1. 谈谈你对工程化的初步认识，结合你之前遇到过的问题说出三个以上工程化能够解决问题或者带来的价值。

**解答：**我理解的工程化是将一些具有一定标准、规范的工作通过机器来实现。如传统工业中的工程化的体现，由人力流水线生产转变为自动化机械生产线。而放到web前端开这块则是体现为，将一些原本需要开发者手动去完成的工作通过工具自动完成。 如项目初始化阶段，需要我们构建项目的初始结构。不使用其他工具的情况下，我们一般通过拷贝以前项目的代码、或者从基准的项目拷贝，然后针对项目进行修改。而如果我们通过脚手架工具来初始化项目的话，则可以根据项目的需求，通过配置的方式来实现我们项目的初始化构建，不用再去做代码的改动。

再比如在编码阶段，多人协作时，由于每个人使用的编辑器、以及个人的编码习惯和风格都不同，导致提交的代码风格排版多种多样，这样后期代码的可读性及可维护性都比较差，不使用工具时一般通过制定详细的代码规范，并要求开发人员遵守开发规范，以及通过代码review来检验代码质量，这样花费的时间比较多，质量也得不到保证，全靠开发人员自觉。而使用工具的话我们可以通过ESLint这样的代码检测工具来保持代码风格统一，并通过git hook在代码提交之前对代码进行检测，来保证代码质量。

此外在开发阶段，我们可以通过工具来实现开发阶段的预览和热更新，而不用每次修改代码之后重新去构建项目，然后再去刷新页面查看最新的界面效果。在部署阶段，我们可以通过jenkins这样的工具来自动完成代码提交到部署的过程，不需要开发人员去手动代码，然后上传文件到服务器，大大的节省了开发人员的时间和精力。

1. 你认为脚手架除了为我们创建项目结构，还有什么更深的意义？

**解答：**我认为脚手架除了为我们创建项目结构之外，更重要的意义是为后期前端智能化编码提供基础。这个基础包含两个方面，一个是思想方面的。一个是编码方面的。 在思想方面，开发者如果都具有工程化的思想，就都会去思考如何利用工具来减少重复工作，开发者会更加注重功能的标准化，细粒度化，然后通过组合细粒度的功能来实现更加庞大，更加复杂的功能。当前端界面的风格、布局、以及业务流程的标准慢慢趋向于稳定，功能的实现粒度足够细时，我们就可以通过配置文件、命令行或页面交互的方式来完成项目及功能的初始化，如果足够智能的话，开发者在使用脚手架完成项目初始化的同时，就能完成部分功能开发。举个例子，我们公司是做项目的，项目中经常会使用到的系统管理，但各个项目中用到的模块不尽相同，这时我们在初始化项目的时候就可以通过自己开发的脚手架来完成一些模块的内置。当这样的功能模块足够多时，相对的需要手动开发的功能模块就少了。

编码方面，主要体现在插件功能的发展，插件功能越来越多，越来越完善，为开发者实现功能组合提供助力。当前插件能够帮助开发者实现代码的编译，压缩，以后就可能帮助开发者实现业务功能的整合。