# 1.Element Plus组件库P186

## 1.什么是UI组件库

|  |
| --- |
|  |

## 2.element-plus库

|  |
| --- |
|  |
|  |
| 3 |

# 2.Element Plus按钮组件p187

## 关于组件的使用,可以多看规范文档

### <https://element-plus.org/en-US/component/button>

# 3. Element Plus基础表单1-p188

|  |
| --- |
|  |

# 4. Element Plus基础表单2-p189

## 还是上面的例子,不过我们把它给我我们需要的样子.

|  |
| --- |
|  |

# 5. Element Plus行内表单-p190

## 其实就是把所有的控件都放在一行的表单,只需要修改一个inline属性为true,其他和基础表单是一样的.

|  |
| --- |
|  |

# 6. Element Plus表单校验-p191

|  |
| --- |
|  |

## 这里我们会学习到元素里面的ref属性,其实这是vue3用来代替原生js的id属性的.

## 有一个rules属性是指表单的验证规则

## 还有一个prop属性,用来把规则绑定到指定的元素上面

## 必填的元素上面还需要家required关键字

|  |
| --- |
|  |

# 7. Element Plus-Dialog对话框-p192

# 8. Element Plus信息提示框-p193

# 演练1,element-plus的基本使用

## 1>用前面的组合式api模板新建一个项目,然后输入npm install element-plus –save安装element-plus库,然后把项目把node\_modules文件夹删除,那么它就是一个项目模板.我们用这个模板创建一个项目: lesson23-elementplus\_basic,然后进入项目,安装依赖,然后输入npm run dev把它运行起来.

|  |
| --- |
|  |

## 2>我们需要在main.js里面配置elementPlus

|  |
| --- |
|  |

## 3>然后我们就可以使用这些组件了,其实可以到element-plus官网里面复制,不过有些组件需要导入有些资源,你必须把import语句也复制到script标签中.下面我们使用有些比较简单的组件

|  |
| --- |
|  |

### 刷新网页,效果如下

|  |
| --- |
|  |

## 4>我们可以去官网复制有些带有icon的按钮,注意需要导入这些icon

|  |
| --- |
|  |
|  |

### 效果:

|  |
| --- |
|  |

# 演练2-element-plus表单案例

## 1.用模板新建一个项目: lesson24-elementplus\_form1,安装相关依赖(包括element-plus),然后把项目跑起来.

|  |
| --- |
|  |

## 2.在main.js里面使用element-plus,这一步非常重要,否则程序不能正常运行

|  |
| --- |
|  |

## 3>去Element Plus官网把一个form表单复制过来,粘贴到App.vue里面,代码如下

|  |
| --- |
| <**template**>  <**el-form :model="form" label-width="auto" style="max-width**: 600**px"**>  <**el-form-item label="Activity name"**>  <**el-input v-model="form**.**name"** />  </**el-form-item**>  <**el-form-item label="Activity zone"**>  <**el-select v-model="form**.**region" placeholder="please select your zone"**>  <**el-option label="Zone one" value="shanghai"** />  <**el-option label="Zone two" value="beijing"** />  </**el-select**>  </**el-form-item**>  <**el-form-item label="Activity time"**>  <**el-col :span="**11**"**>  <**el-date-picker  v-model="form**.**date1"  type="date"  placeholder="Pick a date"  style="width**: 100%**"** />  </**el-col**>  <**el-col :span="**2**" class="text-center"**>  <**span class="text-gray-500"**>-</**span**>  </**el-col**>  <**el-col :span="**11**"**>  <**el-time-picker  v-model="form**.**date2"  placeholder="Pick a time"  style="width**: 100%**"** />  </**el-col**>  </**el-form-item**>  <**el-form-item label="Instant delivery"**>  <**el-switch v-model="form**.**delivery"** />  </**el-form-item**>  <**el-form-item label="Activity type"**>  <**el-checkbox-group v-model="form**.**type"**>  <**el-checkbox value="Online activities" name="type"**>  Online activities  </**el-checkbox**>  <**el-checkbox value="Promotion activities" name="type"**>  Promotion activities  </**el-checkbox**>  <**el-checkbox value="Offline activities" name="type"**>  Offline activities  </**el-checkbox**>  <**el-checkbox value="Simple brand exposure" name="type"**>  Simple brand exposure  </**el-checkbox**>  </**el-checkbox-group**>  </**el-form-item**>  <**el-form-item label="Resources"**>  <**el-radio-group v-model="form**.**resource"**>  <**el-radio value="Sponsor"**>Sponsor</**el-radio**>  <**el-radio value="Venue"**>Venue</**el-radio**>  </**el-radio-group**>  </**el-form-item**>  <**el-form-item label="Activity form"**>  <**el-input v-model="form**.**desc" type="textarea"** />  </**el-form-item**>  <**el-form-item**>  <**el-button type="primary" @click="onSubmit"**>Create</**el-button**>  <**el-button**>Cancel</**el-button**>  </**el-form-item**>  </**el-form**> </**template**>  <**script setup**> **import** { *reactive* } **from 'vue'** *// do not use same name with ref* **const *form*** = *reactive*({  **name**: **''**,  **region**: **''**,  **date1**: **''**,  **date2**: **''**,  **delivery**: **false**,  **type**: [],  **resource**: **''**,  **desc**: **''**, })  **const** *onSubmit* = () => {  ***console***.log(**'submit!'**) }  </**script**>  <**style scoped**>  </**style**> |

### 刷新页面,效果如下

|  |
| --- |
|  |

## 4>我们可以给el-form添加一个类名,然后给他添加样式,比如这里我们把它居中显示,并且显示边框

|  |
| --- |
|  |
|  |

### 效果如下

|  |
| --- |
|  |

## 5>其实我们可以把这个表单改为用户注册表单,当然需要修改一些属性和添加一些控件修改后的代码如下

|  |
| --- |
| <**template**>  <**h1 style="text-align**: **center"**>用户注册</**h1**>  <**el-form :model="form" label-width="auto" style="max-width**: 600**px**;**" class="myForm"**>  <**el-form-item label="用户名"**>  <**el-input v-model="form**.**userName"** />  </**el-form-item**>  <**el-form-item label="密 码"**>  <**el-input type="password" v-model="form**.**passWord"** />  </**el-form-item**>  <**el-form-item label="学历"**> *<!-- <el-select v-model="form.edu" placeholder="请选择你的学历">--> <!-- <el-option label="本科以上" value="本科以上" />--> <!-- <el-option label="大专" value="大专" />--> <!-- <el-option label="高中" value="高中" />--> <!-- <el-option label="中专" value="中专" />--> <!-- <el-option label="初中" value="初中" />--> <!-- </el-select>-->* <**el-select v-model="form**.**edu" placeholder="请选择你的学历"**>  *<!-- 可以使用v-for来遍历我们的数组来生成option value需要动态绑定-->* <**el-option v-for="**edu **in edus" :value="**edu**"** />  </**el-select**>  </**el-form-item**>  <**el-form-item label="出生日期"**>  <**el-date-picker  v-model="form**.**birthday"  type="date"  placeholder="选择出生日期"  style="width**: 100%**"** />  </**el-form-item**>   <**el-form-item label="兴趣爱好"**>  <**el-checkbox-group v-model="form**.**hobbies"**>  <**el-checkbox value="爬山" name="hobbies"**>  爬山  </**el-checkbox**>  <**el-checkbox value="游泳" name="hobbies"**>  游泳  </**el-checkbox**>  <**el-checkbox value="看电影" name="hobbies"**>  看电影  </**el-checkbox**>  <**el-checkbox value="泡妞" name="hobbies"**>  泡妞  </**el-checkbox**>  </**el-checkbox-group**>  </**el-form-item**>  <**el-form-item label="性别"**>  <**el-radio-group v-model="form**.**gender"**>  <**el-radio value="男"**>男</**el-radio**>  <**el-radio value="女"**>女</**el-radio**>  </**el-radio-group**>  </**el-form-item**>  <**el-form-item label="个人简介"**>  <**el-input v-model="form**.**desc" type="textarea"** />  </**el-form-item**>  <**el-form-item label="记住密码"**>  <**el-switch v-model="form**.**remember"** />  </**el-form-item**>  <**el-form-item**>  <**el-button type="primary" @click="onSubmit"**>注册</**el-button**>  <**el-button**>取消</**el-button**>  </**el-form-item**>  </**el-form**>  {{**form**}} </**template**>  <**script setup**> **import** { *reactive* } **from 'vue'** *// do not use same name with ref* **const *form*** = *reactive*({  **userName**: **''**,  **passWord**:**""**,  **edu**: **''**,  **birthday**: **''**,  **remember**: **false**,  **hobbies**: [],  **gender**: **''**,  **desc**: **''**, })  **let *edus*** = *reactive*([**"初中"**,**"高中"**,**"大专"**,**"本科以上"**])  **const** *onSubmit* = () => {  ***console***.log(**'submit!'**,***form***) }  </**script**>  <**style scoped**>  .**myForm**{  **border**: 2**px dashed deeppink**;  **padding**:20**px**;  **margin**: 10**px auto**;  } </**style**> |

### 效果如下

|  |
| --- |
|  |

### 点击注册,会在控制台输出信息.

|  |
| --- |
|  |

# 演练3.-element-plus行内表单案例

## 1>用elementplus案例模板新建一个项目:lesson25-element-plus-inline-form,安装依赖,然后跑起来

|  |
| --- |
|  |

## 2>然后到element plus官网复制一个行内表单,粘贴到App.vue中

|  |
| --- |
| <**template**>  <**el-form :inline="true" :model="formInline" class="demo-form-inline"**>  <**el-form-item label="Approved by"**>  <**el-input v-model="formInline**.**user" placeholder="Approved by" clearable** />  </**el-form-item**>  <**el-form-item label="Activity zone"**>  <**el-select  v-model="formInline**.**region"  placeholder="Activity zone"  clearable** >  <**el-option label="Zone one" value="shanghai"** />  <**el-option label="Zone two" value="beijing"** />  </**el-select**>  </**el-form-item**>  <**el-form-item label="Activity time"**>  <**el-date-picker  v-model="formInline**.**date"  type="date"  placeholder="Pick a date"  clearable** />  </**el-form-item**>  <**el-form-item**>  <**el-button type="primary" @click="onSubmit"**>Query</**el-button**>  </**el-form-item**>  </**el-form**> </**template**>  <**script setup**> **import** { *reactive* } **from 'vue'  const *formInline*** = *reactive*({  **user**: **''**,  **region**: **''**,  **date**: **''**, })  **const** *onSubmit* = () => {  ***console***.log(**'submit!'**) } </**script**>  <**style**> .**demo-form-inline** .**el-input** {  **--el-input-width**: 220**px**; }  .**demo-form-inline** .**el-select** {  **--el-select-width**: 220**px**; } </**style**> |

### 刷新页面,效果如下

|  |
| --- |
|  |

## 3>我们也可以把它改为我们需要的样子

|  |
| --- |
| <**template**>  <**el-form :inline="true" :model="formInline" class="demo-form-inline"**>  <**el-form-item label="用户名称"**>  <**el-input v-model="formInline**.**userName" placeholder="用户名称" clearable** />  </**el-form-item**>  <**el-form-item label="手机号码"**>  <**el-input v-model="formInline**.**phone" placeholder="手机号码" clearable** />  </**el-form-item**>  <**el-form-item label="用户状态"**>  <**el-select  v-model="formInline**.**status"  placeholder="用户状态"  clearable** >  <**el-option value="正常"** />  <**el-option value="禁用"** />  </**el-select**>  </**el-form-item**>  <**el-form-item label="创建日期"**>  <**el-date-picker  v-model="formInline**.**date"  type="daterange"  range-separator="To"  placeholder="Pick a date"  start-placeholder="开始日期"  end-placeholder="结束日期"  clearable** />  </**el-form-item**>  <**el-form-item**>  <**el-button type="primary" @click="onSubmit"**>Query</**el-button**>  </**el-form-item**>  </**el-form**>  {{**formInline**}} </**template**>  <**script setup**> **import** { *reactive* } **from 'vue'  const *formInline*** = *reactive*({  **userName**: **''**,  **phone**: **''**,  **status**:**""**,  **date**: **''**, })  **const** *onSubmit* = () => {  ***console***.log(**'submit!'**) } </**script**>  <**style**> .**el-form-item label**{  **font-weight**: **bold**; } .**demo-form-inline** .**el-input** {  **--el-input-width**: 220**px**; }  .**demo-form-inline** .**el-select** {  **--el-select-width**: 220**px**; } </**style**> |

### 效果如下

|  |
| --- |
|  |

# 演练4.element plus表单校验案例

## 1>用element plus项目模板创建一个项目,然后进入项目安装依赖,在main.js中引入ElementPlus和导入index.css, 然后就可以把项目先运行起来了

|  |
| --- |
|  |
|  |

## 2>然后我们到element plus官网复制一个有验证功能的表单粘贴到App.vue里面,默认代码是这样子的

|  |
| --- |
| <template>  <el-form  ref="ruleFormRef"  style="max-width: 600px"  :model="ruleForm"  :rules="rules"  label-width="auto"  >  <el-form-item label="Activity name" prop="name">  <el-input v-model="ruleForm.name" />  </el-form-item>  <el-form-item label="Activity zone" prop="region">  <el-select v-model="ruleForm.region" placeholder="Activity zone">  <el-option label="Zone one" value="shanghai" />  <el-option label="Zone two" value="beijing" />  </el-select>  </el-form-item>  <el-form-item label="Activity count" prop="count">  <el-select-v2  v-model="ruleForm.count"  placeholder="Activity count"  :options="options"  />  </el-form-item>  <el-form-item label="Activity time" required>  <el-col :span="11">  <el-form-item prop="date1">  <el-date-picker  v-model="ruleForm.date1"  type="date"  aria-label="Pick a date"  placeholder="Pick a date"  style="width: 100%"  />  </el-form-item>  </el-col>  <el-col class="text-center" :span="2">  <span class="text-gray-500">-</span>  </el-col>  <el-col :span="11">  <el-form-item prop="date2">  <el-time-picker  v-model="ruleForm.date2"  aria-label="Pick a time"  placeholder="Pick a time"  style="width: 100%"  />  </el-form-item>  </el-col>  </el-form-item>  <el-form-item label="Instant delivery" prop="delivery">  <el-switch v-model="ruleForm.delivery" />  </el-form-item>  <el-form-item label="Activity location" prop="location">  <el-segmented v-model="ruleForm.location" :options="locationOptions" />  </el-form-item>  <el-form-item label="Activity type" prop="type">  <el-checkbox-group v-model="ruleForm.type">  <el-checkbox value="Online activities" name="type">  Online activities  </el-checkbox>  <el-checkbox value="Promotion activities" name="type">  Promotion activities  </el-checkbox>  <el-checkbox value="Offline activities" name="type">  Offline activities  </el-checkbox>  <el-checkbox value="Simple brand exposure" name="type">  Simple brand exposure  </el-checkbox>  </el-checkbox-group>  </el-form-item>  <el-form-item label="Resources" prop="resource">  <el-radio-group v-model="ruleForm.resource">  <el-radio value="Sponsorship">Sponsorship</el-radio>  <el-radio value="Venue">Venue</el-radio>  </el-radio-group>  </el-form-item>  <el-form-item label="Activity form" prop="desc">  <el-input v-model="ruleForm.desc" type="textarea" />  </el-form-item>  <el-form-item>  <el-button type="primary" @click="submitForm(ruleFormRef)">  Create  </el-button>  <el-button @click="resetForm(ruleFormRef)">Reset</el-button>  </el-form-item>  </el-form>  </template>  <script lang="ts" setup>  import { reactive, ref } from 'vue'  import type { FormInstance, FormRules } from 'element-plus'  interface RuleForm {  name: string  region: string  count: string  date1: string  date2: string  delivery: boolean  location: string  type: string[]  resource: string  desc: string  }  const ruleFormRef = ref<FormInstance>()  const ruleForm = reactive<RuleForm>({  name: 'Hello',  region: '',  count: '',  date1: '',  date2: '',  delivery: false,  location: '',  type: [],  resource: '',  desc: '',  })  const locationOptions = ['Home', 'Company', 'School']  const rules = reactive<FormRules<RuleForm>>({  name: [  { required: true, message: 'Please input Activity name', trigger: 'blur' },  { min: 3, max: 5, message: 'Length should be 3 to 5', trigger: 'blur' },  ],  region: [  {  required: true,  message: 'Please select Activity zone',  trigger: 'change',  },  ],  count: [  {  required: true,  message: 'Please select Activity count',  trigger: 'change',  },  ],  date1: [  {  type: 'date',  required: true,  message: 'Please pick a date',  trigger: 'change',  },  ],  date2: [  {  type: 'date',  required: true,  message: 'Please pick a time',  trigger: 'change',  },  ],  location: [  {  required: true,  message: 'Please select a location',  trigger: 'change',  },  ],  type: [  {  type: 'array',  required: true,  message: 'Please select at least one activity type',  trigger: 'change',  },  ],  resource: [  {  required: true,  message: 'Please select activity resource',  trigger: 'change',  },  ],  desc: [  { required: true, message: 'Please input activity form', trigger: 'blur' },  ],  })  const submitForm = async (formEl: FormInstance | undefined) => {  if (!formEl) return  await formEl.validate((valid, fields) => {  if (valid) {  console.log('submit!')  } else {  console.log('error submit!', fields)  }  })  }  const resetForm = (formEl: FormInstance | undefined) => {  if (!formEl) return  formEl.resetFields()  }  const options = Array.from({ length: 10000 }).map((\_, idx) => ({  value: `${idx + 1}`,  label: `${idx + 1}`,  }))  </script> |

### 默认效果如下

|  |
| --- |
|  |

## 3>我们需要把它改造为一个有验证功能的注册表单,修改后的代码如下

|  |
| --- |
| <**template**>  <**h1 class="title"**>用户注册</**h1**>  <**el-form  ref="ruleFormRef"  style="max-width**: 550**px"  :model="ruleForm"  :rules="rules"  label-width="auto"  class="regForm"** >  <**el-form-item label="用户名" prop="userName"**>  <**el-input v-model="ruleForm**.**userName"** />  </**el-form-item**>  <**el-form-item label="密 码" prop="password"**>  <**el-input v-model="ruleForm**.**password" type="password"**/>  </**el-form-item**>  <**el-form-item label="学历" prop="edu"**>  <**el-select v-model="ruleForm**.**edu" placeholder="选择学历"**>  <**el-option v-for="**edu **in edus" :value="**edu**"** />  </**el-select**>  </**el-form-item**>  <**el-form-item label="出生日期" required**>  <**el-form-item prop="birthday"**>  <**el-date-picker  v-model="ruleForm**.**birthday"  type="date"  aria-label="选择出生日期"  placeholder="选择出生日期"  style="width**: 100%**"** />  </**el-form-item**>  </**el-form-item**>   <**el-form-item label="兴趣爱好" prop="hobbies"**>  <**el-checkbox-group v-model="ruleForm**.**hobbies"**>  <**el-checkbox v-for="**h **in hobbies" :value="**h**" name="type"**>  {{h}}  </**el-checkbox**>  </**el-checkbox-group**>  </**el-form-item**>  <**el-form-item label="性别" prop="gender"**>  <**el-radio-group v-model="ruleForm**.**gender"**>  <**el-radio v-for="**g **in genders" :value="**g**"**>{{g}}</**el-radio**>  </**el-radio-group**>  </**el-form-item**>  <**el-form-item label="个人简介" prop="desc"**>  <**el-input v-model="ruleForm**.**desc" type="textarea"** />  </**el-form-item**>  <**el-form-item label="记住密码" prop="remember"**>  <**el-switch v-model="ruleForm**.**remember"** />  </**el-form-item**>  <**el-form-item**>  <**el-button type="primary" @click="submitForm**(**ruleFormRef**)**"**>  注册  </**el-button**>  <**el-button @click="resetForm**(**ruleFormRef**)**"**>重置</**el-button**>  </**el-form-item**>  </**el-form**> </**template**>  <**script lang="ts" setup**> **import** { *reactive*, *ref* } **from 'vue'  import type** { FormInstance, FormRules } **from 'element-plus'  interface** RuleForm {  **userName**: **string  password**:**string  edu**: **string  birthday**: **string  remember**: **boolean  hobbies**: **string**[]  **gender**: **string  desc**: **string** }  *//响应式学历数组* **let *edus*** = *reactive*([**"初中"**,**"高中"**,**"大专"**,**"本科以上"**]) *//兴趣爱好数组* **let *hobbies*** = *reactive*([**"篮球"**,**"足球"**,**'羽毛球'**,**'乒乓球'**,**'游泳'**,**"看书"**,**'爬山'**,**"打游戏"**,**"学编程"**,**"聊天"**]) *//性别数组* **let *genders*** = *reactive*([**'男'**,**"女"**,**"保密"**])  **const *ruleFormRef*** = *ref*<FormInstance>() **const *ruleForm*** = *reactive*<RuleForm>({  **userName**: **'Hello'**,  **password**:**''**,  **edu**: **''**,  **birthday**: **''**,  **remember**: **false**,  **hobbies**: [],  **gender**: **''**,  **desc**: **''**, })  **const *locationOptions*** = [**'Home'**, **'Company'**, **'School'**]  **const *rules*** = *reactive*<FormRules<RuleForm>>({  **userName**: [  { **required**: **true**, **message**: **'姓名不能为空'**, **trigger**: **'blur'** },  { **min**: 2, **max**: 5, **message**: **'姓名必须是2-5位'**, **trigger**: **'blur'** },  ],  **password**: [  { **required**: **true**, **message**: **'密码不能为空'**, **trigger**: **'blur'** },  { **min**: 6, **max**: 12, **message**: **'密码必须是6-12位'**, **trigger**: **'blur'** },  ],  **edu**: [  {  **required**: **true**,  **message**: **'请选择你的学历'**,  **trigger**: **'change'**,  },  ],  **birthday**: [  {  **type**: **'date'**,  **required**: **true**,  **message**: **'请选择出生日期'**,  **trigger**: **'change'**,  },  ],   **hobbies**: [  {  **type**: **'array'**,  **required**: **true**,  **message**: **'请选择至少一个兴趣爱好'**,  **trigger**: **'change'**,  },  ],  **gender**: [  {  **required**: **true**,  **message**: **'请选择性别'**,  **trigger**: **'change'**,  },  ],  **desc**: [  { **required**: **true**, **message**: **'请填写个人简介'**, **trigger**: **'blur'** },  {**min**:10,**max**:200, **message**: **'个人简介的长度是10-200个字符'**, **trigger**: **'blur'**}  ], })  **const** *submitForm* = **async** (formEl: FormInstance | **undefined**) => {  **if** (!formEl) **return  await** formEl.**validate**((valid, fields) => {  **if** (valid) {  ***console***.log(**'submit!'**)  } **else** {  ***console***.log(**'error submit!'**, fields)  }  }) }  **const** *resetForm* = (formEl: FormInstance | **undefined**) => {  **if** (!formEl) **return** formEl.**resetFields**() }  *// const options = Array.from({ length: 10000 }).map((\_, idx) => ({ // value: `${idx + 1}`, // label: `${idx + 1}`, // }))* </**script**>  <**style scoped**>  .**regForm**{  **margin**: 10**px auto**;  }  .**title**{  **text-align**: **center**;  } </**style**> |

### 刷新页面,效果如下

|  |
| --- |
|  |

### 如果你没有填写用户名然后点击另外一个地方,就会显示需要填写用户名的提示

|  |
| --- |
|  |

### 如果填写的长度不符合要求,也会报错

|  |
| --- |
|  |

### 密码的效果也是一样的.

|  |
| --- |
|  |
|  |

### 如果我们没有填写所有内容就点击注册按钮,没有填写的地方都会报错,然后提交也会失败

|  |
| --- |
|  |

### 填写个人简介的时候也会有长度限制

|  |
| --- |
|  |

### 所有内容都正确,点击注册,在控制台输出注册,说明模拟注册成功

|  |
| --- |
|  |

## 至此,一个有简单验证功能的表单就做好了.我们在设计开发在一般只需要做简单验证,无需太复杂