



面试多个公司总结出的套路

腾讯音乐面试

其它公司面试感受

面试感受一

面试感受二

面试感受三

面试经验总结

原生 JavaScript 组合拳

布局和页面性能

算法设计模式

## 面试多个公司总结出的套路

### 腾讯音乐面试

面试官非常 nice，善于引导，不会的问题就会善于提示。面试官先让做了自我介绍，然后又问了以前做过的项目，根据做过的项目进行提问，登陆过程后端怎么实现，还有冒泡，import 与 require 对循环引用的处理，还有有没有遇到兼容性问题，还问了 es6 新特性这样。

问题：

- 介绍项目
- 前端安全
- 前端优化
- 内存泄漏
- css 清除浮动；
- this 指向；
- css 实现垂直居中；
- 实现 BFC。

主要问了项目经历，网站性能优化，https、http2.0、负载均衡、平常的爱好。面试官深入考察了 TCP 对 3 次握手和 4 次挥手，问了各种状态码，主要的 http 头部。考察了一些经典数据结构以及算法对实现。深入问了一些 js 的底层运行原理。

### 其它公司面试感受

#### 面试感受一

一面：面试官和蔼，气氛比较轻松。围绕之前的工作项目问了一些内容，移动开发、ssr、pwa 之类的。再就是一些基础问题 es6、协商缓存、vue 基础和原理，css。二面：同事面，问了下具体工作中遇到的问题和解决方法，异步事件等等。总体下来，面试气氛都挺棒的。主要个人没准备好就去面了，下次有准备了还会再去试试。

#### 面试感受二

一面：

- 大致介绍一下你的前端项目经历
- 如何使低版本的浏览器（如 IE8 等）支持新的语言特性（比如 数组的 forEach 方法）—提示：使用原型链的相关知识
- JS 为什么是单线程？这样做有什么好处是多线程语言无法实现的？—（解释进程和线程的区别和联系）
- 项目中有使用过 react 和 vue 那么请比较说明一下两者的优劣
- vue 是如何实现数据的双向绑定的？（数据劫持和 Object.defineProperty 中的 set 和 get 函数） react 中的单向数据流动有什么好处？
- 对框架源码的熟悉程度 react 向一个位于很深节点的子组件传值，除了 props 还有没有别的方法？（我说的是 redux 里面的 provider，但是被告诉说这个不对，然后我也不知道是什么还没有百度出来）
- 简单说明一下你对 ES6 中箭头函数的理解
- 跨域问题：如何解决跨域问题？设置请求头应该如何设置？那么服务器返回的报文的响应头中有什么不一样的地方
- 讲一下 TCP 三次握手的过程，输入一个 url 到页面呈现到浏览器上的全过程
- 怎样以对象为构造函数创建一个对象
- 使用过 webpack 吗？讲一下插件（plugin）和 loader 的区别。

#### 面试感受三

一面：

- 性能优化：项目里做过哪些性能优化，怎么检测性能，有没有实际测过优化的数据有多少提—网络安全方面的问题，有哪些安全问题
- http2.0 有哪些特性？https 是怎么加密的，http 强缓存和协商缓存在项目里怎么用的；强缓存一般设置多久
- JS 基础：闭包，事件循环，作用域；
- http 响应状态码有哪些，cookie，localStorage 的大小限制及超出的异常如何处理；
- 错误监控有哪些方式；在项目里用过吗
- 队列和栈的特性；
- 算法题:反转单链表，给 6 分钟时间，说出代码思路或伪代码；
- 跨域怎么解决，跨域的复杂请求和简单请求的区别，简单请求响应头会不会返回 Access-Control-Allow-Origin、Access-Control-Allow-Origin 设置为\*和具体的域名有什么区别；
- 如何保证用户一定是访问 https 而非 http 的网站。
- service work 离线存储失败的情况有遇见过吗？怎么处理
- 聊了小程序的性能优化，setData 大数组会有什么问题

### 面试经验总结

首先聊一下腾讯的背景， 腾讯前端是以原生 JavaScript 为主，所以对 JavaScript 的要求一定会很高， 那么这个高要求和高标准大概有多高那， 面试官通常会以丰富的经验给你各种各样的组合拳， 这一套又一套的组合拳着实打的新手摸不着头脑？我们先来分析一套原生组合拳：

#### 原生 JavaScript 组合拳

- 问：基本类型和引用类型的区别。
- 答：基本类型是存储在栈之中的值，引用类型是存储在堆中数据的地址。
- 问：可以举例说明一下 JavaScript 之中的引用类型有哪些么？
- 答：Object，function，Array，Set，Map

当然了，面试 JavaScript 肯定离不开原型链和作用域链，那么什么是作用域链那？这个其实是说来话长的一个话题，因为在这里给大家阐述下我个人的一些理解：函数在运行的过程之中会产生一个临时存储数据的对象，这个对象我们称之为 activeObject，简称 AO，每个函数在执行时都会在栈里面存上这么一个东西，以便于储存期中的局部变量。但是 js 有全局变量的概念，也就是在局部中可以访问全局的数据，怎么访问那？当然是由内自外的冒泡访问，但是如何由内自外，这个以包含关系决定访问顺序就是传说中的作用域链。

接着来说原型链，这个是实例对象访问方法和共用数据的一个顺序，也是由内自外的。接着会顺着继续向下问一些关于闭包，内存泄漏的问题。

闭包问题，其实就是两个函数嵌套，产生了两个嵌套的活动对象，子活动对象存储了父级活动对象的属性，从而导致外部函数的局部变量被子活动对象保存的一种 JavaScript 特性。

用代码表示一下：

```
function outer(){  
    var freeVar = "hello"  
}
```

这是一个外部函数会产生一个 AO，这个 AO 是因为 outer 函数执行而创建出来的，所以我们暂时命名为 AO(outer).

闭包自然少不了函数的嵌套：

```
function outer(){  
    var freeVar = "hello";  
    function inner(){  
        console.log(freeVar);  
    }  
    return inner;  
}
```

但凡是在 AO(outer)内部创建的函数引用了外部的变量那么妥了，闭包就产生了。

收起你的黑人问号脸，来看一下我的解释：

Inner 函数执行时当然也会产生 AO，那么这个 AO 我们取名叫做 AO(inner), ok 在这个 ao 中我们干了啥？是不是引用了外部函数的变量 freeVar？对啊，肯定用了，但是你只是用了 freeVar 这么简单么？no,no,no，其实在内部函数之中我们引用的是 AO(outer).freeVar。

好了我们这个时候来看一下，AO 是个啥？是个对象啊，既然是对象那这货是不是一个引用类型，引用类型存在于内存之中不？存在，那么恭喜你，这个 AO.freeVar 不会想普通的局部变量在函数结尾被删除了，而是被保留在了内存之中。这，就是闭包。

基本上这一套下来，你对 JavaScript 的理解，对 JavaScript 原理和机制的理解面试官也就了解的比较透彻了，接下来会考察一些知识的广度，当然这些问题都是比较简单随意也是比较容易得到答案的，所以在后面的部分我就只列举问题。

#### 布局和页面性能

这个就是考察基本功的时候了，布局可以说是一个前端最常用的技能，所以会根据各种需求提问,当然同样重要的还有一些优化细节，比如：

- 重绘和回流分别都是什么，哪个性能消耗的多？应该如何去提高性能？
- 各种居中。
- 命名空间和命名方式。
- 移动端 rem 布局，百分比布局，响应式布局。
- HTML 标签的语义化
- JavaScript 多线程。
- http 请求优化

#### 算法设计模式

基本排序算法的理解，比如快排，冒泡等排序算法总要知道一个吧，这种东西提前准备总是没坏处。设计模式这种东西就是一个没啥卵用的大框框，但是有写设计模式确实很有借鉴的意义，比如观察者模式，策略模式，可以让代码更加优(zhuang)雅(bi)清晰。