# 官网

https://element.eleme.cn/#/zh-CN

# 基本概念



# 特点

## ****1、丰富的组件****

Element UI 包含了大量常用的 UI 组件，如按钮、表单、表格、对话框、导航菜单、轮播图等，覆盖了大部分常见的界面元素和交互组件。

## ****2、响应式设计****

Element UI 的组件都经过了响应式设计，在不同设备上都能够提供良好的显示效果和用户体验，支持移动端和桌面端的应用开发。

## ****3、主题定制****

Element UI 提供了丰富的主题定制选项，开发者可以根据项目需求定制自己的主题样式，以满足不同项目的设计风格要求。

## ****4、文档详实****

Element UI 官方提供了详细的文档和示例，帮助开发者快速上手并深入了解每个组件的用法和属性。

## ****5、社区活跃****

作为一款流行的前端 UI 框架，Element UI 拥有庞大的开发者社区，提供了丰富的第三方插件和扩展，同时也有很多社区贡献的解决方案和教程。

## ****6、与 Vue.js 紧密集成****

Element UI 是专为 Vue.js 开发的 UI 框架，完美地结合了 Vue.js 的数据驱动和组件化开发思想，使得开发者可以更高效地构建复杂的前端应用。

# 事件与方法的区别

## 基本概念

在 Element UI 中，事件（Events）和方法（Methods）是两个不同的概念，它们用于处理用户交互和组件逻辑的不同方面。

## 事件

1、定义：事件是在组件中触发的动作或交互行为，比如点击、输入等用户操作。

2、使用方式：通常通过监听组件提供的事件来响应用户的操作，例如使用 @event-name 或 v-on:event-name 来绑定事件处理函数。

3、作用：事件用于捕获用户的交互操作，然后执行相应的逻辑或处理。

## 方法

1、定义：方法是组件内部的函数，用于定义组件的具体行为和逻辑。

2、使用方式：在 Vue.js 组件中，可以在 methods 对象中定义各种方法，然后在模板或其他方法中调用这些方法。

3、作用：方法用于实现组件的具体功能，可以包含业务逻辑、数据处理、状态更新等操作。

## 区别总结

1、事件主要用于捕获用户的操作行为，如点击、输入等，触发相应的处理逻辑。

2、方法用于定义组件的具体行为和逻辑，实现功能和处理数据的操作。

## 示例代码



在上面的示例中，@click="handleClick" 绑定了按钮的点击事件，当按钮被点击时，会触发 handleClick 方法执行相应的逻辑。

# 总结

事件和方法在 Element UI 中扮演着不同的角色，事件用于捕获用户的交互操作，而方法则用于定义组件的具体行为和逻辑。

# Icon图标

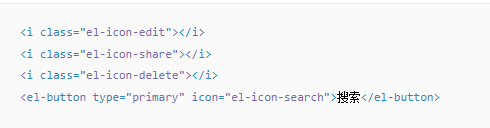
## 基本概念

提供了一套常用的图标集合。

## 使用方法

直接通过设置类名为 el-icon-iconName 来使用即可。

## 代码示例



# button按钮

## 基本概念

常用的操作按钮。

## 使用方法

使用type、plain、round和circle属性来定义 Button 的样式。



## 属性

### type

定义按钮的类型

### plain

减弱按钮的背景色

### round

定义圆角按钮

### circle

定义圆形按钮

### disabled

可以使用disabled属性来定义按钮是否可用，它接受一个Boolean值。



### size

尺寸

### loading

是否加载中状态

### icon

图标类名

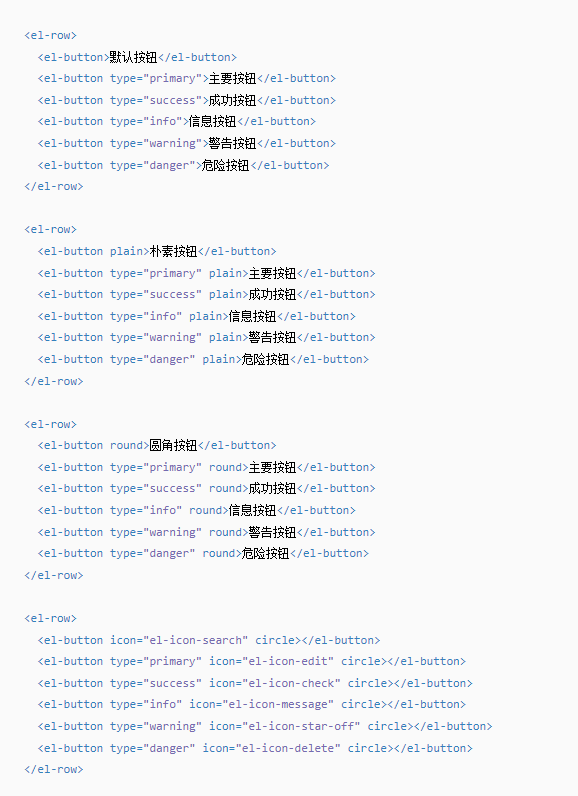
### autofocus

是否默认聚焦

### native-type

原生 type 属性

## 代码示例



# Link文字链接

## 基本概念

文字超链接

## 使用方法



## 属性

### type

类型

### underline

是否下划线

### disabled

是否禁用状态

### href

原生 href 属性

### icon

图标类名

# Radio单选框

## 基本概念

在一组备选项中进行单选

## 使用方法

由于选项默认可见，不宜过多，若选项过多，建议使用 Select 选择器。



要使用 Radio 组件，只需要设置v-model绑定变量，选中意味着变量的值为相应 Radio label属性的值，label可以是String、Number或Boolean。



## Radio属性

### value / v-model

绑定值

### label

radio的value

### disabled

是否禁用

### border

是否显示边框

### size

Radio 的尺寸，仅在 border 为真时有效

### name

原生 name 属性

## Radio事件

### change

绑定值变化时触发的事件。

回调参数：选中的 Radio label 值

## 代码示例



# Checkbox 多选框

## 基本概念

一组备选项中进行多选

## 使用方法

单独使用可以表示两种状态之间的切换，写在标签中的内容为 checkbox 按钮后的介绍。

在el-checkbox元素中定义v-model绑定变量，单一的checkbox中，默认绑定变量的值会是Boolean，选中为true。



## Checkbox属性

### value/v-model

绑定的值

### label

选中状态的值（只有在checkbox-group或者绑定对象类型为array时有效）

### true-label

选中时的值

### false-label

没有选中时的值

### disabled

是否禁用

### border

### size

### name

### checked

### indeterminate

## Checkbox事件

### change

当绑定值变化时触发的事件

回调参数：更新后的值

# input输入框

## 基本概念

通过鼠标或键盘输入字符。

通常情况下，应当处理 input 事件，并更新组件的绑定值（或使用v-model）。否则，输入框内显示的值将不会改变。不支持 v-model 修饰符。

## 使用方法



## input属性

### type

类型

### value/v-model

绑定值

### maxlength

原生属性，最大输入长度

### minlength

原生属性，最小输入长度

### show-word-limit

是否显示输入字数统计，只在 type = "text" 或 type = "textarea" 时有效

### placeholder

输入框占位文本

### clearable

是否可清空

### show-password

是否显示切换密码图标

### disabled

禁用

### size

输入框尺寸，只在 type!="textarea" 时有效

### prefix-icon

输入框头部图标

### suffix-icon

输入框尾部图标

### rows

输入框行数，只对 type="textarea" 有效

## input插槽

### prefix

输入框头部内容，只对 type="text" 有效

### suffix

输入框尾部内容，只对 type="text" 有效

### prepend

输入框前置内容，只对 type="text" 有效

### append

输入框后置内容，只对 type="text" 有效

## input事件

### change

仅在输入框失去焦点或用户按下回车时触发

回调参数：(value: string | number)

### input

在 Input 值改变时触发

回调参数：(value: string | number)

## 示例代码



# Form表单

## 基本概念

由输入框、选择器、单选框、多选框等控件组成，用以收集、校验、提交数据。

## 使用方法

在 Form 组件中，每一个表单域由一个 Form-Item 组件构成，表单域中可以放置各种类型的表单控件，包括 Input、Select、Checkbox、Radio、Switch、DatePicker、TimePicker

## 行内表单

当垂直方向空间受限且表单较简单时，可以在一行内放置表单。

设置 inline 属性可以让表单域变为行内的表单域





## form属性

### model

表单数据对象

### rules

表单验证规则

### inline

行内表单模式

### hide-required-asterisk

是否显示必填字段的标签旁边的红色星号

### show-message

是否显示校验错误信息

## form方法

## form事件

### validate

任一表单项被校验后触发

回调函数：

被校验的表单项 prop 值，校验是否通过，错误消息（如果存在）

# Table表格

## table属性

### data

显示的数据

### height

Table 的高度，默认为自动高度。如果 height 为 number 类型，单位 px；如果 height 为 string 类型，则这个高度会设置为 Table 的 style.height 的值，Table 的高度会受控于外部样式。

### max-height

Table 的最大高度。合法的值为数字或者单位为 px 的高度。

### stripe

是否为斑马纹 table

### border

是否带有纵向边框

### size

Table 的尺寸

### show-header

是否显示表头

## table column属性

### type

对应列的类型。如果设置了 selection 则显示多选框；如果设置了 index 则显示该行的索引（从 1 开始计算）；如果设置了 expand 则显示为一个可展开的按钮

### index

如果设置了 type=index，可以通过传递 index 属性来自定义索引

### label

显示的标题

### prop

对应列内容的字段名，也可以使用 property 属性

# Pagination 分页

## 基本概念

当数据量过多时，使用分页分解数据。

## 使用方法

设置layout，表示需要显示的内容，用逗号分隔，布局元素会依次显示。prev表示上一页，next为下一页，pager表示页码列表，除此以外还提供了jumper和total，size和特殊的布局符号->，->后的元素会靠右显示，jumper表示跳页元素，total表示总条目数，size用于设置每页显示的页码数量。

## 属性

### background

是否为分页按钮添加背景色

### page-size

每页显示条目个数，支持 .sync 修饰符，默认值：10

### page-sizes

每页显示个数选择器的选项设置，默认值：[10, 20, 30, 40, 50, 100]

### total

总条目数

### page-count

总页数，total 和 page-count 设置任意一个就可以达到显示页码的功能；如果要支持 page-sizes 的更改，则需要使用 total 属性

### pager-count

页码按钮的数量，当总页数超过该值时会折叠

### layout

组件布局，子组件名用逗号分隔

默认值：'prev, pager, next, jumper, ->, total'

### current-page

当前页数，支持 .sync 修饰符，默认值是1。

## 事件

### size-change

pageSize 改变时会触发

### current-change

currentPage 改变时会触发

## 示例代码



# DatePicker 日期选择器

## 基本概念

用于选择或输入日期

## 使用方法

基本单位由type属性指定。快捷选项需配置picker-options对象中的shortcuts，禁用日期通过 disabledDate 设置，传入函数。

## 属性

### value / v-model

绑定值

### readonly

完全只读

### disabled

禁用

### clearable

是否显示清除按钮

### size

输入框尺寸

### placeholder

非范围选择时的占位内容

## 示例代码



