前言

首先感谢大家购买并支持本课程，因为我工作经验有限，写作时难免会存在一些不好或错误的地方。若大家在阅读时发现可改善的地方请在每章下方评论提出建议，我会认真采纳并提升本课程的体验质量。

很多同学经过一个多月的学习， 前端工程化 相关技能得到很大的提升。在此表扬下有二十几位同学，他们几乎每天都私聊找我解决问题，我有时间也会尽自己一份力帮助他们。

我花了一年时间去创作本课程，总会帮助到很多同学，经过很多同学的反馈得知事实也是如此，论付出与收入，几乎无可比性，收入在付出面前不值得一提。我拿这些时间跟我爸做生意都比本课程收入多得多，但我更喜欢学习与分享，所以选择了创作本课程，与大家一起探讨 前端工程化 。

经过考虑，后续我会将本课程进一步优化并增加更多 前端工程化 相关内容，出版一本纸质书，希望通过更多途径让更多同学从零到一学习 前端工程化 。

发展：前端不再是曾经的前端

在 2020年 前，整个前端行业对 前端工程化 还是处于一个探索与完善的阶段。在互联网早期发展历史中， 后端开发者 一直挑起工程化的整个大梁， 前端开发者 顶多从事 Web开发 相关工作。相信接触过 ASP/JSP/PHP 工作的 前端开发者 能体验到当时的无奈。

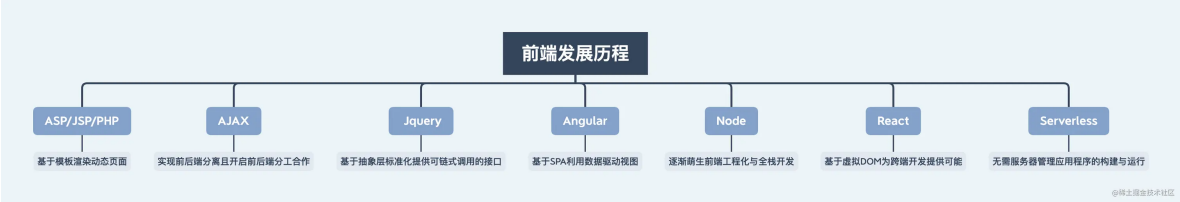
这种传统模式下的协作效率很低，即使及时发现一个 Bug ，其解决效率相比现在也是相当低下。有时前端与后端争得面红耳赤，谁也不想背锅。我作为一个从事前端超过七年的过来人，这些情况都不知遇到多少次了，当然这只是每个 前端开发者 曾经的一个缩影，真正让前端兴起是 JS 的蓬勃发展。

曾经的 JS 只是作为丰富网页效果的脚本语言，通过植入预设逻辑就能让网页生动起来，提升用户体验。 JS 自 1995年 诞生以来，我觉得有六种前端技术让前端在短时间内产生了从量变到质变的跳跃。

- **AJAX**：诞生于 2005年 ，无需刷新即可快速动态更新局部网页的Web开发技术

- **Jquery**：诞生于 2006年 ，提供简便 JS设计模式 且优化 DOM操作 、 语言增强 、 事件处理 、 动画设计 、 AJAX交互 等功能的 JS框架
- **Angular**：诞生于 2009年 ，提供 MVC 、 模块化 、 双向绑定 、 依赖注入 、 语义标签 等功能的 JS框架
- **Node**：诞生于 2009年 ，基于 Chrome V8引擎 使用 事件驱动 与 非阻塞式I/O模型 让 JS 运行在服务端的 JS运行环境
- **React**：诞生于 2013年 ，采用声明范式轻松描述应用，通过 虚拟DOM 最大限度减少与 DOM 交互的 JS框架
- **Serverless**：诞生于 2015年 ，无需服务器管理应用程序的构建与运行的概念

基于上述六种前端技术，我将前端发展历程划分为以下阶段。每种前端技术都为当前阶段提供了推进作用，将前端从一个层次推升到另一个层次。每种前端技术在出现时都可能不受重视，随着时间推移与项目实践，它可能从众多技术中突围而出。这些热门的前端技术都能形成一个庞大且热闹的生态社区，各路开发者都踊跃贡献自身一份力量，这些前端技术慢慢地就会沉淀出一条主流的技术栈或工具链，以推动 前端工程化 的发展。

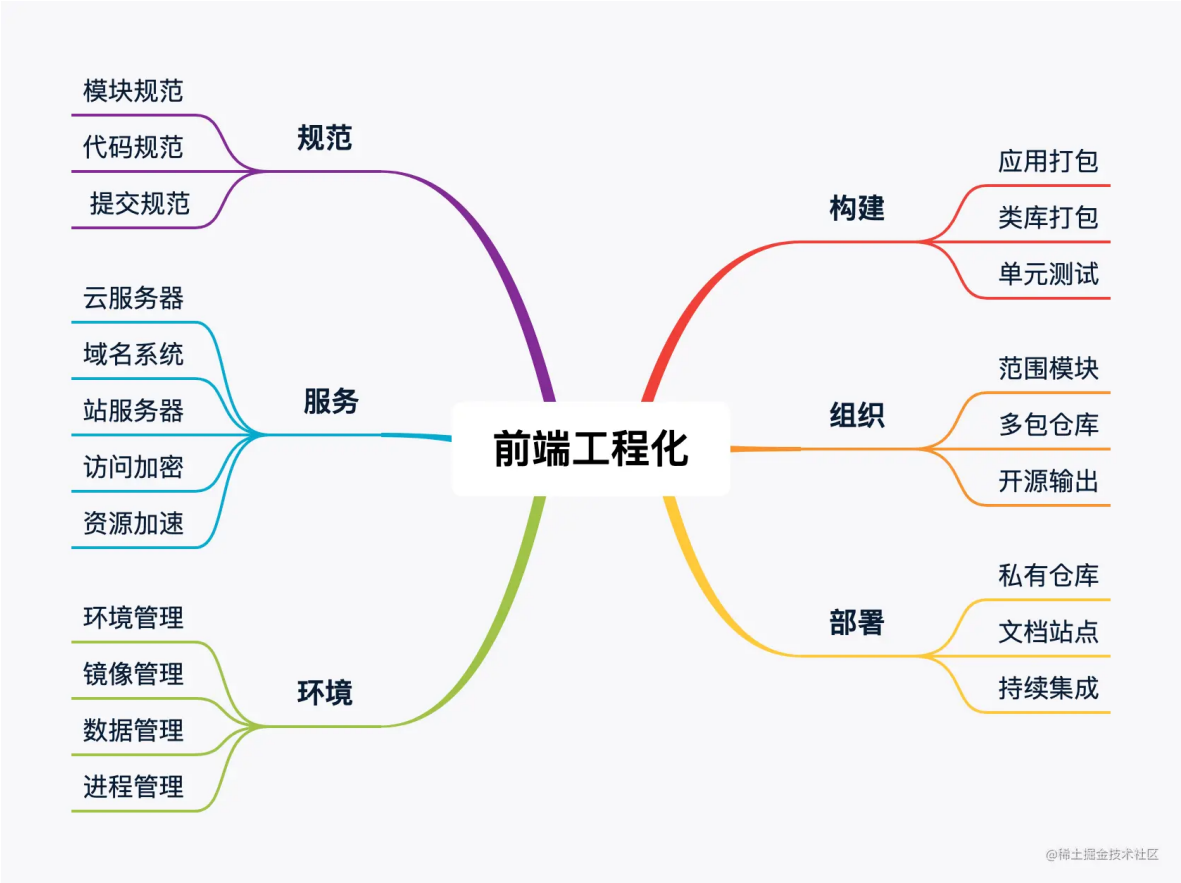


曾经前端只是简单地开发网页，在同行中一直被认为是门槛低技术烂的岗位，很多时候被其他研发岗位鄙视，一直游走于边缘。随着近几年前端技术的大力发展，再加上 Node 在不同领域中的不断尝试，使得基于 HTML/CSS/JS 能干很多事情！

回顾：大家都学会了什么

有些同学可能因为跳读导致在后续章节中发现衔接不上，因此私聊找我“兴师问罪”。在学习本课程前，我已明确本课程根据六大模块践行，因此每章都是承上启下地互相衔接着，跳读只能让你在某些知识点中产生断层效应，因此建议大家还是跟着课程章节一章一章学习下去，遇到不懂问题再私聊我也不迟。

在设计本课程时，我考虑到大众需求，因此以一个前端小白的身份切入到每章中。很多时候都会提出一些甚至连官网文档都不会提到的注意事项或细节，这些知识点都是我在平时实践中总结所得，因此在学习时要时刻注意。



在 规范篇 中，熟悉 模块/代码/提交 三大开发阶段规范，通过规范约束自己，保障工作质量与提升开发效率。

在 服务篇 中，熟悉 云服务器/域名系统/站服务器 部署服务环境，掌握整体部署与工具配置，学会独立上线应用与服务。

在 环境篇 中，熟悉 Node/Nvm/Npm 部署开发环境，独立搭建一个 接口服务 ，实践 环境/镜像/数据/进程 四种 Node 应用方式。

在 构建篇 中，熟悉 构建工具 打包类库模块，独立封装一个 类库模块 ，结合 测试用例 保障代码的生产质量。

在 组织篇 中，熟悉 Monorepo模式 管理多包仓库，独立维护一个 多包仓库 ，结合 Npm Scope 发布模块到公共仓库。

在 部署篇 中，熟悉 自动化工具 部署前端项目，独立打造一个 私有仓库 与 文档站点 ，结合 CI/CD 在提交代码时自动部署到公网。

第2到4章部署三大开发阶段规范，虽然在后续再也未提到，但在开发过程(包括 模块化编码 与 提交代码)中也一直无形地保驾护航。第5到9章部署服务器相关配置与工具，应用与服务最终都会发布的外网或内网，所以必须要有服务器的加持。第9到13

章部署 Node 相关配置与工具，通过不同方面实践 Node 的应用场景，为后续应用与服务的部署打下坚实基础。第14到22章展开性地探讨与学习 前端工程化 在日常开发中的应用，基于服务器与 Node 完成相关部署流程并最终产出以下站点。

- **主站**： https://yangzw.vip
- **接口站点**： https://api.yangzw.vip
- **文档站点**： https://doc.yangzw.vip
- **仓库站点**： https://npm.yangzw.vip
- **资源站点**： https://static.yangzw.vip

当然你有更好的想法可继续实施更多站点的开发与部署，学以致用才是最关键的。说些题外话，可能本课程设计的应用场景并不能覆盖 100% ，但大家可从中学习一些解决问题的思路与技巧，而不是让我 100% 为你解决所有问题，我相信任何一本书籍或一门课程都做不到。

授人以鱼不如授人以渔的道理相信你都懂， 套路千篇一律，唯有独自弄懂 。

总结

最后感谢一路以来大家对我的支持与认可，也感谢很多同学给予我充分的理解与体谅。码字不易，从整体的课程设计、示例规划、内容衔接和语言表达，我都花了很多时间，遇到不合理的内容我甚至会推翻重做。本课程所有代码都在以下仓库中，麻烦顺手给我一个 Star ，让我更有动力开源更好用的项目哈！

- [x] 示例项目：fe-engineering
- [x] 正式项目：bruce

在本课程完结后，我又有了以下课题的规划，大家投票看看更喜欢哪个课题，我会根据大众喜好选择适合的课题创作。

- **CSS方面**： 《玩转CSS的技术之美：CSS高级玩法》
- **Node方面**： 《Node模块实战开发通关秘籍》
- **架构方面**： 《从0到1落地前端架构设计》

在即将到来的 30岁 ，我也慢慢地从享受写作分享到认清其在我人生的重要性。可能 30岁 就是人生的一个重要转折点，希望我能接触更多方向而不是目前的 编程写作方向 。任何时刻都要做到多元化发展，这样才会有更多的进步与想法。

最后再次感谢大家对我的支持与认可，纸质书我会继续准备，第三本掘金课程我也会继续准备，期待再见！！！

> 可通过以下方式找到我

- 微信：YangzwVIP
- 公众号：IQ前端

留言

输入评论（Enter换行，Ctrl + Enter发送）

发表评论

全部评论（8）

小萌新入场 JY.4 IT 23天前

架构架构

1 回复

coder_lbw JY.4 2月前

来点高级css

2 回复

console_man JY.3 Web前端 2月前

小孩儿才做选择题，我全都要

点赞 回复

NewName LV.4 JY.5 前端CV 2月前

node吧，大佬

点赞 回复

肖麦麦 JY.4 切图崽崽 2月前

要不来个Node吧

点赞 回复

多肉小子 JY.3 2月前

为嘛很多都访问不了，是我网络问题么？
主站：yangzw.vip
接口站点：api.yangzw.vip
文档站点：doc.yangzw.vip
仓库站点：npm.yangzw.vip
资源站点：static.yangzw.vip

点赞 1

JowayYoung （作者） 2月前

不要直接访问，我这里贴上的只是二级域名，后面还要加上后缀才能访问，具体还是要看每一章部署了什么内容

点赞 回复

bigFace JY.4 3月前

三个方向，css和架构都是之前小册的加强，node是一本新的，可以先搞个node来看看

1 回复