# 第五阶段面试题

## 阶段1: UniApp

### 题目1: 什么是条件编译

条件编译是用特殊的注释作为标记，在编译时根据这些特殊的注释，将注释里面的代码编译到不同平台。

分 ifdef 和 ifndef 两种判断方式.

#ifdef APP-PLUS  
在App平台下执行的代码  
#endif  
  
#ifndef H5  
在非H5平台下执行的代码  
#endif

### 题目2: 有哪些生命周期, 简述几个

共有3类生命周期

* 应用级: 书写在 App.vue 中
  + onLaunch 应用初始化完毕
  + onShow 应用显示, 从后台进入前台
  + onHide 应用从前台进入后台
* 页面级: 同微信小程序的周期
  + onLoad 页面加载, 可以接收路由参数
  + onPullDownRefresh 下拉刷新时
  + onReachBottom 触底时
  + onPageScroll 页面滚动时
* 组件级: 同vue组件周期
  + mounted 组件挂载完毕
  + destroyed 组件销毁后
  + created 组件创建完毕

### 题目3: 谈谈你对uniapp的理解

uniapp采用vue框架实现页面的制作, 采用微信小程序的组件库和api库;

目标是书写一次代码, 即可编译到不同的平台. 兼容 App Web 各类小程序;

能够用最低的成本实现多平台项目开发.

缺点是: 目前生态还不够完善, 官方遗留了很多BUG等待解决. 目前还无法完美实现所有需求.

### 题目4: 如何实现页面之间的跳转操作

有两种方式可以实现

* 组件方式: 使用uni-navigator组件
* 编程方式: 使用uni.navigateTo这个Api

### 题目5: 如何实现下拉刷新操作

首先, 必须在pages.json文件中, 为需要开启下拉刷新操作的页面开启下拉刷新功能: enablePullDownRefresh

然后, 利用页面的生命周期函数onPullDownRefresh监听下拉刷新操作触发的时机

最后, 利用uni.stopPullDownRefresh方法结束下拉刷新动画

## 阶段2: React

### 题目1: 什么是JSX

JSX是一种JavaScript的语法扩展, 本质是DOM操作的语法糖写法.

使用JSX编写DOM代码更加简单快速.

例如:

const element = <div>Hello World!</div>;

### 题目2: 什么是Redux

Redux是基于JavaScript的一款前端开发库, 用于整个应用的状态管理.

类似于Vue中的VueX.

### 题目3: 什么是Hook

Hook是React16.8的新增特性. 可以在不编写class的情况下使用state以及其他的React特性.

例如:

* useState: 用于向函数组件新增state状态值
* useEffect: 用于向函数组件新增生命周期相似的特性

### 题目4: 在代码中如何获取某个子元素的索引

方式1:

abc = React.createRef();   
   
render(){   
 return <el ref={this.abc} />   
}   
   
//使用时:   
this.abc.current

方式2:

render(){   
 return <el ref={el => this.abc = el} />   
}   
   
//使用时:   
this.abc

### 题目5: state 和 props 的区别

* state: 配合setState函数使用, 保存在state中的值可以绑定到页面上, 使用setState进行更新时可以同步刷新页面.
* props: 组件通过属性传入的参数会保存在props属性中

### 题目6: 生命周期有哪些

* componentDidMount 挂载完毕
* componentWillUnmount 将要卸载
* shouldComponentUpdate 组件是否要更新
* componentDidUpdate 组件更新完毕

### 题目7: 书写一段Redux代码, 实现更新 name 属性

//1.安装 npm i redux --save   
import {createStore} from "redux"   
   
//2.初始化数据 state   
const initialState = {   
 name: "妲己",   
}   
   
//3.reducer   
function reducer(state = initialState, action) {   
 switch (action.type) {   
 case "changeName":   
 return {   
 ...state,   
 name: action.name   
 }   
 default:   
 return state;   
 }   
}   
//4. 创建仓库   
const store=createStore(reducer);   
   
//5. 仓库监听   
store.subscribe(()=>{   
 console.log(store.getState())   
})   
   
//6. 更改name   
store.dispatch({type:"changeName", name:'东东'})

## 阶段3: Angular

### 题目1: 内置的过滤器有哪些?

* date 日期格式化
* currency 千进制钱
* lowercase 变小写
* uppercase 变大写
* percent 百分数

### 题目2: 什么是依赖注入机制

依赖注入机制分两部分理解:

* 依赖: 组件的构造方法中可以声明依赖 constructor(权限词 变量名: 类型)
* 注入: 系统会根据组件的构造方法中的变量类型要求, 提供对应的变量. 关联 服务 特性;

### 题目3: 有哪些快捷命令

* ng g c 组件
* ng g p 管道
* ng g d 指令
* ng g s 服务
* ng s -o 启动服务 并 打开
* ng n 项目名 创建项目

### 题目4: 有哪些数据绑定写法

* 插值: {{ 变量 }}
* 属性: <el [属性名]="值" />
* 事件: <el (事件名)="方法名()" />
* 双向数据绑定: <el [(ngModel)]="变量" />

### 题目5: 生命周期

* ngOnChanges: 接受外部传入属性
* ngOnInit: 组件初始化时
* ngDoCheck: 组件外部传入的属性发生变化
* ngAfterContentInit: 组件数据初始化完毕
* ngAfterViewInit: 组件界面初始化完毕
* ngAfterContentChecked: 组件数据更新
* ngAfterViewChecked: 组件界面更新
* ngOnDestroy: 组件销毁时

### 题目6: 如何获取子元素的索引

首先为子元素绑定一个唯一标识: <el #唯一标识 />

然后利用@ViewChild('唯一标识') 变量名: 类型名 的方式关联变量和元素

## 阶段4: Vue项目

### 题目1: 什么是SPA项目

单页应用 Single Page web Application

单页应用的流畅性让 Web 应用更像桌面端或 Native 应用了。相对于传统的 Web 应用，单页应用做到了前后端分离，后端只负责处理数据提供接口，页面逻辑和页面渲染都交给了前端。前端发展到现在，单页应用的使用已经很广泛，目前时兴的 React、Vue、Angular 等前端框架均采用了 SPA 原则。单页应用意味着前端掌握了主动权，也让前端代码更复杂和庞大，模块化、组件化以及架构设计都变得越来越重要

### 题目2: v-show 和 v-if 的区别

两个指令都可以实现元素的隐藏和显示.

不同点在于:

* v-show: 利用 css 的 display:none 实现元素隐藏; 效率高, 适合频繁切换.
* v-if: 利用 DOM 的移除操作实现元素隐藏; 效率低, 适合一次性操作. 例如页面初始加载时忽略某些元素的渲染, 当网络操作完毕后再进行渲染.

### 题目3: axios的集成方式

* 安装: 项目目录下执行安装命令 npm i axios
* 集成: 在 main.js 中
  + 使用 import axios from 'axios' 引入
  + 利用原型进行注入: Vue.prototype.axios = axios
* 使用: 在组件中利用 this.axios 进行使用

### 题目4: 监听路由中参数的变化

* 监听器
* watch:{   
   '$route.params.xxx'(to, from){   
     
   }   
  }
* 路由守卫
* beforeRouteUpdate(to, from, next) {   
   next()   
  }

### 题目5: $route 和 $router 的区别

* $route: 存储了路由中存储的相关值, 例如路由传参, 路由名, meta 等
* $router: 存储了操作路由的各种方法, 例如 $router.push() 可以实现跳转