

Dialog 对话框

在保留当前页面状态的情况下，告知用户并承载相关操作。

基本用法

Dialog 弹出一个对话框，适合需要定制性更大的场景。

::demo 需要设置 `visible` 属性，它接收 `Boolean`，当为 `true` 时显示 Dialog。Dialog 分为两个部分：`body` 和 `footer`，`footer` 需要具名为 `footer` 的 `slot`。`title` 属性用于定义标题，它是可选的，默认值为空。最后，本例还展示了 `before-close` 的用法。

```
<el-button type="text" @click="dialogVisible = true">点击打开 Dialog</el-button>

<el-dialog
  title="提示"
  :visible.sync="dialogVisible"
  width="30%"
  :before-close="handleClose">
  <span>这是一段信息</span>
  <span slot="footer" class="dialog-footer">
    <el-button @click="dialogVisible = false">取 消</el-button>
    <el-button type="primary" @click="dialogVisible = false">确 定</el-button>
  </span>
</el-dialog>

<script>
  export default {
    data() {
      return {
        dialogVisible: false
      };
    },
    methods: {
      handleClose(done) {
        this.$confirm('确认关闭? ')
          .then(_ => {
            done();
          })
          .catch(_ => {});
      }
    }
  };
</script>
```

...

::tip `before-close` 仅当用户通过点击关闭图标或遮罩关闭 Dialog 时起效。如果你在 `footer` 具名 slot 里添加了用于关闭 Dialog 的按钮，那么可以在按钮的点击回调函数里加入 `before-close` 的相关逻辑。...

自定义内容

Dialog 组件的内容可以是任意的，甚至可以是表格或表单，下面是应用了 Element Table 和 Form 组件的两个样例。

:::demo

```
<!-- Table -->
<el-button type="text" @click="dialogTableVisible = true">打开嵌套表格的 Dialog</el-button>

<el-dialog title="收货地址" :visible.sync="dialogTableVisible">
  <el-table :data="gridData">
    <el-table-column property="date" label="日期" width="150"></el-table-column>
    <el-table-column property="name" label="姓名" width="200"></el-table-column>
    <el-table-column property="address" label="地址"></el-table-column>
  </el-table>
</el-dialog>

<!-- Form -->
<el-button type="text" @click="dialogFormVisible = true">打开嵌套表单的 Dialog</el-button>

<el-dialog title="收货地址" :visible.sync="dialogFormVisible">
  <el-form :model="form">
    <el-form-item label="活动名称" :label-width="formLabelWidth">
      <el-input v-model="form.name" autocomplete="off"></el-input>
    </el-form-item>
    <el-form-item label="活动区域" :label-width="formLabelWidth">
      <el-select v-model="form.region" placeholder="请选择活动区域">
        <el-option label="区域一" value="shanghai"></el-option>
        <el-option label="区域二" value="beijing"></el-option>
      </el-select>
    </el-form-item>
  </el-form>
  <div slot="footer" class="dialog-footer">
    <el-button @click="dialogFormVisible = false">取 消</el-button>
    <el-button type="primary" @click="dialogFormVisible = false">确 定</el-button>
  </div>
</el-dialog>

<script>
export default {
  data() {
    return {
      gridData: [{
        date: '2016-05-02',
        name: '王小虎',
        address: '上海市普陀区金沙江路 1518 弄'
      }, {
        date: '2016-05-04',
        name: '王小虎',
        address: '上海市普陀区金沙江路 1518 弄'
      }, {
        date: '2016-05-01',
```

```

        name: '王小虎',
        address: '上海市普陀区金沙江路 1518 弄'
      }, {
        date: '2016-05-03',
        name: '王小虎',
        address: '上海市普陀区金沙江路 1518 弄'
      }
    ],
    dialogTableVisible: false,
    dialogFormVisible: false,
    form: {
      name: '',
      region: '',
      date1: '',
      date2: '',
      delivery: false,
      type: [],
      resource: '',
      desc: ''
    },
    formLabelWidth: '120px'
  };
}
};
</script>

```

...

嵌套的 Dialog

如果需要在 一个 Dialog 内部嵌套另一个 Dialog，需要使用 `append-to-body` 属性。:::demo 正常情况下，我们不建议使用嵌套的 Dialog，如果需要在页面上同时显示多个 Dialog，可以将它们平级放置。对于确实需要嵌套 Dialog 的场景，我们提供了 `append-to-body` 属性。将内层 Dialog 的该属性设置为 `true`，它就会插入至 `body` 元素上，从而保证内外层 Dialog 和遮罩层级关系的正确。

```

<template>
  <el-button type="text" @click="outerVisible = true">点击打开外层 Dialog</el-button>

  <el-dialog title="外层 Dialog" :visible.sync="outerVisible">
    <el-dialog
      width="30%"
      title="内层 Dialog"
      :visible.sync="innerVisible"
      append-to-body>
    </el-dialog>
    <div slot="footer" class="dialog-footer">
      <el-button @click="outerVisible = false">取 消</el-button>
      <el-button type="primary" @click="innerVisible = true">打开内层 Dialog</el-
button>
    </div>
  </el-dialog>
</template>

```

```

<script>
  export default {
    data() {
      return {
        outerVisible: false,
        innerVisible: false
      };
    }
  }
}
</script>

```

...

居中布局

标题和底部可水平居中

::demo 将 `center` 设置为 `true` 即可使标题和底部居中。`center` 仅影响标题和底部区域。Dialog 的内容是任意的，在一些情况下，内容并不适合居中布局。如果需要内容也水平居中，请自行为其添加 CSS。

```

<el-button type="text" @click="centerDialogVisible = true">点击打开 Dialog</el-button>

<el-dialog
  title="提示"
  :visible.sync="centerDialogVisible"
  width="30%"
  center>
  <span>需要注意的是内容是默认不居中的</span>
  <span slot="footer" class="dialog-footer">
    <el-button @click="centerDialogVisible = false">取 消</el-button>
    <el-button type="primary" @click="centerDialogVisible = false">确 定</el-button>
  </span>
</el-dialog>

<script>
  export default {
    data() {
      return {
        centerDialogVisible: false
      };
    }
  };
</script>

```

...

::tip Dialog 的内容是懒渲染的，即在第一次被打开之前，传入的默认 slot 不会被渲染到 DOM 上。因此，如果需要执行 DOM 操作，或通过 `ref` 获取相应组件，请在 `open` 事件回调中进行。...

:::tip 如果 `visible` 属性绑定的变量位于 Vuex 的 store 内，那么 `.sync` 不会正常工作。此时需要去除 `.sync` 修饰符，同时监听 Dialog 的 `open` 和 `close` 事件，在事件回调中执行 Vuex 中对应的 mutation 更新 `visible` 属性绑定的变量的值。 :::

Attributes

参数	说明	类型	可选值	默认值
visible	是否显示 Dialog，支持 .sync 修饰符	boolean	—	false
title	Dialog 的标题，也可通过具名 slot （见下表）传入	string	—	—
width	Dialog 的宽度	string	—	50%
fullscreen	是否为全屏 Dialog	boolean	—	false
top	Dialog CSS 中的 margin-top 值	string	—	15vh
modal	是否需要遮罩层	boolean	—	true
modal-append-to-body	遮罩层是否插入至 body 元素上，若为 false，则遮罩层会插入至 Dialog 的父元素上	boolean	—	true
append-to-body	Dialog 自身是否插入至 body 元素上。嵌套的 Dialog 必须指定该属性并赋值为 true	boolean	—	false
lock-scroll	是否在 Dialog 出现时将 body 滚动锁定	boolean	—	true
custom-class	Dialog 的自定义类名	string	—	—
close-on-click-modal	是否可以通过点击 modal 关闭 Dialog	boolean	—	true
close-on-press-escape	是否可以通过按下 ESC 关闭 Dialog	boolean	—	true
show-close	是否显示关闭按钮	boolean	—	true
before-close	关闭前的回调，会暂停 Dialog 的关闭	function(done), done 用于关闭 Dialog	—	—
center	是否对头部和底部采用居中布局	boolean	—	false
destroy-on-close	关闭时销毁 Dialog 中的元素	boolean	—	false

Slot

name	说明
—	Dialog 的内容
title	Dialog 标题区的内容

footer	Dialog 按钮操作区的内容
--------	-----------------

Events

事件名称	说明	回调参数
open	Dialog 打开的回调	—
opened	Dialog 打开动画结束时的回调	—
close	Dialog 关闭的回调	—
closed	Dialog 关闭动画结束时的回调	—