

综合

1 谈谈你对重构的理解

- 网站重构：在不改变外部行为的前提下，简化结构、添加可读性，而在网站前端保持一致的行为。也就是说是在不改变UI的情况下，对网站进行优化，在扩展的同时保持一致的UI
- 对于传统的网站来说重构通常是：
 - 表格(table)布局改为 DIV+CSS
 - 使网站前端兼容于现代浏览器(针对于不合规范的 CSS、如对IE6有效的)
 - 对于移动平台的优化
 - 针对于 SEO 进行优化

2 什么样的前端代码是好的

- 高复用低耦合，这样文件小，好维护，而且好扩展。
- 具有可用性、健壮性、可靠性、宽容性等特点
- 遵循设计模式的六大原则

3 对前端工程师这个职位是怎么样理解的？它的前景会怎么样

- 前端是最贴近用户的程序员，比后端、数据库、产品经理、运营、安全都近
 - 实现界面交互
 - 提升用户体验
 - 基于NodeJS，可跨平台开发
- 前端是最贴近用户的程序员，前端的能力就是能让产品从 90分进化到 100 分，甚至更好，
- 与团队成员， UI 设计，产品经理的沟通；
- 做好的页面结构，页面重构和用户体验；

4 你觉得前端工程的价值体现在哪

- 为简化用户使用提供技术支持（交互部分）
- 为多个浏览器兼容性提供支持
- 为提高用户浏览速度（浏览器性能）提供支持
- 为跨平台或者其他基于webkit或其他渲染引擎的应用提供支持
- 为展示数据提供支持（数据接口）

5 平时如何管理你的项目

- 先期团队必须确定好全局样式（`globe.css`），编码模式(`utf-8`)等；
- 编写习惯必须一致（例如都是采用继承式的写法，单样式都写成一行）；
- 标注样式编写人，各模块都及时标注（标注关键样式调用的地方）；
- 页面进行标注（例如 页面 模块 开始和结束）；
- `CSS` 跟 `HTML` 分文件夹并行存放，命名都得统一（例如 `style.css`）；
- `JS` 分文件夹存放 命名以该 `JS` 功能为准的英文翻译。
- 图片采用整合的 `images.png png8` 格式文件使用 - 尽量整合在一起使用方便将来的管理
- 规定全局样式、公共脚本
- 严格要求代码注释(html/js/css)
- 严格要求静态资源存放路径
- `Git` 提交必须填写说明

6 组件封装

目的：为了重用，提高开发效率和代码质量 注意：低耦合，单一职责，可复用性，可维护性 常用操作

- 分析布局
- 初步开发
- 化繁为简
- 组件抽象

7 Web 前端开发的注意事项

- 特别设置 `meta` 标签 `viewport`
- 百分比布局宽度，结合 `box-sizing: border-box;`
- 使用 `rem` 作为计算单位。`rem` 只参照跟节点 `html` 的字体大小计算
- 使用 `css3` 新特性。弹性盒模型、多列布局、媒体查询等
- 多机型、多尺寸、多系统覆盖测试

8 在设计 Web APP 时，应当遵循以下几点

- 简化不重要的动画/动效/图形文字样式
- 少用手势，避免与浏览器手势冲突
- 减少页面内容，页面跳转次数，尽量在当前页面显示
- 增强 `Loading` 趣味性，增强页面主次关系

9 你怎么看待 Web App/hybrid App/Native App? (移动端前端 和 Web 前端区别?)

- Web App(HTML5): 采用HTML5生存在浏览器中的应用, 不需要下载安装
 - 优点: 开发成本低, 迭代更新容易, 不需用户升级, 跨多个平台和终端
 - 缺点: 消息推送不够及时, 支持图形和动画效果较差, 功能使用限制 (相机、GPS等)
- Hybrid App(混合开发): UI WebView, 需要下载安装
 - 优点: 接近 Native App 的体验, 部分支持离线功能
 - 缺点: 性能速度较慢, 未知的部署时间, 受限于技术尚不成熟
- Native App(原生开发): 依托于操作系统, 有很强的交互, 需要用户下载安装使用
 - 优点: 用户体验完美, 支持离线工作, 可访问本地资源 (通讯录, 相册)
 - 缺点: 开发成本高 (多系统), 开发成本高 (版本更新), 需要应用商店的审核

10 页面重构怎么操作

网站重构: 不改变UI的情况下, 对网站进行优化, 在扩展的同时保持一致的UI。

- 页面重构可以考虑的方面:
 - 升级第三方依赖
 - 使用 HTML5、CSS3、ES6 新特性
 - 加入响应式布局
 - 统一代码风格规范
 - 减少代码间的耦合
 - 压缩/合并静态资源
 - 程序的性能优化
 - 采用 CDN 来加速资源加载
 - 对于 JS DOM 的优化
 - HTTP服务器的文件缓存