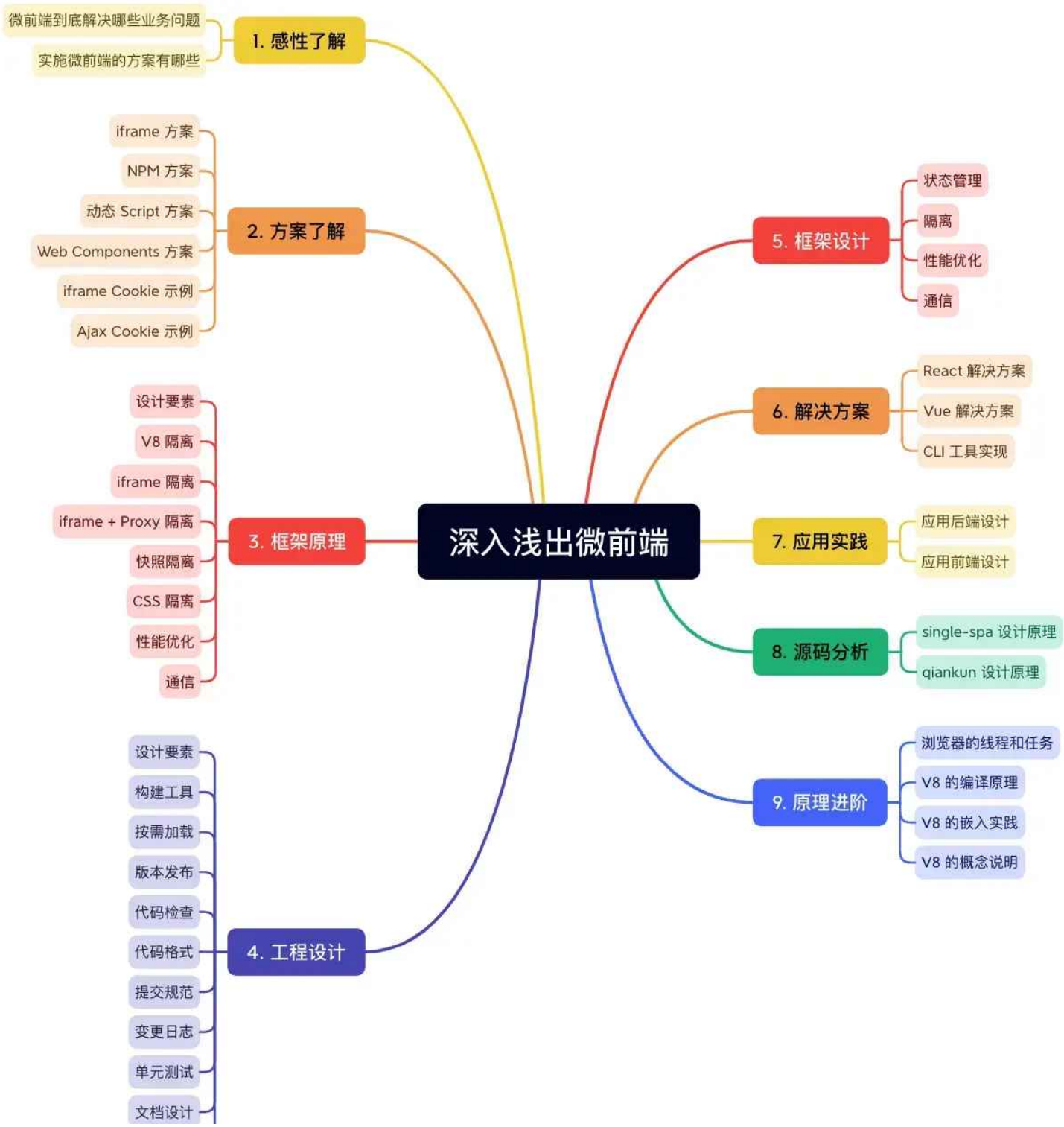
微前端学习指南



感性了解：微前端到底解决了哪些业务问题

- 什么是微前端？
- 微前端具备哪些特性？
- 在哪些业务场景中推荐（不推荐）使用微前端？

- 微前端中 MPA 和 SPA 方案各自有什么优缺点？

方案了解：iframe 方案

- 简述浏览器多进程架构的优点？
- Chrome 浏览器中存在哪些进程？
- 浏览器中的 Browser 主进程和 Renderer 进程有什么作用？
- 从新开一个标签页开始，讲述一下网页生成的整个过程？
- 简述浏览器的沙箱设计？
- Chrome 浏览器的插件进程是否已经沙箱化？
- 什么是浏览器的同源策略？它有什么作用？
- 多个跨站的 Web 应用处于同一个 Renderer 进程会有什么安全风险？
- 什么是站点隔离？它有什么作用？
- 什么情况下浏览器会给标签页（iframe）分配不同的 Renderer 进程？
- 为什么浏览器设计成站点隔离，而不是同源策略的源隔离？
- 跨站和跨域有什么区别？跨域的应用一定跨站吗？
- iframe 和浏览上下文存在什么关系？
- 在微前端中如何识别微应用是在 iframe 中打开？
- 微前端中的 iframe 方案有什么优点和缺点？
- 在浏览器中绕过同源策略限制的解决方案有哪些？是否存在安全风险？

方案了解：NPM 方案

- 为什么 JavaScript 需要模块化规范？
- 为什么在浏览器中使用 ES Module 规范开发需要通过 HTTP 请求的形式？
- 在浏览器中使用 ES Module 规范进行开发的优势有哪些？
- 浏览器中如何设置 ES Module 引入路径的别名？
- CommonJS 模块和 ES Module 模块有什么区别？
- Vite 开发态热更新速度快的主要原因是什么？
- 为什么需要在 Web 应用的开发中使用 Webpack、Vite、ESBuild 等工具？
- 大部分浏览器兼容的最高 ECMAScript 标准是多少？
- 为什么需要对业务组件进行产物构建发布？如果不构建直接发布源码会有什么问题吗？
- 业务组件和 Web 应用的产物构建存在哪些差异？
- 为什么大部分应用使用 babel-loader 进行转义时推荐屏蔽 node_modules 目录？

- 在什么情况下对应用和业务组件进行构建时需要使用 Polyfill?
- 使用 Rollup 和 Webpack 等构建输出的 ES Module 模块和 ES6+ 源码存在什么区别?
- 使用构建工具构建 ES Module 模块的好处是什么?
- 微前端中的微应用和业务组件的构建存在什么差异?
- 微前端 NPM 方案有什么优点和缺点?
- 微前端 NPM 方案适用于哪些业务场景?
- 如何解决在主应用中实时调试微应用 NPM 包的源码问题?

方案了解：动态 Script 方案

- 微前端中的动态 Script 方案如何实现?
- 在动态 Script 方案中如何实现 CSS 样式的激活和失效?
- 通过 `script` 标签动态加载 JS 文件后删除 `script` 标签，JS 文件的代码还会执行吗?
- 针对上述问题，如果删除 JS 文件后代码不执行，会有什么副作用?
- 删除 JS 文件和删除后再次添加有什么区别?
- 通过 `link` 标签动态加载 CSS 文件后删除 `link` 标签，CSS 文件对应的样式还会生效吗?
- 微前端动态 Script 方案相对于 NPM 方案有哪些优势?
- 微前端动态 Script 方案存在哪些问题?

方案了解：Web Components 方案

- 什么是 Web Components?
- Web Components 的浏览器兼容情况？如何处理浏览器的兼容性问题?
- Web Components 的生命周期回调函数有哪些?
- 微前端中的 Web Components 方案如何实现?
- Web Components 相对于动态 Script 方案有哪些优势?

方案了解：Cookie 处理

- 实现微前端中子应用免登的方案有哪些?
- 跨域情况下 LocalStorage 和 Cookie 能否共享?
- 主子应用同域的情况下使用 Cookie 可能存在什么潜在风险?
- 如何实现本地 IP 地址和自定义域名的关系映射?
- 本地系统的 hosts 文件匹配和 DNS 域名解析的差异是什么?
- 同一个 IP 地址可以映射不同的域名吗?

- 主应用和 iframe 子应用跨域同站的情况下能否共享 Cookie?
- Cookie 的 SameSite 属性有什么作用?
- Chrome 浏览器将 SameSite 默认值设置为 Lax 是为了解决什么问题?
- Chrome 浏览器中主应用和 iframe 子应用跨站时，iframe 子应用默认能够携带 Cookie 吗?
- Chrome 浏览器中 iframe 子应用跨站时，如何才能携带 Cookie?
- 为本地的 Node 服务建立 HTTPS 连接的方式有哪些?
- 主应用和 iframe 子应用跨站的情况下能否共享 Cookie? 为什么呢?
- 主子应用如何做到不同的服务地址前端应用不跨域?
- Web 前端应用跨站发送 Ajax 请求时能否携带 Cookie? 如何才能携带 Cookie?

原理解析：引言

- 简述在地址栏中输入 URL 地址后浏览器的整个运行过程?
- 浏览器中的网络服务是一个独立的进程还是线程?
- 标签页应用的生命周期状态和事件有哪些?
- 浏览器中如何做到应用内的 DOM（CSS） 和 JS 隔离?
- 站点隔离和跨源隔离的区别是什么?
- 浏览器内部提升加载性能的功能有哪些?
- 简单描述一下标签页应用的多级缓存设计以及缓存的触发顺序?
- 一个通用的微前端框架大致需要包含哪些设计要素?

原理解析：V8 隔离

- 在同一个 SPA 应用中动态加载两个不同的 Script 时为什么会产生同名属性冲突?
- 如何解决动态 Script 中的同名属性冲突问题?
- 不同的标签页应用、标签页和 iframe 应用为什么可以隔离 JS 运行时?
- V8 是如何做到 JS 运行时隔离的?
- V8 中的 Context 和 Isolate 有什么作用?
- V8 中为什么需要 Context?
- 什么是全局执行上下文栈？它有什么作用?
- V8 中的 Isolate 和 Context 隔离有什么差异?
- 浏览器的 Web 应用中有可以实现 Isolate 或者 Context 隔离的 Web API 吗?
- 浏览器的 Web 应用中隔离 JS 运行时的功能有哪些?

原理解析：iframe 隔离

- 什么是 iframe 隔离？
- iframe 隔离和 iframe 方案有什么区别？
- 使用 iframe 隔离存在哪些优势？
- iframe 的 `src` 设置成 `about:blank` 后会和主应用形成跨域吗？
- 将 iframe 的 `src` 设置成 `about:blank` 后内部的 JS 运行时环境会存在什么限制吗？
- Vue 或者 React 框架的 Hash 路由和 History 路由有什么差异？
- 如何生成一个和主应用完全同域的 iframe 页面？
- iframe 隔离相对于 iframe 方案存在哪些优势？
- 如何解决 iframe 隔离中的白屏问题？
- 如何解决 iframe 和主应用的 URL 同步问题？
- 如何解决 iframe 中的模态框无法相对于主应用进行居中的问题？
- iframe 和主应用同域后有哪些优势和注意事项？

原理解析：iframe + Proxy 隔离

- 同域的 iframe 隔离和 `about:blank` 的 iframe 隔离相比有什么优点和缺点？
- 如果想要解决同源 iframe 隔离中需要网络请求和服务端网关的问题，有没有什么解决方案呢？
- 如何避免主子应用的路由产生冲突？
- iframe 的 `src` 设置成 `about:blank` 后如何解决 History API 报错的问题？
- 为什么 iframe 的 `src` 设置成 `about:blank` 后 History API 会报错？
- Proxy 常用的拦截操作有哪些？
- 模块作用域中声明的 `var` 变量是全局变量还是局部变量？
- 在函数作用域内声明的变量 `var` 是全局变量还是局部变量？是否有方式可以将 `var` 声明的变量通过 Proxy 进行拦截处理？
- `withProxy` 可以阻止作用域链查找吗？如何才能做到？
- 代理后的 `window` 对象执行 `alert`、`addEventListener` 等原生函数会存在什么问题？如何解决该问题？
- 执行 `bind` 操作有什么副作用吗？

原理解析：快照隔离

- 快照隔离的思路是什么？

- 自执行匿名函数有什么特点?
- 将微应用放在自执行的匿名函数中运行可以解决隔离的哪些问题?又会产生什么问题?
- 通过 HTTP 请求获取的 JS 文本字符串有哪些手动执行的方式?
- `eval` 和 `Function` 存在哪些差异?
- 如何使 `eval` 在全局作用域生效?
- 简单描述快照隔离的实现过程?
- 快照隔离的限制有哪些?

原理解析:CSS 隔离

- 主应用和微应用在同一个 DOM 上下文时,实现 CSS 样式隔离的思路有哪些?
- 如果一个时刻只运行一个微应用,实现 CSS 样式隔离的方式有哪些?
- 通过动态删除微应用的 Style 标签来减少样式影响面有什么限制的使用场景吗?
- 在 Shadow DOM 中隔离 CSS 的优势是什么?有什么缺点吗?
- 在 Shadow DOM 中运行低版本的 React 框架为什么会导致 React 事件失效?

原理解析:通信原理

- 观察者模式和发布 / 订阅模式的区别是什么?
- 列举浏览器中常见的观察者模式设计的通信案例?
- 列举常见的发布 / 订阅模式的通信案例?
- MVC、MVP 和 MVVM 中会使用哪种模式进行通信?
- Vue.js 的响应式设计中使用了哪种模式进行设计?
- 简单描述一下观察者模式的概念?
- 简单描述一下发布 / 订阅模式的概念?
- 在微前端中应该使用哪种模式进行通信?
- `postMessage` 属于观察者模式还是发布 / 订阅模式?

原理解析:性能优化

- 简述 HTTP 缓存的作用
- `Expires` 和 `Cache-Control: max-age=N` 有什么区别?
- `Cache-Control: no-cache` 和 `Cache-Control: no-store` 有什么区别?
- `Cache-Control: max-age=N` 和 `Cache-Control: s-maxage=N` 有什么区别?
- `Expires` 在什么情况下缓存可能会计算错误?

- `Last-Modified` 和 `Etag` 有什么区别?
- 使用 `Last-Modified` 进行缓存可能会有哪些问题?
- 使用 `Etag` 进行缓存有什么缺点吗?
- 什么是 Resource Hints?
- 什么是 Early Hints?
- SPA 的微前端应用中能使用 Resource Hints 中的 Prerender 功能吗?
- 什么是导航预加载，如何实现导航预加载?
- Prefetch 缓存能被重复命中吗?
- 简述浏览器一帧（Frame）的执行过程?
- 浏览器自带 Prefetch 的空闲时间 和 `requestIdleCallback` 空闲时间有什么区别吗?
- `requestAnimationFrame` 和 `requestIdleCallback` 分别是在 Frame 的什么阶段执行?
- 递归调用 `requestAnimationFrame` 会阻塞当前 Frame 的页面渲染吗?
- 当前并行执行的微任务会阻塞 `requestAnimationFrame` 的执行吗?
- `requestIdleCallback` 回调中的 `requestIdleCallback` 是在当前帧还是下一帧执行?
- `requestAnimationFrame` 和 `requestIdleCallback` 回调中的微任务会在什么时机执行?
- `requestIdleCallback` 的最大执行时间为什么约为 50ms? 超过该时间会有什么影响吗?
- 在 `requestIdleCallback` 的回调中执行任务时有哪些注意事项?
- 在 `requestAnimationFrame` 和 `requestIdleCallback` 中更改 DOM 合适吗?
- 多个 `requestIdleCallback` 并行运行时浏览器是如何分配运行时间的?
- React 为什么需要任务调度? 它的时间切片任务是在 `requestIdleCallback` 中执行的吗?
- 如何通过 JavaScript 手动模拟 Resource Hints 中的 Prefetch 和 Prerender 功能?

框架解析：single-spa 的 NPM 示例

- 什么是 single-spa，它的作用是什么?
- 简述一下 single-spa 的运行机制?
- 在主应用中加载微应用的方式有哪些?
- single-spa 的注册 API 有哪些参数? 这些参数的作用是什么?
- 在 single-spa 中微应用可以分为哪几种类型? 各自的作用是什么?
- 在什么情况下需要使用 Parcel? 什么情况下不推荐使用 Parcel?
- 简述在 single-spa 中激活和失活微应用的执行流程?
- 在 single-spa 中首次激活激活微应用和再次激活微应用的执行流程有什么差异?

- 在 single-spa 的微应用中可以提供哪些生命周期函数？
- 如果同时存在需要激活和失活的微应用，那么会优先执行失活微应用的 `unmount` 函数吗？
- 什么情况下会执行 `unload` 生命周期函数，执行 `unload` 的作用是什么？
- single-spa 微应用和普通应用的入口设计存在什么差异？
- 为什么普通应用不需要卸载应用的 DOM 处理？例如执行 Vue 实例的 `unmount`？
- 为什么 single-spa 微应用需要提供生命周期函数？如果不提供会有什么问题吗？
- 如何构建微应用可以使得主应用在引入微应用时能够识别对应的生命周期函数？
- 通过 NPM、动态 Script 以及 Fetch 请求加载微应用时，哪些加载方式可以支持 chunk 分离？
- 什么是 Monorepo，它的作用是什么？什么是 Lerna，它有哪些功能？
- 是否可以在 Lerna 中结合 Git 的 submodule 实现 single-spa 的源码引入调试？
- 如何结合 React 的路由实现 single-spa 的 NPM 微前端示例？
- 如何实现 single-spa NPM 微前端示例的按需加载微应用能力？
- Webpack 的 `import()` 动态加载模块时能够使用完全动态的表达式吗？例如 `import(变量)`？
- Webpack 的 `import()` 本质上是利用什么技术实现动态加载的？
- single-spa 的 NPM 方案中需要如何构建微应用？它的构建和库构建存在什么差异？
- Web 应用和依赖的组件库都使用了 Lodash，如何使得 Web 应用只打包一次 Lodash？
- 组件库的构建需要排除所有的 NPM 包依赖吗？

框架解析：single-spa 的 Script 示例

- **方案了解：动态 Script 方案**从通用性角度出发存在哪些设计缺陷？
- 在**方案了解：动态 Script 方案**中主应用如何获取微应用的生命周期函数？该获取方式会存在哪些问题？如何实现微应用的动态增加和删除？
- 使用 single-spa 设计动态 Script 方案存在哪些优势？
- 在设计动态 Script 方案时，如果微应用需要遵循 single-spa 的生命周期函数命名规范，有哪些通用的设计方案可以使得主应用获取微应用的生命周期函数？这些设计方案各自有什么优缺点？
- 主应用和 `<script>` 脚本加载的微应用如何实现通信？
- 主应用和 `<script>` 脚本加载的微应用可以通过 `LocalStorage` 实现通信吗？`LocalStorage` 能够存储微应用的生命周期函数吗？
- 如何对全局属性进行创建之前后的防冲突处理？
- 除了使用 `<script>` 的 `onload` 事件，主应用有什么方式可以更快的感知到微应用的脚本执行情况？

框架解析：single-spa 的 Fetch 示例

- 在 qiankun 中配置 Webpack 的 `output.library` 和 `output.libraryTarget` 有什么作用？
- 在 Webpack 中 `output.library` 配置的作用是什么？
- 简述构建 `output.library` 之后 Webpack 的 ES Module 导出的运行时原理？
- 在 Webpack 构建后的运行时代码中 `webpack_modules`、`webpack_require` 和 `webpack_exports` 各自的作用是什么？
- Webpack 构建后的代码是如何建立模块化作用域的？
- Webpack 开发态构建后源代码模块对应的构建代码为什么在 `eval` 中执行？
- Chrome DevTools 如何识别出 `eval` 中的代码并实现调试能力？
- Webpack 如何配置才能实现和源码一致的调试能力？
- Webpack 的 `devtool` 有哪些配置选项，这些选项有什么特性？
- 构建后生成的 `.map` 后缀文件有什么作用？
- 在生产模式下建议生成 `.map` 文件吗？为什么？
- 在 Webpack 中 `output.libraryTarget` 配置的作用是什么？
- 什么是 UMD？它有什么作用？
- Webpack 构建时可以引入 UMD 规范的模块吗？它是通过什么方式来识别该模块的？
- 如何按属性的添加顺序遍历 `window` 对象的属性？
- 为什么配置 `output.libraryTarget` 后可以实现更加解耦的微应用生命周期函数获取？
- `Object.keys` 和 `for...in` 的区别是什么？
- 在 Fetch 示例中获取微应用的 JS 文本后，除了使用 `eval` 还有哪些可行的执行脚本方案？
- 在 qiankun 中为什么使用 `eval` 而不是 `<script>` 执行脚本？
- 如何确保微应用配置 `output.library` 时的唯一性？
- 使用 Fetch 请求微应用的 JS 资源时需要考虑跨域吗？

框架解析：single-spa 的 Code Splitting 示例

- 什么是代码分割？它的作用是什么？
- 常见的代码分割包含哪几种情况？各自有什么使用场景？
- 有什么可视化工具可以指导我们对构建产物进行分析和构建优化？
- 使用服务端 Node 托管静态资源时，重启 Node 服务会对资源的 HTTP 缓存产生影响吗？

- 多应用多入口时可以使用 Webpack 的哪些配置抽离复用代码？如何实现自定义分割？
- 多应用多入口情况下分离可复用代码有哪些优势？再次构建又会有什么优势？
- 如何实现 Vue 或者 React 框架的路由按需加载？
- 使用 `import()` 动态导入进行代码分割需要进行 Webpack 配置吗？
- 在 Webpack 的运行时原理中为什么需要对模块进行缓存？如果模块被多次导入缓存能起到什么作用？
- 使用 `import()` 动态导入进行代码构建后最终是通过什么方式加载分割的脚本的？
- 在 Webpack 运行时中如果使用 `import()` 动态导入后的分割代码加载失败，会进行加载重试吗？
- 在 Webpack 运行时中如何计算分割脚本的 `publicPath`？该配置是否可以指定？
- 在 Webpack 运行时中主文件的模块化映射对象如何添加异步分割文件的模块化映射对象？
- 在 Webpack 运行时中主文件和异步分割文件是如何实现通信的？
- 在 Webpack 运行时中如果加载分割脚本超时，会进行超时重试吗？
- 简述使用 `import()` 动态导入进行代码分割的 Webpack 运行时原理？
- Webpack 的 `output.chunkLoadingGlobal` 配置在代码分割中有什么作用？
- 按需加载的本质是什么？是通过 `fetch` 或者 `<script>` 进行动态加载？
- 在微前端中加载具备代码分割的微应用时，有哪些注意事项？
- 微应用中代码分割的脚本是否需要按顺序执行？
- 在微前端中微应用使用 `import()` 进行加载时，有哪些注意事项？
- 在微前端中如果两个微应用配置了相同的 `output.chunkLoadingGlobal` 会有什么问题？
- 如何确保微应用的 `output.chunkLoadingGlobal` 配置的唯一性？

工程设计：构建工具

- Web 应用的构建需要具备哪些构建特性？
- 在 Webapck 中进行语法转译可以使用什么工具？
- Web 应用在进行打包时为什么希望忽略 `node_modules` 中的库包转译？
- Webpack 构建的 JavaScript 脚本具有什么特性？
- Webpack 进行代码分离的作用是什么？
- Webpack 默认能对 ES6+ 进行 ES5 编译吗？
- 库的设计为什么需要构建工具？如果不构建会有什么影响？
- 库的设计在进行构建时需要具备哪些特性？

- 如何设计一个库可以支持应用在构建时使用 Tree Shaking 特性?
- 构建工具的类型有哪些?
- 列举常用的打包工具和转译工具?
- 如果使用打包工具设计库构建，需要考虑哪些构建配置处理，对比 Web 应用的构建存在什么差异?
- 打包工具和转译工具有什么差异?
- 组件库和工具库的构建存在什么差异?
- 使用转译工具设计库包的按需加载有哪些优势?
- 支持 TypeScript 转译的工具有哪些? 各自有什么特点?

工程设计：按需加载

- 为什么在库的设计中需要考虑按需加载?
- 哪些类型的库不需要考虑实现按需加载?
- 哪些打包工具可以识别 `package.json` 中的 `module` 字段呢?
- 如何使 Node 脚本支持 TypeScript 的语法设计? 此时它的 `tsconfig.json` 如何配置呢?
- `gulp-typescript` 具备类型检查的能力吗? 在构建时能够自动产出声明文件吗?
- 如何向 `package.json` 的 script 中的 node 脚本设置环境变量或者传递参数?
- 如何实现平铺构建产物? 需要考虑哪些内容的平铺处理?

工程设计：版本发布

- `package.json` 中的字段有哪些? 各自有什么作用?
- 描述一下发布 NPM 库包的完整流程?
- 如何查询库包已经发布的最新版本? 如何进行本地版本和发布版本的比较?
- 在多人协作的过程中如何对库包的发布进行管控从而提高发布的稳定性?
- 在 Github 中如何设置一个受保护的分支? 使其不能被 push 或者强制 push 代码到远程?
- 如何设计一个 Node 脚本来判断本地和远程的 master 分支代码一致?
- semver 2.0 中的三位数版本号中 X.Y.Z 各自代表什么含义?
- 什么情况下需要发布 X、Y 和 Z 版本号?
- `package-lock.json` 在什么情况下会生成?
- 库包中的依赖类型有哪些? 生产依赖和开发依赖相对于发布的库包有什么区别?
- semver 通配符中常见的 `^1.0.0`、`~1.0.0` 是什么含义?
- `npm i` 安装库包时能够自动升级库包的 X 版本号吗?

- `package-lock.json` 的作用是什么，使用它有什么好处？
- `package-lock.json` 建议被提交到远程仓库吗？
- `npm i` 和 `npm ci` 的区别是什么？
- 在有 `package-lock.json` 的情况下如何对单独的库包进行版本升级？

工程设计：代码检查

- 什么是代码检查？为什么需要在 JavaScript 中进行代码检查？
- 代码检查的工具有哪些？
- ESLint 和 TSLint 有什么区别？
- TypeScript 为什么要全面从 TSLint 转向 ESLint？
- ESLint 中的配置文件类型有哪些？各自的优先级是什么？
- ESLint 中的层叠配置有什么作用？能否举个可实践的例子？
- ESLint 的自定义解析器有什么作用？
- ESLint 的插件有什么作用？如何基于插件进行规则设置？
- ESLint 的日志风格有哪些？
- ESLint 的共享配置有什么作用？能否举一个最佳实践的例子？
- ESLint 的共享配置能否配合 ESLint 插件使用？
- ESLint 中的解析器、插件和共享配置的差异是什么？
- 如果让你设计一套代码风格指南，并配合 ESLint 工具进行约束，你会如何设计？
- 如何在写代码时实时获取 ESLint 校验提示信息？有哪些可行的方案？
- 在 VS Code 的 ESLint 插件中如果有校验错误信息，VS Code 会如何表现出来？
- 如何实现 VS Code 的 ESLint 插件的自动保存格式化功能？
- VS Code 的配置有哪两种类型？如何在团队内共享 VS Code 的 ESLint 配置？
- 如何确保构建的代码没有 ESLint 错误？

工程设计：代码格式

- 为什么需要使用 Prettier 进行代码格式化？
- ESLint 和 Prettier 有什么区别？
- ESLint 代码检查的规则可以分为哪两种类型？
- ESLint 中哪种类型的规则可以通过格式化一键修复？
- ESLint 和 Prettier 组合使用时如果有规则冲突，应该如何处理？

- 如何检查项目中的 ESLint 和 Prettier 是否存在规则冲突?
- Prettier 在 VS Code 中会有提示信息吗?
- 如何实现 Prettier 插件的保存自动格式化功能?
- Prettier 在库构建时建议一键格式化处理吗?

工程设计：提交规范

- 什么是 Git 钩子?
- 如何才能跳过本地的 Git 钩子执行?
- 什么是 Shebang? 它有什么作用?
- 在 Git 钩子中可以使用 Node 脚本进行设计吗?
- Git 钩子的类型有哪些? Git 钩子有什么作用?
- 团队协作中在本地设计 Git 钩子会有什么问题?
- 社区用于增强 Git 钩子的工具有哪些? 为什么需要设计这些增强工具?
- 如何设计一个 Git 钩子可以在检测 ESLint 失败后放弃提交代码?
- Git 钩子的默认生效目录 `.git/hooks` 可以被更改吗?
- 如何设计一个可共享 Git 钩子目录的 Git 项目?
- NPM 的 Life Cycle Scripts（Script 钩子）有哪些? 各自有什么作用?
- 如何将 NPM 包封住成 CLI 工具? 如何基于 Node 设计 CLI 工具?
- npm 和 npx 有什么区别?
- 简述 husky 的工作原理?
- 如何共享一个 Git 钩子，可以使得 ESLint 校验失败后放弃代码提交?
- 简述 lint-staged 的工作原理?
- lint-staged 检测是暂存区所在的文件还是暂存区中变化的代码?
- 如何使用 Node 脚本获取 Git 暂存区的文件列表?
- 相对于普通的 ESLint 校验，使用 lint-staged 的好处是什么?

工程设计：变更日志

- 什么是 Git 的提交说明，它有什么作用?
- 提交说明和变更日志有什么关联关系?
- 什么是 Angular 规范的提交说明? 它有什么作用?
- Angular 规范的提交说明包含哪几个部分?

- Angualr 规范的 Header 中包含哪几个字段信息?
- Angular 规范中 Header 的提交类型 type 有哪些？哪些类型必须放入变更日志?
- Angular 规范中 Header 的 scope 可以根据哪些维度进行范围划分?
- Angular 规范中 Header 的 subject 书写有哪些注意事项?
- Angular 规范中 Body 是必须说明的吗？Body 的书写有哪些注意事项?
- Angular 规范中 Footer 的作用是什么?
- 如何使用 commitizen 才能生成符合 Angular 规范的提交说明?
- commitizen 为什么需要适配器?
- cz-conventional-changelog 适配器有什么作用?
- commitizen 可以同时使用多个适配器吗?
- 简述 commitizen 的工作原理?
- git 命令中 `--allow-empty`、`--amend`、`--retry` 参数的作用是什么?
- 简述 cz-conventional-changelog 适配器的工作原理?
- cz-conventional-changelog 中 Angular 规范的 type 可以定制吗?
- 结合 commitizen、Webpack 以及 Vue CLI 描述一下通用插件的设计思路?
- 如何对 Angular 规范的提交说明进行汉化处理?
- 如何在团队开发的过程中使大家都可以强制遵循 Angular 规范的提交说明?
- commitlint 的作用是什么?
- 使用 commitlint 校验的好处是什么?
- 开发者有什么方式可以绕过 commitlint 校验?
- 如何自动生成变更日志?
- 在 Git 中如果使用相同的提交说明重复提交，有没有什么方式可以合并提交说明?
- 合并提交说明的好处是什么?
- 变更日志的作用是什么?

工程设计：单元测试

- 为什么需要单元测试？它的作用有哪些?
- 除了单元测试，你还知道哪些和前端息息相关的测试类型?
- 单元测试和 e2e 测试的区别是什么?
- 你所了解的测试框架有哪些?
- Jest 测试框架有哪些特性?

- 如何设计插入排序算法的测试用例?
- TDD 和 BDD 的差异是什么?
- 测试套件、测试用例和断言各自有什么作用?
- Jest 中断言函数提供的匹配器有哪些?
- Jest 有哪些支持 TypeScrip 开发的配置类型?各自有什么优缺点?
- 测试目录的放置风格有哪些?各自有什么优缺点?
- 在 TypeScript 和 Jest 中如何配置路径映射?路径映射的好处是什么?
- 在 Jest 中如何确保大部分源码可以被测试覆盖?
- 在 Jest 中测试覆盖率的覆盖类型有哪些?
- 在 ESLint 中如何支持校验 Jest 测试代码?
- VS Code 插件 Jest 有哪些功能?
- 如何使得源代码可以实时展示测试覆盖率百分比?
- 如何实时知道源代码中哪些代码没有被测试覆盖?
- 如何使得测试代码实现保存自动执行测试用例?
- 如何防止含有失败测试用例的源代码被提交到远程仓库?
- 在提交代码时如何实现只执行改动源码相关的测试用例?
- 如何防止含有失败测试用例的代码被发布版本?

工程设计:文档设计

- 库的文档一般需要包含哪些内容说明?
- 列举你所了解的静态网站生成器?
- 如何将编写的 Markdown 文档转换成 HTML?
- VuePress 具有哪些特点?和其它站点生成器相比有什么独特的地方?
- 如何将之前设计的变更日志快速集成到 VuePress 中?
- VuePress 支持在 Markdown 中使用 Vue 有什么作用?
- 如何对 Markdown 文档进行格式规范校验?
- 是否有必要在代码提交时对 Markdown 进行格式校验?
- 列举你所了解的注释标准?
- TSDoc 注释标准有什么特点?
- TSDoc 和 JSDoc 有什么区别?
- TSDoc 具备自己的文档生成器吗?

- TSDoc 的注释标签类型有哪些？各自有什么特点？
- 如何快速生成支持 TSDoc 标准的注释？
- 支持 TSDoc 标准的文档生成器有哪些？
- TypeDoc 文档生成器有什么特点？
- 如何将 TypeDoc 生成的文档集成到 VuePress 中？
- 如何在 Github 中托管静态网页？

工程设计：CI / CD

- 什么是 CI 和 CD，它们有什么区别？
- 使用 CI 和 CD 的好处有哪些？
- 持续交付和持续部署有什么区别？
- Web 应用和 NPM 包开发在 CI 和 CD 流程各自可执行的动作包括哪些？
- Web 应用和 NPM 包开发在 CD 流程的设计上有什么差异？
- 列举你所了解的 CI / CD 平台？
- Github Actions 有什么特点？相对于其它 CI / CD 平台的优势是什么？
- Github Actions 包含哪些组成部分？
- Github Actions 中的工作流有哪些触发事件？
- Github Actions 中的 Job 默认是并行还是串行执行？
- Github Actions 中的 Step 和 Action 可以并行执行吗？
- Github Actions 中的 Action 可以使用 Node 进行设计吗？
- 列举你认为可以封装的通用型 Action？
- CI 工作流程可以基于哪些事件进行触发？
- 在 CI 流程中使用 `npm ci` 和 `npm i` 有什么区别？
- 如何加快 CI 流程的执行速度？
- 项目在本地利用 Git 钩子检测和在远程使用 CI / CD 检测有什么区别？
- 如何实时感知 CI / CD 的执行状态？
- Github Action 失败后能否禁用 Merge Pull Request 操作？
- 在开发中可以设计哪些和 Issues 或者 Pull Request 息息相关的自动化操作？
- 在 Coveralls 上传 Jest 测试报告的好处有哪些？
- 在 CD 中进行文档部署的好处有哪些？
- 在远程的 CD 服务器上如何实现 NPM 免登？

- NPM 的 token 主要分为哪几种类型？适合 CD 流程的是哪一种 token 类型？
- 如何在本地进行 NPM 非登录态的 token 认证？
- 如何避免在 CD 流程中暴露 NPM 的 token 信息？
- 在 Github Actions 中合并分支可以使用什么事件触发工作流？
- 库包的 CI / CD 设计有哪些流程是可以重叠的？

工程设计：README 说明

- 什么是 README，它有什么作用？
- 一个完整的 README 文档应该包含哪些说明信息？
- 如何给 README 说明添加徽章？可以添加哪些徽章？
- 如何给 Github Actions 的 CI / CD 流程添加执行结果徽章？
- 如何生成单元测试覆盖率徽章？
- 如何生成版本信息、NPM 包下载量、包大小等徽章？
- 如何生成自定义徽章？

原理进阶：浏览器的线程和任务

- 简述 Chrome 浏览器多线程架构的优点？
- Renderer 进程中有哪些线程？
- 简述 Chrome 线程中的任务执行机制？
- Renderer 主线程组要负责哪些任务工作？
- 如何查看浏览器中页面的执行任务？
- Web 页面中的 JavaScript 为什么是单线程的运行模式？
- Web 页面中的 JavaScript 编译执行和 UI 渲染为什么需要互斥？

原理进阶：V8 的编译原理

- 什么是解释器？什么是编译器？两者有什么区别？
- 前端和后端编译器的作用是什么？
- 中间表示 IR 的作用是什么？
- 优化编译器如何实现对代码的编译优化？
- 什么是 JIT，它有什么优缺点？
- 解释器、传统编译器和 JIT 编译器有什么区别？
- 解释器和 JIT 编译器的混合使用有什么优势？

- 早期的 JavaScript 为什么采用解释执行的方式运行?
- JavaScript 是解释型语言还是编译型语言?
- 简述 JavaScript 的编译原理?
- V8 在编译的过程中将 JavaScript 转换成字节码有什么优势?
- 弱类型和强类型、静态类型和动态类型语言的区别?
- 一二三地址指令寄存器架构各自有什么优缺点?
- V8 在什么情况下会对 JavaScript 代码进行优化和去优化?

原理进阶：V8 的嵌入实践

- 将 C++ 源代码编译成可执行文件需要经历哪几个编译步骤?
- C++ 的编译工具有哪些?
- GNU 的 gcc 和 clang 有什么区别?
- clang 和 LLVM 有什么关联关系?
- g++ 和 gcc 有什么区别?
- C++ 中静态库和动态库有什么区别？各自有什么优缺点?
- 如何制作 V8 的静态库?
- 如何在 Mac 的命令终端中添加环境变量?
- g++ 有哪些编译参数？宏定义参数有什么作用?
- 什么是 Homebrew？它有什么作用?
- 如何下载和使用 V8 动态库?
- 如何使用 CMake 编译 V8 应用?
- V8 可以独立使用吗?

原理进阶：V8 的概念说明

- V8 除了可以在 Chrome 和 Node.js 中使用，还可以在哪些应用场景中使用?
- 小程序中应用的双线程架构的优缺点是什么?
- 列举你所知道的 JavaScript 引擎?
- 在 Andriod 和 iOS 中的 JavaScript 引擎都是 V8 吗?
- 如何在 C++ 应用中执行 JavaScript 文件?
- 什么是 Isolate，它的作用是什么?
- 什么是 HandleScope，什么是 Handle，它们的作用是什么?
- 什么是 Context，什么是全局上下文栈，它们的作用是什么?

- Isolate 隔离和 Context 隔离的区别是什么？为什么需要这两个不同的概念？
- 全局上下文栈和执行上下文栈的关联关系是什么？
- 执行上下文栈和作用域链有什么关联关系？