### Vue3新特性

* 1. Composition API（组合式API）
  2. Teleport：控制渲染位置
  3. Suspense：实现延迟加载和错误处理
  4. Fragment：承载多个子元素的虚拟组件，换言之，单文件组件，可以包含多个根节点
  5. Vue3响应式更新系统有哪些
  6. Proxy：拦截对象中任意属性的变化，包括：属性值的读写、添加、删除
  7. Reflect：对源对象的属性进行操作
  8. weakMap

### Vue3与vue2区别

* 1. 响应式系统重构：使用proxy代替object.defineProperty，优势
     1. Proxy直接监听对象（添加、删除）属性
     2. 直接监听数组
     3. 性能提升，proxy直接监听对象，object.defineProperty是遍历每个属性
  2. Vue3组合式API，更好的逻辑复用和代码组织
     1. Setup配置
     2. Ref和reactive
     3. Provide/inject
  3. V-if 和v-for优先级
     1. Vue2 v-for优先级高
     2. Vue3 是v-if优先级高
  4. 重构虚拟dom，diff算法
     1. Vue2 diff算法比较虚拟节点，并返回的一个patch对象，存储两个节点不同的地方，最后用patch记录去局部更新dom；属于全量diff
     2. Vue3 diff 算法初始化时会给每个虚拟节点添加一个patchFlags做静态标记，只会比较patchFlags发生变化的虚拟节点，进行更新视图，没有变化的，直接复用渲染，属于非全量diff
  5. 生命周期的变化
     1. Vue3 setup 代替beforCreate和created
     2. onBeforeMount ---> beforeMount;
     3. onMounted --> mounted
     4. onBeforeUpdate --> beforeUpdate
     5. onUpdated --> updated
     6. onBeforeUnMount --> beforeDestroy
     7. onUnMounted --> destoryed
  6. Vue3新内置组件：teleport、suspense、fragment

### Vue2响应式出现的问题

* 1. 新增、删除属性，界面不会更新
  2. 直接通过下标修改数组，界面不会自动更新

解决方案：使用$set

### Vue2第一次加载触发哪些钩子函数

* 1. BeforeCreate -> created -> beforeMount -> mounted

### hook 和mixin区别

* 1. 两者都是提供实现代码的可重用性和功能复用
  2. Mixin缺点：
     1. 命名冲突：多个mixin中可能包含相同名称的属性和方法，容易造成命名冲突
     2. 隐式依赖：组件行为可能因为引入mixin，导致代码难以理解和维护

c） hook优点：

i. 易于组合和复用：可以将不同的钩子函数组合使用，提高代码的高度复用

Ii. 易于测试和调试：由于逻辑和组件解耦，可以更容易地进行单元测试和调试

1. hook缺点：学习成本高和使用需要vue3版本及以上

### mixin

* 1. 不同方法名，正常调用，相同方法名，优先调用本组件的方法和属性，mixin失效
  2. 事件的构造函数（如created等生命周期）调用顺序，mixin优先
  3. 多组件引用mixin， 不会互相污染改变变量

### watch和watchEffect的区别

* 1. Watch需要传入监听数据元，watchEffect会自动追踪响应式数据作为数据源
  2. Watch可以访问新旧数据，watchEffect只能访问新数据
  3. Watch运行不会立即执行，需要值改变才进行，watchEffect运行后立即执行
  4. Watch监听reactive，需要使用torefs转换成ref数据，watchEffect无需而外处理
  5. Watch可以同时监听多个数据源，watchEffect不行，需要在回调函数内部使用响应式数据进行处理
  6. Watch可以监听对象中某个属性，watchEffect不行

8、