, context: PopupVisibleChangeContext)", type: "下拉框显示或隐藏时触发。详细类型定义。import { PopupVisibleChangeContext } from '@Popup'", {attribute: "tag-change", description: "(value: TagInputValue, context: SelectInputChangeContext)", type: "值变化时触发, 参数 context.trigger 表示数据变化的触发来源; context.index 指当前变化项的下标; context.item 指当前变化项; context.e 表示事件参数。详细类型定义。type SelectInputChangeContext = TagInputChangeContext"}]]}], {componentName: "slider", apiData: [{docTitle: "Slider Props说明", tableRows: [{attribute: "disabled", description: "Boolean", type: "false", default: "是否禁用组件",
{attribute: "inputNumberProps", description: "Boolean / Object", type: "false", default: "用于控制数字输入框组件, 值为 false 表示不显示数字输入框; 值为 true 表示呈现默认数字输入框; 值类型为 Object 表示透传属性到数字输入框组件。TS 类型: boolean | InputNumberProps, InputNumber API Documents。详细类型定义", {attribute: "label", description: "String / Boolean / Slot / Function", type: "true", default: "滑块当前值文本。值为 true 显示默认文案; 值为 false 不显示滑块当前值文本; 值为 ${value}% 则表示组件会根据占位符渲染文案; 值类型为函数时, 参数 value 标识滑块值, 参数 position=start 表示范围滑块的起始值, 参数 position=end 表示范围滑块的终点值。TS 类型: string | boolean | TNode<{ value: SliderValue; position?: 'start' | 'end' }>。通用类型定义",
{attribute: "layout", description: "String", type: "horizontal", default: "滑块布局方向。可选项: vertical/horizontal", {attribute: "marks", description: "Object / Array", type: "-", default: "刻度标记, 示例: [0, 10, 40, 200] 或者 { 10: (val) => val + '%', 50: (h) => <button>50</button> }。TS 类型: Array<number> | SliderMarks interface SliderMarks { [mark: number]: string | TNode<{ value: number }> }。通用类型定义。详细类型定义", {attribute: "max", description: "Number", type: "100", default: "滑块范围最大值", {attribute: "min", description: "Number", type: "0", default: "滑块范围最小值",
{attribute: "range", description: "Boolean", type: "false", default: "双游标滑块",
{attribute: "showStep", description: "Boolean", type: "false", default: "控制步长刻度值显示",
{attribute: "step", description: "Number", type: "1", default: "步长",
{attribute: "tooltipProps", description: "Object", type: "-", default: "透传提示组件属性。TS 类型: TooltipProps, Tooltip API Documents。详细类型定义", {attribute: "value", description: "Number / Array", type: "-", default: "滑块值。支持语法糖 v-model 或 v-model:value。TS 类型: SliderValue type SliderValue = number | Array<number>。详细类型定义", {attribute: "defaultValue", description: "Number /
```


Array", type: "-", default: "滑块值。非受控属性。TS 类型: SliderValue type SliderValue = number | Array<number>。详细类型定义"), {attribute: "onChange", description: "Function", type: "", default: "TS 类型: (value: SliderValue) => void 滑块值变化时触发"}], {docTitle: "Slider Events说明", tableRows: [{attribute: "change", description: "(value: SliderValue)", type: "滑块值变化时触发"}]}], {componentName: "switch", apiData: [{docTitle: "Switch Props说明", tableRows: [{attribute: "customValue", description: "Array", type: "-", default: "用于自定义开关的值, [打开时的值, 关闭时的值]。默认为 [true, false]。示例: [1, 0], ['open', 'close']。TS 类型: Array<SwitchValue>"}, {attribute: "disabled", description: "Boolean", type: "-", default: "是否禁用组件"}, {attribute: "label", description: "Array / Slot / Function", type: "[]", default: "开关内容, [开启时内容, 关闭时内容]。示例: ['开', '关'] 或 (value) => value ? '开' : '关'。TS 类型: Array<string | TNode> | TNode<{ value: SwitchValue }>。通用类型定义"}, {attribute: "loading", description: "Boolean", type: "false", default: "是否处于加载中状态"}, {attribute: "size", description: "String", type: "medium", default: "开关尺寸。可选项: small/medium/large"}, {attribute: "value", description: "String / Number / Boolean", type: "-", default: "开关值。支持语法糖 v-model 或 v-model:value。TS 类型: SwitchValue type SwitchValue = string | number | boolean。详细类型定义"}, {attribute: "defaultValue", description: "String / Number / Boolean", type: "-", default: "开关值。非受控属性。TS 类型: SwitchValue type SwitchValue = string | number | boolean。详细类型定义"}, {attribute: "onChange", description: "Function", type: "", default: "TS 类型: (value: SwitchValue) => void 数据发生变化时触发"}]}], {docTitle: "Switch Events说明", tableRows: [{attribute: "change", description: "(value: SwitchValue)", type: "数据发生变化时触发"}]}], {componentName: "tag-input", apiData: [{docTitle: "TagInput Props说明", tableRows: [{attribute: "autoWidth", description: "Boolean", type: "false", default: "宽度随内容自适应"}, {attribute: "clearable", description: "Boolean", type: "false", default: "是否可清空"}, {attribute: "collapsedItems", description: "Slot / Function", type: "-", default: "标签过多的情况下, 折叠项内容, 默认为 +N。如果需要悬浮就显示其他内容, 可以使用 collapsedItems 自定义。value 表示当前存在的所有标签, collapsedTags 表示折叠的标签, count 表示折叠的数量。TS 类型: TNode<{ value: TagInputValue; collapsedTags: TagInputValue; count: number }>。通用类型定义"}, {attribute: "disabled", description: "Boolean", type: "-", default: "是否禁用标签输入框"}, {attribute: "dragSort", description: "Boolean", type: "false", default: "拖拽调整标签顺序"}, {attribute: "excessTagsDisplayType", description: "String", type: "break-line", default: "标签超出时的呈现方式, 有两种: 横向滚动显示 和 换行显示。可选项: scroll/break-line"}, {attribute: "inputProps", description: "Object", type: "-", default: "透传 Input 输入框组件全部属性。TS 类型: InputProps, Input API Documents。详细类型定义"}, {attribute: "inputValue", description: "String / Number", type: "''", default: "输入框的值。支持语法糖 v-model:inputValue。TS 类型: InputValue, Input API Documents。详细类型定义"}, {attribute: "defaultInputValue", description: "String / Number", type: "''", default: "输入框的值。非受控属性。TS 类型: InputValue, Input API Documents。详细类型定义"}, {attribute: "label", description: "String / Slot / Function", type: "-", default: "左侧文本。TS 类型: string | TNode。通用类型定义"}, {attribute: "max", description: "Number", type: "-", default: "最大允许输入的标签数量"}, {attribute: "minCollapsedNum", description: "Number", type: "0", default: "最小折叠数量, 用于标签数量过多的情况下折叠选中项, 超出该数值的选中项折叠。值为 0 则表示不折叠"}, {attribute: "placeholder", description: "String", type: "undefined", default: "占位符"}, {attribute: "readonly", description: "Boolean", type: "false", default: "只读状态, 值为真会隐藏标签移除按钮和输入框"}, {attribute: "size", description: "String", type: "medium", default: "尺寸。可选项: small/medium/large"}, {attribute: "status", description: "String", type: "-", default: "输入框状态。可选项: default/success/warning/error"}, {attribute: "suffix", description: "String / Slot / Function", type: "-", default: "后置图标前的后置内容。TS 类型: string | TNode。通用类型定义"}, {attribute: "suffixIcon", description: "Slot / Function", type: "-", default: "组件后置图标。TS 类型: TNode。通用类型定义"}, {attribute: "tag", description: "String / Slot / Function", type: "-", default: "自定义标签的内部内容, 每个标签的当前值。注意和 valueDisplay 区分, valueDisplay 是用来定义全部标签内容, 而非某一个标签。TS 类型: string | TNode<{ value: string | number }>。通用类型定义"}, {attribute: "tagProps", description: "Object", type: "-", default: "透传 Tag 组件全部属性。TS 类型: TagProps, Tag API Documents。详细类型定义"}, {attribute: "tips", description: "String / Slot / Function", type: "-", default: "输入框下方提示文本, 会根据不同的 status 呈现不同的样式。TS 类型: string | TNode。通用类型定义"}, {attribute: "value", description: "Array", type: "[]", default: "值。支持语法糖 v-model 或 v-model:value。TS 类型: TagInputValue type TagInputValue = Array<string | number>。详细类型定义"}, {attribute: "defaultValue", description: "Array", type: "[]", default: "值。非受控属性。TS 类型: TagInputValue type TagInputValue = Array<string | number>。详细类型定义"}, {attribute: "valueDisplay", description: "String / Slot / Function", type: "-", default: "自定义值呈现的全部内容, 参数为所有标签的值。TS 类型: string | TNode<{ value: TagInputValue; onClose: (index: number, item?: any) => void }>。通用类型定义"}, {attribute: "onBlur", description: "Function", type: "", default: "TS 类型: (value: TagInputValue, context: { inputValue: InputValue; e: FocusEvent }) => void 失去焦点时触发"}, {attribute: "onChange", description: "Function", type: "", default: "TS 类型: (value: TagInputValue, context: TagInputChangeContext) => void 值变化时触发, 参数 context.trigger 表示数据变化的触发来源; context.index 指当前变化项的下标; context.item 指当前变化项; context.e 表示事件参数。详细类型定义。interface TagInputChangeContext { trigger: TagInputTriggerSource; index?: number; item?: string | number; e?: MouseEvent | KeyboardEvent } type TagInputTriggerSource = 'enter' | 'tag-remove' | 'backspace' | 'clear'"}, {attribute: "onClear", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent }) => void 清空按钮点击时触发"},

```

{attribute:"onClick",description:"Function",type:"",default:"TS 类型: (context: { e: MouseEvent }) => void 点击组件时触发"}, {attribute:"onDragSort",description:"Function",type:"",default:"TS 类型: (context: TagInputDragSortContext) => void 【开发中】拖拽排序时触发。详细类型定义。interface TagInputDragSortContext { newTags: TagInputValue; currentIndex: number; current: string | number; targetIndex: number; target: string | number }"}, {attribute:"onEnter",description:"Function",type:"",default:"TS 类型: (value: TagInputValue, context: { e: KeyboardEvent; inputValue: InputValue }) => void 按键按下 Enter 时触发"}, {attribute:"onFocus",description:"Function",type:"",default:"TS 类型: (value: TagInputValue, context: { inputValue: InputValue; e: FocusEvent }) => void 聚焦时触发"}, {attribute:"onInputChange",description:"Function",type:"",default:"TS 类型: (value: InputValue, context?: InputValueChangeContext) => void 输入框值发生变化时触发, context.trigger 表示触发输入框值变化的来源: 文本输入触发、清除按钮触发、回车键触发等。详细类型定义。interface InputValueChangeContext { e?: InputEvent | MouseEvent | CompositionEvent | KeyboardEvent; trigger: 'input' | 'clear' | 'enter' | 'blur' }"}, {attribute:"onMouseenter",description:"Function",type:"",default:"TS 类型: (context: { e: MouseEvent }) => void 进入输入框时触发"}, {attribute:"onMouseleave",description:"Function",type:"",default:"TS 类型: (context: { e: MouseEvent }) => void 离开输入框时触发"}, {attribute:"onPaste",description:"Function",type:"",default:"TS 类型: (context: { e: ClipboardEvent; pasteValue: string }) => void 粘贴事件, pasteValue 表示粘贴板的内容"}, {attribute:"onRemove",description:"Function",type:"",default:"TS 类型: (context: TagInputRemoveContext) => void 移除单个标签时触发。详细类型定义。interface TagInputRemoveContext { value: TagInputValue; index: number; item: string | number; e?: MouseEvent | KeyboardEvent; trigger: TagInputRemoveTrigger } type TagInputRemoveTrigger = 'tag-remove' | 'backspace' }"}, {docTitle:"TagInput Events说明",tableRows:[{attribute:"blur",description:"(value: TagInputValue, context: { inputValue: InputValue; e: FocusEvent })",type:"失去焦点时触发"}, {attribute:"change",description:"(value: TagInputValue, context: TagInputChangeContext)",type:"值变化时触发, 参数 context.trigger 表示数据变化的触发来源; context.index 指当前变化项的下标; context.item 指当前变化项; context.e 表示事件参数。详细类型定义。interface TagInputChangeContext { trigger: TagInputTriggerSource; index?: number; item?: string | number; e?: MouseEvent | KeyboardEvent } type TagInputTriggerSource = 'enter' | 'tag-remove' | 'backspace' | 'clear'"}, {attribute:"clear",description:"(context: { e: MouseEvent })",type:"清空按钮点击时触发"}, {attribute:"click",description:"(context: { e: MouseEvent })",type:"点击组件时触发"}, {attribute:"drag-sort",description:"(context: TagInputDragSortContext)",type:"【开发中】拖拽排序时触发。详细类型定义。interface TagInputDragSortContext { newTags: TagInputValue; currentIndex: number; current: string | number; targetIndex: number; target: string | number }"}, {attribute:"enter",description:"(value: TagInputValue, context: { e: KeyboardEvent; inputValue: InputValue })",type:"按键按下 Enter 时触发"}, {attribute:"focus",description:"(value: TagInputValue, context: { inputValue: InputValue; e: FocusEvent })",type:"聚焦时触发"}, {attribute:"input-change",description:"(value: InputValue, context?: InputValueChangeContext)",type:"输入框值发生变化时触发, context.trigger 表示触发输入框值变化的来源: 文本输入触发、清除按钮触发、回车键触发等。详细类型定义。interface InputValueChangeContext { e?: InputEvent | MouseEvent | CompositionEvent | KeyboardEvent; trigger: 'input' | 'clear' | 'enter' | 'blur' }"}, {attribute:"mouseenter",description:"(context: { e: MouseEvent })",type:"进入输入框时触发"}, {attribute:"mouseleave",description:"(context: { e: MouseEvent })",type:"离开输入框时触发"}, {attribute:"paste",description:"(context: { e: ClipboardEvent; pasteValue: string })",type:"粘贴事件, pasteValue 表示粘贴板的内容"}, {attribute:"remove",description:"(context: TagInputRemoveContext)",type:"移除单个标签时触发。详细类型定义。interface TagInputRemoveContext { value: TagInputValue; index: number; item: string | number; e?: MouseEvent | KeyboardEvent; trigger: TagInputRemoveTrigger } type TagInputRemoveTrigger = 'tag-remove' | 'backspace' }"}], {componentName:"textarea",apiData:[{docTitle:"Textarea Props说明",tableRows:[{attribute:"allowInputOverMax",description:"Boolean",type:"false",default:"超出maxlength或maxcharacter之后是否还允许输入"}, {attribute:"autofocus",description:"Boolean",type:"false",default:"自动聚焦, 拉起键盘"}, {attribute:"autosize",description:"Boolean / Object",type:"false",default:"高度自动撑开。 autosize = true 表示组件高度自动撑开, 同时, 依旧允许手动拖高度。如果设置了 autosize.maxRows 或者 autosize.minRows 则不允许手动调整高度。TS 类型: boolean | { minRows?: number; maxRows?: number }"}, {attribute:"disabled",description:"Boolean",type:"false",default:"是否禁用文本框"}, {attribute:"label",description:"String / Slot / Function",type:"-",default:"左侧文本。TS 类型: string | TNode。通用类型定义"}, {attribute:"maxcharacter",description:"Number",type:"-",default:"用户最多可以输入的字符个数, 一个中文汉字表示两个字符长度"}, {attribute:"maxlength",description:"Number",type:"-",default:"用户最多可以输入的字符个数"}, {attribute:"name",description:"String",type:"-",default:"名称, HTML 元素原生属性"}, {attribute:"placeholder",description:"String",type:"undefined",default:"占位符"}, {attribute:"readonly",description:"Boolean",type:"false",default:"只读状态"}, {attribute:"status",description:"String",type:"-",default:"文本框状态。可选项: default/success/warning/error"}, {attribute:"tips",description:"String / Slot / Function",type:"-",default:"输入框下方提示文本, 会根据不同的 status 呈现不同的样式。TS 类型: string | TNode。通用类型定义"}, {attribute:"value",description:"String / Number",type:"-",default:"文本框值。支持语法糖 v-model 或 v-model:value。TS 类型: TextareaValue type TextareaValue = string。详细类型定义"}, {attribute:"defaultValue",description:"String / Number",type:"-",default:"文本框值。非受控属性。TS

```

类型: TextareaValue type TextareaValue = string. 详细类型定义",
 {attribute:"onBlur",description:"Function",type:"",default:"TS 类型: (value: TextareaValue, context: { e: FocusEvent }) => void失去焦点时触发"},
 {attribute:"onChange",description:"Function",type:"",default:"TS 类型: (value: TextareaValue, context?: { e?: InputEvent }) => void输入内容变化时触发"},
 {attribute:"onFocus",description:"Function",type:"",default:"TS 类型: (value: TextareaValue, context: { e: FocusEvent }) => void获得焦点时触发"},
 {attribute:"onKeyDown",description:"Function",type:"",default:"TS 类型: (value: TextareaValue, context: { e: KeyboardEvent }) => void键盘按下时触发"},
 {attribute:"onKeyPress",description:"Function",type:"",default:"TS 类型: (value: TextareaValue, context: { e: KeyboardEvent }) => void按下字符键时触发 (keydown -> keypress -> keyup) "},
 {attribute:"onKeyUp",description:"Function",type:"",default:"TS 类型: (value: TextareaValue, context: { e: KeyboardEvent }) => void释放键盘时触发"}], {docTitle:"Textarea Events说明",tableRows: [{attribute:"blur",description:"(value: TextareaValue, context: { e: FocusEvent })",type:"失去焦点时触发"}, {attribute:"change",description:"(value: TextareaValue, context?: { e?: InputEvent })",type:"输入内容变化时触发"}, {attribute:"focus",description:"(value: TextareaValue, context: { e: FocusEvent })",type:"获得焦点时触发"}, {attribute:"keydown",description:"(value: TextareaValue, context: { e: KeyboardEvent })",type:"键盘按下时触发"}, {attribute:"keypress",description:"(value: TextareaValue, context: { e: KeyboardEvent })",type:"按下字符键时触发 (keydown -> keypress -> keyup) "}, {attribute:"keyup",description:"(value: TextareaValue, context: { e: KeyboardEvent })",type:"释放键盘时触发"}]}, {componentName:"transfer",apiData:[{docTitle:"Transfer Props说明",tableRows:[{attribute:"checkboxProps",description:"Object",type:"-","default:"用于控制复选框属性。TS 类型: CheckboxProps, Checkbox API Documents. 详细类型定义"}, {attribute:"checked",description:"Array",type:"[]",default:"数据列表选中项。支持语法糖 v-model:checked。TS 类型: Array<TransferValue>"}, {attribute:"defaultChecked",description:"Array",type:"[]",default:"数据列表选中项。非受控属性。TS 类型: Array<TransferValue>"}, {attribute:"data",description:"Array",type:"[]",default:"全量数据。TS 类型: Array<T>"}, {attribute:"direction",description:"String",type:"both",default:"穿梭框可操作方向。可选项: left/right/both"}, {attribute:"disabled",description:"Boolean / Array",type:"false",default:"禁用全部操作: 搜索、选中、移动、分页等。[源列表, 目标列表], 示例: [true, false] 或者 true。TS 类型: boolean | Array<boolean>"}, {attribute:"empty",description:"String / Array / Slot / Function",type:"","",default:"列表为空时呈现的内容。值类型为数组, 则表示分别控制源列表和目标列表数据为空的呈现内容。TS 类型: EmptyType | Array<EmptyType> | TNode type EmptyType = string | TNode。通用类型定义。详细类型定义"}, {attribute:"footer",description:"Array / Slot / Function",type:"-","default:"穿梭框底部内容。TS 类型: Array<string | TNode> | TNode<{ type: TransferListType }>。通用类型定义"}, {attribute:"keys",description:"Object",type:"-","default:"用来定义选项文本和选项值字段, 示例: { label: 'text', value: 'id' }, 表示选项文本取 text 字段, 选项值取 id 字段。TS 类型: KeysType。通用类型定义"}, {attribute:"operation",description:"Array / Slot / Function",type:"-","default:"方向操作按钮。默认显示组件内置操作图标。自定义操作图标示例: ['向左', '向右'] 或者 [() => <i class='left' />, () => <i class='right' />] 或者 (h, direction) => direction === 'left' ? '《' : '》'。TS 类型: Array<string | TNode> | TNode<{ direction: 'left' | 'right' }>。通用类型定义"}, {attribute:"pagination",description:"Object / Array",type:"-","default:"分页配置, 值为空则不显示。具体 API 参考分页组件。值类型为数组, 表示可分别控制源列表和目标列表分页组件。TS 类型: PaginationProps | Array<PaginationProps>, Pagination API Documents. 详细类型定义"}, {attribute:"search",description:"Boolean / Object / Array",type:"false",default:"搜索框配置, 值为 false 表示不显示搜索框; 值为 true 表示显示默认搜索框; 值类型为对象, 用于透传 Props 到 Input 组件; 值类型为数组, 则分别表示控制两侧搜索框。TS 类型: SearchOption | Array<SearchOption> type SearchOption = boolean | InputProps, Input API Documents. 详细类型定义"}, {attribute:"showCheckAll",description:"Boolean / Array",type:"true",default:"是否显示全选, 值类型为数组则表示分别控制源列表和目标列表。TS 类型: boolean | Array<boolean>"}, {attribute:"targetSort",description:"String",type:"original",default:"目标数据列表排列顺序。可选项: original/push/unshift"}, {attribute:"title",description:"Array / Slot / Function",type:"[]",default:"穿梭框标题, 示例: ['源列表', '目标列表'] 或者 [() => 'A', () => 'B'] 或者 ({ type }) => type === 'source' ? '源' : '目标'。TS 类型: Array<TitleType> | TNode<{ type: TransferListType }> type TitleType = string | TNode type TransferListType = 'source' | 'target'。通用类型定义。详细类型定义"}, {attribute:"transferItem",description:"Slot / Function",type:"-","default:"自定义渲染节点。TS 类型: TNode<TransferItem<T>> interface TransferItem<T extends DataOption = DataOption> { data: T; index: number; type: TransferListType}。通用类型定义。详细类型定义"}, {attribute:"tree",description:"Slot / Function",type:"传入 Tree 组件定义树形结构",default:"TS 类型: (tree: TreeProps) => TNode, Tree API Documents. 通用类型定义。详细类型定义"}, {attribute:"value",description:"Array",type:"[]",default:"目标数据列表数据。支持语法糖 v-model 或 v-model:value。TS 类型: Array<TransferValue>"}, {attribute:"defaultValue",description:"Array",type:"[]",default:"目标数据列表数据。非受控属性。TS 类型: Array<TransferValue>"}, {attribute:"onChange",description:"Function",type:"",default:"TS 类型: (targetValue: Array<TransferValue>, context: TargetParams) => void数据列表发生变化时触发, type 值为 source, 表示源列表移动到目标列表, 值为 target 表示目标列表移动到源列表, movedValue 则表示被移动的选项。TS 类型: TNode。通用类型定义。详细类型定义。interface TargetParams { type: TransferListType; movedValue: Array<TransferValue> }"}, {attribute:"onCheckedChange",description:"Function",type:"",default:"TS 类型: (options: CheckedOptions) => void源数据列表或目标数据列表的选中项发生变化时触发, context.type 可

以区分触发来源是目标列表，还是源列表。TS 类型：TNode。通用类型定义。详细类型定义。interface CheckedOptions { checked: Array<TransferValue>; sourceChecked: Array<TransferValue>; targetChecked: Array<TransferValue>; type: TransferListType }”，

{attribute: "onPageChange", description: "Function", type: "", default: "TS 类型：(page: PageInfo, context: { type: TransferListType }) => void 分页发生变化时触发”，

{attribute: "onScroll", description: "Function", type: "", default: "TS 类型：(options: { e: Event; bottomDistance: number; type: TransferListType }) => void 列表滚动时触发，bottomDistance 表示元素滚动到底部的距离。TS 类型：TNode。通用类型定义”，

{attribute: "onSearch", description: "Function", type: "", default: "TS 类型：(options: SearchContext) => void 搜索时触发，options.query 表示用户输入的内容。TS 类型：TNode。通用类型定义。详细类型定义。interface SearchContext { query: string; type: TransferListType; trigger: 'input' | 'enter'; e: InputEvent | KeyboardEvent }”}}，{docTitle: "Transfer Events 说明", tableRows: [{attribute: "change", description: "(targetValue: Array<TransferValue>, context: TargetParams)", type: "数据列表发生变化时触发，type 值为 source，表示源列表移动到目标列表，值为 target 表示目标列表移动到源列表，movedValue 则表示被移动的选项。TS 类型：TNode。通用类型定义。详细类型定义。interface TargetParams { type: TransferListType; movedValue: Array<TransferValue> }”，

{attribute: "checked-change", description: "(options: CheckedOptions)", type: "源数据列表或目标数据列表的选中项发生变化时触发，context.type 可以区分触发来源是目标列表，还是源列表。TS 类型：TNode。通用类型定义。详细类型定义。interface CheckedOptions { checked: Array<TransferValue>; sourceChecked: Array<TransferValue>; targetChecked: Array<TransferValue>; type: TransferListType }”，

{attribute: "page-change", description: "(page: PageInfo, context: { type: TransferListType })”，type: "分页发生变化时触发”，{attribute: "scroll", description: "(options: { e: Event; bottomDistance: number; type: TransferListType })”，type: "列表滚动时触发，bottomDistance 表示元素滚动到底部的距离。TS 类型：TNode。通用类型定义”，{attribute: "search", description: "(options: SearchContext)", type: "搜索时触发，options.query 表示用户输入的内容。TS 类型：TNode。通用类型定义。详细类型定义。interface SearchContext { query: string; type: TransferListType; trigger: 'input' | 'enter'; e: InputEvent | KeyboardEvent }”}}}}，{componentName: "time-picker", apiData: [{docTitle: "TimePicker Props 说明", tableRows: [{attribute: "allowInput", description: "Boolean", type: "false", default: "是否允许直接输入时间”，

{attribute: "clearable", description: "Boolean", type: "false", default: "是否允许清除选中值”，

{attribute: "disableTime", description: "Function", type: "-", default: "禁用时间项。TS 类型：(h: number, m: number, s: number) => Partial<{ hour: Array<number>, minute: Array<number>, second: Array<number> }>”，

{attribute: "disabled", description: "Boolean", type: "false", default: "是否禁用组件”，

{attribute: "format", description: "String", type: "HH:mm:ss", default: "用于格式化时间，详细文档”，

{attribute: "hideDisabledTime", description: "Boolean", type: "true", default: "是否隐藏禁用状态的时间项”，

{attribute: "inputProps", description: "Object", type: "-", default: "透传给输入框（Input）组件的参数。TS 类型：InputProps, Input API Documents。详细类型定义”，

{attribute: "placeholder", description: "String", type: "undefined", default: "占位符”，

{attribute: "popupProps", description: "Object", type: "-", default: "透传给 popup 组件的参数。TS 类型：PopupProps, Popup API Documents。详细类型定义”，{attribute: "presets", description: "Object", type: "-", default: "预设快捷时间选择，示例：{ '前一小时': '11:00:00' }。TS 类型：PresetTime interface PresetTime { [presetName: string]: TimePickerValue | (() => TimePickerValue) }。详细类型定义”，

{attribute: "size", description: "String", type: "medium", default: "尺寸。可选项：small/medium/large”，

{attribute: "status", description: "String", type: "-", default: "输入框状态。可选项：default/success/warning/error”，{attribute: "steps", description: "Array", type: "[1, 1, 1]", default: "时间间隔步数，数组排列 [小时, 分钟, 秒]，示例：[2, 1, 1] 或者 ['2', '1', '1']。TS 类型：Array<string | number>”，{attribute: "tips", description: "String / Slot / Function", type: "-", default: "输入框下方提示文本，会根据不同的 status 呈现不同的样式。TS 类型：string | TNode。通用类型定义”，

{attribute: "value", description: "String", type: "-", default: "选中值。支持语法糖 v-model 或 v-model:value。TS 类型：TimePickerValue type TimePickerValue = string。详细类型定义”，

{attribute: "defaultValue", description: "String", type: "-", default: "选中值。非受控属性。TS 类型：TimePickerValue type TimePickerValue = string。详细类型定义”，

{attribute: "onBlur", description: "Function", type: "", default: "TS 类型：(context: { value: TimePickerValue; e: FocusEvent }) => void 当输入框失去焦点时触发，value 表示组件当前有效值”，

{attribute: "onChange", description: "Function", type: "", default: "TS 类型：(value: TimePickerValue) => void 选中值发生变化时触发”，{attribute: "onClose", description: "Function", type: "", default: "TS 类型：(context: { e: MouseEvent }) => void 面板关闭时触发”，

{attribute: "onFocus", description: "Function", type: "", default: "TS 类型：(context: { value: TimePickerValue; e: FocusEvent }) => void 输入框获得焦点时触发，value 表示组件当前有效值”，

{attribute: "onInput", description: "Function", type: "", default: "TS 类型：(context: { value: TimePickerValue; e: InputEvent }) => void 当输入框内容发生变化时触发，参数 value 表示组件当前有效值”，{attribute: "onOpen", description: "Function", type: "", default: "TS 类型：(context: { e: MouseEvent }) => void 面板打开时触发”，{attribute: "onPick", description: "Function", type: "", default: "TS 类型：(value: TimePickerValue, context: { e: MouseEvent }) => void 面板选中值后触发”}}，

{docTitle: "TimePicker Events 说明", tableRows: [{attribute: "blur", description: "(context: { value: TimePickerValue; e: FocusEvent })”，type: "当输入框失去焦点时触发，value 表示组件当前有效值”，

{attribute: "change", description: "(value: TimePickerValue)", type: "选中值发生变化时触发”，

{attribute: "close", description: "(context: { e: MouseEvent })”，type: "面板关闭时触发”，

{attribute: "focus", description: "(context: { value: TimePickerValue; e: FocusEvent })”，type: "输入框获

得焦点时触发, value 表示组件当前有效值"}, {attribute: "input", description: "(context: { value: TimePickerValue; e: InputEvent })", type: "当输入框内容发生变化时触发, 参数 value 表示组件当前有效值"}, {attribute: "open", description: "(context: { e: MouseEvent })", type: "面板打开时触发"}, {attribute: "pick", description: "(value: TimePickerValue, context: { e: MouseEvent })", type: "面板选中值后触发"}], {docTitle: "TimeRangePicker Props说明", tableRows: [{attribute: "allowInput", description: "Boolean", type: "false", default: "是否允许直接输入时间"}, {attribute: "clearable", description: "Boolean", type: "false", default: "是否允许清除选中值"}, {attribute: "disableTime", description: "Function", type: "-", default: "禁用时间项。TS 类型: (h: number, m: number, s: number, context: { partial: TimeRangePickerPartial }) => Partial<{ hour: Array<number>, minute: Array<number>, second: Array<number>}> type TimeRangePickerPartial = 'start' | 'end'。详细类型定义"}, {attribute: "disabled", description: "Boolean / Array", type: "false", default: "是否禁用组件, 值为数组表示可分别控制开始日期和结束日期是否禁用。TS 类型: boolean | Array<boolean>"}, {attribute: "format", description: "String", type: "HH:mm:ss", default: "用于格式化时间, 详细文档"}, {attribute: "hideDisabledTime", description: "Boolean", type: "true", default: "是否隐藏禁用状态的时间项"}, {attribute: "placeholder", description: "String / Array", type: "undefined", default: "占位符, 值为数组表示可分别为开始日期和结束日期设置占位符。TS 类型: string | Array<string>"}, {attribute: "popupProps", description: "Object", type: "-", default: "透传给 popup 组件的参数。TS 类型: PopupProps, Popup API Documents。详细类型定义"}, {attribute: "presets", description: "Object", type: "-", default: "预设快捷时间范围选择, 示例: { '下午': ['13:00:00', '18:00:00'] }。TS 类型: PresetTimeRange interface PresetTimeRange { [presetRangeName: string]: TimeRangeValue | (() => TimeRangeValue)。详细类型定义"}, {attribute: "rangeInputProps", description: "Object", type: "-", default: "透传给范围输入框 RangeInput 组件的参数。TS 类型: RangeInputProps, RangeInput API Documents。详细类型定义"}, {attribute: "size", description: "String", type: "medium", default: "尺寸。可选项: small/medium/large"}, {attribute: "status", description: "String", type: "-", default: "输入框状态。可选项: default/success/warning/error"}, {attribute: "steps", description: "Array", type: "[1, 1, 1]", default: "时间间隔步数, 数组排列 [小时, 分钟, 秒], 示例: [2, 1, 1] 或者 ['2', '1', '1']。TS 类型: Array<string | number>"}, {attribute: "tips", description: "String / Slot / Function", type: "-", default: "输入框下方提示文本, 会根据不同的 status 呈现不同的样式。TS 类型: string | TNode。通用类型定义"}, {attribute: "value", description: "Array", type: "-", default: "选中值。支持语法糖 v-model 或 v-model:value。TS 类型: TimeRangeValue type TimeRangeValue = Array<string>。详细类型定义"}, {attribute: "defaultValue", description: "Array", type: "-", default: "选中值。非受控属性。TS 类型: TimeRangeValue type TimeRangeValue = Array<string>。详细类型定义"}, {attribute: "onBlur", description: "Function", type: "", default: "TS 类型: (context: { value: TimeRangeValue; e?: FocusEvent; position?: TimeRangePickerPartial }) => void当输入框失去焦点时触发。详细类型定义。type TimeRangePickerPartial = 'start' | 'end'"}, {attribute: "onChange", description: "Function", type: "", default: "TS 类型: (value: TimeRangeValue) => void选中值发生变化时触发"}, {attribute: "onFocus", description: "Function", type: "", default: "TS 类型: (context?: { value: TimeRangeValue; e?: FocusEvent; position?: TimeRangePickerPartial }) => void范围输入框获得焦点时触发。详细类型定义。type TimeRangePickerPartial = 'start' | 'end'"}, {attribute: "onInput", description: "Function", type: "", default: "TS 类型: (context: { value: TimeRangeValue; e?: InputEvent; position?: TimeRangePickerPartial }) => void当输入框内容发生变化时触发, 参数 input 表示输入内容, value 表示组件当前有效值。详细类型定义。type TimeRangePickerPartial = 'start' | 'end'"}, {attribute: "onPick", description: "Function", type: "", default: "TS 类型: (value: TimeRangeValue, context: { e: MouseEvent, position?: TimeRangePickerPartial }) => void面板选中值后触发"}], {docTitle: "TimeRangePicker Events说明", tableRows: [{attribute: "blur", description: "(context: { value: TimeRangeValue; e?: FocusEvent; position?: TimeRangePickerPartial })", type: "当输入框失去焦点时触发。详细类型定义。type TimeRangePickerPartial = 'start' | 'end'"}, {attribute: "change", description: "(value: TimeRangeValue)", type: "选中值发生变化时触发"}, {attribute: "focus", description: "(context?: { value: TimeRangeValue; e?: FocusEvent; position?: TimeRangePickerPartial })", type: "范围输入框获得焦点时触发。详细类型定义。type TimeRangePickerPartial = 'start' | 'end'"}, {attribute: "input", description: "(context: { value: TimeRangeValue; e?: InputEvent; position?: TimeRangePickerPartial })", type: "当输入框内容发生变化时触发, 参数 input 表示输入内容, value 表示组件当前有效值。详细类型定义。type TimeRangePickerPartial = 'start' | 'end'"}, {attribute: "pick", description: "(value: TimeRangeValue, context: { e: MouseEvent, position?: TimeRangePickerPartial })", type: "面板选中值后触发"}], {componentName: "tree-select", apiData: [{docTitle: "TreeSelect Props说明", tableRows: [{attribute: "autoWidth", description: "Boolean", type: "false", default: "宽度随内容自适应"}, {attribute: "borderless", description: "Boolean", type: "false", default: "【开发中】无边框模式"}, {attribute: "clearable", description: "Boolean", type: "false", default: "是否允许清空"}, {attribute: "collapsedItems", description: "Slot / Function", type: "-", default: "多选情况下, 用于设置折叠项内容, 默认为 +N。如果需要悬浮就显示其他内容, 可以使用 collapsedItems 自定义。TS 类型: TNode<{ value: DataOption[]; collapsedSelectedItems: DataOption[]; count: number }>。通用类型定义"}, {attribute: "data", description: "Array", type: "[[]]", default: "数据。TS 类型: Array<DataOption>"}, {attribute: "disabled", description: "Boolean", type: "-", default: "是否禁用组件"}, {attribute: "empty", description: "String / Slot / Function", type: "", default: "当下拉列表为空时显示的内容。TS 类型: string | TNode。通用类型定义"}, {attribute: "filter", description: "Function", type: "-", default: "过滤方法, 用于对现有数据进行搜索过滤, 判断是否过滤某一项数据。TS 类型: (filterWords: string, option: DataOption) => boolean"}, {attribute: "filterable", description: "Boolean", type: "false", default: "是否可搜索"}],

{attribute: "inputProps", description: "Object", type: "-", default: "透传给 Input 组件的全部属性。TS 类型: InputProps, Input API Documents。详细类型定义"}, {attribute: "inputValue", description: "String / Number", type: "-", default: "输入框的值。支持语法糖 v-model:inputValue。TS 类型: InputValue, Input API Documents。详细类型定义"}, {attribute: "defaultInputValue", description: "String / Number", type: "-", default: "输入框的值。非受控属性。TS 类型: InputValue, Input API Documents。详细类型定义"}, {attribute: "loading", description: "Boolean", type: "false", default: "是否正在加载数据"}, {attribute: "loadingText", description: "String / Slot / Function", type: "-", default: "远程加载时显示的文字，支持自定义。如加上超链接。TS 类型: string | TNode。通用类型定义"}, {attribute: "max", description: "Number", type: "0", default: "用于控制多选数量，值为 0 则不限制"}, {attribute: "minCollapsedNum", description: "Number", type: "0", default: "最小折叠数量，用于多选情况下折叠选中项，超出该数值的选中项折叠。值为 0 则表示不折叠"}, {attribute: "multiple", description: "Boolean", type: "false", default: "是否允许多选"}, {attribute: "placeholder", description: "String", type: "undefined", default: "占位符"}, {attribute: "popupProps", description: "Object", type: "-", default: "透传给 popup 组件的全部属性。TS 类型: PopupProps, Popup API Documents。详细类型定义"}, {attribute: "popupVisible", description: "Boolean", type: "-", default: "是否显示下拉框"}, {attribute: "prefixIcon", description: "Slot / Function", type: "-", default: "组件前置图标。TS 类型: TNode。通用类型定义"}, {attribute: "readonly", description: "Boolean", type: "false", default: "只读状态，值为真会隐藏输入框，且无法打开下拉框"}, {attribute: "selectInputProps", description: "Object", type: "-", default: "【开发中】透传 SelectInput 筛选器输入框组件的全部属性。TS 类型: SelectInputProps, SelectInput API Documents。详细类型定义"}, {attribute: "size", description: "String", type: "medium", default: "尺寸。可选项: small/medium/large"}, {attribute: "status", description: "String", type: "-", default: "输入框状态。可选项: default/success/warning/error"}, {attribute: "tagProps", description: "Object", type: "-", default: "【开发中】透传 Tag 标签组件全部属性。TS 类型: TagProps, Tag API Documents。详细类型定义"}, {attribute: "tips", description: "String / Slot / Function", type: "-", default: "输入框下方提示文本，会根据不同的 status 呈现不同的样式。TS 类型: string | TNode。通用类型定义"}, {attribute: "treeProps", description: "Object", type: "-", default: "透传 Tree 组件的全部属性。TS 类型: TreeProps, Tree API Documents。详细类型定义"}, {attribute: "value", description: "String / Number / Object / Array", type: "-", default: "选中值。支持语法糖 v-model 或 v-model:value。TS 类型: TreeSelectValue type TreeSelectValue = string | number | object | Array<TreeSelectValue>。详细类型定义"}, {attribute: "defaultValue", description: "String / Number / Object / Array", type: "-", default: "选中值。非受控属性。TS 类型: TreeSelectValue type TreeSelectValue = string | number | object | Array<TreeSelectValue>。详细类型定义"}, {attribute: "valueDisplay", description: "Slot / Function", type: "-", default: "自定义选中项呈现方式。TS 类型: string | TNode<{ value: TreeSelectValue; onClose: (index: number, item?: any) => void }>。通用类型定义"}, {attribute: "valueType", description: "String", type: "value", default: "用于控制选中值的类型。假设数据选项为: [{ label: '姓名', value: 'name' }], value 表示值仅返回数据选项中的 value，object 表示值返回全部数据。可选项: value/object"}, {attribute: "onBlur", description: "Function", type: "", default: "TS 类型: (context: { value: TreeSelectValue; e: FocusEvent }) => void 输入框失去焦点时触发"}, {attribute: "onChange", description: "Function", type: "", default: "TS 类型: (value: TreeSelectValue, context: { node: TreeNodeModel<DataOption>; trigger: TreeSelectValueChangeTrigger; e?: MouseEvent | KeyboardEvent }) => void 节点选中状态变化时触发，context.node 表示当前变化的选项，context.trigger 表示触发变化的来源。详细类型定义。type TreeSelectValueChangeTrigger = 'clear' | 'tag-remove' | 'backspace' | 'check' | 'uncheck'"}, {attribute: "onClear", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent }) => void 点击清除按钮时触发"}, {attribute: "onFocus", description: "Function", type: "", default: "TS 类型: (context: { value: TreeSelectValue; e: FocusEvent }) => void 输入框获得焦点时触发"}, {attribute: "onInputChange", description: "Function", type: "", default: "TS 类型: (value: InputValue, context?: SelectInputValueChangeContext) => void 输入框值发生变化时触发，context.trigger 表示触发输入框值变化的来源: 文本输入触发、清除按钮触发、失去焦点等。详细类型定义。import { SelectInputValueChangeContext } from '@SelectInput'"}, {attribute: "onPopupVisibleChange", description: "Function", type: "", default: "TS 类型: (visible: boolean, context: PopupVisibleChangeContext) => void 下拉框显示或隐藏时触发。详细类型定义。import { PopupVisibleChangeContext } from '@Popup'"}, {attribute: "onRemove", description: "Function", type: "", default: "TS 类型: (options: RemoveOptions<DataOption>) => void 多选模式下，选中数据被移除时触发。详细类型定义。interface RemoveOptions<T> { value: string | number | object; data: T; e?: MouseEvent }"}, {attribute: "onSearch", description: "Function", type: "", default: "TS 类型: (filterWords: string) => void 输入值变化时，触发搜索事件。主要用于远程搜索新数据"}], {docTitle: "TreeSelect Events说明", tableRows: [{attribute: "blur", description: "(context: { value: TreeSelectValue; e: FocusEvent })", type: "输入框失去焦点时触发"}, {attribute: "change", description: "(value: TreeSelectValue, context: { node: TreeNodeModel<DataOption>; trigger: TreeSelectValueChangeTrigger; e?: MouseEvent | KeyboardEvent })", type: "节点选中状态变化时触发，context.node 表示当前变化的选项，context.trigger 表示触发变化的来源。详细类型定义。type TreeSelectValueChangeTrigger = 'clear' | 'tag-remove' | 'backspace' | 'check' | 'uncheck'"}, {attribute: "clear", description: "(context: { e: MouseEvent })", type: "点击清除按钮时触发"}, {attribute: "focus", description: "(context: { value: TreeSelectValue; e: FocusEvent })", type: "输入框获得焦点时触发"}, {attribute: "input-change", description: "(value: InputValue, context?: SelectInputValueChangeContext)", type: "输入框值发生变化时触发，context.trigger 表示触发输入框值变化的"}]

来源：文本输入触发、清除按钮触发、失去焦点等。详细类型定义。import { SelectInputValueChangeContext } from 'SelectInput'"}, {attribute: "popup-visible-change", description: "(visible: boolean, context: PopupVisibleChangeContext)", type: "下拉框显示或隐藏时触发。详细类型定义。import { PopupVisibleChangeContext } from '@Popup'"}, {attribute: "remove", description: "(options: RemoveOptions<DataOption>)", type: "多选模式下，选中数据被移除时触发。详细类型定义。interface RemoveOptions<T> { value: string | number | object; data: T; e?: MouseEvent }"}, {attribute: "search", description: "(filterWords: string)", type: "输入值变化时，触发搜索事件。主要用于远程搜索新数据"}]]], {componentName: "upload", apiData: [{docTitle: "Upload Props说明", tableRows: [{attribute: "abridgeName", description: "Array", type: "-", default: "文件名过长时，需要省略中间的文本，保留首尾文本。示例：[10, 7]，表示首尾分别保留的文本长度。TS 类型：Array<number>"}, {attribute: "accept", description: "String", type: "-", default: "接受上传的文件类型，查看 W3C示例，查看 MDN 示例"}, {attribute: "action", description: "String", type: "-", default: "上传接口。设接口响应数据为字段 response，那么 response.error 存在时会判断此次上传失败，并显示错误文本信息；response.url 会作为文件上传成功后的地址，并使用该地址显示图片或文件"}, {attribute: "allowUploadDuplicateFile", description: "Boolean", type: "false", default: "是否允许重复上传相同文件名的文件"}, {attribute: "autoUpload", description: "Boolean", type: "true", default: "是否在选择文件后自动发起请求上传文件"}, {attribute: "beforeAllFilesUpload", description: "Function", type: "-", default: "如果是自动上传模式 autoUpload=true，表示全部文件上传之前的钩子函数，函数参数为上传的文件，函数返回值决定是否继续上传，若返回值为 false 则终止上传。如果是非自动上传模式 autoUpload=false，则函数返回值为 false 时表示不触发文件变化。TS 类型：(file: UploadFile[]) => boolean | Promise<boolean>"}, {attribute: "beforeUpload", description: "Function", type: "-", default: "如果是自动上传模式 autoUpload=true，表示单个文件上传之前的钩子函数，若函数返回值为 false 则表示不上传当前文件。如果是非自动上传模式 autoUpload=false，函数返回值为 false 时表示从上传文件中剔除当前文件。TS 类型：(file: UploadFile) => boolean | Promise<boolean>"}, {attribute: "data", description: "Object", type: "-", default: "上传请求所需的额外字段，默认字段有 file，表示文件信息。可以添加额外的文件名字段，如：{file_name: "custom-file-name.txt"}。autoUpload=true 时有效。也可以使用 formatRequest 完全自定义上传请求的字段。TS 类型：Record<string, any> | ((files: UploadFile[]) => Record<string, any>)"}, {attribute: "default", description: "String / Slot / Function", type: "-", default: "非拖拽场景，指触发上传的元素，如：“选择文件”。如果是拖拽场景，则是指拖拽区域。TS 类型：string | TNode。通用类型定义"}, {attribute: "disabled", description: "Boolean", type: "-", default: "是否禁用"}, {attribute: "dragContent", description: "String / Slot / Function", type: "-", default: "用于自定义拖拽区域，theme=custom 且 draggable=true 时有效。TS 类型：TNode | TNode<TriggerContext>。通用类型定义"}, {attribute: "draggable", description: "Boolean", type: "undefined", default: "是否启用拖拽上传，不同的组件风格默认值不同。theme=file 或 theme=image 时有效"}, {attribute: "fileListDisplay", description: "Slot / Function", type: "-", default: "用于完全自定义文件列表内容。TS 类型：TNode<{ files: UploadFile[]; dragEvents?: UploadDisplayDragEvents }>。通用类型定义"}, {attribute: "files", description: "Array", type: "[]", default: "已上传文件列表，同 value。TS 类型：UploadFile。支持语法糖 v-model:files。TS 类型：Array<T>"}, {attribute: "defaultFiles", description: "Array", type: "[]", default: "已上传文件列表，同 value。TS 类型：UploadFile。非受控属性。TS 类型：Array<T>"}, {attribute: "format", description: "Function", type: "-", default: "转换文件 UploadFile 的数据结构，可新增或修改 UploadFile 的属性，注意不能删除 UploadFile 属性。action 存在时有效。TS 类型：(file: File) => UploadFile"}, {attribute: "formatRequest", description: "Function", type: "-", default: "用于新增或修改文件上传请求参数。action 存在时有效。一个请求上传一个文件时，默认请求字段有 file；一个请求上传多个文件时，默认字段有 file[0]/file[1]/file[2]/.../length，其中 length 表示本次上传的文件数量。⚠️ 非常注意，此处的 file[0]/file[1] 仅仅是一个字段名，并非表示 file 是一个数组，接口获取字段时注意区分。可以使用 name 定义 file 字段的别名，也可以使用 formatRequest 自定义任意字段。TS 类型：(requestData: { [key: string]: any }) => { [key: string]: any }"}, {attribute: "formatResponse", description: "Function", type: "-", default: "用于格式化文件上传后的接口响应数据，response 便是接口响应的原始数据。action 存在时有效。此函数的返回值 error 或 response.error 会作为错误文本提醒，如果存在会判定为本次上传失败。此函数的返回值 url 或 response.url 会作为上传成功后的链接。TS 类型：(response: any, context: FormatResponseContext) => ResponseType type ResponseType = { error?: string; url?: string } & Record<string, any> interface FormatResponseContext { file: UploadFile; currentFiles?: UploadFile[] }。详细类型定义"}, {attribute: "headers", description: "Object", type: "-", default: "设置上传的请求头部，action 存在时有效。TS 类型：{[key: string]: string}"}, {attribute: "inputAttributes", description: "Object", type: "-", default: "用于添加属性到 HTML 元素 input。TS 类型：CSSProperties"}, {attribute: "isBatchUpload", description: "Boolean", type: "false", default: "多个文件是否作为一个独立文件包，整体替换，整体删除。不允许追加文件，只允许替换文件。theme=file-flow 时有效"}, {attribute: "locale", description: "Object", type: "-", default: "上传组件文本语言配置，支持自定义配置组件中的全部文本。优先级高于全局配置中语言。TS 类型：UploadConfig import { UploadConfig } from '../config-provider/type'。详细类型定义"}, {attribute: "max", description: "Number", type: "0", default: "用于控制文件上传数量，值为 0 则不限制"}, {attribute: "method", description: "String", type: "POST", default: "HTTP 请求类型。可选项：POST/GET/PUT/OPTION/PATCH/post/get/put/option/patch"}, {attribute: "mockProgressDuration", description: "Number", type: "-", default: "模拟进度间隔时间，单位：毫秒，默认：300。由于原始上传请求，小文件上传进度只有 0 和 100，故而新增模拟进度，每间隔 mockProgressDuration 毫秒刷新一次模拟进度。小文件设置小一点，大文件设置大一点。注意：当 useMockProgress 为真时，当前设置有效"}, {attribute: "multiple", description: "Boolean", type: "false", default: "支持多文件上传"}],

```

{attribute:"name",description:"String",type:"file",default:"文件上传时的名称"},
{attribute:"placeholder",description:"String",type:"-",default:"占位符"},
{attribute:"requestMethod",description:"Function",type:"-",default:"自定义上传方法。返回值 status 表示上传成功或失败; error 或 response.error 表示上传失败的原因; response 表示请求上传成功后的返回数据, response.url 表示上传成功后的图片/文件地址, response.files 表示一个请求上传多个文件/图片后的返回值。示例一: { status: 'fail', error: '上传失败', response }。示例二: { status: 'success', response: { url: 'https://tdesign.gtimg.com/site/avatar.jpg' } }。示例三: { status: 'success', files: [{ url: 'https://xxx.png', name: 'xxx.png' }] }。TS 类型: (files: UploadFile | UploadFile[]) => Promise<RequestMethodResponse> interface RequestMethodResponse { status: 'success' | 'fail'; error?: string; response: { url?: string; files?: UploadFile[]; [key: string]: any } }。详细类型定义",
{attribute:"showUploadProgress",description:"Boolean",type:"true",default:"是否显示上传进度"},
{attribute:"sizeLimit",description:"Number / Object",type:"-",default:"图片文件大小限制, 默认单位 KB。可选单位有: 'B' | 'KB' | 'MB' | 'GB'。示例一: 1000。示例二: { size: 2, unit: 'MB', message: '图片大小不超过 {sizeLimit} MB' }。TS 类型: number | SizeLimitObj interface SizeLimitObj { size: number; unit: SizeUnit; message?: string } type SizeUnitArray = ['B', 'KB', 'MB', 'GB'] type SizeUnit = SizeUnitArray[number]。详细类型定义",
{attribute:"status",description:"String",type:"-",default:"文件上传提示文本状态。可选项: default/success/warning/error"},
{attribute:"theme",description:"String",type:"file",default:"组件风格。custom 表示完全自定义风格; file 表示默认文件上传风格; file-input 表示输入框形式的文件上传; file-flow 表示文件批量上传; image 表示默认图片上传风格; image-flow 表示图片批量上传。可选项: custom/file/file-input/file-flow/image/image-flow"},
{attribute:"tips",description:"String / Slot / Function",type:"-",default:"组件下方文本提示, 可以使用 status 定义文本。TS 类型: string | TNode。通用类型定义",
{attribute:"trigger",description:"Slot / Function",type:"-",default:"触发上传的元素, files 指本次显示的全部文件。TS 类型: TNode<TriggerContext> interface TriggerContext { dragActive?: boolean; files: UploadFile[] }。通用类型定义。详细类型定义",
{attribute:"triggerButtonProps",description:"Object",type:"-",default:"透传选择按钮全部属性。TS 类型: ButtonProps, Button API Documents。详细类型定义",
{attribute:"uploadAllFilesInOneRequest",description:"Boolean",type:"false",default:"是否在同一个请求中上传全部文件, 默认一个请求上传一个文件。多文件上传时有效",
{attribute:"useMockProgress",description:"Boolean",type:"true",default:"是否在请求时间超过 300ms 后显示模拟进度。上传进度有模拟进度和真实进度两种。一般大小的文件上传, 真实的上传进度只有 0 和 100, 不利于交互呈现, 因此组件内置模拟上传进度。真实上传进度一般用于大文件上传。"},
{attribute:"value",description:"Array",type:"[]",default:"已上传文件列表, 同 files。TS 类型: UploadFile。支持语法糖 v-model 或 v-model:value。TS 类型: Array<T>",
{attribute:"defaultValue",description:"Array",type:"[]",default:"已上传文件列表, 同 files。TS 类型: UploadFile。非受控属性。TS 类型: Array<T>",
{attribute:"withCredentials",description:"Boolean",type:"false",default:"上传请求时是否携带 cookie"},
{attribute:"onCancelUpload",description:"Function",type:"",default:"TS 类型: () => void 点击「取消上传」时触发"},
{attribute:"onChange",description:"Function",type:"",default:"TS 类型: (value: Array<T>, context: UploadChangeContext) => void 已上传文件列表发生变化时触发, trigger 表示触发本次的来源。详细类型定义。interface UploadChangeContext { e?: MouseEvent | ProgressEvent; response?: any; trigger: UploadChangeTrigger; index?: number; file?: UploadFile; files?: UploadFile[] } type UploadChangeTrigger = 'add' | 'remove' | 'abort' | 'progress-success' | 'progress' | 'progress-fail'",
{attribute:"onDragenter",description:"Function",type:"",default:"TS 类型: (context: { e: DragEvent }) => void 进入拖拽区域时触发"},
{attribute:"onDragleave",description:"Function",type:"",default:"TS 类型: (context: { e: DragEvent }) => void 离开拖拽区域时触发"},
{attribute:"onDrop",description:"Function",type:"",default:"TS 类型: (context: { e: DragEvent }) => void 拖拽结束时触发"},
{attribute:"onFail",description:"Function",type:"",default:"TS 类型: (options: UploadFailContext) => void 上传失败后触发。response 指接口响应结果, response.error 会作为错误文本提醒。如果希望判定为上传失败, 但接口响应数据不包含 error 字段, 可以使用 formatResponse 格式化 response 数据结构。如果是多文件多请求上传场景, 请到事件 onOneFileFail 中查看 response。详细类型定义。interface UploadFailContext { e?: ProgressEvent; failedFiles: UploadFile[]; currentFiles: UploadFile[]; response?: any; file: UploadFile; XMLHttpRequest?: XMLHttpRequest }",
{attribute:"onOneFileFail",description:"Function",type:"",default:"TS 类型: (options: UploadFailContext) => void 多文件/图片场景下, 单个文件上传失败后触发, 如果一个请求上传一个文件, 则会触发多次。单文件/图片不会触发"},
{attribute:"onOneFileSuccess",description:"Function",type:"",default:"TS 类型: (context: Pick<SuccessContext, 'e' | 'file' | 'response' | 'XMLHttpRequest'>) => void 单个文件上传成功后触发, 在多文件场景下会触发多次。context.file 表示当前上传成功的单个文件, context.response 表示上传请求的返回数据"},
{attribute:"onPreview",description:"Function",type:"",default:"TS 类型: (options: { file: UploadFile; index: number; e: MouseEvent }) => void 点击图片预览时触发, 文件没有预览"},
{attribute:"onProgress",description:"Function",type:"",default:"TS 类型: (options: ProgressContext) => void 上传进度变化时触发, 真实进度和模拟进度都会触发。type=real 表示真实上传进度, type=mock 表示模拟上传进度。详细类型定义。interface ProgressContext { e?: ProgressEvent; file?: UploadFile; currentFiles: UploadFile[]; percent: number; type: UploadProgressType; XMLHttpRequest?: XMLHttpRequest } type UploadProgressType = 'real' | 'mock'",
{attribute:"onRemove",description:"Function",type:"",default:"TS 类型: (context: UploadRemoveContext) => void 移除文件时触发。详细类型定义。interface UploadRemoveContext { index?:

```

```

number; file?: UploadFile; e: MouseEvent }"},
{attribute: "onSelectChange", description: "Function", type: "", default: "TS 类型: (files: File[], context: UploadSelectChangeContext) => void选择文件或图片之后, 上传之前, 触发该事件。详细类型定义。interface UploadSelectChangeContext { currentSelectedFiles: UploadFile[] }"},
{attribute: "onSuccess", description: "Function", type: "", default: "TS 类型: (context: SuccessContext) => void上传成功后触发。context.currentFiles 表示当次请求上传的文件(无论成功或失败), context.fileList 表示上传成功后的文件, context.response 表示上传请求的返回数据。context.results 表示单次选择全部文件上传成功后的响应结果, 可以在这个字段存在时提醒用户上传成功或失败。。详细类型定义。interface SuccessContext { e?: ProgressEvent; file?: UploadFile; fileList?: UploadFile[]; currentFiles?: UploadFile[]; response?: any; results?: SuccessContext[]; XMLHttpRequest?: XMLHttpRequest }"},
{attribute: "onValidate", description: "Function", type: "", default: "TS 类型: (context: { type: UploadValidateType, files: UploadFile[] }) => void文件上传校验结束事件, 文件数量超出、文件大小超出限制、文件同名、beforeAllFilesUpload 返回值为假、beforeUpload 返回值为假等场景会触发。注意: 如果设置允许上传同名文件, 即 allowUploadDuplicateFile=true, 则不会因为文件重名触发该事件。结合 status 和 tips 可以在组件中呈现不同类型的错误(或告警)提示。详细类型定义。type UploadValidateType = 'FILE_OVER_SIZE_LIMIT' | 'FILES_OVER_LENGTH_LIMIT' | 'FILTER_FILE_SAME_NAME' | 'BEFORE_ALL_FILES_UPLOAD' | 'CUSTOM_BEFORE_UPLOAD'"},
{attribute: "onWaitingUploadFilesChange", description: "Function", type: "", default: "TS 类型: (context: { files: Array<UploadFile>, trigger: 'validate' | 'remove' | 'uploaded' }) => void待上传文件列表发生变化时触发。context.files 表示事件参数为待上传文件, context.trigger 引起此次变化的触发来源"}],
{docTitle: "Upload Events说明", tableRows: [{attribute: "cancel-upload", description: "-", type: "点击「取消上传」时触发"}, {attribute: "change", description: "(value: Array<T>, context: UploadChangeContext)", type: "已上传文件列表发生变化时触发, trigger 表示触发本次的来源。详细类型定义。interface UploadChangeContext { e?: MouseEvent | ProgressEvent; response?: any; trigger: UploadChangeTrigger; index?: number; file?: UploadFile; files?: UploadFile[] }type UploadChangeTrigger = 'add' | 'remove' | 'abort' | 'progress-success' | 'progress' | 'progress-fail'"}, {attribute: "dragenter", description: "(context: { e: DragEvent })", type: "进入拖拽区域时触发"}, {attribute: "dragleave", description: "(context: { e: DragEvent })", type: "离开拖拽区域时触发"}, {attribute: "drop", description: "(context: { e: DragEvent })", type: "拖拽结束时触发"}, {attribute: "fail", description: "(options: UploadFailContext)", type: "上传失败后触发。response 指接口响应结果, response.error 会作为错误文本提醒。如果希望判定为上传失败, 但接口响应数据不包含 error 字段, 可以使用 formatResponse 格式化 response 数据结构。如果是多文件多请求上传场景, 请到事件 onOneFileFail 中查看 response。详细类型定义。interface UploadFailContext { e?: ProgressEvent; failedFiles: UploadFile[]; currentFiles: UploadFile[]; response?: any; file: UploadFile; XMLHttpRequest?: XMLHttpRequest }"}, {attribute: "one-file-fail", description: "(options: UploadFailContext)", type: "多文件/图片场景下, 单个文件上传失败后触发, 如果一个请求上传一个文件, 则会触发多次。单文件/图片不会触发"}, {attribute: "one-file-success", description: "(context: Pick<SuccessContext, 'e' | 'file' | 'response' | 'XMLHttpRequest'>)", type: "单个文件上传成功后触发, 在多文件场景下会触发多次。context.file 表示当前上传成功的单个文件, context.response 表示上传请求的返回数据"}, {attribute: "preview", description: "(options: { file: UploadFile; index: number; e: MouseEvent })", type: "点击图片预览时触发, 文件没有预览"}, {attribute: "progress", description: "(options: ProgressContext)", type: "上传进度变化时触发, 真实进度和模拟进度都会触发。type=real 表示真实上传进度, type=mock 表示模拟上传进度。详细类型定义。interface ProgressContext { e?: ProgressEvent; file?: UploadFile; currentFiles: UploadFile[]; percent: number; type: UploadProgressType; XMLHttpRequest?: XMLHttpRequest }type UploadProgressType = 'real' | 'mock'"}, {attribute: "remove", description: "(context: UploadRemoveContext)", type: "移除文件时触发。详细类型定义。interface UploadRemoveContext { index?: number; file?: UploadFile; e: MouseEvent }"}, {attribute: "select-change", description: "(files: File[], context: UploadSelectChangeContext)", type: "选择文件或图片之后, 上传之前, 触发该事件。详细类型定义。interface UploadSelectChangeContext { currentSelectedFiles: UploadFile[] }"}, {attribute: "success", description: "(context: SuccessContext)", type: "上传成功后触发。context.currentFiles 表示当次请求上传的文件(无论成功或失败), context.fileList 表示上传成功后的文件, context.response 表示上传请求的返回数据。context.results 表示单次选择全部文件上传成功后的响应结果, 可以在这个字段存在时提醒用户上传成功或失败。。详细类型定义。interface SuccessContext { e?: ProgressEvent; file?: UploadFile; fileList?: UploadFile[]; currentFiles?: UploadFile[]; response?: any; results?: SuccessContext[]; XMLHttpRequest?: XMLHttpRequest }"}, {attribute: "validate", description: "(context: { type: UploadValidateType, files: UploadFile[] })", type: "文件上传校验结束事件, 文件数量超出、文件大小超出限制、文件同名、beforeAllFilesUpload 返回值为假、beforeUpload 返回值为假等场景会触发。注意: 如果设置允许上传同名文件, 即 allowUploadDuplicateFile=true, 则不会因为文件重名触发该事件。结合 status 和 tips 可以在组件中呈现不同类型的错误(或告警)提示。详细类型定义。type UploadValidateType = 'FILE_OVER_SIZE_LIMIT' | 'FILES_OVER_LENGTH_LIMIT' | 'FILTER_FILE_SAME_NAME' | 'BEFORE_ALL_FILES_UPLOAD' | 'CUSTOM_BEFORE_UPLOAD'"}, {attribute: "waiting-upload-files-change", description: "(context: { files: Array<UploadFile>, trigger: 'validate' | 'remove' | 'uploaded' })", type: "待上传文件列表发生变化时触发。context.files 表示事件参数为待上传文件, context.trigger 引起此次变化的触发来源"}],
{docTitle: "Upload InstanceFunctions 组件实例方法说明", tableRows: [{attribute: "triggerUpload", description: "-", type: "-", default: "必需。组件实例方法, 打开文件选择器"}, {attribute: "uploadFiles", description: "(files?: UploadFile[])", type: "-", default: "必需。组件实例方法, 默认上传未成功上传过的所有文件。带参数时, 表示上传指定文件"}], {docTitle: "UploadFile说明", tableRows: [{attribute: "lastModified", description: "Number", type: "-", default: "上一次变更的时间"},

```

```

{attribute:"name",description:"String",type:"-","default:"文件名称"},
{attribute:"percent",description:"Number",type:"-","default:"下载进度"},
{attribute:"raw",description:"Object",type:"-","default:"原始文件对象。TS 类型: File"},
{attribute:"response",description:"Object",type:"-","default:"上传接口返回的数据。response.error 存在
时会判断此次上传失败,并显示错误文本信息;response.url 会作为文件上传成功后的地址,并使用该地址显示
图片。TS 类型: { [key: string]: any }"},{attribute:"size",description:"Number",type:"-","default:"文
件大小"},{attribute:"status",description:"String",type:"-","default:"文件上传状态: 上传成功,上传失
败,上传中,等待上传。TS 类型: 'success' | 'fail' | 'progress' | 'waiting'"},
{attribute:"type",description:"String",type:"-","default:"文件类型"},
{attribute:"uploadTime",description:"String",type:"-","default:"上传时间"},
{attribute:"url",description:"String",type:"-","default:"文件上传成功后的下载/访问地址"},
{attribute:"PlainObject",description:"-","type:"-","default:"PlainObject 不是 UploadFile 中的属性,而
表示 UploadFile 本身支持添加任意属性,type PlainObject = {[key: string]: any}'"}]]],
{componentName:"avatar",apiData:[{docTitle:"Avatar Props说明",tableRows:
[ {attribute:"alt",description:"String",type:"-","default:"头像替换文本,仅当图片加载失败时有效"},
{attribute:"content",description:"String / Slot / Function",type:"-","default:"子元素内容。TS 类型:
string | TNode。通用类型定义"},{attribute:"default",description:"String / Slot / Function",type:"-
",default:"子元素内容,同 content。TS 类型: string | TNode。通用类型定义"},
{attribute:"hideOnLoadFailed",description:"Boolean",type:"false",default:"加载失败时隐藏图片"},
{attribute:"icon",description:"Slot / Function",type:"-","default:"图标。TS 类型: TNode。通用类型定
义"},{attribute:"image",description:"String",type:"-","default:"图片地址"},
{attribute:"imageProps",description:"Object",type:"-","default:"透传至 Image 组件。TS 类型:
ImageProps, Image API Documents。详细类型定义"},
{attribute:"shape",description:"String",type:"circle",default:"形状。可选项: circle/round。TS 类型:
ShapeEnum type ShapeEnum = 'circle' | 'round'。详细类型定义"},
{attribute:"size",description:"String",type:"-","default:"尺寸,示例值: small/medium/large/24px/38px
等。优先级高于 AvatarGroup.size。Avatar 单独存在时,默认值为 medium。如果父组件存在 AvatarGroup,默
认值便由 AvatarGroup.size 决定"},{attribute:"onError",description:"Function",type:"","default:"TS 类
型: (context: { e: Event }) => void图片加载失败时触发"}]],{docTitle:"Avatar Events说明",tableRows:
[ {attribute:"error",description:"(context: { e: Event })",type:"图片加载失败时触发"}]],
{docTitle:"AvatarGroup Props说明",tableRows:
[ {attribute:"cascading",description:"String",type:"'right-up'",default:"图片之间的层叠关系,可选值:
左侧图片在上和右侧图片在上。可选项: left-up/right-up。TS 类型: CascadingValue type CascadingValue =
'left-up' | 'right-up'。详细类型定义"},{attribute:"collapseAvatar",description:"String / Slot /
Function",type:"-","default:"头像数量超出时,会出现一个头像折叠元素。该元素内容可自定义。默认为 +N。
示例: +5, ..., 更多。TS 类型: string | TNode。通用类型定义"},
{attribute:"max",description:"Number",type:"-","default:"能够同时显示的最多头像数量"},
{attribute:"popupProps",description:"Object",type:"-","default:"头像右上角提示信息。TS 类型:
PopupProps, Popup API Documents。详细类型定义"},{attribute:"size",description:"String",type:"-
",default:"尺寸,示例值: small/medium/large/24px/38px 等。优先级低于 Avatar.size"}]]],
{componentName:"badge",apiData:[{docTitle:"Badge Props说明",tableRows:
[ {attribute:"color",description:"String",type:"-","default:"颜色"},
{attribute:"content",description:"String / Slot / Function",type:"-","default:"徽标内容。TS 类型:
string | TNode。通用类型定义"},{attribute:"count",description:"String / Number / Slot /
Function",type:"0",default:"徽标右上角内容。可以是数字,也可以是文字。如: 'new' / 3/99+。TS 类型:
string | number | TNode。通用类型定义"},{attribute:"default",description:"String / Slot /
Function",type:"-","default:"徽标内容,默认插槽,同 content。TS 类型: string | TNode。通用类型定义"},
{attribute:"dot",description:"Boolean",type:"false",default:"是否为红点"},
{attribute:"maxCount",description:"Number",type:"99",default:"封顶的数字值"},
{attribute:"offset",description:"Array",type:"-","default:"设置状态点的位置偏移,示例: [-10, 20] 或
['10em', '8rem']。TS 类型: Array<string | number>"},
{attribute:"shape",description:"String",type:"circle",default:"形状。可选项: circle/round"},
{attribute:"showZero",description:"Boolean",type:"false",default:"当数值为 0 时,是否展示徽标"},
{attribute:"size",description:"String",type:"medium",default:"尺寸。可选项: small/medium"}]]],
{componentName:"calendar",apiData:[{docTitle:"Calendar Props说明",tableRows:
[ {attribute:"cell",description:"String / Slot / Function",type:"-","default:"单元格插槽。TS 类型:
string | TNode<CalendarCell>。通用类型定义"},{attribute:"cellAppend",description:"String / Slot /
Function",type:"-","default:"单元格插槽,在原来的内容之后追加。TS 类型: string |
TNode<CalendarCell>。通用类型定义"},{attribute:"controllerConfig",description:"Boolean /
Object",type:"-","default:"右上角控制器配置。值为 false 则表示不显示控制器,值为 true 则显示控制器默
认配置,值类型为 CalendarController 则显示为自定义控制器配置。TS 类型: boolean |
CalendarController"},{attribute:"fillWithZero",description:"Boolean",type:"true",default:"小于 10 的
日期,是否使用 '0' 填充。默认表现为 01 02,值为 false 表现为 1 2 9"},
{attribute:"firstDayOfWeek",description:"Number",type:"1",default:"第一天从星期几开始,仅在日历展示
维度为月份时(mode = month)有效。默认为 1。可选项: 1/2/3/4/5/6/7"},
{attribute:"format",description:"String",type:"'YYYY-MM-DD'",default:"用于格式化日期,决定事件参数
formattedFilterDate 的输出值。详细文档"},{attribute:"head",description:"String / Slot /
Function",type:"-","default:"头部插槽(左上角处,默认不显示任何内容)。TS 类型: string |
TNode<ControllerOptions>。通用类型定义"},

```

```

{attribute:"isShowWeekendDefault",description:"Boolean",type:"true",default:"默认是否显示周末"},
{attribute:"mode",description:"String",type:"month",default:"日历展示维度。可选项: month/year"},
{attribute:"month",description:"String / Number",type:"-",default:"控制当前面板展示月份, 优先级高于 controllerConfig.month"}, {attribute:"multiple",description:"Boolean",type:"-",default:"是否高亮多个日期单元格"}, {attribute:"preventCellContextMenu",description:"Boolean",type:"false",default:"是否禁用单元格右键默认系统菜单"}, {attribute:"range",description:"Array",type:"-",default:"用于设置日历的年份月份显示范围, [范围开始, 范围结束]。TS 类型: Array<CalendarValue>"},
{attribute:"theme",description:"String",type:"full",default:"日历风格。可选项: full/card"},
{attribute:"value",description:"String / Date",type:"-",default:"当前高亮的日期。TS 类型: CalendarValue | CalendarValue[] type CalendarValue = string | Date。详细类型定义"},
{attribute:"week",description:"Array / Slot / Function",type:"-",default:"用于自定义日历星期呈现方式。CalendarWeek.day 表示当前是星期几。示例一: ['周一', '周二', '周三', '周四', '周五', '星期六', '星期天']。示例二: ({ day }) => '周' + day。TS 类型: Array<string> | TNode<CalendarWeek> interface CalendarWeek { day: WeekDay } type WeekDay = 1 | 2 | 3 | 4 | 5 | 6 | 7。通用类型定义。详细类型定义"}, {attribute:"year",description:"String / Number",type:"-",default:"控制当前面板展示年份, 优先级高于 controllerConfig.year"}, {attribute:"onCellClick",description:"Function",type:"",default:"TS 类型: (options: { cell: CalendarCell; e: MouseEvent }) => void 日历单元格点击时触发"}, {attribute:"onCellDoubleClick",description:"Function",type:"",default:"TS 类型: (options: { cell: CalendarCell; e: MouseEvent }) => void 日历单元格双击时触发"}, {attribute:"onCellRightClick",description:"Function",type:"",default:"TS 类型: (options: { cell: CalendarCell; e: MouseEvent }) => void 日历单元格右击时触发"}, {attribute:"onControllerChange",description:"Function",type:"",default:"TS 类型: (options: ControllerOptions) => void 右上角控件组选中值有变化的时候触发。详细类型定义。interface ControllerOptions { filterDate: Date; formattedFilterDate: string; mode: string; isShowWeekend: boolean }"}, {attribute:"onMonthChange",description:"Function",type:"",default:"TS 类型: (options: { month: string; year: string }) => void 月份切换时触发"}], {docTitle:"Calendar Events说明",tableRows: [{attribute:"cell-click",description:"(options: { cell: CalendarCell; e: MouseEvent })",type:"日历单元格点击时触发"}, {attribute:"cell-double-click",description:"(options: { cell: CalendarCell; e: MouseEvent })",type:"日历单元格双击时触发"}, {attribute:"cell-right-click",description:"(options: { cell: CalendarCell; e: MouseEvent })",type:"日历单元格右击时触发"}, {attribute:"controller-change",description:"(options: ControllerOptions)",type:"右上角控件组选中值有变化的时候触发。详细类型定义。interface ControllerOptions { filterDate: Date; formattedFilterDate: string; mode: string; isShowWeekend: boolean }"}, {attribute:"month-change",description:"(options: { month: string; year: string })",type:"月份切换时触发"}], {docTitle:"CalendarController说明",tableRows: [{attribute:"current",description:"Object",type:"-",default:"“今天(本月)”按钮控制器。TS 类型: { visible?: boolean; currentDayButtonProps?: ButtonProps; currentMonthButtonProps?: ButtonProps }"}, {attribute:"disabled",description:"Boolean",type:"false",default:"是否禁用右上角控制器"}, {attribute:"mode",description:"Object",type:"-",default:"日历展示维度控制器。TS 类型: { visible?: boolean; radioGroupProps?: RadioGroupProps }, Radio API Documents。详细类型定义"}, {attribute:"month",description:"Object",type:"-",default:"日历月份控制器。TS 类型: { visible?: boolean; selectProps?: SelectProps }"}, {attribute:"weekend",description:"Object",type:"-",default:"隐藏/显示周末控制器。TS 类型: { visible?: boolean; showWeekendButtonProps?: CheckTagProps; hideWeekendButtonProps?: CheckTagProps }, Tag API Documents。详细类型定义"}, {attribute:"year",description:"Object",type:"-",default:"日历年份控制器。TS 类型: { visible?: boolean; selectProps?: SelectProps }, Select API Documents。详细类型定义"}], {docTitle:"CalendarCell说明",tableRows: [{attribute:"belongTo",description:"Number",type:"-",default:"用于表示日期单元格属于哪一个月份。值为 0 表示是当前日历显示的月份中的日期, 值为 -1 表示是上个月的, 值为 1 表示是下个月的 (日历展示维度是“月”时有值)"}, {attribute:"date",description:"Object",type:"-",default:"日历单元格日期。TS 类型: Date"}, {attribute:"day",description:"Number",type:"-",default:"日期单元格对应的星期, 值为 1~7, 表示周一到周日。 (日历展示维度是“月”时有值)"}, {attribute:"formattedDate",description:"String",type:"-",default:"日历单元格日期字符串 (输出日期的格式和 format 有关)"}, {attribute:"isCurrent",description:"Boolean",type:"-",default:"日期单元格是否为当前高亮日期或高亮月份"}, {attribute:"weekOrder",description:"Number",type:"-",default:"日期在本月的第几周 (日历展示维度是“月”时有值)"}, {attribute:"ControllerOptions",description:"-",type:"-",default:"继承 ControllerOptions 中的全部 API"}], {componentName:"card",apiData:[{docTitle:"Card Props说明",tableRows: [{attribute:"actions",description:"String / Slot / Function",type:"-",default:"卡片操作区。TS 类型: string | TNode。通用类型定义"}, {attribute:"avatar",description:"String / Slot / Function",type:"-",default:"卡片中的用户头像, 仅在海报风格的卡片中有效。TS 类型: string | TNode。通用类型定义"}, {attribute:"bordered",description:"Boolean",type:"true",default:"是否有边框"}, {attribute:"content",description:"String / Slot / Function",type:"-",default:"卡片内容。TS 类型: string | TNode。通用类型定义"}, {attribute:"cover",description:"String / Slot / Function",type:"-",default:"卡片封面图。值类型为字符串, 会自动使用 img 标签输出封面图; 也可以完全自定义封面图。TS 类型: string | TNode。通用类型定义"}, {attribute:"default",description:"String / Slot / Function",type:"-",default:"卡片内容, 同 content。TS 类型: string | TNode。通用类型定义"}, {attribute:"description",description:"String / Slot / Function",type:"-",default:"卡片描述文案。TS 类型: string | TNode。通用类型定义"}, {attribute:"footer",description:"String / Slot / Function",type:"-",default:"卡片底部内容, 可完全自定义。TS 类型: string | TNode。通用类型定义"}, {attribute:"header",description:"String / Slot / Function",type:"-",default:"卡片顶部内容, 优先级高

```


于其他所有元素。TS 类型: string | TNode。通用类型定义"},
 {attribute:"headerBordered",description:"Boolean",type:"false",default:"头部是否带分割线, 仅在在有 header 时有效"}, {attribute:"hoverShadow",description:"Boolean",type:"false",default:"hover 时是否有阴影"}, {attribute:"loading",description:"Boolean / Slot / Function",type:"false",default:"加载状态, 值为 true 会根据不同的布局显示不同的加载状态, 值为 false 则表示非加载状态。也可以使用 Sketon 组件完全自定义加载态呈现内容。TS 类型: boolean | TNode。通用类型定义"},
 {attribute:"shadow",description:"Boolean",type:"false",default:"是否显示卡片阴影, 默认不显示"},
 {attribute:"size",description:"String",type:"medium",default:"尺寸。可选项: medium/small"},
 {attribute:"status",description:"String",type:"-",default:"卡片状态内容, 仅在操作区域不在顶部时有效 (即 theme=poster2) "}, {attribute:"subtitle",description:"String / Slot / Function",type:"-",default:"卡片副标题。TS 类型: string | TNode。通用类型定义"},
 {attribute:"theme",description:"String",type:"normal",default:"卡片风格: 普通风格、海报风格1 (操作区域在顶部)、海报风格2 (操作区域在底部)。可选项: normal/poster1/poster2"},
 {attribute:"title",description:"String / Slot / Function",type:"-",default:"卡片标题。TS 类型: string | TNode。通用类型定义"}]]], {componentName:"collapse",apiData:[{docTitle:"Collapse Props 说明",tableRows:[{attribute:"borderless",description:"Boolean",type:"false",default:"是否为无边框模式"}, {attribute:"defaultExpandAll",description:"Boolean",type:"false",default:"默认是否展开全部"}, {attribute:"disabled",description:"Boolean",type:"-",default:"是否禁用面板展开/收起操作"}, {attribute:"expandIcon",description:"Boolean / Slot / Function",type:"true",default:"展开图标。值为 undefined 或 false 则不显示展开图标; 值为 true 显示默认图标; 值类型为函数, 则表示完全自定义展开图标。TS 类型: boolean | TNode。通用类型定义"}, {attribute:"expandIconPlacement",description:"String",type:"left",default:"展开图标的位置, 左侧或右侧。可选项: left/right"}, {attribute:"expandMutex",description:"Boolean",type:"false",default:"每个面板互斥展开, 每次只展开一个面板"}, {attribute:"expandOnRowClick",description:"Boolean",type:"true",default:"是否允许点击整行标题展开面板"}, {attribute:"value",description:"Array",type:"[]",default:"展开的面板集合。支持语法糖 v-model 或 v-model:value。TS 类型: CollapseValue type CollapseValue = Array<string | number>。详细类型定义"}, {attribute:"defaultValue",description:"Array",type:"[]",default:"展开的面板集合。非受控属性。TS 类型: CollapseValue type CollapseValue = Array<string | number>。详细类型定义"}, {attribute:"onChange",description:"Function",type:"",default:"TS 类型: (value: CollapseValue) => void 切换面板时触发, 返回变化的值"}]]], {docTitle:"Collapse Events 说明",tableRows:[{attribute:"change",description:"(value: CollapseValue)",type:"切换面板时触发, 返回变化的值"}]]], {docTitle:"CollapsePanel Props 说明",tableRows:[{attribute:"content",description:"String / Slot / Function",type:"-",default:"折叠面板内容。TS 类型: string | TNode。通用类型定义"}, {attribute:"default",description:"String / Slot / Function",type:"-",default:"折叠面板内容, 同 content。TS 类型: string | TNode。通用类型定义"}, {attribute:"destroyOnCollapse",description:"Boolean",type:"false",default:"当前面板处理折叠状态时, 是否销毁面板内容"}, {attribute:"disabled",description:"Boolean",type:"undefined",default:"禁止当前面板展开, 优先级大于 Collapse 的同名属性"}, {attribute:"expandIcon",description:"Boolean / Slot / Function",type:"undefined",default:"当前折叠面板展开图标, 优先级大于 Collapse 的同名属性。TS 类型: boolean | TNode。通用类型定义"}, {attribute:"header",description:"String / Slot / Function",type:"-",default:"面板头内容。TS 类型: string | TNode。通用类型定义"}, {attribute:"headerRightContent",description:"String / Slot / Function",type:"-",default:"面板头的右侧区域, 一般用于呈现面板操作。TS 类型: string | TNode。通用类型定义"}, {attribute:"value",description:"String / Number",type:"-",default:"必需。当前面板唯一标识, 如果值为空则取当前面下标兜底作为唯一标识"}]]]], {componentName:"comment",apiData:[{docTitle:"Comment Props 说明",tableRows:[{attribute:"actions",description:"Array",type:"-",default:"操作。TS 类型: Array<TNode>。通用类型定义"}, {attribute:"author",description:"String / Slot / Function",type:"-",default:"作者。TS 类型: string | TNode。通用类型定义"}, {attribute:"avatar",description:"String / Object / Slot / Function",type:"-",default:"头像。TS 类型: string | AvatarProps | TNode, Avatar API Documents。通用类型定义。详细类型定义"}, {attribute:"content",description:"String / Slot / Function",type:"-",default:"内容。TS 类型: string | TNode。通用类型定义"}, {attribute:"datetime",description:"String / Slot / Function",type:"-",default:"时间。TS 类型: string | TNode。通用类型定义"}, {attribute:"quote",description:"String / Slot / Function",type:"-",default:"引用。TS 类型: string | TNode。通用类型定义"}, {attribute:"reply",description:"String / Slot / Function",type:"-",default:"回复。TS 类型: string | TNode。通用类型定义"}]]]], {componentName:"image",apiData:[{docTitle:"Image Props 说明",tableRows:[{attribute:"alt",description:"String",type:"-",default:"图片描述"}, {attribute:"error",description:"String / Slot / Function",type:"-",default:"自定义图片加载失败状态下的显示内容。TS 类型: string | TNode。通用类型定义"}, {attribute:"fit",description:"String",type:"fill",default:"图片填充模式。可选项: contain/cover/fill/none/scale-down"}, {attribute:"gallery",description:"Boolean",type:"false",default:"是否展示为图集样式"}, {attribute:"lazy",description:"Boolean",type:"false",default:"是否开启图片懒加载"}, {attribute:"loading",description:"String / Slot / Function",type:"-",default:"自定义加载中状态的图片内容, 如: “加载中”。TS 类型: string | TNode。通用类型定义"}, {attribute:"overlayContent",description:"String / Slot / Function",type:"-",default:"图片上方的浮层内容。TS 类型: string | TNode。通用类型定义"}, {attribute:"overlayTrigger",description:"String",type:"always",default:"浮层 overlayContent 出现的时

机。可选项: always/hover"}, {attribute: "placeholder", description: "String / Slot / Function", type: "-", default: "占位元素, 展示层级低于 loading error 和图片本身, 值类型为字符串时表示占位图片地址。TS 类型: string | TNode。通用类型定义"}, {attribute: "position", description: "String", type: "center", default: "等同于原生的 object-position 属性, 可选值为 top right bottom left 或 string, 可以自定义任何单位, px 或者 百分比"}, {attribute: "shape", description: "String", type: "square", default: "图片圆角类型。可选项: circle/round/square"}, {attribute: "src", description: "String", type: "-", default: "图片链接"}, {attribute: "srcset", description: "Object", type: "-", default: "图片地址, 支持特殊格式的图片, 如 .avif 和 .webp。TS 类型: ImageSrcset interface ImageSrcset { 'image/avif': string; 'image/webp': string; }。详细类型定义"}, {attribute: "onError", description: "Function", type: "", default: "TS 类型: (context: { e: ImageEvent }) => void 图片加载失败时触发"}, {attribute: "onLoad", description: "Function", type: "", default: "TS 类型: (context: { e: ImageEvent }) => void 图片加载完成时触发"}], {docTitle: "Image Events说明", tableRows: [{attribute: "error", description: "(context: { e: ImageEvent })", type: "图片加载失败时触发"}, {attribute: "load", description: "(context: { e: ImageEvent })", type: "图片加载完成时触发"}]}, {componentName: "image-viewer", apiData: [{docTitle: "ImageViewer Props说明", tableRows: [{attribute: "closeBtn", description: "Boolean / Slot / Function", type: "true", default: "是否展示关闭按钮, 值为 true 显示默认关闭按钮; 值为 false 则不显示关闭按钮; 也可以完全自定义关闭按钮。TS 类型: boolean | TNode。通用类型定义"}, {attribute: "closeOnOverlay", description: "Boolean", type: "-", default: "是否在点击遮罩层时, 触发预览关闭"}, {attribute: "draggable", description: "Boolean", type: "undefined", default: "是否允许拖拽调整位置。mode=modal 时, 默认不允许拖拽; mode=modelless 时, 默认允许拖拽"}, {attribute: "imageScale", description: "Object", type: "-", default: "图片缩放相关配置。imageScale.max 缩放的 最大比例; imageScale.min 缩放的最小比例; imageScale.step 缩放的步长速度。TS 类型: ImageScale interface ImageScale { max: number; min: number; step: number }。详细类型定义"}, {attribute: "images", description: "Array", type: "[]", default: "图片数组。mainImage 表示主图, 必传; thumbnail 表示缩略图, 如果不存在, 则使用主图显示; download 是否允许下载图片, 默认允许下载。示例: ['img_url_1', 'img_url_2'], [{thumbnail: 'small_image_url', mainImage: 'big_image_url', download: false}]。TS 类型: Array<string | ImageInfo> interface ImageInfo { mainImage: string; thumbnail?: string; download?: boolean }。详细类型定义"}, {attribute: "index", description: "Number", type: "-", default: "当前预览图片所在的下标。支持语法糖 v-model:index"}, {attribute: "defaultIndex", description: "Number", type: "-", default: "当前预览图片所在的下标。非受控属性"}, {attribute: "mode", description: "String", type: "modal", default: "模态预览 (modal) 和非模态预览 (modelless)。可选项: modal/modelless"}, {attribute: "navigationArrow", description: "Boolean / Slot / Function", type: "true", default: "切换预览图片的左图标, 可自定义。TS 类型: boolean | TNode。通用类型定义"}, {attribute: "showOverlay", description: "Boolean", type: "undefined", default: "是否显示遮罩层。mode=modal 时, 默认显示; mode=modelless 时, 默认不显示"}, {attribute: "title", description: "String / Slot / Function", type: "-", default: "预览标题。TS 类型: string | TNode。通用类型定义"}, {attribute: "trigger", description: "String / Slot / Function", type: "-", default: "触发图片预览的元素, 可能是一个预览按钮, 可能是一张缩略图, 完全自定义。TS 类型: string | TNode<{ open: () => void }>。通用类型定义"}, {attribute: "viewerScale", description: "Object", type: "-", default: "限制预览器缩放的最小宽度和最小高度, 仅 mode=modelless 时有效。TS 类型: ImageViewerScale interface ImageViewerScale { minWidth: number; minHeight: number }。详细类型定义"}, {attribute: "visible", description: "Boolean", type: "false", default: "隐藏/显示预览。支持语法糖 v-model 或 v-model:visible"}, {attribute: "defaultVisible", description: "Boolean", type: "false", default: "隐藏/显示预览。非受控属性"}, {attribute: "zIndex", description: "Number", type: "-", default: "层级, 默认为 2000"}, {attribute: "onClose", description: "Function", type: "", default: "TS 类型: (context: { trigger: 'close-btn' | 'overlay' | 'esc'; e: MouseEvent | KeyboardEvent }) => void 关闭时触发, 事件参数包含触发关闭的来源: 关闭按钮、遮罩层、ESC 键"}, {attribute: "onIndexChange", description: "Function", type: "", default: "TS 类型: (index: number, context: { trigger: 'prev' | 'next' }) => void 预览图片切换时触发, context.prev 切换到上一张图片, context.next 切换到下一张图片"}], {docTitle: "ImageViewer Events说明", tableRows: [{attribute: "close", description: "(context: { trigger: 'close-btn' | 'overlay' | 'esc'; e: MouseEvent | KeyboardEvent })", type: "关闭时触发, 事件参数包含触发关闭的来源: 关闭按钮、遮罩层、ESC 键"}, {attribute: "index-change", description: "(index: number, context: { trigger: 'prev' | 'next' })", type: "预览图片切换时触发, context.prev 切换到上一张图片, context.next 切换到下一张图片"}]}, {componentName: "list", apiData: [{docTitle: "List Props说明", tableRows: [{attribute: "asyncLoading", description: "String / Slot / Function", type: "-", default: "自定义加载中。值为空不显示加载中, 值为 'loading' 显示加载中状态, 值为 'load-more' 显示加载更多状态。值类型为函数, 则表示自定义加载状态呈现内容。TS 类型: string | TNode。通用类型定义"}, {attribute: "footer", description: "String / Slot / Function", type: "-", default: "底部。TS 类型: string | TNode。通用类型定义"}, {attribute: "header", description: "String / Slot / Function", type: "-", default: "头部。TS 类型: string | TNode。通用类型定义"}, {attribute: "layout", description: "String", type: "horizontal", default: "排列方式 (待设计稿输出)。可选项: horizontal/vertical"}, {attribute: "size", description: "String", type: "medium", default: "尺寸。可选项: small/medium/large"}, {attribute: "split", description: "Boolean", type: "false", default: "是否展示分割线"}, {attribute: "stripe", description: "Boolean", type: "false", default: "是否展示斑马纹"}, {attribute: "onLoadMore", description: "Function", type: "", default: "TS 类型: (options: { e: MouseEvent }) => void 点击加载更多时触发"}, {attribute: "onScroll", description: "Function", type: "", default: "TS 类型: (options: { e: Event | WheelEvent; scrollTop: number; scrollBottom: number }) => void 列表滚动时触发, scrollTop 表示顶部滚动距离, scrollBottom 表示底部滚动距离"}]}, {docTitle: "List Events说

明",tableRows:[{attribute:"load-more",description:"(options: { e: MouseEvent })",type:"点击加载更多时触发"},{attribute:"scroll",description:"(options: { e: Event | WheelEvent; scrollTop: number; scrollTop: number })",type:"列表滚动时触发, scrollTop 表示顶部滚动距离, scrollTop 表示底部滚动距离"}],{docTitle:"ListItem Props说明",tableRows:[{attribute:"action",description:"String / Slot / Function",type:"-","default:"操作栏。TS 类型: string | TNode。通用类型定义"},{attribute:"content",description:"String / Slot / Function",type:"-","default:"内容。TS 类型: string | TNode。通用类型定义"},{attribute:"default",description:"String / Slot / Function",type:"-","default:"内容, 同 content。TS 类型: string | TNode。通用类型定义"}]}, {docTitle:"ListItemMeta Props说明",tableRows:[{attribute:"avatar",description:"String / Slot / Function",type:"-","default:"已废弃。列表项图片。TS 类型: string | TNode。通用类型定义"},{attribute:"description",description:"String / Slot / Function",type:"-","default:"列表项内容。TS 类型: string | TNode。通用类型定义"},{attribute:"image",description:"String / Slot / Function",type:"-","default:"列表项图片。TS 类型: string | TNode。通用类型定义"},{attribute:"title",description:"String / Slot / Function",type:"-","default:"列表项标题。TS 类型: string | TNode。通用类型定义"}]}],{componentName:"loading",apiData:[{docTitle:"Loading Props说明",tableRows:[{attribute:"attach",description:"String / Function",type:"'",default:"挂载元素, 默认挂载到组件本身所在的位置。数据类型为 String 时, 会被当作选择器处理, 进行节点查询。示例: 'body' 或 () => document.body。TS 类型: AttachNode。通用类型定义"},{attribute:"content",description:"String / Slot / Function",type:"-","default:"子元素。TS 类型: string | TNode。通用类型定义"},{attribute:"default",description:"String / Slot / Function",type:"-","default:"子元素, 同 content。TS 类型: string | TNode。通用类型定义"},{attribute:"delay",description:"Number",type:"0",default:"延迟显示加载效果的时间, 用于防止请求速度过快引起的加载闪烁, 单位: 毫秒"},{attribute:"fullscreen",description:"Boolean",type:"false",default:"是否显示为全屏加载"},{attribute:"indicator",description:"Boolean / Slot / Function",type:"true",default:"加载指示符, 值为 true 显示默认指示符, 值为 false 则不显示, 也可以自定义指示符。TS 类型: boolean | TNode。通用类型定义"},{attribute:"inheritColor",description:"Boolean",type:"false",default:"是否继承父元素颜色"},{attribute:"loading",description:"Boolean",type:"true",default:"是否处于加载状态"},{attribute:"preventScrollThrough",description:"Boolean",type:"true",default:"防止滚动穿透, 全屏加载模式有效"},{attribute:"showOverlay",description:"Boolean",type:"true",default:"是否需要遮罩层, 遮罩层对包裹元素才有效"},{attribute:"size",description:"String",type:"medium",default:"尺寸, 示例: small/medium/large/12px/56px/0.3em"},{attribute:"text",description:"String / Slot / Function",type:"-","default:"加载提示文案。TS 类型: string | TNode。通用类型定义"},{attribute:"zIndex",description:"Number",type:"-","default:"消息通知层级, 样式默认为 3500"}]}, {docTitle:"LoadingPlugin说明",tableRows:[]}],{componentName:"progress",apiData:[{docTitle:"Progress Props说明",tableRows:[{attribute:"color",description:"String / Object / Array",type:"'",default:"进度条颜色。示例: '#ED7B2F' 或 'orange' 或 ['#f00', '#0ff', '#f0f'] 或 { '0%': '#f00', '100%': '#0ff' } 或 { from: '#000', to: '#000' } 等。TS 类型: string | Array<string> | Record<string, string>"},{attribute:"label",description:"String / Boolean / Slot / Function",type:"true",default:"进度百分比, 可自定义。TS 类型: string | boolean | TNode。通用类型定义"},{attribute:"percentage",description:"Number",type:"0",default:"进度条百分比"},{attribute:"size",description:"String / Number",type:"medium",default:"进度条尺寸, 示例: small/medium/large/240。small 值为 72; medium 值为 112; large 值为 160"},{attribute:"status",description:"String",type:"-","default:"进度条状态。可选项: success/error/warning/active。TS 类型: StatusEnum type StatusEnum = 'success' | 'error' | 'warning' | 'active'。详细类型定义"},{attribute:"strokeWidth",description:"String / Number",type:"-","default:"进度条线宽。宽度数值不能超过 size 的一半, 否则不能输出环形进度"},{attribute:"theme",description:"String",type:"line",default:"进度条风格。值为 line, 标签 (label) 显示在进度条右侧; 值为 plump, 标签 (label) 显示在进度条里面; 值为 circle, 标签 (label) 显示在进度条正中间。可选项: line/plump/circle。TS 类型: ThemeEnum type ThemeEnum = 'line' | 'plump' | 'circle'。详细类型定义"},{attribute:"trackColor",description:"String",type:"'",default:"进度条未完成部分颜色"}]}],{componentName:"skeleton",apiData:[{docTitle:"Skeleton Props说明",tableRows:[{attribute:"animation",description:"String",type:"none",default:"动画效果, 有「渐变加载动画」和「闪烁加载动画」两种。值为 'none' 则表示没有动画。可选项: gradient/flashed/none"},{attribute:"delay",description:"Number",type:"0",default:"【开发中】延迟显示加载效果的时间, 用于防止请求速度过快引起的加载闪烁, 单位: 毫秒"},{attribute:"loading",description:"Boolean",type:"true",default:"是否为加载状态, 如果是则显示骨架图, 如果不是则显示加载完成的内容"},{attribute:"rowCol",description:"Array",type:"-","default:"高级设置, 用于自定义行列数量、宽度高度、间距等。【示例一】, [1, 1, 2] 表示输出三行骨架图, 第一行一列, 第二行一列, 第三行两列。【示例二】, [1, 1, { width: '100px' }] 表示自定义第三行的宽度为 100px。【示例三】, [1, 2, [{ width, height }, { width, height, marginLeft }]] 表示第三行有两列, 且自定义宽度、高度、尺寸 (圆形或方形使用)、间距、内容等。TS 类型: SkeletonRowCol type SkeletonRowCol = Array<Number | SkeletonRowColObj | Array<SkeletonRowColObj>> interface SkeletonRowColObj { width?: string; height?: string; size?: string; marginRight?: string; marginLeft?: string; margin?: string; content?: string | TNode; type?: 'rect' | 'circle' | 'text' }。通用类型定义"},{attribute:"theme",description:"String",type:"text",default:"快捷定义骨架图风格, 有基础、头像组合等, 具体参看代码示例。可选项: text/avatar/paragraph/avatar-text/tab/article"}]}],{componentName:"swiper",apiData:[{docTitle:"Swiper Props说明",tableRows:[{attribute:"animation",description:"String",type:"slide",default:"轮播切换动画效果类型: 滑动、淡入淡出等。可选项: slide/fade"},{attribute:"autoplay",description:"Boolean",type:"true",default:"是否自

动播放"}, {attribute: "current", description: "Number", type: "0", default: "当前轮播在哪一项（下标）。支持语法糖 v-model 或 v-model:current"}, {attribute: "defaultCurrent", description: "Number", type: "0", default: "当前轮播在哪一项（下标）。非受控属性"}, {attribute: "direction", description: "String", type: "horizontal", default: "轮播滑动方向，包括横向滑动和纵向滑动两个方向。可选项: horizontal/vertical"}, {attribute: "duration", description: "Number", type: "300", default: "滑动动画时长"}, {attribute: "height", description: "Number", type: "-", default: "当使用垂直方向滚动时的高度"}, {attribute: "interval", description: "Number", type: "5000", default: "轮播间隔时间"}, {attribute: "loop", description: "Boolean", type: "true", default: "是否循环播放"}, {attribute: "navigation", description: "Object / Slot / Function", type: "-", default: "导航器全部配置。TS 类型: SwiperNavigation | TNode。通用类型定义"}, {attribute: "stopOnHover", description: "Boolean", type: "true", default: "是否悬浮时停止轮播"}, {attribute: "theme", description: "String", type: "light", default: "深色模式和浅色模式。可选项: light/dark"}, {attribute: "trigger", description: "String", type: "hover", default: "触发切换的方式: 悬浮、点击等。可选项: hover/click"}, {attribute: "type", description: "String", type: "default", default: "样式类型: 默认样式、卡片样式。可选项: default/card"}, {attribute: "onChange", description: "Function", type: "", default: "TS 类型: (current: number, context: { source: SwiperChangeSource }) => void 轮播切换时触发。详细类型定义。type SwiperChangeSource = 'autoplay' | 'click' | 'hover'"}], {docTitle: "Swiper Events说明", tableRows: [{attribute: "change", description: "(current: number, context: { source: SwiperChangeSource })", type: "轮播切换时触发。详细类型定义。type SwiperChangeSource = 'autoplay' | 'click' | 'hover'"}], {docTitle: "SwiperNavigation说明", tableRows: [{attribute: "placement", description: "String", type: "inside", default: "导航器位置，位于主体的内侧或是外侧。可选项: inside/outside"}, {attribute: "showSlideBtn", description: "String", type: "always", default: "何时显示导航器的翻页按钮: 始终显示、悬浮显示、永不显示。可选项: always/hover/never"}, {attribute: "size", description: "String", type: "medium", default: "导航器尺寸。可选项: small/medium/large"}, {attribute: "type", description: "String", type: "-", default: "导航器类型，点状(dots)、点条状(dots-bar)、条状(bars)、分式(fraction)等。TS 类型: SwiperNavigationType type SwiperNavigationType = 'dots' | 'dots-bar' | 'bars' | 'fraction'。详细类型定义"}]}], {componentName: "table", apiData: [{docTitle: "BaseTable Props说明", tableRows: [{attribute: "allowResizeColumnWidth", description: "Boolean", type: "undefined", default: "已废弃。是否允许调整列宽。请更为使用 resizable"}, {attribute: "attach", description: "String / Function", type: "-", default: "超出省略等所有浮层元素统一绑定到 attach，可根据实际情况调整挂载元素。TS 类型: AttachNode。通用类型定义"}, {attribute: "bordered", description: "Boolean", type: "false", default: "是否显示表格边框"}, {attribute: "bottomContent", description: "String / Slot / Function", type: "-", default: "表格底部内容，可以用于自定义列设置等。TS 类型: string | TNode。通用类型定义"}, {attribute: "cellEmptyContent", description: "String / Slot / Function", type: "-", default: "单元格数据为空时呈现的内容。TS 类型: string | TNode<BaseTableCellParams<T>>。通用类型定义"}, {attribute: "columns", description: "Array", type: "[]", default: "列配置，泛型 T 指表格数据类型。TS 类型: Array<BaseTableCol<T>>"}, {attribute: "data", description: "Array", type: "[]", default: "数据源，泛型 T 指表格数据类型。TS 类型: Array<T>"}, {attribute: "disableDataPage", description: "Boolean", type: "false", default: "是否禁用本地数据分页。当 data 数据长度超过分页大小时，会自动进行本地数据分页。如果 disableDataPage 设置为 true，则无论何时，都不会进行本地数据分页"}, {attribute: "empty", description: "String / Slot / Function", type: "", default: "空表格呈现样式，支持全局配置 GlobalConfigProvider。TS 类型: string | TNode。通用类型定义"}, {attribute: "firstFullRow", description: "String / Slot / Function", type: "-", default: "首行内容，横跨所有列。TS 类型: string | TNode。通用类型定义"}, {attribute: "fixedRows", description: "Array", type: "-", default: "固定行（冻结行），示例: [M, N]，表示冻结表头 M 行和表尾 N 行。M 和 N 值为 0 时，表示不冻结行。TS 类型: Array<number>"}, {attribute: "footData", description: "Array", type: "[]", default: "表尾数据源，泛型 T 指表格数据类型。TS 类型: Array<T>"}, {attribute: "footerAffixProps", description: "Object", type: "-", default: "已废弃。请更为使用 footerAffixedBottom。表尾吸底基于 Affix 组件开发，透传全部 Affix 组件属性。。TS 类型: Partial<AffixProps>"}, {attribute: "footerAffixedBottom", description: "Boolean / Object", type: "false", default: "表尾吸底。使用此向功能，需要非常注意表格是相对于哪一个父元素进行滚动。值为 true，则表示相对于整个窗口吸底。如果表格滚动的父元素不是整个窗口，请通过 footerAffixedBottom.container 调整固钉的吸顶范围。基于 Affix 组件开发，透传全部 Affix 组件属性。TS 类型: boolean | Partial<AffixProps>"}, {attribute: "footerSummary", description: "String / Slot / Function", type: "-", default: "表尾总结行。TS 类型: string | TNode。通用类型定义"}, {attribute: "headerAffixProps", description: "Object", type: "-", default: "已废弃。请更为使用 headerAffixedTop。表头吸顶基于 Affix 组件开发，透传全部 Affix 组件属性。TS 类型: Partial<AffixProps>, Affix API Documents。详细类型定义"}, {attribute: "headerAffixedTop", description: "Boolean / Object", type: "false", default: "表头吸顶。使用该功能，需要非常注意表格是相对于哪一个父元素进行滚动。值为 true，表示相对于整个窗口吸顶。如果表格滚动的父元素不是整个窗口，请通过 headerAffixedTop.container 调整吸顶的位置。基于 Affix 组件开发，透传全部 Affix 组件属性。。TS 类型: boolean | Partial<AffixProps>"}, {attribute: "height", description: "String / Number", type: "-", default: "表格高度，超出后会出现滚动条。示例: 100, '30%', '300'。值为数字类型，会自动加上单位 px。如果不是绝对固定表格高度，建议使用 maxHeight"}, {attribute: "horizontalScrollAffixedBottom", description: "Boolean / Object", type: "-"}]

", default: "滚动条吸底。基于 Affix 组件开发, 透传全部 Affix 组件属性。TS 类型: boolean | Partial<AffixProps>", {attribute: "hover", description: "Boolean", type: "false", default: "是否显示鼠标悬浮状态"}, {attribute: "lastFullRow", description: "String / Slot / Function", type: "-", default: "尾行内容, 横跨所有列。TS 类型: string | TNode。通用类型定义"}, {attribute: "loading", description: "Boolean / Slot / Function", type: "undefined", default: "加载中状态。值为 true 会显示默认加载中样式, 可以通过 Function 和 插槽 自定义加载状态呈现内容和样式。值为 false 则会取消加载状态。TS 类型: boolean | TNode。通用类型定义"}, {attribute: "loadingProps", description: "Object", type: "-", default: "透传加载组件全部属性。TS 类型: Partial<LoadingProps>, Loading API Documents。详细类型定义"}, {attribute: "maxHeight", description: "String / Number", type: "-", default: "表格最大高度, 超出后会出现滚动条。示例: 100, '30%', '300'。值为数字类型, 会自动加上单位 px"}, {attribute: "pagination", description: "Object", type: "-", default: "分页配置, 值为空则不显示。具体 API 参考分页组件。当 data 数据长度超过分页大小时, 会自动对本地数据 data 进行排序, 如果不希望对于 data 进行排序, 可以设置 disableDataPage = true。TS 类型: PaginationProps, Pagination API Documents。详细类型定义"}, {attribute: "paginationAffixedBottom", description: "Boolean / Object", type: "-", default: "分页吸底。基于 Affix 组件开发, 透传全部 Affix 组件属性。TS 类型: boolean | Partial<AffixProps>"}, {attribute: "resizable", description: "Boolean", type: "false", default: "是否允许调整列宽。如果想要配置宽度可调整的最小值和最大值, 请使用 column.resize, 示例: columns: [{ resize: { minWidth: 120, maxWidth: 300 } }]。默认规则: 因列宽超出存在横向滚动条时, 列宽调整仅影响当前列宽和总列宽; 表格列较少没有横向滚动条时, 列宽调整表现为自身宽度和相邻宽度变化"}, {attribute: "rowAttributes", description: "Object / Array / Function", type: "-", default: "HTML 标签 tr 的属性。类型为 Function 时, 参数说明: params.row 表示行数据; params.rowIndex 表示行下标; params.type=body 表示属性作用于 tbody 中的元素; params.type=foot 表示属性作用于 tfoot 中的元素。示例一: { draggable: true }, 示例二: [{ draggable: true }, { title: '超出省略显示' }]。示例三: () => [{ draggable: true }]。TS 类型: TableRowAttributes<T> type TableRowAttributes<T> = HTMLElementAttributes | ((params: { row: T; rowIndex: number; type: 'body' | 'foot' }) => HTMLElementAttributes) | Array<TableRowAttributes<T>>。通用类型定义。详细类型定义"}, {attribute: "rowClassName", description: "String / Object / Array / Function", type: "-", default: "行类名, 泛型 T 指表格数据类型。params.row 表示行数据; params.rowIndex 表示行下标; params.type=body 表示类名作用于 tbody 中的元素; params.type=tfoot 表示类名作用于 tfoot 中的元素。TS 类型: ClassName | ((params: RowClassNameParams<T>) => ClassName) interface RowClassNameParams<T> { row: T; rowIndex: number; type?: 'body' | 'tfoot' }。通用类型定义。详细类型定义"}, {attribute: "rowKey", description: "String", type: "id", default: "必需。唯一标识一行数据的字段名, 来源于 data 中的字段。如果是字段嵌套多层, 可以设置形如 item.a.id 的方法"}, {attribute: "rowspanAndColspan", description: "Function", type: "-", default: "用于自定义合并单元格, 泛型 T 指表格数据类型。示例: ({ row, col, rowIndex, colIndex }) => { rowspan: 2, colspan: 3 }。TS 类型: TableRowspanAndColspanFunc<T> type TableRowspanAndColspanFunc<T> = (params: BaseTableCellParams<T>) => RowspanColspan interface RowspanColspan { colspan?: number; rowspan?: number }。详细类型定义"}, {attribute: "rowspanAndColspanInFooter", description: "Function", type: "-", default: "用于自定义表尾的合并单元格, 泛型 T 指表格数据类型。示例: ({ row, col, rowIndex, colIndex }) => { rowspan: 2, colspan: 3 }。TS 类型: TableRowspanAndColspanFunc<T>"}, {attribute: "scroll", description: "Object", type: "-", default: "懒加载和虚拟滚动。为保证组件收益最大化, 当数据量小于阈值 scroll.threshold 时, 无论虚拟滚动的配置是否存在, 组件内部都不会开启虚拟滚动, scroll.threshold 默认为 100。TS 类型: TScroll。通用类型定义"}, {attribute: "showHeader", description: "Boolean", type: "true", default: "是否显示表头"}, {attribute: "size", description: "String", type: "medium", default: "表格尺寸。可选项: small/medium/large。TS 类型: SizeEnum。通用类型定义"}, {attribute: "stripe", description: "Boolean", type: "false", default: "是否显示斑马纹"}, {attribute: "tableContentWidth", description: "String", type: "-", default: "表格内容的总宽度, 注意不是表格可见宽度。主要应用于 table-layout: auto 模式下的固定列显示。tableContentWidth 内容宽度的值必须大于表格可见宽度"}, {attribute: "tableLayout", description: "String", type: "fixed", default: "表格布局方式。可选项: auto/fixed"}, {attribute: "topContent", description: "String / Slot / Function", type: "-", default: "表格顶部内容, 可以用于自定义列设置、顶部查询条件等。TS 类型: string | TNode。通用类型定义"}, {attribute: "verticalAlign", description: "String", type: "middle", default: "行内容上下方向对齐。可选项: top/middle/bottom"}, {attribute: "onCellClick", description: "Function", type: "", default: "TS 类型: (context: BaseTableCellEventContext<T>) => void 单元格点击时触发。详细类型定义。interface BaseTableCellEventContext<T> { row: T; col: BaseTableCol; rowIndex: number; colIndex: number; e: MouseEvent }"}, {attribute: "onPageChange", description: "Function", type: "", default: "TS 类型: (pageInfo: PageInfo, newDataSource: Array<T>) => void 分页发生变化时触发。参数 newDataSource 表示分页后的数据。本地数据进行分页时, newDataSource 和源数据 data 会不一样。泛型 T 指表格数据类型"}, {attribute: "onRowClick", description: "Function", type: "", default: "TS 类型: (context: RowEventContext<T>) => void 行点击时触发, 泛型 T 指表格数据类型。详细类型定义。interface RowEventContext<T> { row: T; index: number; e: MouseEvent }"}, {attribute: "onRowDbclick", description: "Function", type: "", default: "TS 类型: (context: RowEventContext<T>) => void 行双击时触发, 泛型 T 指表格数据类型"}, {attribute: "onRowMouseDown", description: "Function", type: "", default: "TS 类型: (context: RowEventContext<T>) => void 鼠标在表格行按下时触发, 泛型 T 指表格数据类型"}, {attribute: "onRowMouseenter", description: "Function", type: "", default: "TS 类型: (context: RowEventContext<T>) => void 鼠标在表格行进入时触发, 泛型 T 指表格数据类型"}, {attribute: "onRowMouseleave", description: "Function", type: "", default: "TS 类型: (context: RowEventContext<T>) => void 鼠标在表格行离开时触发, 泛型 T 指表格数据类型"},


```

{attribute:"onRowMouseover",description:"Function",type:"",default:"TS 类型: (context: RowEventContext<T>) => void鼠标悬浮到行时触发, 泛型 T 指表格数据类型"},
{attribute:"onRowMouseup",description:"Function",type:"",default:"TS 类型: (context: RowEventContext<T>) => void鼠标在表格行按下又弹起时触发, 泛型 T 指表格数据类型"},
{attribute:"onScroll",description:"Function",type:"",default:"TS 类型: (params: { e: WheelEvent }) => void表格内容滚动时触发"}, {attribute:"onScrollX",description:"Function",type:"",default:"TS 类型: (params: { e: WheelEvent }) => void已废弃。表格内容横向滚动时触发。请更为使用 onScroll 事件"},
{attribute:"onScrollY",description:"Function",type:"",default:"TS 类型: (params: { e: WheelEvent }) => void已废弃。表格内容纵向滚动时触发。当内容超出高度(height)或最大高度(max-height)时, 会出现纵向滚动条。请更为使用 onScroll 事件"}], {docTitle:"BaseTable Events说明",tableRows:[{attribute:"cell-click",description:"(context: BaseTableCellEventContext<T>)",type:"单元格点击时触发。详细类型定义。interface BaseTableCellEventContext<T> { row: T; col: BaseTableCol; rowIndex: number; colIndex: number; e: MouseEvent }"}, {attribute:"page-change",description:"(pageInfo: PageInfo, newDataSource: Array<T>)",type:"分页发生变化时触发。参数 newDataSource 表示分页后的数据。本地数据进行分页时, newDataSource 和源数据 data 会不一样。泛型 T 指表格数据类型"}, {attribute:"row-click",description:"(context: RowEventContext<T>)",type:"行点击时触发, 泛型 T 指表格数据类型。详细类型定义。interface RowEventContext<T> { row: T; index: number; e: MouseEvent }"}, {attribute:"row-dblclick",description:"(context: RowEventContext<T>)",type:"行双击时触发, 泛型 T 指表格数据类型"},
{attribute:"row-mousedown",description:"(context: RowEventContext<T>)",type:"鼠标在表格行按下时触发, 泛型 T 指表格数据类型"}, {attribute:"row-mouseenter",description:"(context: RowEventContext<T>)",type:"鼠标在表格行进入时触发, 泛型 T 指表格数据类型"}, {attribute:"row-mouseleave",description:"(context: RowEventContext<T>)",type:"鼠标在表格行离开时触发, 泛型 T 指表格数据类型"}, {attribute:"row-mouseover",description:"(context: RowEventContext<T>)",type:"鼠标悬浮到行时触发, 泛型 T 指表格数据类型"}, {attribute:"row-mouseup",description:"(context: RowEventContext<T>)",type:"鼠标在表格行按下又弹起时触发, 泛型 T 指表格数据类型"},
{attribute:"scroll",description:"(params: { e: WheelEvent })",type:"表格内容滚动时触发"}, {attribute:"scroll-x",description:"(params: { e: WheelEvent })",type:"已废弃。表格内容横向滚动时触发。请更为使用 onScroll 事件"}, {attribute:"scroll-y",description:"(params: { e: WheelEvent })",type:"已废弃。表格内容纵向滚动时触发。当内容超出高度(height)或最大高度(max-height)时, 会出现纵向滚动条。请更为使用 onScroll 事件"}], {docTitle:"BaseTableInstanceFunctions 组件实例方法说明",tableRows:[{attribute:"refreshTable",description:"-","type":"-",default:"必需。全部重新渲染表格"}, {attribute:"scrollColumnIntoView",description:"(colKey: string)","type":"-",default:"必需。横向滚动到指定列, 呈现在可视范围内"}, {attribute:"scrollToElement",description:"(params: ScrollToElementParams)","type":"-",default:"必需。虚拟滚动场景, 纵向滚动到指定行。示例: scrollToElement({ index: 100, top: 80, time: 200, behavior: 'smooth' })"}], {docTitle:"BaseTableCol 说明",tableRows:[{attribute:"align",description:"String",type:"left",default:"列横向对齐方式。可选项: left/right/center"}, {attribute:"attrs",description:"Object / Function",type":"-",default:"透传 HTML 属性到列元素。TS 类型: BaseTableColumnAttributes<T> type BaseTableColumnAttributes<T> = { [key: string]: any } | ((context: CellData<T>) => { [key: string]: any })。详细类型定义"}, {attribute:"cell",description:"String / Function",type":"-",default:"自定义单元格渲染。值类型为 Function 表示以函数形式渲染单元格。值类型为 string 表示使用插槽渲染, 插槽名称为 cell 的值。默认使用 colKey 作为插槽名称。优先级高于 render。泛型 T 指表格数据类型。TS 类型: string | TNode<BaseTableCellParams<T>> interface BaseTableCellParams<T> { row: T; rowIndex: number; col: BaseTableCol<T>; colIndex: number }。通用类型定义。详细类型定义"}, {attribute:"children",description:"Array",type":"-",default:"用于多级表头, 泛型 T 指表格数据类型。TS 类型: Array<BaseTableCol<T>>"}, {attribute:"className",description:"String / Object / Array / Function",type":"-",default:"列类名, 值类型是 Function 使用返回值作为列类名; 值类型不为 Function 时, 值用于整列类名(含表头)。泛型 T 指表格数据类型。TS 类型: TableColumnClassName<T> | TableColumnClassName<T>[] type TableColumnClassName<T> = ClassName | ((context: CellData<T>) => ClassName) interface CellData<T> extends BaseTableCellParams<T> { type: 'th' | 'td' }。通用类型定义。详细类型定义"}, {attribute:"colKey",description:"String",type":"-",default:"渲染列所需字段, 值为 serial-number 表示当前列为「序号」列"}, {attribute:"colspan",description:"Number",type":"-",default:"单行表头合并列。多行表头请参考「多级表头」文档示例"}, {attribute:"ellipsis",description:"Boolean / Object / Slot / Function",type:"false",default:"单元格和表头内容超出时, 是否显示省略号。如果仅希望单元格超出省略, 可设置 ellipsisTitle = false。值为 true, 则超出省略浮层默认显示单元格内容; 值类型为 Function 则自定义超出省略浮层显示的内容; 值类型为 Object, 则自动透传属性到 Tooltip 组件, 可用于调整浮层背景色和方向等特性。同时透传 Tooltip 属性和自定义浮层内容, 请使用 { props: { theme: 'light' }, content: () => 'something' }。TS 类型: boolean | TNode<BaseTableCellParams<T>> | TooltipProps | { props: TooltipProps; content: TNode<BaseTableCellParams<T>> }}, Tooltip API Documents。通用类型定义。详细类型定义"}, {attribute:"ellipsisTitle",description:"Boolean / Object / Slot / Function",type:"undefined",default:"表头内容超出时, 是否显示省略号。优先级高于 ellipsis。值为 true, 则超出省略的浮层默认显示表头全部内容; 值类型为 Function 用于自定义超出省略浮层显示的表头内容; 值类型为 Object, 则自动透传属性到 Tooltip 组件, 则自动透传属性到 Tooltip 组件, 可用于调整浮层背景色和方向等特性。同时透传 Tooltip 属性和自定义浮层内容, 请使用 { props: { theme: 'light' }, content: () => 'something' }。TS 类型: boolean | TNode<BaseTableColParams<T>> | TooltipProps | { props: TooltipProps; content: TNode<BaseTableColParams<T>> } interface BaseTableColParams<T> { col: BaseTableCol<T>; colIndex: number }。通用类型定义。详细类型定义"}, {attribute:"fixed",description:"String",type:"left",default:"固定列显示位置。可选项: left/right"}],

```

{attribute:"foot",description:"String / Function",type:"-","default:"自定义表尾表尾。值类型为 Function 表示以函数形式渲染表尾内容。值类型为 string 表示使用插槽渲染,插槽名称为 foot 值。TS 类型: string | TNode<{ col: BaseTableCol; colIndex: number }>。通用类型定义"},

{attribute:"minWidth",description:"String / Number",type:"-","default:"透传 CSS 属性 min-width 到 <col> 元素。⚠ 仅少部分浏览器支持,如:使用 TablesNG 渲染的 Chrome 浏览器支持 minWidth",

{attribute:"render",description:"Function",type:"-","default:"自定义表头或单元格,泛型 T 指表格数据类型。TS 类型: TNode<BaseTableRenderParams<T>> interface BaseTableRenderParams<T> extends BaseTableCellParams<T> { type: RenderType } type RenderType = 'cell' | 'title'。通用类型定义。详细类型定义"},

{attribute:"resizable",description:"Boolean",type:"true",default:"是否允许调整当前列列宽"},

{attribute:"resize",description:"Object",type:"-","default:"限制拖拽调整的最小宽度和最大宽度。resize.minWidth 默认为 80,resize.maxWidth 默认为 600。TS 类型: TableColumnResizeConfig interface TableColumnResizeConfig { minWidth: number; maxWidth: number }。详细类型定义"},

{attribute:"stopPropagation",description:"Boolean",type:"-","default:"是否阻止当列单元格点击事件冒泡"},

{attribute:"thClassName",description:"String / Object / Array / Function",type:"-","default:"列表头类名,值类型是函数时使用返回值作为列类名。泛型 T 指表格数据类型。TS 类型: TableColumnClassName<T> | TableColumnClassName<T>[]。通用类型定义"},

{attribute:"title",description:"String / Function",type:"-","default:"自定义表头渲染。值类型为 Function 表示以函数形式渲染表头。值类型为 string 表示使用插槽渲染,插槽名称为 title 的值。优先级高于 render。TS 类型: string | TNode<{ col: BaseTableCol; colIndex: number }>。通用类型定义"},

{attribute:"width",description:"String / Number",type:"-","default:"列宽,可以作为最小宽度使用。当列宽总和小于 table 元素时,浏览器根据宽度设置情况自动分配宽度;当列宽总和大于 table 元素,表现为定宽。可以同时调整 table 元素的宽度来达到自己想要的效果"}],{docTitle:"PrimaryTable Props说明",tableRows: [{attribute:"asyncLoading",description:"String / Slot / Function",type:"-","default:"异步加载状态。值为 loading 显示默认文字“正在加载中,请稍后”,值为 loading-more 显示“点击加载更多”,值为其他,表示完全自定义异步加载区域内容。TS 类型: 'loading' | 'load-more' | TNode。通用类型定义"},

{attribute:"columnController",description:"Object",type:"-","default:"自定义显示列控制器,值为空不会显示。具体属性请看下方 TableColumnController 文档。TS 类型: TableColumnController"},

{attribute:"columnControllerVisible",description:"Boolean",type:"undefined",default:"是否显示列配置弹框控制器,只要该属性值不为 undefined,弹框的显示/隐藏完全由该属性控制。支持语法糖 v-model:columnControllerVisible"},

{attribute:"defaultColumnControllerVisible",description:"Boolean",type:"undefined",default:"是否显示列配置弹框控制器,只要该属性值不为 undefined,弹框的显示/隐藏完全由该属性控制。非受控属性"},

{attribute:"columns",description:"Array",type:"[]",default:"列配置,泛型 T 指表格数据类型。TS 类型: Array<PrimaryTableCol<T>>"},{attribute:"displayColumns",description:"Array",type:"-","default:"列配置功能中,当前显示的列。支持语法糖 v-model:displayColumns。TS 类型: CheckboxGroupValue"},

{attribute:"defaultDisplayColumns",description:"Array",type:"-","default:"列配置功能中,当前显示的列。非受控属性。TS 类型: CheckboxGroupValue"},{attribute:"dragSort",description:"String",type:"-","default:"拖拽排序方式,值为 row 表示行拖拽排序,这种方式无法进行文本复制,慎用。值为 row-handler 表示通过拖拽手柄进行行拖拽排序。值为 col 表示列顺序拖拽。值为 row-handler-col 表示同时支持行拖拽和列拖拽。⚠ drag-col 已废弃,请勿使用。。可选项: row/row-handler/col/row-handler-col/drag-col"},

{attribute:"dragSortOptions",description:"Object",type:"-","default:"拖拽排序扩展参数,具体参数见 Sortable。TS 类型: SortableOptions"},{attribute:"editableCellState",description:"Function",type:"-","default:"单元格是否允许编辑。返回值为 true 则表示可编辑;返回值为 false 则表示不可编辑,只读状态。TS 类型: EditableCellType<T> type EditableCellType<T> = (params: PrimaryTableCellParams<T>) => boolean。详细类型定义"},{attribute:"editableRowKeys",description:"Array",type:"-","default:"处于编辑状态的行。TS 类型: Array<string | number>"},{attribute:"expandIcon",description:"Boolean / Slot / Function",type:"true",default:"用于控制是否显示「展开图标列」,值为 false 则不会显示。可以精确到某一行是否显示,还可以自定义展开图标内容。expandedRow 存在时,该参数有效。支持全局配置 GlobalConfigProvider。TS 类型: boolean | TNode<ExpandArrowRenderParams<T>> interface ExpandArrowRenderParams<T> { row: T; index: number }。通用类型定义。详细类型定义"},

{attribute:"expandOnRowClick",description:"Boolean",type:"-","default:"是否允许点击行展开"},

{attribute:"expandedRow",description:"String / Slot / Function",type:"-","default:"展开行内容,泛型 T 指表格数据类型。TS 类型: TNode<TableExpandedRowParams<T>> interface TableExpandedRowParams<T> { row: T; index: number; columns: PrimaryTableCol<T>[] | BaseTableCol<T>[] }。通用类型定义。详细类型定义"},

{attribute:"expandedRowKeys",description:"Array",type:"[]",default:"展开行。支持语法糖 v-model:expandedRowKeys。TS 类型: Array<string | number>"},

{attribute:"defaultExpandedRowKeys",description:"Array",type:"[]",default:"展开行。非受控属性。TS 类型: Array<string | number>"},{attribute:"filterIcon",description:"Slot / Function",type:"-","default:"自定义过滤图标,支持全局配置 GlobalConfigProvider。TS 类型: TNode<{ col: PrimaryTableCol<T>; colIndex: number }>。通用类型定义"},{attribute:"filterRow",description:"String / Slot / Function",type:"-","default:"自定义过滤状态行及清空筛选等。TS 类型: string | TNode。通用类型定义"},{attribute:"filterValue",description:"Object",type:"-","default:"过滤数据的值。支持语法糖 v-model:filterValue。TS 类型: FilterValue type FilterValue = { [key: string]: any }。详细类型定义"},

{attribute:"defaultFilterValue",description:"Object",type:"-","default:"过滤数据的值。非受控属性。TS 类型: FilterValue type FilterValue = { [key: string]: any }。详细类型定义"},

{attribute:"hideSortTips",description:"Boolean",type:"-","default:"隐藏排序文本提示,支持全局配置 GlobalConfigProvider,默认全局配置值为 false"},

{attribute:"indeterminateSelectedRowKeys",description:"Array",type:"-","default:"半选状态行。选中行请更为使用 selectedRowKeys 控制。TS 类型: Array<string | number>"},

```

{attribute: "multipleSort", description: "Boolean", type: "false", default: "是否支持多列排序"},
{attribute: "reserveSelectedRowOnPaginate", description: "Boolean", type: "true", default: "行选中功能，是否在分页时保留上一页选中结果不清空，本地数据分页场景下，会全选所有页数据。值为 false 则表示全部选中操作停留在当前页，不跨分页；本地数据分页场景下，全选仅选中当前页"},
{attribute: "selectOnRowClick", description: "Boolean", type: "-", default: "是否在点击整行时选中"},
{attribute: "selectedRowKeys", description: "Array", type: "[]", default: "选中行，控制属性。半选状态行请更为使用 indeterminateSelectedRowKeys 控制。支持语法糖 v-model:selectedRowKeys。TS 类型: Array<string | number>"}, {attribute: "defaultSelectedRowKeys", description: "Array", type: "[]", default: "选中行，控制属性。半选状态行请更为使用 indeterminateSelectedRowKeys 控制。非受控属性。TS 类型: Array<string | number>"}, {attribute: "showSortColumnBgColor", description: "Boolean", type: "false", default: "当前排序列是否显示背景色"}, {attribute: "sort", description: "Object / Array", type: "-", default: "排序控制。sortBy 排序字段; descending 是否进行降序排列。值为数组时，表示正进行多字段排序。支持语法糖 v-model:sort。TS 类型: TableSort type TableSort = SortInfo | Array<SortInfo> interface SortInfo { sortBy: string; descending: boolean }。详细类型定义"}, {attribute: "defaultSort", description: "Object / Array", type: "-", default: "排序控制。sortBy 排序字段; descending 是否进行降序排列。值为数组时，表示正进行多字段排序。非受控属性。TS 类型: TableSort type TableSort = SortInfo | Array<SortInfo> interface SortInfo { sortBy: string; descending: boolean }。详细类型定义"}, {attribute: "sortIcon", description: "Slot / Function", type: "-", default: "自定义排序图标，支持全局配置 GlobalConfigProvider。TS 类型: TNode。通用类型定义"}, {attribute: "sortOnRowDraggable", description: "Boolean", type: "false", default: "已废弃。允许表格行拖拽时排序。请更为使用 dragSort=\\\"row\\\""}, {attribute: "Omit<BaseTableProps<T>, 'columns' | 'onCellClick'>", description: "-", type: "-", default: "继承 Omit<BaseTableProps<T>, 'columns' | 'onCellClick'> 中的全部属性"},
{attribute: "onAsyncLoadingClick", description: "Function", type: "", default: "TS 类型: (context: { status: 'loading' | 'load-more' }) => void异步加载区域被点击时触发"},
{attribute: "onCellClick", description: "Function", type: "", default: "TS 类型: (context: PrimaryTableCellEventContext<T>) => void单元格点击时触发。详细类型定义。interface PrimaryTableCellEventContext<T> { row: T; col: PrimaryTableCol; rowIndex: number; colIndex: number; e: MouseEvent }"}, {attribute: "onChange", description: "Function", type: "", default: "TS 类型: (data: TableChangeData, context: TableChangeContext<T>) => void分页、排序、过滤等内容变化时触发，泛型 T 指表格数据类型，currentData 表示变化后的数据。详细类型定义。interface TableChangeData { sorter?: TableSort; filter?: FilterValue; pagination?: PaginationProps }interface TableChangeContext<T> { trigger: TableChangeTrigger; currentData?: T[] }type TableChangeTrigger = 'filter' | 'sorter' | 'pagination'"}, {attribute: "onColumnChange", description: "Function", type: "", default: "TS 类型: (context: PrimaryTableColumnChange<T>) => void确认操作之前列配置发生变化时触发。context.columns 表示已选中的列; context.currentColumn 表示本次变化操作的列，值不存在表示全选操作; context.type 表示当前操作属于选中列或是取消列。详细类型定义。interface PrimaryTableColumnChange<T> { columns?: CheckboxGroupValue; currentColumn?: PrimaryTableCol<T>; type?: 'check' | 'uncheck'; e?: Event }"}, {attribute: "onColumnControllerVisibleChange", description: "Function", type: "", default: "TS 类型: (visible: boolean, context: { trigger: 'cancel' | 'confirm' }) => void列配置弹窗显示或隐藏变化时触发"}, {attribute: "onDataChange", description: "Function", type: "", default: "TS 类型: (data: Array<T>, context: TableDataChangeContext) => void本地数据排序导致 data 变化时触发，第一个参数指变化后的数据，第二个参数 context.trigger 表示触发本次变化的来源。详细类型定义。interface TableDataChangeContext { trigger: 'sort' }"}, {attribute: "onDisplayColumnsChange", description: "Function", type: "", default: "TS 类型: (value: CheckboxGroupValue) => void确认列配置时触发。详细类型定义。import { CheckboxGroupValue } from '@checkbox'"}, {attribute: "onDragSort", description: "Function", type: "", default: "TS 类型: (context: DragSortContext<T>) => void拖拽排序时触发，data 表示排序前的数据，newData 表示拖拽排序结束后的新数据，sort=row 表示行拖拽事件触发，sort=col 表示列拖拽事件触发。详细类型定义。interface DragSortContext<T> { currentIndex: number; current: T; targetIndex: number; target: T; data: T[]; newData: T[]; currentData?: T[]; e: SortableEvent; sort: 'row' | 'col' }import { SortableEvent, SortableOptions } from 'sortablejs'"}, {attribute: "onExpandChange", description: "Function", type: "", default: "TS 类型: (expandedRowKeys: Array<string | number>, options: ExpandOptions<T>) => void展开行发生变化时触发，泛型 T 指表格数据类型。详细类型定义。interface ExpandOptions<T> { expandedRowData: Array<T>; currentRowData: T }"}, {attribute: "onFilterChange", description: "Function", type: "", default: "TS 类型: (filterValue: FilterValue, context: { col?: PrimaryTableCol<T> }) => void过滤参数发生变化时触发，泛型 T 指表格数据类型"}, {attribute: "onRowEdit", description: "Function", type: "", default: "TS 类型: (context: PrimaryTableRowEditContext<T>) => void行编辑时触发。详细类型定义。type PrimaryTableRowEditContext<T> = PrimaryTableCellParams<T> & { value: any; editedRow: T }"}, {attribute: "onRowValidate", description: "Function", type: "", default: "TS 类型: (context: PrimaryTableRowValidateContext<T>) => void行编辑校验完成后触发，即组件实例方法 validateRowData 执行结束后触发。result 表示校验结果，trigger=self 表示编辑组件内部触发的校验，trigger='parent' 表示表格父组件触发的校验。详细类型定义。type PrimaryTableRowValidateContext<T> = { result: TableRowValidateResult<T>[]; trigger: TableValidateTrigger }type TableValidateTrigger = 'self' | 'parent'export type TableRowValidateResult<T> = PrimaryTableCellParams<T> & { errorList: AllValidateResult[]; value: any }"}, {attribute: "onSelectChange", description: "Function", type: "", default: "TS 类型: (selectedRowKeys: Array<string | number>, options: SelectOptions<T>) => void选中行发生变化时触发，泛型 T 指表格数据类型。两个参数，第一个参数为选中行 keys，第二个参数为更多参数，具体如下: type = uncheck 表示当前行操作为「取消行选中」; type = check 表示当前行操作为「行选中」; currentRowKey 表示当前操作行的 rowKey

```

值；currentRowData 表示当前操作行的行数据。详细类型定义。interface SelectOptions<T> { selectedRowData: Array<T>; type: 'uncheck' | 'check'; currentRowKey?: string; currentRowData?: T }", {attribute: "onSortChange", description: "Function", type: "", default: "TS 类型: (sort: TableSort, options: SortOptions<T>) => void 排序发生变化时触发。其中 sortBy 表示当前排序的字段, sortType 表示排序的方式, currentDataSource 表示 sorter 排序后的结果, col 表示列配置。sort 值类型为数组时表示多字段排序。详细类型定义。interface SortOptions<T> { currentDataSource?: Array<T>; col: PrimaryTableCol }", {attribute: "onValidate", description: "Function", type: "", default: "TS 类型: (context: PrimaryTableValidateContext) => void 可编辑行表格, 全部数据校验完成后触发。即组件实例方法 validateTableData 执行结束后触发。详细类型定义。interface PrimaryTableValidateContext { result: TableErrorListMap } type TableErrorListMap = { [key: string]: AllValidateResult[] }"}], {docTitle: "PrimaryTable Events 说明", tableRows: [{attribute: "async-loading-click", description: "(context: { status: 'loading' | 'load-more' })", type: "异步加载区域被点击时触发"}, {attribute: "cell-click", description: "(context: PrimaryTableCellEventContext<T>)", type: "单元格点击时触发。详细类型定义。interface PrimaryTableCellEventContext<T> { row: T; col: PrimaryTableCol; rowIndex: number; colIndex: number; e: MouseEvent }", {attribute: "change", description: "(data: TableChangeData, context: TableChangeContext<T>)", type: "分页、排序、过滤等内容变化时触发, 泛型 T 指表格数据类型, currentData 表示变化后的数据。详细类型定义。interface TableChangeData { sorter?: TableSort; filter?: FilterValue; pagination?: PaginationProps } interface TableChangeContext<T> { trigger: TableChangeTrigger; currentData?: T[] } type TableChangeTrigger = 'filter' | 'sorter' | 'pagination'"}, {attribute: "column-change", description: "(context: PrimaryTableColumnChange<T>)", type: "确认操作之前列配置发生变化时触发。context.columns 表示已选中的列; context.currentColumn 表示本次变化操作的列, 值不存在表示全选操作; context.type 表示当前操作属于选中列或是取消列。详细类型定义。interface PrimaryTableColumnChange<T> { columns?: CheckboxGroupValue; currentColumn?: PrimaryTableCol<T>; type?: 'check' | 'uncheck'; e?: Event }", {attribute: "column-controller-visible-change", description: "(visible: boolean, context: { trigger: 'cancel' | 'confirm' })", type: "列配置弹窗显示或隐藏变化时触发"}, {attribute: "data-change", description: "(data: Array<T>, context: TableDataChangeContext)", type: "本地数据排序导致 data 变化时触发, 第一个参数指变化后的数据, 第二个参数 context.trigger 表示触发本次变化的来源。详细类型定义。interface TableDataChangeContext { trigger: 'sort' }", {attribute: "display-columns-change", description: "(value: CheckboxGroupValue)", type: "确认列配置时触发。详细类型定义。import { CheckboxGroupValue } from '@checkbox'"}, {attribute: "drag-sort", description: "(context: DragSortContext<T>)", type: "拖拽排序时触发, data 表示排序前的数据, newData 表示拖拽排序结束后的新数据, sort=row 表示行拖拽事件触发, sort=col 表示列拖拽事件触发。详细类型定义。interface DragSortContext<T> { currentIndex: number; current: T; targetIndex: number; target: T; data: T[]; newData: T[]; currentData?: T[]; e: SortableEvent; sort: 'row' | 'col' } import { SortableEvent, SortableOptions } from 'sortablejs'"}, {attribute: "expand-change", description: "(expandedRowKeys: Array<string | number>, options: ExpandOptions<T>)", type: "展开行发生变化时触发, 泛型 T 指表格数据类型。详细类型定义。interface ExpandOptions<T> { expandedRowData: Array<T>; currentRowData: T }", {attribute: "filter-change", description: "(filterValue: FilterValue, context: { col?: PrimaryTableCol<T> })", type: "过滤参数发生变化时触发, 泛型 T 指表格数据类型"}, {attribute: "row-edit", description: "(context: PrimaryTableRowEditContext<T>)", type: "行编辑时触发。详细类型定义。type PrimaryTableRowEditContext<T> = PrimaryTableCellParams<T> & { value: any; editedRow: T }", {attribute: "row-validate", description: "(context: PrimaryTableRowValidateContext<T>)", type: "行编辑校验完成后触发, 即组件实例方法 validateRowData 执行结束后触发。result 表示校验结果, trigger=self 表示编辑组件内部触发的校验, trigger='parent' 表示表格父组件触发的校验。详细类型定义。type PrimaryTableRowValidateContext<T> = { result: TableRowValidateResult<T>[]; trigger: TableValidateTrigger } type TableValidateTrigger = 'self' | 'parent' export type TableRowValidateResult<T> = PrimaryTableCellParams<T> & { errorList: AllValidateResult[]; value: any }", {attribute: "select-change", description: "(selectedRowKeys: Array<string | number>, options: SelectOptions<T>)", type: "选中行发生变化时触发, 泛型 T 指表格数据类型。两个参数, 第一个参数为选中行 keys, 第二个参数为更多参数, 具体如下: type = uncheck 表示当前行操作为「取消行选中」; type = check 表示当前行操作为「行选中」; currentRowKey 表示当前操作行的 rowKey 值; currentRowData 表示当前操作行的行数据。详细类型定义。interface SelectOptions<T> { selectedRowData: Array<T>; type: 'uncheck' | 'check'; currentRowKey?: string; currentRowData?: T }", {attribute: "sort-change", description: "(sort: TableSort, options: SortOptions<T>)", type: "排序发生变化时触发。其中 sortBy 表示当前排序的字段, sortType 表示排序的方式, currentDataSource 表示 sorter 排序后的结果, col 表示列配置。sort 值类型为数组时表示多字段排序。详细类型定义。interface SortOptions<T> { currentDataSource?: Array<T>; col: PrimaryTableCol }", {attribute: "validate", description: "(context: PrimaryTableValidateContext)", type: "可编辑行表格, 全部数据校验完成后触发。即组件实例方法 validateTableData 执行结束后触发。详细类型定义。interface PrimaryTableValidateContext { result: TableErrorListMap } type TableErrorListMap = { [key: string]: AllValidateResult[] }"}], {docTitle: "PrimaryTable Instance Functions 组件实例方法说明", tableRows: [{attribute: "validateRowData", description: "(rowValue: any)", type: "Promise<{ trigger: TableValidateTrigger, result: ErrorListObjectType<T>[] }>", default: "必需。校验行信息, 校验完成后, 会触发事件 onRowValidate。参数 rowValue 表示行唯一标识的值。详细类型定义。type ErrorListObjectType<T> = PrimaryTableRowEditContext<T> & { errorList: AllValidateResult[] }", {attribute: "validateTableData", description: "-", type: "Promise<TableErrorListMap>", default: "必需。校验表格全部数据, 校验完成后, 会触发事件 onValidate"}], {docTitle: "PrimaryTable Col 说明", tableRows: [{attribute: "cell", description: "String / Function", type: "-", default: "自定义单元格渲染。值类型为

Function 表示以函数形式渲染单元格。值类型为 string 表示使用插槽渲染，插槽名称为 cell 的值。默认使用 colKey 作为插槽名称。优先级高于 render。泛型 T 指表格数据类型。TS 类型：string | TNode<PrimaryTableCellParams<T>> interface PrimaryTableCellParams<T> { row: T; rowIndex: number; col: PrimaryTableCol<T>; colIndex: number }。通用类型定义。详细类型定义”，

{attribute: "checkProps", description: "Object / Function", type: "-", default: "透传参数，colKey 值为 row-select 时，配置有效。具体定义参考 Checkbox 组件 和 Radio 组件。泛型 T 指表格数据类型。TS 类型：CheckProps<T> type CheckProps<T> = CheckboxProps | RadioProps | ((options: { row: T; rowIndex: number }) => CheckboxProps | RadioProps) import { CheckboxProps } from '@Checkbox', Radio API Documents。详细类型定义”，

{attribute: "children", description: "Array", type: "-", default: "用于多级表头，泛型 T 指表格数据类型。TS 类型：Array<PrimaryTableCol<T>>”，

{attribute: "colKey", description: "String", type: "-", default: "渲染列所需字段，必须唯一。值为 row-select 表示当前列为行选中操作列。值为 drag 表示当前列为拖拽排序操作列。值为 serial-number 表示当前列为「序号」列”，

{attribute: "disabled", description: "Function", type: "-", default: "是否禁用行选中，colKey 值为 row-select 时，配置有效。TS 类型：(options: { row: T; rowIndex: number }) => boolean”，

{attribute: "edit", description: "Object", type: "-", default: "可编辑单元格配置项，具体属性参考文档 TableEditableCellConfig 描述。TS 类型：TableEditableCellConfig<T>”，

{attribute: "filter", description: "Object", type: "-", default: "过滤规则，支持多选(multiple)、单选(single)、输入框(input) 等三种形式。想要自定义过滤组件，可通过 filter.component 实现，自定义过滤组件需要包含参数 value 和事件 change。更多信息请查看当前页面中 TableColumnFilter 的详细文档。TS 类型：TableColumnFilter”，

{attribute: "render", description: "Function", type: "-", default: "自定义表头或单元格，泛型 T 指表格数据类型。TS 类型：TNode<PrimaryTableRenderParams<T>> interface PrimaryTableRenderParams<T> extends PrimaryTableCellParams<T> { type: RenderType }。通用类型定义。详细类型定义”，

{attribute: "sortType", description: "String", type: "all", default: "当前列支持排序的方式，desc 表示当前列只能进行降序排列；asc 表示当前列只能进行升序排列；all 表示当前列既可升序排列，又可以降序排列。可选项：desc/asc/all。TS 类型：SortType type SortType = 'desc' | 'asc' | 'all'。详细类型定义”，

{attribute: "sorter", description: "Boolean / Function", type: "false", default: "该列是否支持排序。值为 true 表示该列支持排序；值类型为函数，表示对本地数据 data 进行排序，返回值参考 MDN Array.sort。泛型 T 指表格数据类型。TS 类型：boolean | SorterFun<T> type SorterFun<T> = (a: T, b: T) => number。详细类型定义”，

{attribute: "title", description: "String / Function", type: "-", default: "自定义表头渲染。值类型为 Function 表示以函数形式渲染表头。值类型为 string 表示使用插槽渲染，插槽名称为 title 的值。优先级高于 render。TS 类型：string | TNode<{ col: PrimaryTableCol; colIndex: number }>。通用类型定义”，

{attribute: "type", description: "String", type: "single", default: "colKey 值为 row-select 时表示行选中列，有两种模式：单选和多选。type=single 表示单选，type=multiple 表示多选。可选项：single/multiple”，

{attribute: "omit", description: "Omit<BaseTableCol, 'cell' | 'title' | 'render' | 'children'>", type: "-", default: "继承 Omit<BaseTableCol, 'cell' | 'title' | 'render' | 'children'> 中的全部属性”}], {docTitle: "EnhancedTable Props说明", tableRows: [

{attribute: "beforeDragSort", description: "Function", type: "-", default: "树形结构中，拖拽排序前控制，返回值为 true 则继续排序；返回值为 false 则中止排序还原数据。TS 类型：(context: DragSortContext<T>) => boolean”，

{attribute: "tree", description: "Object", type: "-", default: "树形结构相关配置。具体属性文档查看 TableTreeConfig 相关描述。TS 类型：TableTreeConfig”，

{attribute: "treeExpandAndFoldIcon", description: "Function", type: "-", default: "自定义树形结构展开图标，支持全局配置 GlobalConfigProvider。TS 类型：TNode<{ type: 'expand' | 'fold' }>。通用类型定义”，

{attribute: "PrimaryTableProps<T>", description: "-", type: "-", default: "继承 PrimaryTableProps<T> 中的全部属性”，

{attribute: "onAbnormalDragSort", description: "Function", type: "", default: "TS 类型：(context: TableAbnormalDragSortContext<T>) => void异常拖拽排序时触发，如：树形结构中，非同层级之间的交换。context.code 指交换异常错误码，固定值；context.reason 指交换异常的原因。详细类型定义。interface TableAbnormalDragSortContext<T> { code: number; reason: string }”，

{attribute: "onTreeExpandChange", description: "Function", type: "", default: "TS 类型：(context: TableTreeExpandChangeContext<T>) => void树形结构，用户操作引起节点展开或收起时触发，代码操作不会触发。详细类型定义。interface TableTreeExpandChangeContext<T> { row: T; rowIndex: number; rowState: TableRowState<T>; trigger?: 'expand-fold-icon' | 'row-click' }”}], {docTitle: "EnhancedTable Events说明", tableRows: [

{attribute: "abnormal-drag-sort", description: "(context: TableAbnormalDragSortContext<T>)", type: "异常拖拽排序时触发，如：树形结构中，非同层级之间的交换。context.code 指交换异常错误码，固定值；context.reason 指交换异常的原因。详细类型定义。interface TableAbnormalDragSortContext<T> { code: number; reason: string }”，

{attribute: "tree-expand-change", description: "(context: TableTreeExpandChangeContext<T>)", type: "树形结构，用户操作引起节点展开或收起时触发，代码操作不会触发。详细类型定义。interface TableTreeExpandChangeContext<T> { row: T; rowIndex: number; rowState: TableRowState<T>; trigger?: 'expand-fold-icon' | 'row-click' }”}], {docTitle: "EnhancedTable Instance Functions 组件实例方法说明", tableRows: [

{attribute: "appendTo", description: "(key: TableRowValue, newData: T)", type: "-", default: "必需。树形结构中，为当前节点添加子节点。如果 key 为空，则表示为根节点添加子节点”，

{attribute: "expandAll", description: "-", type: "-", default: "必需。展开全部行”，

{attribute: "foldAll", description: "-", type: "-", default: "必需。折叠全部行”，

{attribute: "getData", description: "(key: TableRowValue)", type: "TableRowState<T>", default: "必需。树形结构中，用于获取行数据所有信息。泛型 T 表示行数据类型。详细类型定义。type TableRowValue = string | number”，

{attribute: "getTreeExpandedRow", description: "(type: 'unique' | 'data' | 'all')", type: "-", default: "必需。获取展开的树形节点。type=unique 标识获取展开节点的行唯一标识值，type=data 表示获取展开节点的数据，type=all 表示获取行节点包含展开状态的全部数据”，

{attribute: "getTreeNode", description: "-", type: "T[]", default: "必需。树形结构中，获取完整的树形结构”，


```

{attribute:"insertAfter",description:"(key: TableRowValue, newData: T)",type:"-",default:"必需。树形结构中，在当前节点之后添加子节点"},{attribute:"insertBefore",description:"(key: TableRowValue, newData: T)",type:"-",default:"必需。树形结构中，在当前节点之前添加子节点"},{attribute:"remove",description:"(key: TableRowValue)",type:"-",default:"必需。树形结构中，移除指定节点"},{attribute:"resetData",description:"(newData: T[])",type:"-",default:"必需。重置或更新整个表格数据"},{attribute:"setData",description:"(key: TableRowValue, newRowData: T)",type:"-",default:"必需。树形结构中，用于更新行数据。泛型 T 表示行数据类型"},{attribute:"swapData",description:"(params: SwapParams<T>)",type:"-",default:"必需。树形结构中，交换两个节点的顺序。详细类型定义。interface SwapParams<T> { current: T; target: T; currentIndex: number; targetIndex: number }"},{attribute:"toggleExpandData",description:"(p: { row: T, rowIndex: number})",type:"-",default:"必需。展开或收起树形行"}]],{docTitle:"TableRowState说明",tableRows:[{attribute:"disabled",description:"Boolean",type:"false",default:"表格行是否禁用选中"},{attribute:"expandChildrenLength",description:"Number",type:"-",default:"当前节点展开的子节点数量"},{attribute:"expanded",description:"Boolean",type:"false",default:"必需。表格行是否展开"},{attribute:"id",description:"String / Number",type:"-",default:"必需。唯一标识"},{attribute:"level",description:"Number",type:"-",default:"当前节点层级。TS 类型: number"},{attribute:"parent",description:"-",type:"-",default:"父节点。TS 类型: TableRowState<T>"},{attribute:"path",description:"Array",type:"-",default:"当前节点路径。TS 类型: TableRowState<T>[]"},{attribute:"row",description:"-",type:"-",default:"必需。原始表格行数据。TS 类型: T"},{attribute:"rowIndex",description:"Number",type:"-",default:"必需。表格行下标，值为 -1 标识当前行未展开显示"}]],{docTitle:"TableColumnFilter说明",tableRows:[{attribute:"component",description:"Slot / Function",type:"-",default:"用于自定义筛选器，只要保证自定义筛选器包含 value 属性和 change 事件，即可像内置筛选器一样正常使用。示例: component: DatePicker。TS 类型: ComponentType。通用类型定义"},{attribute:"confirmEvents",description:"Array",type:"-",default:"哪些事件触发后会进行过滤搜索（确认按钮无需配置，会默认触发搜索）。输入框组件示例: confirmEvents: ['onEnter']。TS 类型: string[]"},{attribute:"list",description:"Array",type:"-",default:"用于配置当前筛选器可选值有哪些，仅当 filter.type 等于 single 或 multiple 时有效。TS 类型: Array<OptionData>。通用类型定义"},{attribute:"popupProps",description:"Object",type:"-",default:"透传 Popup 组件全部属性到筛选器浮层。TS 类型: PopupProps, Popup API Documents。详细类型定义"},{attribute:"props",description:"Array",type:"-",default:"用于透传筛选器属性，可以对筛选器进行任何原组件支持的属性配置。TS 类型: FilterProps type FilterProps = RadioProps | CheckboxProps | InputProps | { [key: string]: any }, Input API Documents。详细类型定义"},{attribute:"resetValue",description:"-",type:"-",default:"重置时设置的值，示例: '' 或 []。TS 类型: any"},{attribute:"showConfirmAndReset",description:"Boolean",type:"false",default:"是否显示重置和确认。值为真，过滤事件（filter-change）会在确定时触发；值为假，则数据变化时会立即触发过滤事件"},{attribute:"type",description:"String",type:"-",default:"用于设置筛选器类型：单选按钮筛选器、复选框筛选器、输入框筛选器。TS 类型: FilterType type FilterType = 'input' | 'single' | 'multiple'。详细类型定义"}]],{docTitle:"TableColumnController说明",tableRows:[{attribute:"buttonProps",description:"Object",type:"-",default:"自定义列配置按钮，包括 Button 组件的全部属性。比如：按钮颜色和文本。TS 类型: ButtonProps, Button API Documents。详细类型定义"},{attribute:"checkboxProps",description:"Object",type:"-",default:"透传复选框组件全部特性。TS 类型: CheckboxGroupProps, Checkbox API Documents。详细类型定义"},{attribute:"dialogProps",description:"Object",type:"-",default:"透传弹框组件全部特性，如：防止滚动穿透。TS 类型: DialogProps, Dialog API Documents。详细类型定义"},{attribute:"displayType",description:"String",type:"auto-width",default:"指列配置弹框中，各列的字段平铺方式：fixed-width 表示固定宽度，每行固定数量，横向和纵向均对齐，auto-width 表示宽度随列标题数量自由显示，横向铺满，纵向不要求对齐。可选项: fixed-width/auto-width"},{attribute:"fields",description:"Array",type:"-",default:"用于设置允许用户对哪些列进行显示或隐藏的控制，默认为全部字段。TS 类型: string[]"},{attribute:"hideTriggerButton",description:"Boolean",type:"false",default:"是否隐藏表格组件内置的“列配置”按钮"},{attribute:"placement",description:"String",type:"top-right",default:"列配置按钮基于表格的放置位置：左上角、右上角、左下角、右下角等。可选项: top-left/top-right/bottom-left/bottom-right"}]],{docTitle:"TableEditableCellConfig说明",tableRows:[{attribute:"abortEditOnEvent",description:"Array",type:"-",default:"除了点击非自身元素退出编辑态之外，还有哪些事件退出编辑态。示例: abortEditOnEvent: ['onChange']。TS 类型: string[]"},{attribute:"component",description:"-",type:"-",default:"组件定义，如：Input Select。对于完全自定义的组件（非组件库内的组件），组件需要支持 value 和 onChange；如果还需要支持校验规则，则组件还需实现 tips 和 status 两个 API，实现规则可参考 Input 组件。TS 类型: ComponentType。通用类型定义"},{attribute:"defaultEditable",description:"Boolean",type:"false",default:"单元格默认状态是否为编辑态"},{attribute:"on",description:"Function",type:"-",default:"透传给编辑组件的事件。TS 类型: (context: TableEditableCellPropsParams<T>) => { [eventName: string]: Function }"},{attribute:"onEdited",description:"Function",type:"-",default:"编辑完成后，退出编辑模式时触发。TS 类型: (context: PrimaryTableOnEditedContext<T>) => void type PrimaryTableOnEditedContext<T> = PrimaryTableCellParams<T> & { trigger: string; newRowData: T; }。详细类型定义"},{attribute:"props",description:"Object",type:"-",default:"透传给组件 edit.component 的属性。TS 类型: TableEditableCellProps<T> type TableEditableCellProps<T> = TablePlainObject | ((params: TableEditableCellPropsParams<T>) => TablePlainObject) interface TableEditableCellPropsParams<T> extends PrimaryTableCellParams<T> { editedRow: T } interface TablePlainObject { [key: string]: any }。通用类型定义。详细类型定义"},{attribute:"rules",description:"Array",type:"-",default:"校验规则。

```

TS 类型: TableEditableCellRules<T> type TableEditableCellRules<T> = FormRule[] | ((params: PrimaryTableCellParams<T>) => FormRule[]), Form API Documents. 详细类型定义",

{attribute: "showEditIcon", description: "Boolean", type: "true", default: "是否显示编辑图标"},

{attribute: "validateTrigger", description: "String", type: "'exit' | 'change'", default: "触发校验的时机, 有 2 种: 退出编辑时和数据变化时。TS 类型: 'exit' | 'change'"}], {docTitle: "TableTreeConfig说明", tableRows: [

{attribute: "checkStrictly", description: "Boolean", type: "true", default: "表示树形结构的行选中 (多选), 父子行选中是否独立"}, {attribute: "childrenKey", description: "String", type: "children", default: "树形结构子节点字段, 示例: childrenKey='list'。一般应用在数据 data 的子节点字段不是 children 的场景"},

{attribute: "defaultExpandAll", description: "Boolean", type: "false", default: "是否默认展开全部, 仅默认情况有效。如果希望自由控制树形结构的展开或收起, 可使用实例方法 expandAll 和 foldAll"},

{attribute: "expandTreeNodeOnClick", description: "Boolean", type: "false", default: "是否在点击行时展开树形结构节点"}, {attribute: "indent", description: "Number", type: "24", default: "树结点缩进距离, 单位: px"},

{attribute: "treeNodeColumnIndex", description: "Number", type: "0", default: "树结点在第几列渲染, 默认为第一列"}], {docTitle: "TScroll说明", tableRows: [

{attribute: "bufferSize", description: "Number", type: "20", default: "表示除可视区域外, 额外渲染的行数, 避免快速滚动过程中, 新出现的内容来不及渲染从而出现空白"},

{attribute: "isFixedRowHeight", description: "Boolean", type: "false", default: "表示每行内容是否同一个固定高度, 仅在 scroll.type 为 virtual 时有效, 该属性设置为 true 时, 可用于简化虚拟滚动内部计算逻辑, 提升性能, 此时则需要明确指定 scroll.rowHeight 属性的值"},

{attribute: "rowHeight", description: "Number", type: "-", default: "行高, 不会给<tr>元素添加样式高度, 仅作为滚动时的行高参考。一般情况不需要设置该属性。如果设置, 可尽量将该属性设置为每行平均高度, 从而使得滚动过程更加平滑"}, {attribute: "threshold", description: "Number", type: "100", default: "启动虚拟滚动的阈值。为保证组件收益最大化, 当数据量小于阈值 scroll.threshold 时, 无论虚拟滚动的配置是否存在, 组件内部都不会开启虚拟滚动"}, {attribute: "type", description: "String", type: "-", default: "必需。滚动加载类型, 有两种: 懒加载和虚拟滚动。值为 lazy, 表示滚动时会进行懒加载, 非可视区域内的内容将不会默认渲染, 直到该内容可见时, 才会进行渲染, 并且已渲染的内容滚动到不可见时, 不会被销毁; 值为virtual时, 表示会进行虚拟滚动, 无论滚动条滚动到哪个位置, 同一时刻, 仅渲染该可视区域内的内容, 当需要展示的数据量较大时, 建议开启该特性。可选项: lazy/virtual"}], {componentName: "tag", apiData: [{docTitle: "Tag Props说明", tableRows: [

{attribute: "closable", description: "Boolean", type: "false", default: "标签是否可关闭"},

{attribute: "content", description: "String / Slot / Function", type: "-", default: "组件子元素。TS 类型: string | TNode。通用类型定义"}, {attribute: "default", description: "String / Slot / Function", type: "-", default: "组件子元素, 同 content。TS 类型: string | TNode。通用类型定义"},

{attribute: "disabled", description: "Boolean", type: "false", default: "标签禁用态, 失效标签不能触发事件。默认风格 (theme=default) 才有禁用态"}, {attribute: "icon", description: "Slot / Function", type: "undefined", default: "标签中的图标, 可自定义图标呈现。TS 类型: TNode。通用类型定义"},

{attribute: "maxWidth", description: "String / Number", type: "-", default: "标签最大宽度, 宽度超出后会出现省略号。示例: '50px' / 80"}, {attribute: "shape", description: "String", type: "square", default: "标签类型, 有三种: 方形、圆角方形、标记型。可选项: square/round/mark"},

{attribute: "size", description: "String", type: "medium", default: "标签尺寸。可选项: small/medium/large。TS 类型: SizeEnum。通用类型定义"}, {attribute: "theme", description: "String", type: "default", default: "组件风格, 用于描述组件不同的应用场景。可选项: default/primary/warning/danger/success"},

{attribute: "variant", description: "String", type: "dark", default: "标签风格变体。可选项: dark/light/outline/light-outline"}, {attribute: "onClick", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent }) => void点击时触发"},

{attribute: "onClose", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent }) => void如果关闭按钮存在, 点击关闭按钮时触发"}], {docTitle: "Tag Events说明", tableRows: [

{attribute: "click", description: "(context: { e: MouseEvent })", type: "点击时触发"},

{attribute: "close", description: "(context: { e: MouseEvent })", type: "如果关闭按钮存在, 点击关闭按钮时触发"}], {docTitle: "CheckTag Props说明", tableRows: [

{attribute: "checked", description: "Boolean", type: "-", default: "标签选中的状态, 默认风格 (theme=default) 才有选中态。支持语法糖 v-model 或 v-model:checked"},

{attribute: "defaultChecked", description: "Boolean", type: "-", default: "标签选中的状态, 默认风格 (theme=default) 才有选中态。非受控属性"}, {attribute: "content", description: "String / Number / Array / Slot / Function", type: "-", default: "组件子元素; 传入数组时: [选中内容, 非选中内容]。TS 类型: string | number | string[] | TNode。通用类型定义"}, {attribute: "default", description: "String / Slot / Function", type: "-", default: "组件子元素, 默认插槽。TS 类型: string | TNode。通用类型定义"},

{attribute: "disabled", description: "Boolean", type: "false", default: "标签禁用态, 失效标签不能触发事件。默认风格 (theme=default) 才有禁用态"},

{attribute: "size", description: "String", type: "medium", default: "标签尺寸。可选项: small/medium/large。TS 类型: SizeEnum。通用类型定义"}, {attribute: "onChange", description: "Function", type: "", default: "TS 类型: (checked: boolean) => void状态切换时触发"},

{attribute: "onClick", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent }) => void点击标签时触发"}], {docTitle: "CheckTag Events说明", tableRows: [

{attribute: "change", description: "(checked: boolean)", type: "状态切换时触发"},

{attribute: "click", description: "(context: { e: MouseEvent })", type: "点击标签时触发"}], {componentName: "timeline", apiData: [{docTitle: "Timeline Props说明", tableRows: [

{attribute: "labelAlign", description: "String", type: "left", default: "标签信息放在时间轴的位置, mode='alternate' 时生效。纵向时间轴信息位置: 左侧、右侧或两侧, 默认信息在时间轴右侧。横向时间轴信息位置: 上方、下方、两侧。可选项: left/right/alternate/top/bottom"},

{attribute:"layout",description:"String",type:"vertical",default:"时间轴方向：水平方向、垂直方向。可选项：horizontal/vertical"},{attribute:"mode",description:"String",type:"alternate",default:"标签与内容文本的位置关系，alternate 为展示在轴两侧，same 为展示在同一侧。可选项：alternate/same"},{attribute:"reverse",description:"Boolean",type:"false",default:"时间轴是否表现为倒序"},{attribute:"theme",description:"String",type:"default",default:"步骤条风格。可选项：default/dot"}],{docTitle:"TimelineItem Props说明",tableRows:[{attribute:"content",description:"String / Slot / Function",type:"-",default:"描述内容。TS 类型：string | TNode。通用类型定义"},{attribute:"dot",description:"Slot / Function",type:"-",default:"用于自定义时间轴节点元素。TS 类型：TNode。通用类型定义"},{attribute:"dotColor",description:"String",type:"primary",default:"时间轴颜色，内置 primary/warning/error/default 四种色值，可传入 16 进制颜色码或 RGB 颜色值。TS 类型：string"},{attribute:"label",description:"String / Slot / Function",type:"-",default:"标签文本内容，可完全自定义。TS 类型：string | TNode。通用类型定义"},{attribute:"labelAlign",description:"String",type:"-",default:"标签信息相对于时间轴的位置，在 mode='alternate' 时生效，优先级高于 Timeline.labelAlign。可选项：left/right/top/bottom"},{attribute:"loading",description:"Boolean",type:"-",default:"是否处在加载状态"}]]}],{componentName:"tooltip",apiData:[{docTitle:"Tooltip Props说明",tableRows:[{attribute:"delay",description:"Number",type:"-",default:"【议案讨论中】延迟出现提示，用于异步加载提示信息需要延迟显示的业务场景下"},{attribute:"destroyOnClose",description:"Boolean",type:"true",default:"是否在关闭浮层时销毁浮层"},{attribute:"duration",description:"Number",type:"-",default:"用于设置提示默认显示多长时间之后消失，初始第一次有效，单位：毫秒"},{attribute:"placement",description:"String",type:"top",default:"浮层出现位置。TS 类型：'mouse' | PopupPlacement, Popup API Documents。详细类型定义"},{attribute:"showArrow",description:"Boolean",type:"true",default:"是否显示浮层箭头"},{attribute:"theme",description:"String",type:"default",default:"文字提示风格。可选项：default/primary/success/danger/warning/light"},{attribute:"Omit<PopupProps, 'placement'>",description:"-",type:"-",default:"继承 Omit<PopupProps, 'placement'> 中的全部 API"}]]}],{docTitle:"TooltipLite Props说明",tableRows:[{attribute:"content",description:"String / Slot / Function",type:"-",default:"文字提示内容。TS 类型：string | TNode。通用类型定义"},{attribute:"placement",description:"String",type:"top",default:"提示浮层出现的位置。可选项：top/bottom"},{attribute:"showArrow",description:"Boolean",type:"true",default:"是否显示箭头"},{attribute:"showShadow",description:"Boolean",type:"true",default:"文字提示浮层是否需要阴影"},{attribute:"theme",description:"String",type:"default",default:"组件风格，有亮色模式和暗色模式两种。可选项：light/default"},{attribute:"triggerElement",description:"String / Slot / Function",type:"-",default:"触发元素。TS 类型：string | TNode。通用类型定义"}]]}],{componentName:"tree",apiData:[{docTitle:"Tree Props说明",tableRows:[{attribute:"activable",description:"Boolean",type:"false",default:"节点是否可高亮"},{attribute:"activeMultiple",description:"Boolean",type:"false",default:"是否允许多个节点同时高亮"},{attribute:"activated",description:"Array",type:"-",default:"高亮的节点值。支持语法糖 v-model:activated。TS 类型：Array<TreeNodeValue>"},{attribute:"defaultActivated",description:"Array",type:"-",default:"高亮的节点值。非受控属性。TS 类型：Array<TreeNodeValue>"},{attribute:"allowFoldNodeOnFilter",description:"Boolean",type:"false",default:"是否允许在过滤时节点折叠节点"},{attribute:"checkProps",description:"Object",type:"-",default:"透传属性到 checkbox 组件。参考 checkbox 组件 API。TS 类型：CheckboxProps, Checkbox API Documents。详细类型定义"},{attribute:"checkStrictly",description:"Boolean",type:"false",default:"父子节点选中状态不再关联，可各自选中或取消"},{attribute:"checkable",description:"Boolean",type:"false",default:"隐藏节点复选框"},{attribute:"data",description:"Array",type:"[]",default:"树数据，泛型 T 表示树节点 TS 类型。TS 类型：Array<T>"},{attribute:"disableCheck",description:"Boolean / Function",type:"false",default:"禁用复选框，可支持禁用不同的行。TS 类型：boolean | ((node: TreeNodeModel<T>) => boolean)"},{attribute:"disabled",description:"Boolean",type:"-",default:"是否禁用树操作"},{attribute:"draggable",description:"Boolean",type:"-",default:"[开发中]节点是否可拖拽"},{attribute:"empty",description:"String / Slot / Function",type:"'",default:"数据为空时展示的文本。TS 类型：string | TNode。通用类型定义"},{attribute:"expandAll",description:"Boolean",type:"false",default:"是否展开全部节点"},{attribute:"expandLevel",description:"Number",type:"0",default:"默认展开的级别，第一层为 0"},{attribute:"expandMutex",description:"Boolean",type:"false",default:"同级别展开互斥，手风琴效果"},{attribute:"expandOnClickNode",description:"Boolean",type:"false",default:"是否支持点击节点也能展开收起"},{attribute:"expandParent",description:"Boolean",type:"false",default:"展开子节点时是否自动展开父节点"},{attribute:"expanded",description:"Array",type:"[]",default:"展开的节点值。支持语法糖 v-model:expanded。TS 类型：Array<TreeNodeValue>"},{attribute:"defaultExpanded",description:"Array",type:"[]",default:"展开的节点值。非受控属性。TS 类型：Array<TreeNodeValue>"},{attribute:"filter",description:"Function",type:"-",default:"节点过滤方法，只呈现返回值为 true 的节点，泛型 T 表示树节点 TS 类型。TS 类型：(node: TreeNodeModel<T>) => boolean"},{attribute:"hover",description:"Boolean",type:"-",default:"节点是否有悬浮状态"},{attribute:"icon",description:"Boolean / Slot / Function",type:"true",default:"节点图标，可自定义。TS 类型：boolean | TNode<TreeNodeModel<T>>。通用类型定义"},{attribute:"keys",description:"Object",type:"-",default:"用来定义 value / label / children 在 options 中对应的字段别名。TS 类型：TreeKeysType interface TreeKeysType { value?: string; label?: string; children?: string }。通用类型定义。详细类型定义"},{attribute:"label",description:"String /

Boolean / Slot / Function", type: "true", default: "自定义节点内容, 值为 false 不显示, 值为 true 显示默认 label, 值为字符串直接输出该字符串。泛型 T 表示树节点 TS 类型。如果期望只有点击复选框才选中, 而点击节点不选中, 可以使用 label 自定义节点, 然后加上点击事件 e.preventDefault(), 通过调整自定义节点的宽度和高度决定禁止点击选中的范围。TS 类型: string | boolean | TNode<TreeNodeModel<T>>。通用类型定义", {attribute: "lazy", description: "Boolean", type: "true", default: "延迟加载 children 为 true 的节点的子节点数据, 即使 expandAll 被设置为 true, 也同样延迟加载"}, {attribute: "line", description: "Boolean / Slot / Function", type: "false", default: "连接线。值为 false 不显示连接线; 值为 true 显示默认连接线; 值类型为 Function 表示自定义连接线。TS 类型: boolean | TNode。通用类型定义", {attribute: "load", description: "Function", type: "-", default: "加载子数据的方法, 在展开节点时调用 (仅当节点 children 为 true 时生效), 泛型 T 表示树节点 TS 类型。TS 类型: (node: TreeNodeModel<T>) => Promise<Array<T>>"}}, {attribute: "operations", description: "Slot / Function", type: "-", default: "自定义节点操作项, 泛型 T 表示树节点 TS 类型。TS 类型: TNode<TreeNodeModel<T>>。通用类型定义", {attribute: "transition", description: "Boolean", type: "true", default: "节点展开折叠时是否使用过渡动画"}, {attribute: "value", description: "Array", type: "[]", default: "选中值 (组件为可选状态时)。支持语法糖 v-model 或 v-model:value。TS 类型: Array<TreeNodeValue> type TreeNodeValue = string | number。详细类型定义", {attribute: "defaultValue", description: "Array", type: "[]", default: "选中值 (组件为可选状态时)。非受控属性。TS 类型: Array<TreeNodeValue> type TreeNodeValue = string | number。详细类型定义", {attribute: "valueMode", description: "String", type: "onlyLeaf", default: "选中值模式。all 表示父节点和子节点全部会出现在选中值里面; parentFirst 表示当子节点全部选中时, 仅父节点在选中值里面; onlyLeaf 表示无论什么情况, 选中值仅呈现叶子节点。可选项: onlyLeaf/parentFirst/all"}, {attribute: "onActive", description: "Function", type: "", default: "TS 类型: (value: Array<TreeNodeValue>, context: { node: TreeNodeModel<T> }) => void 节点激活时触发, 泛型 T 表示树节点 TS 类型"}, {attribute: "onChange", description: "Function", type: "", default: "TS 类型: (value: Array<TreeNodeValue>, context: { node: TreeNodeModel<T> }) => void 节点选中状态变化时触发, context.node 表示当前变化的选项, 泛型 T 表示树节点 TS 类型"}, {attribute: "onClick", description: "Function", type: "", default: "TS 类型: (context: { node: TreeNodeModel<T>; e: MouseEvent }) => void 节点点击时触发, 泛型 T 表示树节点 TS 类型"}, {attribute: "onExpand", description: "Function", type: "", default: "TS 类型: (value: Array<TreeNodeValue>, context: { node: TreeNodeModel<T>; e: MouseEvent }) => void 节点展开或收起时触发, 泛型 T 表示树节点 TS 类型"}, {attribute: "onLoad", description: "Function", type: "", default: "TS 类型: (context: { node: TreeNodeModel<T> }) => void 异步加载后触发, 泛型 T 表示树节点 TS 类型"}]}, {docTitle: "Tree Events说明", tableRows: [{attribute: "active", description: "(value: Array<TreeNodeValue>, context: { node: TreeNodeModel<T> })", type: "节点激活时触发, 泛型 T 表示树节点 TS 类型"}, {attribute: "change", description: "(value: Array<TreeNodeValue>, context: { node: TreeNodeModel<T> })", type: "节点选中状态变化时触发, context.node 表示当前变化的选项, 泛型 T 表示树节点 TS 类型"}, {attribute: "click", description: "(context: { node: TreeNodeModel<T>; e: MouseEvent })", type: "节点点击时触发, 泛型 T 表示树节点 TS 类型"}, {attribute: "expand", description: "(value: Array<TreeNodeValue>, context: { node: TreeNodeModel<T>; e: MouseEvent })", type: "节点展开或收起时触发, 泛型 T 表示树节点 TS 类型"}, {attribute: "load", description: "(context: { node: TreeNodeModel<T> })", type: "异步加载后触发, 泛型 T 表示树节点 TS 类型"}]}, {docTitle: "TreeInstanceFunctions 组件实例方法说明", tableRows: [{attribute: "appendTo", description: "(value: TreeNodeValue, newData: T | Array<T>)", type: "-", default: "为指定节点添加子节点, 默认添加到根节点, 泛型 T 表示树节点 TS 类型"}, {attribute: "getIndex", description: "(value: TreeNodeValue)", type: "number", default: "获取指定节点下标"}, {attribute: "getItem", description: "(value: TreeNodeValue)", type: "TreeNodeModel<T>", default: "获取指定节点所有信息, 泛型 T 表示树节点 TS 类型"}, {attribute: "getItems", description: "(value?: TreeNodeValue)", type: "Array<TreeNodeModel<T>>", default: "获取某节点的全部子孙节点; 参数为空, 则表示获取整棵树的全部节点, 泛型 T 表示树节点 TS 类型"}, {attribute: "getParent", description: "(value: TreeNodeValue)", type: "TreeNodeModel<T>", default: "获取指定节点的直属父节点, 泛型 T 表示树节点 TS 类型"}, {attribute: "getParents", description: "(value: TreeNodeValue)", type: "TreeNodeModel<T>[]", default: "获取指定节点的全部父节点, 泛型 T 表示树节点 TS 类型"}, {attribute: "getPath", description: "(value: TreeNodeValue)", type: "TreeNodeModel<T>[]", default: "自下而上获取全路径数据, 泛型 T 表示树节点 TS 类型"}, {attribute: "insertAfter", description: "(value: TreeNodeValue, newData: T)", type: "-", default: "插入新节点到指定节点后面, 泛型 T 表示树节点 TS 类型"}, {attribute: "insertBefore", description: "(value: TreeNodeValue, newData: T)", type: "-", default: "插入新节点到指定节点前面, 泛型 T 表示树节点 TS 类型"}, {attribute: "remove", description: "(value: TreeNodeValue)", type: "-", default: "移除指定节点"}, {attribute: "setItem", description: "(value: TreeNodeValue, options: TreeNodeState)", type: "-", default: "设置节点状态"}]}, {docTitle: "TreeNodeState 说明", tableRows: [{attribute: "activable", description: "Boolean", type: "false", default: "节点是否允许被激活"}, {attribute: "actived", description: "Boolean", type: "false", default: "节点是否被激活"}, {attribute: "checkable", description: "Boolean", type: "false", default: "节点是否允许被选中"}, {attribute: "checked", description: "Boolean", type: "false", default: "节点是否被选中"}, {attribute: "disabled", description: "Boolean", type: "false", default: "节点是否被禁用"}, {attribute: "expandMutex", description: "Boolean", type: "false", default: "子节点是否互斥展开"}, {attribute: "expanded", description: "Boolean", type: "false", default: "节点是否已展开"}, {attribute: "indeterminate", description: "Boolean", type: "false", default: "节点是否为半选中状态"}, {attribute: "label", description: "String", type: "-", default: "节点标签文案"}, {attribute: "loading", description: "Boolean", type: "false", default: "子节点数据是否在加载中"}, {attribute: "value", description: "String / Number", type: "-", default: "节点值"}, {attribute: "visible", description: "Boolean", type: "false", default: "节点是否可视"}]}, {docTitle: "TreeNodeModel 说明", tableRows: [{attribute: "actived", description: "Boolean", type: "-"}]

```

", default: "必需。当前节点是否处于高亮激活态"}, {attribute: "checked", description: "Boolean", type: "-",
", default: "必需。当前节点是否被选中"}, {attribute: "data", description: "Object", type: "-", default: "必需。节点数据, 泛型 T 表示树节点 TS 类型。TS 类型: T"},
{attribute: "expanded", description: "Boolean", type: "-", default: "必需。当前节点是否展开"},
{attribute: "indeterminate", description: "Boolean", type: "-", default: "必需。当前节点是否处于半选状态"},
{attribute: "loading", description: "Boolean", type: "-", default: "必需。当前节点是否处于加载中状态"},
{attribute: "TreeNodeState", description: "-", type: "-", default: "继承 TreeNodeState 中的全部 API"}],
{docTitle: "TreeNodeModel说明", tableRows: [{attribute: "appendData", description: "(data: T | Array<T>)", type: "-", default: "必需。追加子节点数据, 泛型 T 表示树节点 TS 类型"},
{attribute: "getChildren", description: "(deep: boolean)", type: "Array<TreeNodeModel<T>> | boolean", default: "必需。默认获取当前节点的全部子节点, deep 值为 true 则表示获取全部子孙节点"},
{attribute: "getIndex", description: "-", type: "number", default: "必需。获取节点在父节点的子节点列表中的位置, 如果没有父节点, 则获取节点在根节点列表的位置"}, {attribute: "getLevel", description: "-", type: "number", default: "必需。获取节点所在的层级"}, {attribute: "getParent", description: "-", type: "TreeNodeModel<T>", default: "必需。获取单个父节点"}, {attribute: "getParents", description: "-", type: "Array<TreeNodeModel<T>>", default: "必需。获取所有父节点"}, {attribute: "getPath", description: "-", type: "Array<TreeNodeModel<T>>", default: "必需。获取节点全路径"}, {attribute: "getRoot", description: "-", type: "TreeNodeModel<T>", default: "必需。获取根节点"}, {attribute: "getSiblings", description: "-", type: "Array<TreeNodeModel<T>>", default: "必需。获取兄弟节点, 包含自己在内"},
{attribute: "insertAfter", description: "(newData: T)", type: "-", default: "必需。在当前节点前插入新节点, 泛型 T 表示树节点 TS 类型"}, {attribute: "insertBefore", description: "(newData: T)", type: "-", default: "必需。在当前节点前插入新节点, 泛型 T 表示树节点 TS 类型"},
{attribute: "isFirst", description: "-", type: "boolean", default: "必需。是否为兄弟节点中的第一个节点"},
{attribute: "isLast", description: "-", type: "boolean", default: "必需。是否为兄弟节点中的最后一个节点"},
{attribute: "isLeaf", description: "-", type: "boolean", default: "必需。是否为叶子节点"},
{attribute: "remove", description: "(value?: TreeNodeValue)", type: "-", default: "必需。移除当前节点或当前节点的子节点, 值为空则移除当前节点, 值存在则移除当前节点的子节点"},
{attribute: "setData", description: "(data: T)", type: "-", default: "必需。设置当前节点数据, 数据变化可自动刷新页面, 泛型 T 表示树节点 TS 类型"}]]], {componentName: "watermark", apiData: [{docTitle: "Watermark Props说明", tableRows: [{attribute: "alpha", description: "Number", type: "1", default: "水印整体透明度, 取值范围 [0-1]"}, {attribute: "content", description: "String / Slot / Function", type: "-", default: "水印所覆盖的内容节点。TS 类型: string | TNode。通用类型定义"}, {attribute: "default", description: "String / Slot / Function", type: "-", default: "水印所覆盖的内容节点, 同 content。TS 类型: string | TNode。通用类型定义"}, {attribute: "height", description: "Number", type: "-", default: "水印高度"},
{attribute: "isRepeat", description: "Boolean", type: "true", default: "水印是否重复出现"},
{attribute: "lineSpace", description: "Number", type: "16", default: "行间距, 只作用在多行 (content 配置为数组) 情况下"}, {attribute: "movable", description: "Boolean", type: "false", default: "水印是否可移动"},
{attribute: "moveInterval", description: "Number", type: "3000", default: "水印发生运动位移的间隙, 单位: 毫秒"}, {attribute: "offset", description: "Array", type: "-", default: "水印在画布上绘制的水平和垂直偏移量, 正常情况下水印绘制在中间位置, 即 offset = [gapX / 2, gapY / 2]。TS 类型: Array<number>"},
{attribute: "removable", description: "Boolean", type: "true", default: "水印是否可被删除, 默认会开启水印节点防删"}, {attribute: "rotate", description: "Number", type: "-22", default: "水印旋转的角度, 单位: °"},
{attribute: "watermarkContent", description: "Object / Array", type: "-", default: "水印内容, 需要显示多行情况下可配置为数组。TS 类型: WatermarkText|WatermarkImage|Array<WatermarkText|WatermarkImage>"},
{attribute: "width", description: "Number", type: "-", default: "水印宽度"},
{attribute: "x", description: "Number", type: "-", default: "水印之间的水平间距"},
{attribute: "y", description: "Number", type: "-", default: "水印之间的垂直间距"},
{attribute: "zIndex", description: "Number", type: "-", default: "水印元素的 z-index, 默认值写在 CSS 中"}]], {docTitle: "WatermarkText说明", tableRows: [{attribute: "fontColor", description: "String", type: "rgba(0,0,0,0.1)", default: "水印文本文字颜色"},
{attribute: "fontSize", description: "Number", type: "16", default: "水印文本文字大小"},
{attribute: "fontWeight", description: "String", type: "normal", default: "水印文本文字粗细。可选项: normal/lighter/bold/bolder"}, {attribute: "text", description: "String", type: "-", default: "水印文本内容"}]], {docTitle: "WatermarkImage说明", tableRows: [{attribute: "isGrayscale", description: "Boolean", type: "false", default: "水印图片是否需要灰阶显示"},
{attribute: "url", description: "String", type: "-", default: "水印图片源地址, 为了显示清楚, 建议导出 2 倍或 3 倍图"}]]], {componentName: "rate", apiData: [{docTitle: "Rate Props说明", tableRows: [{attribute: "allowHalf", description: "Boolean", type: "false", default: "是否允许半选"},
{attribute: "color", description: "String / Array", type: "'#ED7B2F'", default: "评分图标的颜色, 样式中默认为 #ED7B2F。一个值表示设置选中高亮的五角星颜色, 示例: [选中颜色]。数组则表示分别设置 选中高亮的五角星颜色 和 未选中暗灰的五角星颜色, [选中颜色, 未选中颜色]。示例: [' #ED7B2F', ' #E3E6EB']。TS 类型: string | Array<string>"}, {attribute: "count", description: "Number", type: "5", default: "评分的数量"},
{attribute: "disabled", description: "Boolean", type: "-", default: "是否禁用评分"},
{attribute: "gap", description: "Number", type: "4", default: "评分图标的间距"},
{attribute: "icon", description: "Slot / Function", type: "-", default: "自定义评分图标。TS 类型: TNode。通用类型定义"}, {attribute: "showText", description: "Boolean", type: "false", default: "是否显示对应的辅助文字"},
{attribute: "size", description: "String", type: "24px", default: "评分图标的大小, 示例: 20px"},
{attribute: "texts", description: "Array", type: "[]", default: "评分等级对应的辅助文字。组件内置默认值为: [' 极差', ' 失望', ' 一般', ' 满意', ' 惊喜']。自定义值示例: [' 1分', ' 2分', ' 3分', ' 4分', ' 5分']。TS 类

```


型: Array<string>"}], {attribute: "value", description: "Number", type: "0", default: "选择评分的值。支持语法糖 v-model 或 v-model:value"}, {attribute: "defaultValue", description: "Number", type: "0", default: "选择评分的值。非受控属性"}, {attribute: "onChange", description: "Function", type: "", default: "TS 类型: (value: number) => void 评分数改变时触发"}]], {docTitle: "Rate Events说明", tableRows: [{attribute: "change", description: "(value: number)", type: "评分数改变时触发"}]]}, {componentName: "alert", apiData: [{docTitle: "Alert Props说明", tableRows: [{attribute: "close", description: "String / Boolean / Slot / Function", type: "false", default: "关闭按钮。值为 true 则显示默认关闭按钮; 值为 false 则不显示按钮; 值类型为 string 则直接显示; 值类型为 Function 则可以自定关闭按钮。TS 类型: string | boolean | TNode。通用类型定义"}, {attribute: "default", description: "String / Slot / Function", type: "-", default: "内容, 同 message。TS 类型: string | TNode。通用类型定义"}, {attribute: "icon", description: "Slot / Function", type: "-", default: "图标。TS 类型: TNode。通用类型定义"}, {attribute: "maxLine", description: "Number", type: "0", default: "内容显示最大行数, 超出的内容会折叠收起, 用户点击后再展开。值为 0 表示不折叠"}, {attribute: "message", description: "String / Slot / Function", type: "-", default: "内容 (子元素)。TS 类型: string | TNode。通用类型定义"}, {attribute: "operation", description: "Slot / Function", type: "-", default: "跟在告警内容后面的操作区。TS 类型: TNode。通用类型定义"}, {attribute: "theme", description: "String", type: "info", default: "组件风格。可选项: success/info/warning/error"}, {attribute: "title", description: "String / Slot / Function", type: "-", default: "标题。TS 类型: string | TNode。通用类型定义"}, {attribute: "onClose", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent }) => void 关闭按钮点击时触发"}, {attribute: "onClosed", description: "Function", type: "", default: "TS 类型: (context: { e: TransitionEvent }) => void 告警提示框关闭动画结束后触发"}]], {docTitle: "Alert Events说明", tableRows: [{attribute: "close", description: "(context: { e: MouseEvent })", type: "关闭按钮点击时触发"}, {attribute: "closed", description: "(context: { e: TransitionEvent })", type: "告警提示框关闭动画结束后触发"}]]}, {componentName: "dialog", apiData: [{docTitle: "Dialog Props说明", tableRows: [{attribute: "attach", description: "String / Function", type: "", default: "对话框挂载的节点, 默认挂在组件本身的位置。数据类型为 String 时, 会被当作选择器处理, 进行节点查询。示例: 'body' 或 () => document.body。TS 类型: AttachNode。通用类型定义"}, {attribute: "body", description: "String / Slot / Function", type: "", default: "对话框内容。TS 类型: string | TNode。通用类型定义"}, {attribute: "cancelBtn", description: "String / Object / Slot / Function", type: "", default: "取消按钮, 可自定义。值为 null 则不显示取消按钮。值类型为字符串, 则表示自定义按钮文本, 值类型为 Object 则表示透传 Button 组件属性。使用 TNode 自定义按钮时, 需自行控制取消事件。TS 类型: string | ButtonProps | TNode | null, Button API Documents。通用类型定义。详细类型定义"}, {attribute: "closeBtn", description: "String / Boolean / Slot / Function", type: "true", default: "关闭按钮, 可以自定义。值为 true 显示默认关闭按钮, 值为 false 不显示关闭按钮。值类型为 string 则直接显示值, 如: '关闭'。值类型为 TNode, 则表示呈现自定义按钮示例。TS 类型: string | boolean | TNode。通用类型定义"}, {attribute: "closeOnEscKeydown", description: "Boolean", type: "true", default: "按下 ESC 时是否触发对话框关闭事件"}, {attribute: "closeOnOverlayClick", description: "Boolean", type: "true", default: "点击蒙层时是否触发关闭事件"}, {attribute: "confirmBtn", description: "String / Object / Slot / Function", type: "", default: "确认按钮。值为 null 则不显示确认按钮。值类型为字符串, 则表示自定义按钮文本, 值类型为 Object 则表示透传 Button 组件属性。使用 TNode 自定义按钮时, 需自行控制确认事件。TS 类型: string | ButtonProps | TNode | null。通用类型定义"}, {attribute: "confirmOnEnter", description: "Boolean", type: "-", default: "是否在按下回车键时, 触发确认事件"}, {attribute: "default", description: "String / Slot / Function", type: "-", default: "对话框内容, 同 body。TS 类型: string | TNode。通用类型定义"}, {attribute: "destroyOnClose", description: "Boolean", type: "false", default: "是否在关闭弹框的时候销毁子元素"}, {attribute: "draggable", description: "Boolean", type: "false", default: "对话框是否可以拖拽 (仅在非模态对话框时有效)"}, {attribute: "footer", description: "Boolean / Slot / Function", type: "true", default: "底部操作栏, 默认会有 '确认' 和 '取消' 两个按钮。值为 true 显示默认操作按钮, 值为 false 不显示任何内容, 值类型为 Function 表示自定义底部内容。TS 类型: boolean | TNode。通用类型定义"}, {attribute: "header", description: "String / Boolean / Slot / Function", type: "true", default: "头部内容。值为 true 显示空白头部, 值为 false 不显示任何内容, 值类型为 string 则直接显示值, 值类型为 Function 表示自定义头部内容。TS 类型: string | boolean | TNode。通用类型定义"}, {attribute: "mode", description: "String", type: "modal", default: "对话框类型, 有 4 种: 模态对话框、非模态对话框、普通对话框、全屏对话框。弹出「模态对话框」时, 只能操作对话框里面的内容, 不能操作其他内容。弹出「非模态对话框」时, 则可以操作页面内所有内容。「普通对话框」是指没有脱离文档流的对话框, 可以在这个基础上开发更多的插件。可选项: modal/modeless/normal/full-screen"}, {attribute: "placement", description: "String", type: "top", default: "对话框位置, 内置两种: 垂直水平居中显示 和 靠近顶部 (top:20%) 显示。默认情况, 为避免贴顶或贴底, 顶部和底部距离最小为 48px, 可通过调整 top 覆盖默认大小。可选项: top/center"}, {attribute: "preventScrollThrough", description: "Boolean", type: "true", default: "防止滚动穿透"}, {attribute: "showInAttachedElement", description: "Boolean", type: "false", default: "仅在挂载元素中显示抽屉, 默认在浏览器可视区域显示。父元素需要有定位属性, 如: position: relative"}, {attribute: "showOverlay", description: "Boolean", type: "true", default: "是否显示遮罩层"}, {attribute: "theme", description: "String", type: "default", default: "对话框风格。可选项: default/info/warning/danger/success"}, {attribute: "top", description: "String / Number", type: "-", default: "用于弹框具体窗口顶部的距离, 优先级大于 placement"}, {attribute: "visible", description: "Boolean", type: "false", default: "控制对话框是否显示"}, {attribute: "width", description: "String / Number", type: "-", default: "对话框宽度, 示例: 320, '500px'"},

```

'80%'`, {attribute: "zIndex", description: "Number", type: "-", default: "对话框层级, Web 侧样式默认为 2500, 移动端和小程序样式默认为 1500"},
{attribute: "onCancel", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent }) => void 如果“取消”按钮存在, 则点击“取消”按钮时触发, 同时触发关闭事件"},
{attribute: "onClose", description: "Function", type: "", default: "TS 类型: (context: DialogCloseContext) => void 关闭事件, 点击取消按钮、点击关闭按钮、点击蒙层、按下 ESC 等场景下触发。详细类型定义。type DialogEventSource = 'esc' | 'close-btn' | 'cancel' | 'overlay' interface DialogCloseContext { trigger: DialogEventSource; e: MouseEvent | KeyboardEvent }"},
{attribute: "onCloseBtnClick", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent }) => void 点击右上角关闭按钮时触发"},
{attribute: "onClosed", description: "Function", type: "", default: "TS 类型: () => void 对话框消失动画效果结束后触发"}, {attribute: "onConfirm", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent | KeyboardEvent }) => void 如果“确认”按钮存在, 则点击“确认”按钮时触发, 或者键盘按下回车键时触发"}, {attribute: "onEscKeydown", description: "Function", type: "", default: "TS 类型: (context: { e: KeyboardEvent }) => void 按下 ESC 时触发事件"},
{attribute: "onOpened", description: "Function", type: "", default: "TS 类型: () => void 对话框弹出动画效果结束后触发"}, {attribute: "onOverlayClick", description: "Function", type: "", default: "TS 类型: (context: { e: MouseEvent }) => void 如果蒙层存在, 点击蒙层时触发"}], {docTitle: "Dialog Events说明", tableRows: [[{attribute: "cancel", description: "(context: { e: MouseEvent })", type: "如果“取消”按钮存在, 则点击“取消”按钮时触发, 同时触发关闭事件"}, {attribute: "close", description: "(context: DialogCloseContext)", type: "关闭事件, 点击取消按钮、点击关闭按钮、点击蒙层、按下 ESC 等场景下触发。详细类型定义。type DialogEventSource = 'esc' | 'close-btn' | 'cancel' | 'overlay' interface DialogCloseContext { trigger: DialogEventSource; e: MouseEvent | KeyboardEvent }"}, {attribute: "close-btn-click", description: "(context: { e: MouseEvent })", type: "点击右上角关闭按钮时触发"}, {attribute: "closed", description: "-", type: "对话框消失动画效果结束后触发"}, {attribute: "confirm", description: "(context: { e: MouseEvent | KeyboardEvent })", type: "如果“确认”按钮存在, 则点击“确认”按钮时触发, 或者键盘按下回车键时触发"}, {attribute: "esc-keydown", description: "(context: { e: KeyboardEvent })", type: "按下 ESC 时触发事件"}, {attribute: "opened", description: "-", type: "对话框弹出动画效果结束后触发"}, {attribute: "overlay-click", description: "(context: { e: MouseEvent })", type: "如果蒙层存在, 点击蒙层时触发"}]], {docTitle: "Dialog Options说明", tableRows: [[{attribute: "attach", description: "String / Function", type: "'body'", default: "对话框挂载的节点。数据类型为 String 时, 会被当作选择器处理, 进行节点查询。示例: 'body' 或 () => document.body。TS 类型: AttachNode。通用类型定义"}, {attribute: "className", description: "String", type: "-", default: "弹框类名, 示例: 't-class-dialog-first t-class-dialog-second'"}, {attribute: "style", description: "String / Object", type: "-", default: "弹框 style 属性, 输入 CSSStyleDeclaration.cssText。TS 类型: string | Styles。通用类型定义"}, {attribute: "Omit<DialogProps, 'attach'>", description: "-", type: "-", default: "继承 Omit<DialogProps, 'attach'> 中的全部 API"}]], {docTitle: "Dialog Instance说明", tableRows: [[{attribute: "destroy", description: "-", type: "-", default: "销毁弹框"}, {attribute: "hide", description: "-", type: "-", default: "隐藏弹框"}, {attribute: "show", description: "-", type: "-", default: "显示弹框"}, {attribute: "update", description: "(props: DialogOptions)", type: "-", default: "更新弹框内容"}]], {docTitle: "Dialog Plugin说明", tableRows: []}, {docTitle: "Dialog Plugin.confirm说明", tableRows: []}, {docTitle: "Dialog Plugin.alert说明", tableRows: []}], {componentName: "drawer", apiData: [[{docTitle: "Drawer Props说明", tableRows: [{attribute: "attach", description: "String / Function", type: ""', default: "抽屉挂载的节点, 默认挂在组件本身的位置。数据类型为 String 时, 会被当作选择器处理, 进行节点查询。示例: 'body' 或 () => document.body。TS 类型: AttachNode。通用类型定义"}, {attribute: "body", description: "String / Slot / Function", type: "-", default: "抽屉内容。TS 类型: string | TNode。通用类型定义"}, {attribute: "cancelBtn", description: "String / Object / Slot / Function", type: ""', default: "取消按钮, 可自定义。值为 null 则不显示取消按钮。值类型为字符串, 则表示自定义按钮文本, 值类型为 Object 则表示透传 Button 组件属性。使用 TNode 自定义按钮时, 需自行控制取消事件。TS 类型: FooterButton"}, {attribute: "closeBtn", description: "String / Boolean / Slot / Function", type: "-", default: "关闭按钮, 可以自定义。值为 true 显示默认关闭按钮, 值为 false 不显示关闭按钮。值类型为 string 则直接显示值, 如: “关闭”。值类型为 TNode, 则表示呈现自定义按钮示例。TS 类型: string | boolean | TNode。通用类型定义"}, {attribute: "closeOnEscKeydown", description: "Boolean", type: "true", default: "按下 ESC 时是否触发抽屉关闭事件"}, {attribute: "closeOnOverlayClick", description: "Boolean", type: "true", default: "点击蒙层时是否触发抽屉关闭事件"}, {attribute: "confirmBtn", description: "String / Object / Slot / Function", type: ""', default: "确认按钮。值类型为字符串, 则表示自定义按钮文本, 值类型为 Object 则表示透传 Button 组件属性。使用 TNode 自定义按钮时, 需自行控制确认事件。TS 类型: FooterButton type FooterButton = string | ButtonProps | TNode, Button API Documents。通用类型定义。详细类型定义"}, {attribute: "default", description: "String / Slot / Function", type: "-", default: "抽屉内容, 同 body。TS 类型: string | TNode。通用类型定义"}, {attribute: "destroyOnClose", description: "Boolean", type: "false", default: "抽屉关闭时是否销毁节点"}, {attribute: "footer", description: "Boolean / Slot / Function", type: "true", default: "底部操作栏, 默认会有“确认”和“取消”两个按钮。值为 true 显示默认操作按钮, 值为 false 或 null 不显示任何内容, 值类型为 TNode 表示自定义底部内容。TS 类型: boolean | TNode。通用类型定义"}, {attribute: "header", description: "String / Boolean / Slot / Function", type: "true", default: "头部内容。值为 true 显示空白头部, 值为 false 不显示头部, 值类型为 string 则直接显示值, 值类型为 TNode 表示自定义头部内容。TS 类型: string | boolean | TNode。通用类型定义"}, {attribute: "mode", description: "String", type: "overlay", default: "展开方式, 有两种: 直接展示在内容上方

```

和 推开内容区域。可选项: overlay/push",
 {attribute:"placement",description:"String",type:"right",default:"抽屉方向。可选项: left/right/top/bottom"},
 {attribute:"preventScrollThrough",description:"Boolean",type:"true",default:"防止滚动穿透"},
 {attribute:"showInAttachedElement",description:"Boolean",type:"false",default:"仅在挂载元素中显示抽屉, 默认在浏览器可视区域显示。父元素需要有定位属性, 如: position: relative"},
 {attribute:"showOverlay",description:"Boolean",type:"true",default:"是否显示遮罩层"},
 {attribute:"size",description:"String",type:"small",default:"尺寸, 支持 'small', 'medium', 'large', '35px', '30%', '3em' 等。纵向抽屉调整的是抽屉宽度, 横向抽屉调整的是抽屉高度"},
 {attribute:"sizeDraggable",description:"Boolean",type:"false",default:"抽屉大小可拖拽调整, 横向抽屉调整宽度, 纵向抽屉调整高度"}, {attribute:"visible",description:"Boolean",type:"false",default:"组件是否可见"}, {attribute:"zIndex",description:"Number",type:"-",default:"抽屉层级, 样式默认为 1500"},
 {attribute:"onCancel",description:"Function",type:"",default:"TS 类型: (context: { e: MouseEvent }) => void如果“取消”按钮存在, 点击“取消”按钮时触发, 同时触发关闭事件"},
 {attribute:"onClose",description:"Function",type:"",default:"TS 类型: (context: DrawerCloseContext) => void关闭事件, 取消按钮点击时、关闭按钮点击时、ESC 按下时、点击蒙层时均会触发。详细类型定义。type DrawerEventSource = 'esc' | 'close-btn' | 'cancel' | 'overlay' interface DrawerCloseContext { trigger: DrawerEventSource; e: MouseEvent | KeyboardEvent }"},
 {attribute:"onCloseBtnClick",description:"Function",type:"",default:"TS 类型: (context: { e: MouseEvent }) => void如果关闭按钮存在, 点击关闭按钮时触发该事件, 同时触发关闭事件"},
 {attribute:"onConfirm",description:"Function",type:"",default:"TS 类型: (context: { e: MouseEvent }) => void如果“确认”按钮存在, 则点击“确认”按钮时触发"},
 {attribute:"onEscKeydown",description:"Function",type:"",default:"TS 类型: (context: { e: KeyboardEvent }) => void按下 ESC 键时触发"},
 {attribute:"onOverlayClick",description:"Function",type:"",default:"TS 类型: (context: { e: MouseEvent }) => void如果蒙层存在, 点击蒙层时触发"}], {docTitle:"Drawer Events说明",tableRows:
 [{attribute:"cancel",description:"(context: { e: MouseEvent })",type:"如果“取消”按钮存在, 点击“取消”按钮时触发, 同时触发关闭事件"}, {attribute:"close",description:"(context: DrawerCloseContext)",type:"关闭事件, 取消按钮点击时、关闭按钮点击时、ESC 按下时、点击蒙层时均会触发。详细类型定义。type DrawerEventSource = 'esc' | 'close-btn' | 'cancel' | 'overlay' interface DrawerCloseContext { trigger: DrawerEventSource; e: MouseEvent | KeyboardEvent }"},
 {attribute:"close-btn-click",description:"(context: { e: MouseEvent })",type:"如果关闭按钮存在, 点击关闭按钮时触发该事件, 同时触发关闭事件"}, {attribute:"confirm",description:"(context: { e: MouseEvent })",type:"如果“确认”按钮存在, 则点击“确认”按钮时触发"}, {attribute:"esc-keydown",description:"(context: { e: KeyboardEvent })",type:"按下 ESC 键时触发"}, {attribute:"overlay-click",description:"(context: { e: MouseEvent })",type:"如果蒙层存在, 点击蒙层时触发"}], {docTitle:"DrawerOptions说明",tableRows:[{attribute:"attach",description:"String / Function",type:"body",default:"抽屉挂载的节点。数据类型为 String 时, 会被当作选择器处理, 进行节点查询。示例: 'body' 或 () => document.body。TS 类型: AttachNode。通用类型定义"}, {attribute:"className",description:"String",type:"-",default:"抽屉类名, 示例: 't-class-drawer-first t-class-drawer-second'"}, {attribute:"style",description:"String / Object",type:"-",default:"弹框 style 属性, 输入 CSSStyleDeclaration.cssText。TS 类型: string | Styles。通用类型定义"}, {attribute:"Omit<DrawerProps, 'attach'>",description:"-",type:"-",default:"继承 Omit<DrawerProps, 'attach'> 中的全部 API"}], {docTitle:"DrawerInstance说明",tableRows:
 [{attribute:"destroy",description:"-",type:"-",default:"销毁抽屉"}, {attribute:"hide",description:"-",type:"-",default:"隐藏抽屉"}, {attribute:"show",description:"-",type:"-",default:"显示抽屉"},
 {attribute:"update",description:"(props: DrawerOptions)",type:"-",default:"更新抽屉内容"}], {docTitle:"DrawerPlugin说明",tableRows:[[]]}, {componentName:"guide",apiData:[{docTitle:"Guide Props说明",tableRows:[{attribute:"counter",description:"Slot / Function",type:"-",default:"用于自定义渲染计数部分。TS 类型: TNode<{ current: number; total: number }>。通用类型定义"},
 {attribute:"current",description:"Number",type:"-",default:"当前步骤, 即整个引导的进度。-1 则不展示, 用于需要中断展示的场景。支持语法糖 v-model 或 v-model:current"},
 {attribute:"defaultCurrent",description:"Number",type:"-",default:"当前步骤, 即整个引导的进度。-1 则不展示, 用于需要中断展示的场景。非受控属性"},
 {attribute:"finishButtonProps",description:"Object",type:"-",default:"透传 完成 的全部属性, 示例: { content: '完成', theme: 'primary' }。TS 类型: ButtonProps"},
 {attribute:"hideCounter",description:"Boolean",type:"false",default:"是否隐藏计数"},
 {attribute:"hidePrev",description:"Boolean",type:"false",default:"是否隐藏上一步按钮"},
 {attribute:"hideSkip",description:"Boolean",type:"false",default:"是否隐藏跳过按钮"},
 {attribute:"highlightPadding",description:"Number",type:"8",default:"高亮框的内边距"},
 {attribute:"mode",description:"String",type:"popup",default:"引导框的类型。可选项: popup/dialog"},
 {attribute:"nextButtonProps",description:"Object",type:"-",default:"透传 下一步按钮 的全部属性, 示例: { content: '下一步', theme: 'primary' }。TS 类型: ButtonProps, Button API Documents。详细类型定义"}, {attribute:"prevButtonProps",description:"Object",type:"-",default:"透传 上一步按钮 的全部属性, 示例: { content: '上一步', theme: 'default' }。TS 类型: ButtonProps"},
 {attribute:"showOverlay",description:"Boolean",type:"true",default:"是否出现遮罩层"},
 {attribute:"skipButtonProps",description:"Object",type:"-",default:"透传 跳过按钮 的全部属性, { content: '跳过', theme: 'default' }。TS 类型: ButtonProps"},
 {attribute:"steps",description:"Array",type:"-",default:"用于定义每个步骤的内容, 包括高亮的节点、相对位置和具体的文案内容等。。TS 类型: Array<GuideStep>"},

```

{attribute:" zIndex",description:"Number",type:"999999",default:"提示框的层级"},
{attribute:"onChange",description:"Function",type:"",default:"TS 类型: (current: number, context?: {
e: MouseEvent, total: number }) => void当前步骤发生变化时触发"},
{attribute:"onFinish",description:"Function",type:"",default:"TS 类型: (context: { e: MouseEvent,
current: number, total: number }) => void点击完成按钮时触发"},
{attribute:"onNextStepClick",description:"Function",type:"",default:"TS 类型: (context: { e:
MouseEvent, next: number, current: number, total: number }) => void点击下一步时触发"},
{attribute:"onPrevStepClick",description:"Function",type:"",default:"TS 类型: (context: { e:
MouseEvent, prev: number, current: number, total: number }) => void点击上一步时触发"},
{attribute:"onSkip",description:"Function",type:"",default:"TS 类型: (context: { e: MouseEvent,
current: number, total: number }) => void点击跳过按钮时触发"}]],{docTitle:"Guide Events说明",
tableRows:[{attribute:"change",description:"(current: number, context?: { e: MouseEvent, total:
number })",type:"当前步骤发生变化时触发"},{attribute:"finish",description:"(context: { e:
MouseEvent, current: number, total: number })",type:"点击完成按钮时触发"},{attribute:"next-step-
click",description:"(context: { e: MouseEvent, next: number, current: number, total: number
})",type:"点击下一步时触发"},{attribute:"prev-step-click",description:"(context: { e: MouseEvent,
prev: number, current: number, total: number })",type:"点击上一步时触发"},
{attribute:"skip",description:"(context: { e: MouseEvent, current: number, total: number
})",type:"点击跳过按钮时触发"}]],{docTitle:"GuideStep说明",tableRows:
[{{attribute:"body",description:"String / Slot / Function",type:"-","default:"当前步骤提示框的内容。TS
类型: string | TNode。通用类型定义"},{attribute:"content",description:"Slot / Function",type:"-
",default:"用户自定义引导弹框的内容,一旦存在,此时除 placement、offset和element 外,其它属性全部失
效)。TS 类型: TNode。通用类型定义"},{attribute:"element",description:"String / Function",type:"-
",default:"必需。高亮的节点。数据类型为 String 时,会被当作选择器处理,进行节点查询。示
例: '#tdesign' 或 () => document.querySelector('#tdesign')。TS 类型: AttachNode。通用类型定义"},
{attribute:"highlightContent",description:"Slot / Function",type:"-","default:"用户自定义的高亮框 (仅
当 mode 为 popup 时生效)。TS 类型: TNode。通用类型定义"},
{attribute:"highlightPadding",description:"Number",type:"-","default:"高亮框的内边距"},
{attribute:"mode",description:"String",type:"-","default:"引导框的类型。可选项: popup/dialog"},
{attribute:"nextButtonProps",description:"Object",type:"-","default:"用于自定义当前引导框的下一步按钮
的内容。TS 类型: ButtonProps"},{attribute:"offset",description:"Array",type:"-","default:"【讨论确认
中】相对于 placement 的偏移量,示例: [-10, 20] 或 ['10px', '8px']。TS 类型: Array<string |
number>"},{attribute:"placement",description:"String",type:"top",default:"引导框相对于高亮元素出现
的位置。TS 类型: StepPopupPlacement | StepDialogPlacement type StepPopupPlacement =
'top' | 'left' | 'right' | 'bottom' | 'top-left' | 'top-right' | 'bottom-left' | 'bottom-right' | 'left-top' | 'left-
bottom' | 'right-top' | 'right-bottom' type StepDialogPlacement = 'top' | 'center'。详细类型定义"},
{attribute:"popupProps",description:"Object",type:"-","default:"透传全部属性到 Popup 组件。mode=popup
时有效。TS 类型: PopupProps, Popup API Documents。详细类型定义"},
{attribute:"prevButtonProps",description:"Object",type:"-","default:"用于自定义当前引导框的上一步按钮
的内容。TS 类型: ButtonProps"},
{attribute:"showOverlay",description:"Boolean",type:"true",default:"是否出现遮罩层"},
{attribute:"skipButtonProps",description:"Object",type:"-","default:"用于自定义当前步骤引导框的跳过按
钮的内容。TS 类型: ButtonProps"},{attribute:"stepOverlayClass",description:"String",type:"-
",default:"覆盖引导框的类名"},{attribute:"title",description:"String",type:"-","default:"当前步骤的标
题内容"}]]]],{componentName:"message",apiData:[{docTitle:"Message Props说明",tableRows:
[{{attribute:"closeBtn",description:"String / Boolean / Slot / Function",type:"undefined",default:"关
闭按钮,可以自定义。值为 true 显示默认关闭按钮,值为 false 不显示关闭按钮。值类型为 string 则直接显
示值,如:“关闭”。也可以完全自定义按钮。TS 类型: string | boolean | TNode。通用类型定义"},
{attribute:"content",description:"String / Slot / Function",type:"-","default:"用于自定义消息弹出内
容。TS 类型: string | TNode。通用类型定义"},
{attribute:"duration",description:"Number",type:"3000",default:"消息内置计时器,计时到达时会触发
duration-end 事件。单位: 毫秒。值为 0 则表示没有计时器。"},{attribute:"icon",description:"Boolean /
Slot / Function",type:"true",default:"用于自定义消息前面的图标,优先级大于 theme 设定的图标。值为
false 则不显示图标,值为 true 显示 theme 设定图标。TS 类型: boolean | TNode。通用类型定义"},
{attribute:"theme",description:"String",type:"info",default:"消息组件风格。可选项:
info/success/warning/error/question/loading。TS 类型: MessageThemeList type MessageThemeList =
'info' | 'success' | 'warning' | 'error' | 'question' | 'loading'。详细类型定义"},
{attribute:"onClose",description:"Function",type:"",default:"TS 类型: (context: { trigger: 'close-
click' | 'duration-end', e?: MouseEvent }) => void关闭消息时触发"},
{attribute:"onCloseBtnClick",description:"Function",type:"",default:"TS 类型: (context: { e:
MouseEvent }) => void当关闭按钮存在时,用户点击关闭按钮触发"},
{attribute:"onDurationEnd",description:"Function",type:"",default:"TS 类型: () => void计时结束后触
发"}]],{docTitle:"Message Events说明",tableRows:[{attribute:"close",description:"(context: {
trigger: 'close-click' | 'duration-end', e?: MouseEvent })",type:"关闭消息时触发"},
{attribute:"close-btn-click",description:"(context: { e: MouseEvent })",type:"当关闭按钮存在时,用户
点击关闭按钮触发"},{attribute:"duration-end",description:"-","type:"计时结束后触发"}]],
{docTitle:"MessageOptions说明",tableRows:[{attribute:"attach",description:"String /
Function",type:"body",default:"指定弹框挂载的父节点。数据类型为 String 时,会被当作选择器处理,进
行节点查询。示例: 'body' 或 () => document.body。TS 类型: AttachNode。通用类型定义"},

```

```

{attribute:"className",description:"String",type:"-",default:"类名"},
{attribute:"offset",description:"Array",type:"-",default:"相对于 placement 的偏移量, 示例: [-10, 20] 或 ['10em', '8rem']。TS 类型: Array<string | number>"},
{attribute:"placement",description:"String",type:"top",default:"弹出消息位置。可选项: center/top/left/right/bottom/top-left/top-right/bottom-left/bottom-right。TS 类型: MessagePlacementList type MessagePlacementList = 'center' | 'top' | 'left' | 'right' | 'bottom' | 'top-left' | 'top-right' | 'bottom-left' | 'bottom-right'。详细类型定义"},
{attribute:"style",description:"Object",type:"-",default:"内敛样式。TS 类型: CSSProperties"},
{attribute:"zIndex",description:"Number",type:"5000",default:"消息层级"},
{attribute:"MessageProps",description:"-",type:"-",default:"继承 MessageProps 中的全部 API"}]],
{docTitle:"MessagePlugin说明",tableRows:[]}, {docTitle:"MessagePlugin.info说明",tableRows:[]},
{docTitle:"MessagePlugin.error说明",tableRows:[]}, {docTitle:"MessagePlugin.warning说明",tableRows:[]},
{docTitle:"MessagePlugin.success说明",tableRows:[]}, {docTitle:"MessagePlugin.loading说明",tableRows:[]},
{docTitle:"MessagePlugin.question说明",tableRows:[]},
{docTitle:"MessagePlugin.close说明",tableRows:[]}, {docTitle:"MessagePlugin.closeAll说明",tableRows:[]},
{docTitle:"MessagePlugin.config说明",tableRows:[]}], {componentName:"notification",apiData:
[ {docTitle:"Notification Props说明",tableRows:[ {attribute:"closeBtn",description:"String / Boolean / Slot / Function",type:"undefined",default:"关闭按钮, 可以自定义。值为 true 显示默认关闭按钮, 值为 false 不显示关闭按钮。值类型为 string 则直接显示值, 如: '关闭'。值类型为 TNode, 则表示呈现自定义按钮示例。TS 类型: string | boolean | TNode。通用类型定义"}, {attribute:"content",description:"String / Slot / Function",type:"-",default:"自定义内容。TS 类型: string | TNode。通用类型定义"}, {attribute:"default",description:"String / Slot / Function",type:"-",default:"自定义内容, 同 content。TS 类型: string | TNode。通用类型定义"}, {attribute:"duration",description:"Number",type:"3000",default:"消息显示时长, 单位: 毫秒。值为 0 表示永久显示"}, {attribute:"footer",description:"String / Slot / Function",type:"-",default:"用于自定义底部内容。TS 类型: string | TNode。通用类型定义"}, {attribute:"icon",description:"Boolean / Slot / Function",type:"true",default:"用于自定义消息通知前面的图标, 优先级大于 theme 设定的图标。值为 false 则不显示图标, 值为 true 显示 theme 设定图标。TS 类型: boolean | TNode。通用类型定义"}, {attribute:"theme",description:"String",type:"info",default:"消息类型。可选项: info/success/warning/error。TS 类型: NotificationThemeList type NotificationThemeList = 'info' | 'success' | 'warning' | 'error'。详细类型定义"}, {attribute:"title",description:"String / Slot / Function",type:"-",default:"标题。TS 类型: string | TNode。通用类型定义"}, {attribute:"onCloseBtnClick",description:"Function",type:"-",default:"TS 类型: (context: { e: MouseEvent }) => void点击关闭按钮时触发"}, {attribute:"onDurationEnd",description:"Function",type:"-",default:"TS 类型: () => void计时结束时触发"}]], {docTitle:"Notification Events说明",tableRows:[ {attribute:"close-btn-click",description:"(context: { e: MouseEvent })",type:"点击关闭按钮时触发"}, {attribute:"duration-end",description:"-",type:"计时结束时触发"}]], {docTitle:"NotificationOptions说明",tableRows:
[ {attribute:"attach",description:"String / Function",type:"body",default:"指定消息通知挂载的父节点。数据类型为 String 时, 会被当作选择器处理, 进行节点查询。示例: 'body' 或 () => document.body。TS 类型: AttachNode。通用类型定义"}, {attribute:"offset",description:"Array",type:"-",default:"相对于 placement 的偏移量, 示例: [-10, 20] 或 ['10em', '8rem']。TS 类型: Array<string | number>"}, {attribute:"placement",description:"String",type:"top-right",default:"消息弹出位置。可选项: top-left/top-right/bottom-left/bottom-right。TS 类型: NotificationPlacementList type NotificationPlacementList = 'top-left' | 'top-right' | 'bottom-left' | 'bottom-right'。详细类型定义"}, {attribute:"zIndex",description:"Number",type:"6000",default:"消息通知层级"}, {attribute:"NotificationProps",description:"-",type:"-",default:"继承 NotificationProps 中的全部 API"}]], {docTitle:"NotificationPlugin说明",tableRows:[]}, {docTitle:"NotificationPlugin.info说明",tableRows:[]}, {docTitle:"NotificationPlugin.warning说明",tableRows:[]}, {docTitle:"NotificationPlugin.error说明",tableRows:[]}, {docTitle:"NotificationPlugin.success说明",tableRows:[]}, {docTitle:"NotificationPlugin.close说明",tableRows:[]}, {docTitle:"NotificationPlugin.closeAll说明",tableRows:[]}, {docTitle:"NotificationPlugin.config说明",tableRows:[]}], {componentName:"popconfirm",apiData:[ {docTitle:"Popconfirm Props说明",tableRows:
[ {attribute:"cancelBtn",description:"String / Object / Slot / Function",type:"-",default:"取消按钮, 可自定义。值为 null 则不显示取消按钮。值类型为字符串, 则表示自定义按钮文本, 值类型为 Object 则表示透传 Button 组件属性。使用 TNode 自定义按钮时, 需自行控制取消事件。TS 类型: string | ButtonProps | TNode, Button API Documents。通用类型定义。详细类型定义"}, {attribute:"confirmBtn",description:"String / Object / Slot / Function",type:"-",default:"确认按钮。值类型为字符串, 则表示自定义按钮文本, 值类型为 Object 则表示透传 Button 组件属性。使用 TNode 自定义按钮时, 需自行控制确认事件。TS 类型: string | ButtonProps | TNode。通用类型定义"}, {attribute:"content",description:"String / Slot / Function",type:"-",default:"确认框内容。TS 类型: string | TNode。通用类型定义"}, {attribute:"default",description:"String / Slot / Function",type:"-",default:"触发元素, 同 triggerElement。TS 类型: string | TNode。通用类型定义"}, {attribute:"destroyOnClose",description:"Boolean",type:"true",default:"是否在关闭浮层时销毁浮层"}, {attribute:"icon",description:"Slot / Function",type:"-",default:"确认框图标。TS 类型: TNode。通用类型定义"}, {attribute:"placement",description:"String",type:"top",default:"浮层出现位置。可选项: top/left/right/bottom/top-left/top-right/bottom-left/bottom-right/left-top/left-bottom/right-top/right-bottom"}, {attribute:"popupProps",description:"Object",type:"-",default:"透传 Popup 组件属性。TS 类型: PopupProps, Popup API Documents。详细类型定义"},

```



```

{attribute:"showArrow",description:"Boolean",type:"true",default:"是否显示浮层箭头"},
{attribute:"theme",description:"String",type:"default",default:"文字提示风格。可选项:
default/warning/danger"},{attribute:"triggerElement",description:"String / Slot / Function",type:"-
",default:"触发元素。TS 类型: string | TNode。通用类型定义"},
{attribute:"visible",description:"Boolean",type:"- ",default:"是否显示气泡确认框。支持语法糖 v-
model:visible"},{attribute:"defaultVisible",description:"Boolean",type:"- ",default:"是否显示气泡确认
框。非受控属性"},{attribute:"onCancel",description:"Function",type:"",default:"TS 类型: (options: {
e: MouseEvent }) => void点击取消按钮时触发"},
{attribute:"onConfirm",description:"Function",type:"",default:"TS 类型: (options: { e: MouseEvent })
=> void点击确认按钮时触发"},{attribute:"onVisibleChange",description:"Function",type:"",default:"TS
类型: (visible: boolean, context?: PopconfirmVisibleChangeContext) => void确认框显示或隐藏时触发。详细类型定义。interface PopconfirmVisibleChangeContext { trigger?: TriggerSource; e?: MouseEvent }type
TriggerSource = 'cancel' | 'confirm' | 'document' | 'trigger-element-click'"}]],
{docTitle:"Popconfirm Events说明",tableRows:[{attribute:"cancel",description:"(options: { e:
MouseEvent })",type:"点击取消按钮时触发"},{attribute:"confirm",description:"(options: { e:
MouseEvent })",type:"点击确认按钮时触发"},{attribute:"visible-change",description:"(visible:
boolean, context?: PopconfirmVisibleChangeContext)",type:"确认框显示或隐藏时触发。详细类型定义。
interface PopconfirmVisibleChangeContext { trigger?: TriggerSource; e?: MouseEvent }type
TriggerSource = 'cancel' | 'confirm' | 'document' | 'trigger-element-click'"}]]}],
{componentName:"popup",apiData:[{docTitle:"Popup Props说明",tableRows:
[ {attribute:"attach",description:"String / Function",type:"body",default:"制定挂载节点。数据类型为
String 时,会被当作选择器处理,进行节点查询。示例: 'body' 或 () => document.body。TS 类型:
AttachNode。通用类型定义"},{attribute:"content",description:"String / Slot / Function",type:"-
",default:"浮层里面的内容。TS 类型: string | TNode。通用类型定义"},
{attribute:"default",description:"String / Slot / Function",type:"- ",default:"触发元素,同
triggerElement。TS 类型: string | TNode。通用类型定义"},{attribute:"delay",description:"Number /
Array",type:"- ",default:"延时显示或隐藏浮层,[延迟显示的时间,延迟隐藏的时间],单位:毫秒。如果只有
一个时间,则表示显示和隐藏的延迟时间相同。示例 '300' 或者 [200, 200]。默认为: [250, 150]。TS 类型:
number | Array<number>"},{attribute:"destroyOnClose",description:"Boolean",type:"false",default:"是
否在关闭浮层时销毁浮层"},{attribute:"disabled",description:"Boolean",type:"- ",default:"是否禁用组
件"},{attribute:"hideEmptyPopup",description:"Boolean",type:"false",default:"浮层是否隐藏空内容,默
认不隐藏"},{attribute:"overlayClassName",description:"String / Object / Array",type:"- ",default:"浮
层类名,示例: 'name1 name2 name3' 或 ['name1', 'name2'] 或 [{ 'name1': true } ]。TS 类型:
ClassName。通用类型定义"},{attribute:"overlayInnerClassName",description:"String / Object / Array",type:"-
",default:"浮层内容部分类名,示例: 'name1 name2 name3' 或 ['name1', 'name2'] 或 [{ 'name1': true
} ]。TS 类型: ClassName。通用类型定义"},{attribute:"overlayInnerStyle",description:"Boolean / Object
/ Function",type:"- ",default:"浮层内容部分样式,第一个参数 triggerElement 表示触发元素 DOM 节点,第
二个参数 popupElement 表示浮层元素 DOM 节点。TS 类型: Styles | ((triggerElement: HTMLElement,
popupElement: HTMLElement) => Styles)。通用类型定义"},{attribute:"overlayStyle",description:"Boolean
/ Object / Function",type:"- ",default:"浮层样式,第一个参数 triggerElement 表示触发元素 DOM 节点,第
二个参数 popupElement 表示浮层元素 DOM 节点。TS 类型: Styles | ((triggerElement: HTMLElement,
popupElement: HTMLElement) => Styles)。通用类型定义"},
{attribute:"placement",description:"String",type:"top",default:"浮层出现位置。TS 类型:
PopupPlacement type PopupPlacement = 'top' | 'left' | 'right' | 'bottom' | 'top-left' | 'top-right' | 'bottom-
left' | 'bottom-right' | 'left-top' | 'left-bottom' | 'right-top' | 'right-bottom'。详细类型定义"},
{attribute:"popperOptions",description:"Object",type:"- ",default:"popper 初始化配置,详情参考
https://popper.js.org/docs/"},{attribute:"showArrow",description:"Boolean",type:"false",default:"是
否显示浮层箭头"},{attribute:"trigger",description:"String",type:"hover",default:"触发浮层出现的方
式。可选项: hover/click/focus/mousedown/context-menu"},
{attribute:"triggerElement",description:"String / Slot / Function",type:"- ",default:"触发元素。值类
型为字符串表示元素选择器。TS 类型: string | TNode。通用类型定义"},
{attribute:"visible",description:"Boolean",type:"- ",default:"是否显示浮层。支持语法糖 v-model 或 v-
model:visible。TS 类型: boolean"},{attribute:"defaultVisible",description:"Boolean",type:"-
",default:"是否显示浮层。非受控属性。TS 类型: boolean"},
{attribute:"zIndex",description:"Number",type:"- ",default:"组件层级,Web 侧样式默认为 5500,移动端和
小程序样式默认为 1500"},{attribute:"onScroll",description:"Function",type:"",default:"TS 类型:
(context: { e: WheelEvent }) => void下拉选项滚动事件"},
{attribute:"onScrollToBottom",description:"Function",type:"",default:"TS 类型: (context: { e:
WheelEvent }) => void下拉滚动触底事件,常用于滚动到底执行具体业务逻辑"},
{attribute:"onVisibleChange",description:"Function",type:"",default:"TS 类型: (visible: boolean,
context: PopupVisibleChangeContext) => void当浮层隐藏或显示时触发,trigger=document 表示点击非浮层元
素触发;trigger=context-menu 表示右击触发。详细类型定义。interface PopupVisibleChangeContext { e?:
PopupTriggerEvent; trigger?: PopupTriggerSource }type PopupTriggerEvent = MouseEvent | FocusEvent |
KeyboardEventtype PopupTriggerSource = 'document' | 'trigger-element-click' | 'trigger-element-
hover' | 'trigger-element-blur' | 'trigger-element-focus' | 'trigger-element-mousedown' | 'context-
menu' | 'keydown-esc'"}]],{docTitle:"Popup Events说明",tableRows:[{attribute:"scroll",description:"
(context: { e: WheelEvent })",type:"下拉选项滚动事件"},{attribute:"scroll-to-bottom",description:"
(context: { e: WheelEvent })",type:"下拉滚动触底事件,常用于滚动到底执行具体业务逻辑"},
{attribute:"visible-change",description:"(visible: boolean, context:

```

```
PopupVisibleChangeContext)", type: "当浮层隐藏或显示时触发, trigger=document 表示点击非浮层元素触发; trigger=context-menu 表示右击触发。详细类型定义。interface PopupVisibleChangeContext { e?:  
PopupTriggerEvent; trigger?: PopupTriggerSource }type PopupTriggerEvent = MouseEvent | FocusEvent |  
KeyboardEventtype PopupTriggerSource = 'document' | 'trigger-element-click' | 'trigger-element-  
hover' | 'trigger-element-blur' | 'trigger-element-focus' | 'trigger-element-mousedown' | 'context-  
menu' | 'keydown-esc'"]}]}}]};
```