



ByteDance

字节跳动



首先总结一下，本身之前是有过 字节跳动 data部门的半年实习，这次投递的同样是 **base杭州的基础架构部门**。总的来说对字节的扁平化管理以及技术氛围还是比较喜欢的，这次的面试体验也挺好的，八股不太多，都是顺着你讲的点一步步深入，直到达到你的**盲区**🤔

下边是面试的总结，如果有回答的不好的地方敬请指教~ 🐶

一面 60 分钟

一开始都是看着简历问问题,可能因为是一面的关系,后边问的都比较基础,有些篇幅长比较常规的就没具体写上

1. 自我介绍 -- balabala
2. 实习经历 -- balabala
3. 看你简历上写了解 vite, 介绍一下

答: 预构建、核心是依赖现代浏览器支持 ESM 的特性,针对开发体验优化、真正按需加载...

4. vite 预构建这块做了什么, 为什么他冷启动快, 和 webpack 的区别是什么

答:

1. 预构建主要是通过 esbuild 将浏览器原生不支持的 UMD 和 CJS 构建成 ESM, 还有一些比如 vue 的 sfc 解析成 js
2. 把裸模块自动解析成相对路径, 如 import
... from 'vue' =>from
'node_modules..', 这样子浏览器后边引入的时候就不会报错
3. 整合一些依赖的单独文件, 减少请求次数
4. 冷启动快是因为不需要打包, 做好预构建后等待浏览器按需请求即可
5. webpack 需要编译打包等等

5. 一个基本上都用 ESM 写的项目，vite 和 webpack 在开发上都一样？

答：不一样 vite 开发环境不需要打包，在预构建之后直接等待浏览器按需请求即可，而 webpack 还要从入口解析依赖然后编译打包等等

6. 看你还提到 pnpm，它和 npm 还有 yarn 有什么不同呢

答：

1. 更新包最大复用老版本
2. 一类依赖包硬连接维护同一个磁盘镜像
3. 对于 npm3+、yarn 而言不会有包分身的问题
4. 表面扁平化管理依赖，实际上是软连接到 .pnpm 文件下真实依赖的一个嵌套结构，使得 pnpm 不会有幻影依赖的问题，只会把项目的直接依赖提升到最顶层级（幻影依赖的具体原因简单的说是由于依赖提升以及 node 的 require 方法的向父级递归寻找依赖的机制，具体可看 [Phantom dependencies](#)）

7. 诶？那 npm 或者 yarn 为啥会有幻影依赖的问题呢？

答：因为他们是扁平化管理依赖，所以有依赖提升，并且因为 node 的 require 机制的原因

7. 诶？那 npm 或者 yarn 为啥会有幻影依赖的问题呢？

答：因为他们是扁平化管理依赖，所以有依赖提升，并且因为 node 的 require 机制的原因

8. 那他们做不到最大可能复用文件包或者维护同一个磁盘镜像吗？

答：yarn2 的话有一个 pnp 的模式就是维护同一个磁盘镜像的包，并且提供了对特定依赖包更改调试的方法，但是这种模式过于激进，直接取消了 node_modules，这样会产生一系列兼容性问题

9. 诶？那假如 pnpm 依赖了一个 npm 包，这个 npm 包会有幻影依赖的问题吗（默认的话）

答：默认的话还是存在幻影依赖的问题的，因为 pnpm 的配置中默认有一个参数 `hoist: true`，默认是会提升子依赖的。

10. 你是更熟悉 vue 还是 react

答：vue

11. vue 的生命周期有哪些

答：常规

12. 父子组件的生命周期过程手写一下

答：常规

13. vue3 了解过吗，跟 vue2 的响应式原理有啥区别和提升 -- defineProperty 的问题所在（对象、数组），proxy 做了什么

答：

1. defineProperty 监听不到对象属性的增删，数组监听不到以索引修改内容以及更改 length 等等
2. Proxy 相当于直接在对象上层架设一层拦截，解决了 defineProperty 中的一系列弊端，并且因为操作粒度更大，占用的内存更小

14. Vue.nextTick 了解吗？

答：了解的，在 created 里边操作 DOM 的话会失败，因为 DOM 还没有挂载，因此需要将操作放入 nextTick 回调函数，他会在 DOM 挂载渲染好后触发；或者在状态改变后想要操作渲染过后的 DOM 也需要把操作放在 nextTick 后边

15. 那如果我直接 element.innerHTML 操作完后，再用 js 去获取元素，能够获取到渲染之后的结果吗？

答：这样子是可以的，因为他是个同步的操作

16. 那为什么 vue 要将渲染更新设置成异步呢？

答：因为有虚拟 DOM 比对是需要一定的开销的，并且对同个节点的多个操作可以合并到一起异步更新，优化性能

17. 移动端做过吗?

答: 做过但是时间很久远了

18. 移动端的响应式布局怎么做?

答: lib-flexible 🐶

19. lib-flexible 具体原理

答: 设置rem大小, 并且所有的元素大小按照与设计高的比例进行转化, 转化为rem单位

20. 根元素的字体大小怎么设置, 如果有多个终端, 怎么保证屏幕显示比例一样?

答: 直接设置html的font-size属性; 媒介查询?

21. 有了解过屏幕像素和物理像素的关系吗

答: 忘记了, 逻辑像素实际上对应的是多个物理像素?

22. 移动端开发的比较久远了吧

答: 啊啊对对对 🐶

23. flex 布局了解过吗, flex-grow

答: 了解过

24. flex-grow 和 width 有关系吗, 假如说一个 1000px 的容器, 然后内部有两个宽 100px 400px 的子元素 flex-grow 分别设置为 2 1, 那么剩下的 500px 怎么分配?

答: 跟 width 没关系 剩下的 500 按比例分配 07/23/1410

25. `display: none` 和 `visibility: hidden` 了解过吗?

答: 有的有的, 这个的主要的区别就是 `visibility: hidden` 他会继续保留元素位置, 并且 `visibility` 为 `hidden` 的父元素的子元素的 `visibility` 属性设置成 `true` 的话, 还是可以显示出来的

26. `cors` 跨域有了解过吗?

答: 有的有的, 服务端和客户端配合, 简单请求、复杂请求, 响应头等等 (略)

27. 那么 `localhost:8080` 请求一个跨域的站内测试 `api`, 没有人配合你 `cors` 怎么办呢? 页面上的请求是请求到哪里的?

答: 可以使用正向代理, 配置 `devServe`, 请求到 `localhost:8080`

28. 正向代理原理是什么呢?

答: 服务端没有跨域的问题, 同源策略是浏览器层面才有的; 请求拦截, 服务端层面转发;

29. 讲讲 `commonJS` 和 `ESM` 有啥区别

答: `balabala`

30. 来做做题目吧

答: 没问题

31. 版本匹配 算法 12.0.1? 12.9.4

答：简单题吧，不过第一时间想到的不是用 sort，而是用 reduce 做的

32. 说说思路，你这个思路还挺新奇的，第一次见人这么干 🐶

33. 看你用了 let，这里可以用 const 吗，这两个有啥区别

答：常规

34. 手写 Promise.all

答：常规手撕前端题目

35. 反问环节

答：对我的评价？ -- 答得还不错，整体来说还可以，我这边给你过了，不过我们部门一般一天一面，回去等 hr 联系吧