**2018大厂高级前端面试题汇总**

**阿里**

* 使用过的koa2中间件
* koa-body原理
* 介绍自己写过的中间件
* 有没有涉及到Cluster
* 介绍pm2
* master挂了的话pm2怎么处理
* 如何和MySQL进行通信
* React声明周期及自己的理解
* 如何配置React-Router
* 路由的动态加载模块
* 服务端渲染SSR
* 介绍路由的history
* 介绍Redux数据流的流程
* Redux如何实现多个组件之间的通信，多个组件使用相同状态如何进行管理
* 多个组件之间如何拆分各自的state，每块小的组件有自己的状态，它们之间还有一些公共的状态需要维护，如何思考这块
* 使用过的Redux中间件
* 如何解决跨域的问题
* 常见Http请求头
* 移动端适配1px的问题
* 介绍flex布局
* 其他css方式设置垂直居中
* 居中为什么要使用transform（为什么不使用marginLeft/Top）
* 使用过webpack里面哪些plugin和loader
* webpack里面的插件是怎么实现的
* dev-server是怎么跑起来
* 项目优化
* 抽取公共文件是怎么配置的
* 项目中如何处理安全问题
* 怎么实现this对象的深拷贝

**网易**

* 介绍redux，主要解决什么问题
* 文件上传如何做断点续传
* 表单可以跨域吗
* promise、async有什么区别
* 搜索请求如何处理（防抖）
* 搜索请求中文如何请求
* 介绍观察者模式
* 介绍中介者模式
* 观察者和订阅-发布的区别，各自用在哪里
* 介绍react优化
* 介绍http2.0
* 通过什么做到并发请求
* http1.1时如何复用tcp连接
* 介绍service worker
* 介绍css3中position:sticky
* redux请求中间件如何处理并发
* 介绍Promise，异常捕获
* 介绍position属性包括CSS3新增
* 浏览器事件流向
* 介绍事件代理以及优缺点
* React组件中怎么做事件代理
* React组件事件代理的原理
* 介绍this各种情况
* 前端怎么控制管理路由
* 使用路由时出现问题如何解决
* React怎么做数据的检查和变化

**滴滴**

* react-router怎么实现路由切换
* react-router里的<Link>标签和<a>标签有什么区别
* <a>标签默认事件禁掉之后做了什么才实现了跳转
* React层面的性能优化
* 整个前端性能提升大致分几类
* import { Button } from 'antd'，打包的时候只打包button，分模块加载，是怎么做到的
* 使用import时，webpack对node\_modules里的依赖会做什么
* JS异步解决方案的发展历程以及优缺点
* Http报文的请求会有几个部分
* cookie放哪里，cookie能做的事情和存在的价值
* cookie和token都存放在header里面，为什么只劫持前者
* cookie和session有哪些方面的区别
* React中Dom结构发生变化后内部经历了哪些变化
* React挂载的时候有3个组件，textComponent、composeComponent、domComponent，区别和关系，Dom结构发生变化时怎么区分data的变化，怎么更新，更新怎么调度，如果更新的时候还有其他任务存在怎么处理
* key主要是解决哪一类的问题，为什么不建议用索引index（重绘）
* Redux中异步的请求怎么处理
* Redux中间件是什么东西，接受几个参数（两端的柯里化函数）
* 柯里化函数两端的参数具体是什么东西
* 中间件是怎么拿到store和action，然后怎么处理
* state是怎么注入到组件的，从reducer到组件经历了什么样的过程
* koa中response.send、response.rounded、response.json发生了什么事，浏览器为什么能识别到它是一个json结构或是html
* koa-bodyparser怎么来解析request
* webpack整个生命周期，loader和plugin有什么区别
* 介绍AST（Abstract Syntax Tree）抽象语法树
* 安卓Activity之间数据是怎么传递的
* 安卓4.0到6.0过程中WebView对js兼容性的变化
* WebView和原生是如何通信
* 跨域怎么解决，有没有使用过Apache等方案

**今日头条**

* 对async、await的理解，内部原理
* 介绍下Promise，内部实现
* 清除浮动
* 定位问题（绝对定位、相对定位等）
* 从输入URL到页面加载全过程
* tcp3次握手
* tcp属于哪一层（1 物理层 -> 2 数据链路层 -> 3 网络层(ip)-> 4 传输层(tcp) -> 5 应用层(http)）
* redux的设计思想
* 接入redux的过程
* 绑定connect的过程
* connect原理
* webpack介绍
* == 和 ===的区别，什么情况下用相等==
* bind、call、apply的区别
* 动画的了解
* 介绍下原型链（解决的是继承问题吗）
* 对跨域的了解

**有赞**

* Linux 754 介绍
* 介绍冒泡排序，选择排序，冒泡排序如何优化
* transform动画和直接使用left、top改变位置有什么优缺点
* 如何判断链表是否有环
* 介绍二叉搜索树的特点
* 介绍暂时性死区
* ES6中的map和原生的对象有什么区别
* 观察者和发布-订阅的区别
* react异步渲染的概念,介绍Time Slicing 和 Suspense
* 16.X声明周期的改变
* 16.X中props改变后在哪个生命周期中处理
* 介绍纯函数
* 前端性能优化
* pureComponent和FunctionComponent区别
* 介绍JSX
* 如何做RN在安卓和IOS端的适配
* RN为什么能在原生中绘制成原生组件（bundle.js）
* 介绍虚拟DOM
* 如何设计一个localStorage，保证数据的实效性
* 如何设计Promise.all()
* 介绍高阶组件
* sum(2, 3)实现sum(2)(3)的效果
* react性能优化
* 两个对象如何比较

**挖财**

* JS的原型
* 变量作用域链
* call、apply、bind的区别
* 防抖和节流的区别
* 介绍各种异步方案
* react生命周期
* 介绍Fiber
* 前端性能优化
* 介绍DOM树对比
* react中的key的作用
* 如何设计状态树
* 介绍css，xsrf
* http缓存控制
* 项目中如何应用数据结构
* native提供了什么能力给RN
* 如何做工程上的优化
* shouldComponentUpdate是为了解决什么问题
* 如何解决props层级过深的问题
* 前端怎么做单元测试
* webpack生命周期
* webpack打包的整个过程
* 常用的plugins
* pm2怎么做进程管理，进程挂掉怎么处理
* 不用pm2怎么做进程管理

**沪江**

* 介绍下浏览器跨域
* 怎么去解决跨域问题
* jsonp方案需要服务端怎么配合
* Ajax发生跨域要设置什么（前端）
* 加上CORS之后从发起到请求正式成功的过程
* xsrf跨域攻击的安全性问题怎么防范
* 使用Async会注意哪些东西
* Async里面有多个await请求，可以怎么优化（请求是否有依赖）
* Promise和Async处理失败的时候有什么区别
* Redux在状态管理方面解决了React本身不能解决的问题
* Redux有没有做过封装
* react生命周期，常用的生命周期
* 对应的生命周期做什么事
* 遇到性能问题一般在哪个生命周期里解决
* 怎么做性能优化（异步加载组件...）
* 写react有哪些细节可以优化
* React的事件机制（绑定一个事件到一个组件上）
* 介绍下事件代理，主要解决什么问题
* 前端开发中用到哪些设计模式
* React/Redux中哪些功能用到了哪些设计模式
* JS变量类型分为几种，区别是什么
* JS里垃圾回收机制是什么，常用的是哪种，怎么处理的
* 一般怎么组织CSS（Webpack）

**饿了么**

* 小程序里面开页面最多多少
* React子父组件之间如何传值
* Emit事件怎么发，需要引入什么
* 介绍下React高阶组件，和普通组件有什么区别
* 一个对象数组，每个子对象包含一个id和name，React如何渲染出全部的name
* 在哪个生命周期里写
* 其中有几个name不存在，通过异步接口获取，如何做
* 渲染的时候key给什么值，可以使用index吗，用id好还是index好
* webpack如何配sass，需要配哪些loader
* 配css需要哪些loader
* 如何配置把js、css、html单独打包成一个文件
* div垂直水平居中（flex、绝对定位）
* 两个元素块，一左一右，中间相距10像素
* 上下固定，中间滚动布局如何实现
* [1, 2, 3, 4, 5]变成[1, 2, 3, a, b, 5]
* 取数组的最大值（ES5、ES6）
* apply和call的区别
* ES5和ES6有什么区别
* some、every、find、filter、map、forEach有什么区别
* 上述数组随机取数，每次返回的值都不一样
* 如何找0-5的随机数，95-99呢
* 页面上有1万个button如何绑定事件
* 如何判断是button
* 页面上生成一万个button，并且绑定事件，如何做（JS原生操作DOM）
* 循环绑定时的index是多少，为什么，怎么解决
* 页面上有一个input，还有一个p标签，改变input后p标签就跟着变化，如何处理
* 监听input的哪个事件，在什么时候触发

**携程**

* 对React看法，有没有遇到一些坑
* 对闭包的看法，为什么要用闭包
* 手写数组去重函数
* 手写数组扁平化函数
* 介绍下Promise的用途和性质
* Promise和Callback有什么区别
* React生命周期
* 两道手写算法题

**喜马拉雅**

* ES6新的特性
* 介绍Promise
* Promise有几个状态
* 说一下闭包
* React的生命周期
* componentWillReceiveProps的触发条件是什么
* React16.3对生命周期的改变
* 介绍下React的Filber架构
* 画Filber渲染树
* 介绍React高阶组件
* 父子组件之间如何通信
* Redux怎么实现属性传递，介绍下原理
* React-Router版本号
* 网站SEO怎么处理
* 介绍下HTTP状态码
* 403、301、302是什么
* 缓存相关的HTTP请求头
* 介绍HTTPS
* HTTPS怎么建立安全通道
* 前端性能优化（JS原生和React）
* 用户体验做过什么优化
* 对PWA有什么了解
* 对安全有什么了解
* 介绍下数字签名的原理
* 前后端通信使用什么方案
* RESTful常用的Method
* 介绍下跨域
* Access-Control-Allow-Origin在服务端哪里配置
* csrf跨站攻击怎么解决
* 前端和后端怎么联调

**兑吧**

* localStorage和cookie有什么区别
* CSS选择器有哪些
* 盒子模型，以及标准情况和IE下的区别
* 如何实现高度自适应
* prototype和——proto——区别
* \_construct是什么
* new是怎么实现的
* promise的精髓，以及优缺点
* 如何实现H5手机端的适配
* rem、flex的区别（root em）
* em和px的区别
* React声明周期
* 如何去除url中的#号
* Redux状态管理器和变量挂载到window中有什么区别
* webpack和gulp的优缺点
* 如何实现异步加载
* 如何实现分模块打包（多入口）
* 前端性能优化（1js css；2 图片；3 缓存预加载； 4 SSR； 5 多域名加载；6 负载均衡）
* 并发请求资源数上限（6个）
* base64为什么能提升性能，缺点
* 介绍webp这个图片文件格式
* 介绍koa2
* Promise如何实现的
* 异步请求，低版本fetch如何低版本适配
* ajax如何处理跨域
* CORS如何设置
* jsonp为什么不支持post方法
* 介绍同源策略
* React使用过的一些组件
* 介绍Immuable
* 介绍下redux整个流程原理
* 介绍原型链
* 如何继承

**微医**

* 介绍JS数据类型，基本数据类型和引用数据类型的区别
* Array是Object类型吗
* 数据类型分别存在哪里
* var a = {name: "前端开发"}; var b = a; a = null那么b输出什么
* var a = {b: 1}存放在哪里
* var a = {b: {c: 1}}存放在哪里
* 栈和堆的区别
* 垃圾回收时栈和堆的区别
* 数组里面有10万个数据，取第一个元素和第10万个元素的时间相差多少
* 栈和堆具体怎么存储
* 介绍闭包以及闭包为什么没清除
* 闭包的使用场景
* JS怎么实现异步
* 异步整个执行周期
* Promise的三种状态
* Async/Await怎么实现
* Promise和setTimeout执行先后的区别
* JS为什么要区分微任务和宏任务
* Promise构造函数是同步还是异步执行，then呢
* 发布-订阅和观察者模式的区别
* JS执行过程中分为哪些阶段
* 词法作用域和this的区别
* 平常是怎么做继承
* 深拷贝和浅拷贝
* loadsh深拷贝实现原理
* ES6中let块作用域是怎么实现的
* React中setState后发生了什么
* setState为什么默认是异步
* setState什么时候是同步的
* 为什么3大框架出现以后就出现很多native（RN）框架（虚拟DOM）
* 虚拟DOM主要做了什么
* 虚拟DOM本身是什么（JS对象）
* 304是什么
* 打包时Hash码是怎么生成的
* 随机值存在一样的情况，如何避免
* 使用webpack构建时有无做一些自定义操作
* webpack做了什么
* a，b两个按钮，点击aba，返回顺序可能是baa，如何保证是aba（Promise.then）
* node接口转发有无做什么优化
* node起服务如何保证稳定性，平缓降级，重启等
* RN有没有做热加载
* RN遇到的兼容性问题
* RN如何实现一个原生的组件
* RN混原生和原生混RN有什么不同
* 什么是单页项目
* 遇到的复杂业务场景
* Promise.all实现原理

**寺库**

* 介绍Promise的特性，优缺点
* 介绍Redux
* RN的原理，为什么可以同时在安卓和IOS端运行
* RN如何调用原生的一些功能
* 介绍RN的缺点
* 介绍排序算法和快排原理
* 堆和栈的区别
* 介绍闭包
* 闭包的核心是什么
* 网络的五层模型
* HTTP和HTTPS的区别
* HTTPS的加密过程
* 介绍SSL和TLS
* 介绍DNS解析
* JS的继承方法
* 介绍垃圾回收
* cookie的引用为了解决什么问题
* cookie和localStorage的区别
* 如何解决跨域问题
* 前端性能优化

**宝宝树**

* 使用canvas绘图时如何组织成通用组件
* formData和原生的ajax有什么区别
* 介绍下表单提交，和formData有什么关系
* 介绍redux接入流程
* rudux和全局管理有什么区别（数据可控、数据响应）
* RN和原生通信
* 介绍MVP怎么组织
* 介绍异步方案
* promise如何实现then处理
* koa2中间件原理
* 常用的中间件
* 服务端怎么做统一的状态处理
* 如何对相对路径引用进行优化
* node文件查找优先级
* npm2和npm3+有什么区别

**海康威视**

* knex连接数据库响应回调
* 介绍异步方案
* 如何处理异常捕获
* 项目如何管理模块
* 前端性能优化
* JS继承方案
* 如何判断一个变量是不是数组
* 变量a和b，如何交换
* 事件委托
* 多个<li>标签生成的Dom结构是一个类数组
* 类数组和数组的区别
* dom的类数组如何转成数组
* 介绍单页面应用和多页面应用
* redux状态树的管理
* 介绍localstorage的API

**蘑菇街**

* html语义化的理解
* <b>和<strong>的区别
* 对闭包的理解
* 工程中闭包使用场景
* 介绍this和原型
* 使用原型最大的好处
* react设计思路
* 为什么虚拟DOM比真实DOM性能好
* react常见的通信方式
* redux整体的工作流程
* redux和全局对象之间的区别
* Redux数据回溯设计思路
* 单例、工厂、观察者项目中实际场景
* 项目中树的使用场景以及了解
* 工作收获

**酷家乐**

* react生命周期
* react性能优化
* 添加原生事件不移除为什么会内存泄露
* 还有哪些地方会内存泄露
* setInterval需要注意的点
* 定时器为什么是不精确的
* setTimeout(1)和setTimeout(2)之间的区别
* 介绍宏任务和微任务
* promise里面和then里面执行有什么区别
* 介绍pureComponet
* 介绍Function Component
* React数据流
* props和state的区别
* 介绍react context
* 介绍class和ES5的类以及区别
* 介绍箭头函数和普通函数的区别
* 介绍defineProperty方法，什么时候需要用到
* for..in 和 object.keys的区别
* 介绍闭包，使用场景
* 使用闭包特权函数的使用场景
* get和post有什么区别

**百分点**

* React15/16.x的区别
* 重新渲染render会做些什么
* 哪些方法会触发react重新渲染
* state和props触发更新的生命周期分别有什么区别
* setState是同步还是异步
* 对无状态组件的理解
* 介绍Redux工作流程
* 介绍ES6的功能
* let、const以及var的区别
* 浅拷贝和深拷贝的区别
* 介绍箭头函数的this
* 介绍Promise和then
* 介绍快速排序
* 算法：前K个最大的元素

**海风教育**

* 对react看法，它的优缺点
* 使用过程中遇到的问题，如何解决的
* react的理念是什么（拿函数式编程来做页面渲染）
* JS是什么范式语言(面向对象还是函数式编程)
* koa原理，为什么要用koa(express和koa对比)
* 使用的koa中间件
* ES6使用的语法
* Promise 和 async/await 和 callback的区别
* Promise有没有解决异步的问题（promise链是真正强大的地方）
* Promise和setTimeout的区别（Event Loop）
* 进程和线程的区别（一个node实例就是一个进程，node是单线程，通过事件循环来实现异步）
* 介绍下DFS深度优先
* 介绍下观察者模式
* 观察者模式里面使用的数据结构(不具备顺序 ，是一个list)