





前端工程化

(1) 使用Webpack实现项目构建 (2) 使用Babel完成JavaScript编译 (3) CSS预编译: 提供便捷的语法和特性供开发者编写源代码, 随后经过专门的编译工具将源码转化为CSS语法 (4) 模块化开发 (5) 组件化开发 (6) 开发环境的本地服务器与Mock服务 (7) 规范化约束 (8) 项目部署流程化

真题演练

Q. 首屏时间优化的方式

CSR(Client Side Render)

浏览器渲染是最简单, 最符合 Web 应用设计思路的渲染方式。

所谓浏览器渲染, 就是将应用所需的页面展示、前端逻辑、接口请求全都在用户的浏览器中执行。它很好的实现了前后端的解耦, 让前端开发更为独立, 也让后台实现更为简单。

同时, 为了缓解用户的等待焦虑, 我们可以用 **loading 态**, 或者**骨架屏**, 进一步提升异步请求接口时的用户体验。

SSR(Server Side Render)

但在处理首屏体验的问题上, 服务端渲染有着独到的优势。它能提前再服务端中完成页面模板的数据填充, 从而一次性返回完整的首屏内容, 从而面对 SEO 的爬取时能获取到更多有效的关键信息。

NSR(Native Side Render)

Native 渲染的本质其实还是 SSR, 只不过提供服务的 Server 转变为了客户端。由于需要用到客户端机能, 因此此种实现通常应用在移动端 APP, 或者 PWA 下。

Q.两层for循环如何优化

1. 循环次数小的放在外层
2. 中断的条件设为常量会更快
3. 直接从尾巴开始遍历

```
1. for (var i = arr.length - 1; i >= 0; i--) {}
```

2. 只读取了一次.length 也没开辟新的变量空间