实习任务安排：

1. 多条优先顺序的设置与显示
2. 前后端联调接口

任务完成情况：

1. computed计算属性中使用“=”赋值是无效的，需设置getter与setter
2. 对象的拷贝可使用JSON.parse(JSON.stringfy(object))，尽量避免使用“=”使其指向同一个引用（注意引用类型与值类型的区别，例如在for循环中的向数组中push对象

<https://blog.csdn.net/xiaoye319/article/details/78416762>）

数组中增加自定义属性时，在拷贝的时候会丢失，应该讲数组和id封装为对象

1. 表格中设置按钮模板列：作用域插槽slot-scope；

三元运算符可以嵌套；

按钮选中时🡨🡪v-model绑定的值与label值相等，按钮不提供显示的数据时显示label值；

El-table的行事件row-click中处理选中行时单选按钮同时被选中

1. Mixin混用是一种代码复用的方法，适用于不包括template的复用，也就是说可以混method,钩子函数等等，相当于在需要的地方将代码复制一份；而components组件的复用包含所有内容，包括template。

Best practice:在components中复用mixins

使用：a.导入:

import MsgBox from '@/mixins/msgBox'

b.声明：

mixins: [MsgBox]

c.使用：直接调用

混合对象与组件自身合并策略：

数据对象在内部会进行浅合并 (一层属性深度)，在和组件的数据发生冲突时以组件数据优先。

同名钩子函数将混合为一个数组，因此都将被调用。另外，混入对象的钩子将在组件自身钩子**之前**调用。

5、浏览态与编辑态：

使用：

a.路由传参

新建🡪

this.$router.push({name: 'LeaveTypeAdd', params: {ruleId: this.formData.id, typeId: 'new'}})

编辑🡪

this.$router.push({name: 'LeaveTypeAdd', params: {ruleId: row.mainId, typeId: row.id}})

b.在跳转后的页面create中获取数据并初始化

根据传的参数fetchData然后使用Object.assign复制对象

Object.keys(object).foreach(key => { })

6.前后端联调：

a.fetchData使用微服务，二级目录由后台提供,在局域网内连接服务器，配置mockProxy.js中的路径，重启服务

b. 项目中封装的fetchData的data参数为**对象**格式