**工程化的的定义和主要解决的问题**

* 遵循一定标准和规范通过工具来提高效率降低成本的手段
* 通过工程化提升【战斗力】
* 解决问题
* 想使用ES6+新特性，但是兼容问题
* Less/Sass/postCss增江css的编程性 但是运行环境不能直接支持
* 想使用模块化的方式提高项目的可维护性 但是运行环境不能直接支持
* 部署上线前需要手动压缩代码及资源文件 部署过程中需要手动上传代码到服务器
* 多协同开发，无法硬性同意大家的代码风格。从仓库中pull回来的代码质量无法保证

**主要解决问题**

* 传统语言或语法的弊端
* 无法使用模块化/组件化
* 重复的机械式工作
* 代码风格统一、质量保证
* 以来后端服务接口支持
* 整体依赖后端项目

**一个项目过程中工程化的表现**

* 一切以提高效率、降低成本、质量保证为目的的手段都属于【工程化】

**脚手架工具**

* 相同的组织结构
* 相同的开发范式
* 相同的模块依赖
* 相同的工具配置
* 相同的基础代码

**常用的脚手架工具**

* React项目-----create-react-app
* vue---vue-cli
* angular---angular-cli

plop:

创建一个组件、模块所需的文件

**Yeoman**

1. 全局安装yo npm install yo --global yarn global add yo
2. 安装对应的 generator npm i generator-node --global
3. 通过yo运行generator 进入项目 yo node

**使用步骤**

1. 明确需求
2. 找到合适的Generator
3. 全局范围安装找到的Generator
4. 通过yo运行对应的Generator
5. 通过命令行交互填写选项
6. 生成所需的项目结构

* 读取流
* 转换流
* 写入流

css：yarn add gulp-clean-css --dev