# 1.什么是选项式API?

|  |
| --- |
|  |
|  |
|  |
|  |

## 其实我们前面使用的都是选项式API

# 2.什么是组合式API

|  |
| --- |
|  |
|  |

# 3.这两种api各自的特点如下

|  |
| --- |
|  |

# 4.该使用哪一种api风格?

|  |
| --- |
|  |

# 演练1.选项式API演练

# 1.既然我们前面学习的都是选项式api,这里我们就简单做一个学生信息表格作为演示即可,新建一个lesson13项目,既然目录安装依赖.然后在App.vue里面添加下面的代码

|  |
| --- |
| <**template**>  <**div class="box"**>  <**table**>  <**caption**>  <**h2**>学生信息列表</**h2**>  </**caption**>  <**thead**>  <**tr**>  <**th**>ID</**th**>  <**th**>姓名</**th**>  <**th**>年龄</**th**>  <**th**>性别</**th**>  <**th**>爱好</**th**>  </**tr**>  </**thead**>  <**tbody**>  <**tr v-for="**stu **in students" :key="**stu.**id"**>  <**td**>{{stu.**id**}}</**td**>  <**td**>{{stu.**name**}}</**td**>  <**td**>{{stu.**age**}}</**td**>  <**td**>{{stu.**gender**}}</**td**>  <**td**>{{stu.**hobbies**.join(',')}}</**td**>  </**tr**>  </**tbody**>  </**table**>  </**div**> </**template**>  <**script**>  **export default** {  data() {  **return** {  **students**: [  {  **id**:1001,  **name**:**"Jack"**,  **age**:18,  **gender**:**"male"**,  **hobbies**:[**"读书"**,**"爬山"**,**"跑步"**]  },  {  **id**:1002,  **name**:**"Mark"**,  **age**:18,  **gender**:**"male"**,  **hobbies**:[**"读书"**,**"爬山"**,**"跑步"**]  },  {  **id**:1003,  **name**:**"Linda"**,  **age**:18,  **gender**:**"female"**,  **hobbies**:[**"游泳"**,**"跳舞"**,**"跑步"**]  },  {  **id**:1004,  **name**:**"Tracy"**,  **age**:18,  **gender**:**"female"**,  **hobbies**:[**"读书"**,**"听音乐"**,**"打王者"**]  },  ]  }  },  }  </**script**>  <**style scoped**>  **table**{  **border**: 1**px solid #282828**;  **border-collapse**: **collapse**;  **width**: 800**px**;  }  **th**,**td**{  **border**: 1**px solid #f2f2f2**;  **padding**: 8**px**;  }   .**box**{  **width**: 820**px**;  **margin**: 80**px auto**;  } </**style**> |

## 运行程序,效果如下

|  |
| --- |
|  |

# 演练2,组合式API演练

## 1>用模板创建一个lesson14项目,进入目录安装依赖,然后用npm run dev把它运行起来

|  |
| --- |
|  |

## 2>然后我们先在script标签里面定义数据,注意,组合式api的script开标签后面需要跟setup否则不工作.然后以前选项是api的属性统统改为函数,需要从vue库里面导入,响应式数据需要使用ref函数来定义,否则没有响应相关.定义数据和函数的代码如下

|  |
| --- |
|  |

### 需要注意的是,在js代码中操作响应式数据的值需要使用他的value属性,但是在页面的插值和模型绑定是可以之间使用响应式变量的.

## 3>然后在html模板里面的写法是和选项式项目的写法一模一样

|  |
| --- |
|  |

### 刷新页面,效果如下

|  |
| --- |
|  |

### 然后我们来做一点操作

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

## 4>然后我们来调用一下那个生命周期函数,代码如下,注意这个函数在页面加载完成(也就是vue组件挂载完成)就好被调用.

|  |
| --- |
|  |

### 效果:

|  |
| --- |
|  |
|  |

### 注意:这个是一个钩子函数.它会先执行,可以看见此时页面说明东西都没有