

Form 表单

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

典型表单

包括各种表单项，比如输入框、选择器、开关、单选框、多选框等。

:::demo 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

```
<el-form ref="form" :model="form" label-width="80px">
  <el-form-item label="活动名称">
    <el-input v-model="form.name"></el-input>
  </el-form-item>
  <el-form-item label="活动区域">
    <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-item label="活动时间">
    <el-col :span="11">
      <el-date-picker type="date" placeholder="选择日期" v-model="form.date1"
style="width: 100%;"></el-date-picker>
    </el-col>
    <el-col class="line" :span="2"></el-col>
    <el-col :span="11">
      <el-time-picker placeholder="选择时间" v-model="form.date2" style="width:
100%;"></el-time-picker>
    </el-col>
  </el-form-item>
  <el-form-item label="即时配送">
    <el-switch v-model="form.delivery"></el-switch>
  </el-form-item>
  <el-form-item label="活动性质">
    <el-checkbox-group v-model="form.type">
      <el-checkbox label="美食/餐厅线上活动" name="type"></el-checkbox>
      <el-checkbox label="地推活动" name="type"></el-checkbox>
      <el-checkbox label="线下主题活动" name="type"></el-checkbox>
      <el-checkbox label="单纯品牌曝光" name="type"></el-checkbox>
    </el-checkbox-group>
  </el-form-item>
  <el-form-item label="特殊资源">
    <el-radio-group v-model="form.resource">
      <el-radio label="线上品牌商赞助"></el-radio>
      <el-radio label="线下场地免费"></el-radio>
    </el-radio-group>
  </el-form-item>
  <el-form-item label="活动形式">
    <el-input type="textarea" v-model="form.desc"></el-input>
  </el-form-item>
```

```

<el-form-item>
  <el-button type="primary" @click="onSubmit">立即创建</el-button>
  <el-button>取消</el-button>
</el-form-item>
</el-form>
<script>
  export default {
    data() {
      return {
        form: {
          name: '',
          region: '',
          date1: '',
          date2: '',
          delivery: false,
          type: [],
          resource: '',
          desc: ''
        }
      }
    },
    methods: {
      onSubmit() {
        console.log('submit!');
      }
    }
  }
</script>

```

...

tip W3C 标准中有如下[规定](#)：

When there is only one single-line text input field in a form, the user agent should accept Enter in that field as a request to submit the form.

即：当一个 form 元素中只有一个输入框时，在该输入框中按下回车应提交该表单。如果希望阻止这一默认行为，可以在 `<el-form>` 标签上添加 `@submit.native.prevent`。

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。

demo 设置 `inline` 属性可以让表单域变为行内的表单域

```

<el-form :inline="true" :model="formInline" class="demo-form-inline">
  <el-form-item label="审批人">
    <el-input v-model="formInline.user" placeholder="审批人"></el-input>
  </el-form-item>
  <el-form-item label="活动区域">
    <el-select v-model="formInline.region" placeholder="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>

```

```

    </el-select>
  </el-form-item>
  <el-form-item>
    <el-button type="primary" @click="onSubmit">查询</el-button>
  </el-form-item>
</el-form>
<script>
  export default {
    data() {
      return {
        formInline: {
          user: '',
          region: ''
        }
      },
      methods: {
        onSubmit() {
          console.log('submit!');
        }
      }
    }
  }
</script>

```

...

对齐方式

根据具体目标和制约因素，选择最佳的标签对齐方式。

::demo 通过设置 `label-position` 属性可以改变表单域标签的位置，可选值为 `top`、`left`，当设为 `top` 时标签会置于表单域的顶部

```

<el-radio-group v-model="labelPosition" size="small">
  <el-radio-button label="left">左对齐</el-radio-button>
  <el-radio-button label="right">右对齐</el-radio-button>
  <el-radio-button label="top">顶部对齐</el-radio-button>
</el-radio-group>
<div style="margin: 20px;"></div>
<el-form :label-position="labelPosition" label-width="80px" :model="formLabelAlign">
  <el-form-item label="名称">
    <el-input v-model="formLabelAlign.name"></el-input>
  </el-form-item>
  <el-form-item label="活动区域">
    <el-input v-model="formLabelAlign.region"></el-input>
  </el-form-item>
  <el-form-item label="活动形式">
    <el-input v-model="formLabelAlign.type"></el-input>
  </el-form-item>
</el-form>
<script>
  export default {

```

```

    data() {
      return {
        labelPosition: 'right',
        formLabelAlign: {
          name: '',
          region: '',
          type: ''
        }
      };
    }
  }
}
</script>

```

...

表单验证

在防止用户犯错的前提下，尽可能让用户更早地发现并纠正错误。

::demo Form 组件提供了表单验证的功能，只需要通过 `rules` 属性传入约定的验证规则，并将 Form-Item 的 `prop` 属性设置为需校验的字段名即可。校验规则参见 [async-validator](#)

```

<el-form :model="ruleForm" :rules="rules" ref="ruleForm" label-width="100px"
class="demo-ruleForm">
  <el-form-item label="活动名称" prop="name">
    <el-input v-model="ruleForm.name"></el-input>
  </el-form-item>
  <el-form-item label="活动区域" prop="region">
    <el-select v-model="ruleForm.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-item label="活动时间" required>
    <el-col :span="11">
      <el-form-item prop="date1">
        <el-date-picker type="date" placeholder="选择日期" v-model="ruleForm.date1"
style="width: 100%;"></el-date-picker>
      </el-form-item>
    </el-col>
    <el-col class="line" :span="2">-</el-col>
    <el-col :span="11">
      <el-form-item prop="date2">
        <el-time-picker placeholder="选择时间" v-model="ruleForm.date2" style="width:
100%;"></el-time-picker>
      </el-form-item>
    </el-col>
  </el-form-item>
  <el-form-item label="即时配送" prop="delivery">
    <el-switch v-model="ruleForm.delivery"></el-switch>
  </el-form-item>
  <el-form-item label="活动性质" prop="type">

```

```

<el-checkbox-group v-model="ruleForm.type">
  <el-checkbox label="美食/餐厅线上活动" name="type"></el-checkbox>
  <el-checkbox label="地推活动" name="type"></el-checkbox>
  <el-checkbox label="线下主题活动" name="type"></el-checkbox>
  <el-checkbox label="单纯品牌曝光" name="type"></el-checkbox>
</el-checkbox-group>
</el-form-item>
<el-form-item label="特殊资源" prop="resource">
  <el-radio-group v-model="ruleForm.resource">
    <el-radio label="线上品牌商赞助"></el-radio>
    <el-radio label="线下场地免费"></el-radio>
  </el-radio-group>
</el-form-item>
<el-form-item label="活动形式" prop="desc">
  <el-input type="textarea" v-model="ruleForm.desc"></el-input>
</el-form-item>
<el-form-item>
  <el-button type="primary" @click="submitForm('ruleForm')">立即创建</el-button>
  <el-button @click="resetForm('ruleForm')">重置</el-button>
</el-form-item>
</el-form>
<script>
export default {
  data() {
    return {
      ruleForm: {
        name: '',
        region: '',
        date1: '',
        date2: '',
        delivery: false,
        type: [],
        resource: '',
        desc: ''
      },
      rules: {
        name: [
          { required: true, message: '请输入活动名称', trigger: 'blur' },
          { min: 3, max: 5, message: '长度在 3 到 5 个字符', trigger: 'blur' }
        ],
        region: [
          { required: true, message: '请选择活动区域', trigger: 'change' }
        ],
        date1: [
          { type: 'date', required: true, message: '请选择日期', trigger: 'change' }
        ],
        date2: [
          { type: 'date', required: true, message: '请选择时间', trigger: 'change' }
        ],
        type: [
          { type: 'array', required: true, message: '请至少选择一个活动性质', trigger:
'change' }

```

```

    ],
    resource: [
      { required: true, message: '请选择活动资源', trigger: 'change' }
    ],
    desc: [
      { required: true, message: '请填写活动形式', trigger: 'blur' }
    ]
  }
};
},
methods: {
  submitForm(formName) {
    this.$refs[formName].validate((valid) => {
      if (valid) {
        alert('submit!');
      } else {
        console.log('error submit!!');
        return false;
      }
    });
  },
  resetForm(formName) {
    this.$refs[formName].resetFields();
  }
}
}
</script>

```

...

自定义校验规则

这个例子中展示了如何使用自定义验证规则来完成密码的二次验证。

:::demo 本例还使用 `status-icon` 属性为输入框添加了表示校验结果的反馈图标。

```

<el-form :model="ruleForm" status-icon :rules="rules" ref="ruleForm" label-
width="100px" class="demo-ruleForm">
  <el-form-item label="密码" prop="pass">
    <el-input type="password" v-model="ruleForm.pass" autocomplete="off"></el-input>
  </el-form-item>
  <el-form-item label="确认密码" prop="checkPass">
    <el-input type="password" v-model="ruleForm.checkPass" autocomplete="off"></el-
input>
  </el-form-item>
  <el-form-item label="年龄" prop="age">
    <el-input v-model.number="ruleForm.age"></el-input>
  </el-form-item>
  <el-form-item>
    <el-button type="primary" @click="submitForm('ruleForm')">提交</el-button>
    <el-button @click="resetForm('ruleForm')">重置</el-button>
  </el-form-item>

```

```
</el-form>
<script>
  export default {
    data() {
      var checkAge = (rule, value, callback) => {
        if (!value) {
          return callback(new Error('年龄不能为空'));
        }
        setTimeout(() => {
          if (!Number.isInteger(value)) {
            callback(new Error('请输入数字值'));
          } else {
            if (value < 18) {
              callback(new Error('必须年满18岁'));
            } else {
              callback();
            }
          }
        }, 1000);
      };
      var validatePass = (rule, value, callback) => {
        if (value === '') {
          callback(new Error('请输入密码'));
        } else {
          if (this.ruleForm.checkPass !== '') {
            this.$refs.ruleForm.validateField('checkPass');
          }
          callback();
        }
      };
      var validatePass2 = (rule, value, callback) => {
        if (value === '') {
          callback(new Error('请再次输入密码'));
        } else if (value !== this.ruleForm.pass) {
          callback(new Error('两次输入密码不一致!'));
        } else {
          callback();
        }
      };
      return {
        ruleForm: {
          pass: '',
          checkPass: '',
          age: ''
        },
        rules: {
          pass: [
            { validator: validatePass, trigger: 'blur' }
          ],
          checkPass: [
            { validator: validatePass2, trigger: 'blur' }
          ],

```

```

        age: [
          { validator: checkAge, trigger: 'blur' }
        ]
      },
    },
    methods: {
      submitForm(formName) {
        this.$refs[formName].validate((valid) => {
          if (valid) {
            alert('submit!');
          } else {
            console.log('error submit!!');
            return false;
          }
        });
      },
      resetForm(formName) {
        this.$refs[formName].resetFields();
      }
    }
  }
</script>

```

...

::tip 自定义校验 callback 必须被调用。更多高级用法可参考 [async-validator](#)。::

动态增减表单项

::demo 除了在 Form 组件上一次性传递所有的验证规则外还可以在单个的表单域上传递属性的验证规则

```

<el-form :model="dynamicValidateForm" ref="dynamicValidateForm" label-width="100px"
class="demo-dynamic">
  <el-form-item
    prop="email"
    label="邮箱"
    :rules="[
      { required: true, message: '请输入邮箱地址', trigger: 'blur' },
      { type: 'email', message: '请输入正确的邮箱地址', trigger: ['blur', 'change'] }
    ]"
  >
    <el-input v-model="dynamicValidateForm.email"></el-input>
  </el-form-item>
  <el-form-item
    v-for="(domain, index) in dynamicValidateForm.domains"
    :label="'域名' + index"
    :key="domain.key"
    :prop="'domains.' + index + '.value'"
    :rules="{
      required: true, message: '域名不能为空', trigger: 'blur'
    }"
  >

```



```

    >
    <el-input v-model="domain.value"></el-input><el-button
@click.prevent="removeDomain(domain)">删除</el-button>
  </el-form-item>
  <el-form-item>
    <el-button type="primary" @click="submitForm('dynamicValidateForm')">提交</el-
button>
    <el-button @click="addDomain">新增域名</el-button>
    <el-button @click="resetForm('dynamicValidateForm')">重置</el-button>
  </el-form-item>
</el-form>
<script>
  export default {
    data() {
      return {
        dynamicValidateForm: {
          domains: [{
            value: ''
          }],
          email: ''
        }
      };
    },
    methods: {
      submitForm(formName) {
        this.$refs[formName].validate((valid) => {
          if (valid) {
            alert('submit!');
          } else {
            console.log('error submit!!!');
            return false;
          }
        });
      },
      resetForm(formName) {
        this.$refs[formName].resetFields();
      },
      removeDomain(item) {
        var index = this.dynamicValidateForm.domains.indexOf(item)
        if (index !== -1) {
          this.dynamicValidateForm.domains.splice(index, 1)
        }
      },
      addDomain() {
        this.dynamicValidateForm.domains.push({
          value: '',
          key: Date.now()
        });
      }
    }
  }
</script>

```

...

数字类型验证

demo 数字类型的验证需要在 `v-model` 处加上 `.number` 的修饰符，这是 Vue 自身提供的用于将绑定值转化为 `number` 类型的修饰符。

```
<el-form :model="numberValidateForm" ref="numberValidateForm" label-width="100px"
class="demo-ruleForm">
  <el-form-item
    label="年龄"
    prop="age"
    :rules="[
      { required: true, message: '年龄不能为空'},
      { type: 'number', message: '年龄必须为数字值'}
    ]"
  >
    <el-input type="age" v-model.number="numberValidateForm.age" autocomplete="off">
  </el-input>
</el-form-item>
<el-form-item>
  <el-button type="primary" @click="submitForm('numberValidateForm')">提交</el-
button>
  <el-button @click="resetForm('numberValidateForm')">重置</el-button>
</el-form-item>
</el-form>
<script>
export default {
  data() {
    return {
      numberValidateForm: {
        age: ''
      }
    };
  },
  methods: {
    submitForm(formName) {
      this.$refs[formName].validate((valid) => {
        if (valid) {
          alert('submit!');
        } else {
          console.log('error submit!!');
          return false;
        }
      });
    },
    resetForm(formName) {
      this.$refs[formName].resetFields();
    }
  }
}
```

```
}  
</script>
```

...

:::tip 嵌套在 `el-form-item` 中的 `el-form-item` 标签宽度默认为零，不会继承 `el-form` 的 `label-width`。如果需要可以为其单独设置 `label-width` 属性。 ...

表单内组件尺寸控制

通过设置 Form 上的 `size` 属性可以使该表单内所有可调节大小的组件继承该尺寸。Form-Item 也具有该属性。

:::demo 如果希望某个表单项或某个表单组件的尺寸不同于 Form 上的 `size` 属性，直接为这个表单项或表单组件设置自己的 `size` 即可。

```
<el-form ref="form" :model="sizeForm" label-width="80px" size="mini">  
  <el-form-item label="活动名称">  
    <el-input v-model="sizeForm.name"></el-input>  
  </el-form-item>  
  <el-form-item label="活动区域">  
    <el-select v-model="sizeForm.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-item label="活动时间">  
    <el-col :span="11">  
      <el-date-picker type="date" placeholder="选择日期" v-model="sizeForm.date1"  
style="width: 100%;"></el-date-picker>  
    </el-col>  
    <el-col class="line" :span="2"></el-col>  
    <el-col :span="11">  
      <el-time-picker placeholder="选择时间" v-model="sizeForm.date2" style="width:  
100%;"></el-time-picker>  
    </el-col>  
  </el-form-item>  
  <el-form-item label="活动性质">  
    <el-checkbox-group v-model="sizeForm.type">  
      <el-checkbox-button label="美食/餐厅线上活动" name="type"></el-checkbox-button>  
      <el-checkbox-button label="地推活动" name="type"></el-checkbox-button>  
      <el-checkbox-button label="线下主题活动" name="type"></el-checkbox-button>  
    </el-checkbox-group>  
  </el-form-item>  
  <el-form-item label="特殊资源">  
    <el-radio-group v-model="sizeForm.resource" size="medium">  
      <el-radio border label="线上品牌商赞助"></el-radio>  
      <el-radio border label="线下场地免费"></el-radio>  
    </el-radio-group>  
  </el-form-item>  
  <el-form-item size="large">  
    <el-button type="primary" @click="onSubmit">立即创建</el-button>  
    <el-button>取消</el-button>
```

```
</el-form-item>
</el-form>

<script>
  export default {
    data() {
      return {
        sizeForm: {
          name: '',
          region: '',
          date1: '',
          date2: '',
          delivery: false,
          type: [],
          resource: '',
          desc: ''
        }
      };
    },
    methods: {
      onSubmit() {
        console.log('submit!');
      }
    }
  };
</script>
```

⋮

Form Attributes

参数	说明	类型	可选值	默认值
model	表单数据对象	object	—	—
rules	表单验证规则	object	—	—
inline	行内表单模式	boolean	—	false
label-position	表单域标签的位置，如果值为 left 或者 right 时，则需要设置 label-width	string	right/left/top	right
label-width	表单域标签的宽度，例如 '50px'。作为 Form 直接子元素的 form-item 会继承该值。支持 auto。	string	—	—
label-suffix	表单域标签的后缀	string	—	—
hide-required-asterisk	是否隐藏必填字段的标签旁边的红色星号	boolean	—	false
show-message	是否显示校验错误信息	boolean	—	true

inline-message	是否以行内形式展示校验信息	boolean	—	false
status-icon	是否在输入框中显示校验结果反馈图标	boolean	—	false
validate-on-rule-change	是否在 <code>rules</code> 属性改变后立即触发一次验证	boolean	—	true
size	用于控制该表单内组件的尺寸	string	medium / small / mini	—
disabled	是否禁用该表单内的所有组件。若设置为 true，则表单内组件上的 <code>disabled</code> 属性不再生效	boolean	—	false

Form Methods

方法名	说明	参数
validate	对整个表单进行校验的方法，参数为一个回调函数。该回调函数会在校验结束后被调用，并传入两个参数：是否校验成功和未通过校验的字段。若不传入回调函数，则会返回一个 promise	Function(callback: Function(boolean, object))
validateField	对部分表单字段进行校验的方法	Function(props: array string, callback: Function(errorMessage: string))
resetFields	对整个表单进行重置，将所有字段值重置为初始值并移除校验结果	—
clearValidate	移除表单项的校验结果。传入待移除的表单项的 prop 属性或者 prop 组成的数组，如不传则移除整个表单的校验结果	Function(props: array string)

Form Events

事件名称	说明	回调参数
validate	任一表单项被校验后触发	被校验的表单项 prop 值，校验是否通过，错误消息（如果存在）

Form-Item Attributes

参数	说明	类型	可选值	默认值
prop	表单域 model 字段，在使用 validate、resetFields 方法的情况下，该属性是必填的	string	传入 Form 组件的 model 中的字段	—
label	标签文本	string	—	—
label-width	表单域标签的宽度，例如 '50px'。支持 auto。	string	—	—
required	是否必填，如不设置，则会根据校验规则自动生成	boolean	—	false
rules	表单验证规则	object	—	—
error	表单域验证错误信息，设置该值会使表单验证状态变	string	—	—

	为 <code>error</code> ，并显示该错误信息			
show-message	是否显示校验错误信息	boolean	—	true
inline-message	以行内形式展示校验信息	boolean	—	false
size	用于控制该表单域下组件的尺寸	string	medium / small / mini	-

Form-Item Slot

name	说明
—	Form Item 的内容
label	标签文本的内容

Form-Item Scoped Slot

name	说明
error	自定义表单校验信息的显示方式，参数为 { error }

Form-Item Methods

方法名	说明	参数
resetField	对该表单项进行重置，将其值重置为初始值并移除校验结果	-
clearValidate	移除该表单项的校验结果	-