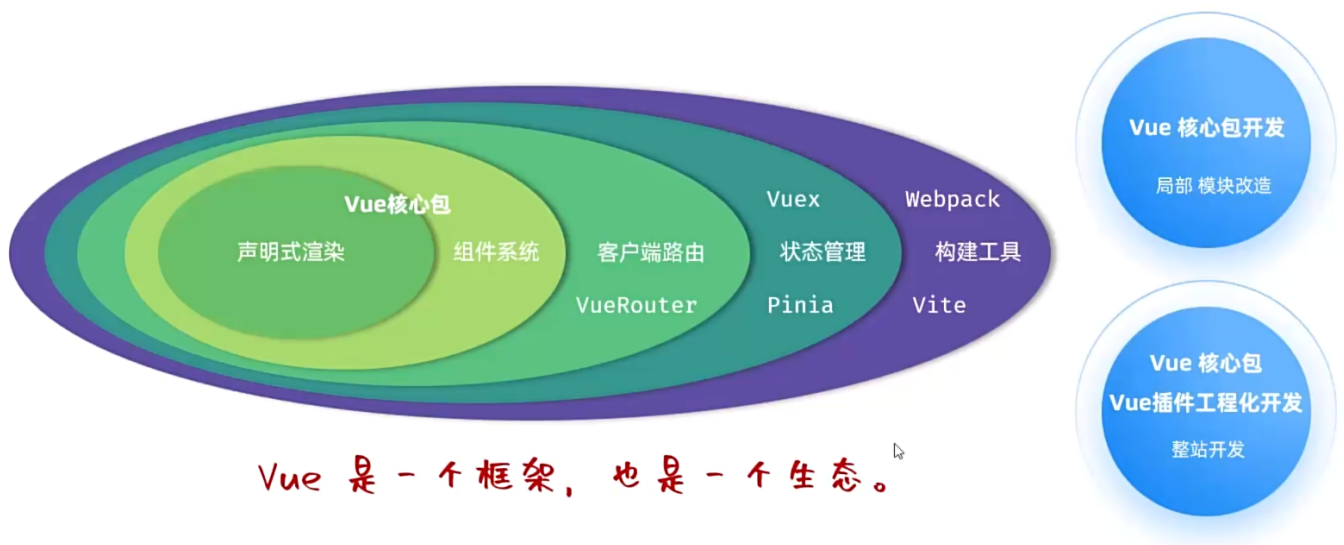


03 Vue3

• 1 概念

- Vue是一款用于**构建用户界面**的**渐进式**的JavaScript**框架**。（官方：<https://cn.vuejs.org/>）
- 框架：就是一套完整的项目解决方案，用于快速构建项目。
- 优点：大大**提升**前端项目的**开发效率**。
- 缺点：需要理解记忆框架的使用规则。（参照官网）



• 2 Vue 快速入门

- 准备
 - 引入Vue模块（官方提供）
 - 创建Vue程序的应用实例，控制视图的元素
 - 准备元素(div)，被Vue控制
- 数据驱动视图
 - 准备数据
 - 通过插值表达式渲染页面



• 3 Vue 常用指令

- 指令：HTML标签上带有 `v-`前缀的特殊属性，不同的指令具有不同的含义，可以实现不同的功能。
- 常用指令：



指令	作用
<code>v-for</code>	列表渲染，遍历容器的元素或者对象的属性
<code>v-bind</code>	为HTML标签绑定属性值，如设置 <code>href</code> ， <code>css</code> 样式等
<code>v-if/v-else-if/v-else</code>	条件性的渲染某元素，判定为 <code>true</code> 时渲染，否则不渲染
<code>v-show</code>	根据条件展示某元素，区别在于切换的是 <code>display</code> 属性的值
<code>v-model</code>	在表单元素上创建双向数据绑定
<code>v-on</code>	为HTML标签绑定事件

• 1. v-for

- 作用：列表渲染，遍历容器的元素或者对象的属性
- 语法： `<tr v-for="(item,index) in items" :key="item.id"> {{item}}</tr>`
- 参数说明：
 - items 为遍历的数组
 - item 为遍历出来的元素
 - index 为索引/下标，从0开始；可以省略，省略index语法： `v-for="item in items"`
- key：
 - 作用：给元素添加的唯一标识，便于vue进行列表项的正确排序复用，提升渲染性能
 - 推荐使用id作为key（唯一），不推荐使用index作为key（会变化，不对应）

注意： 遍历的数组，必须在data中定义； 要想让哪个标签循环展示多次，就在哪个标签上使用 v-for 指令。

● 2. v-bind

- 作用：动态为HTML标签绑定属性值，如设置href, src, style样式等。
- 语法： `v-bind:属性名="属性值"` ``
- 简化： `:属性名="属性值"` ``

注意： 动态的为标签的属性绑定值，不能使用插值表达式，得使用 v-bind 指令。且绑定的数据，必须在data中定义。

- 插值表达式是两个大括号 {{ }}，标签就是尖括号 <>，例如，其中 src 是标签的属性

● 3. v-if & v-show

- 作用：这两类指令，都是用来控制元素的显示与隐藏的
- v-if
 - 语法： `v-if="表达式"`，表达式值为 true，显示； false，隐藏
 - 原理：基于条件判断，来控制创建或移除元素节点（条件渲染）
 - 场景：要么显示，要么不显示，不频繁切换的场景
 - 其它：可以配合 v-else-if / v-else 进行链式调用条件判断
- v-show
 - 语法： `v-show="表达式"`，表达式值为 true，显示； false，隐藏
 - 原理：基于CSS样式display来控制显示与隐藏
 - 场景：频繁切换显示隐藏的场景

注意： v-else-if必须出现在v-if之后，可以出现多个； v-else 必须出现在v-if/v-else-if之后。

● 4. v-model

- 作用：在表元素上使用，双向数据绑定。可以方便的 获取 或 设置 表单项数据

- 语法：v-model="变量名"

```
createApp({
  data() {
    return {
      //定义数据模型，采集员工搜索表单数据
      searchForm: {
        name: '',
        gender: '',
        job: ''
      }
    }
  }
}).mount('#container')
```

姓名:

性别: 请选择 职位: 请选择 查询 清空

姓名	性别	头像	所属部门	职位	入职日期	最后操作时间	操作
张三	男		研发部	班主任	2022-01-01	2022-01-01 12:45:05	<button>编辑</button> <button>删除</button>
李四	女		销售部	讲师	2022-01-01	2022-01-01 12:45:05	<button>编辑</button> <button>删除</button>
王五	男		财务部	学工主管	2022-01-01	2022-01-01 12:45:05	<button>编辑</button> <button>删除</button>

注意： v-model 中绑定的变量，必须在data中定义。

5. v-on

- 作用：为html标签绑定事件（添加事件监听）

- 语法：

- v-on:事件名="方法名"
- 简写为 @事件名="..."

```
<div id="app">
  <button type="button" v-on:click="handle">点我</button>
  <button type="button" @click="handle">再点我</button>
</div>
```

姓名: 性别: 请选择 职位: 请选择 查询 清空

姓名	性别	头像	所属部门	职位	入职日期	最后操作时间	操作
张三	男		研发部	班主任	2022-01-01	2022-01-01 12:45:05	<button>编辑</button> <button>删除</button>
李四	女		销售部	讲师	2022-01-01	2022-01-01 12:45:05	<button>编辑</button> <button>删除</button>
王五	男		财务部	学工主管	2022-01-01	2022-01-01 12:45:05	<button>编辑</button> <button>删除</button>

```
const app = createApp({
  data() {
    //...
  },
  methods: {
    handle() {
      console.log('试试就试试');
    }
  },
}).mount("#app");
```

注意： methods函数中的this指向Vue实例，可以通过this获取到data中定义的数据。

4 Vue 生命周期

- 生命周期：指一个对象从创建到销毁的整个过程。
- 生命周期的八个阶段：每触发一个生命周期事件，会自动执行一个生命周期方法(钩子)。

状态	阶段周期
beforeCreate	创建前
created	创建后
beforeMount	载入前
mounted	挂载完成
beforeUpdate	数据更新前
updated	数据更新后
beforeUnmount	组件销毁前
unmounted	组件销毁后

- 应用场景

在页面加载完毕时，发起异步请求，加载数据，渲染页面。

```
<script type="module">
  import { createApp } from 'https://.../vue.esm-browser.js'
  const app = createApp({
    data() {
      return {
        message: "Hello Vue"
      }
    },
    //生命周期-钩子函数 mounted
    mounted() {
      console.log('Vue挂载完毕，发送请求获取数据 ...');
    }
  }).mount("#app");
</script>
```