Webpack相关面试题

一、谈谈你对webpack的理解？

答：webpack是js应用程序的静态模块打包工具，在处理应用程序的时候，内部会构建一个依赖图对应映射到项目所需要的各个模块

二、webpack的工作流程了解吗？

1. 初始化流程
2. 初始化参数：解析webpack配置参数，合并shell传入的和webpack.config.js文件配置的参数，形成最后的配置结果，各个配置项拷贝到 options 对象中
3. 用上一步得到的参数初始化Compiler编译对象，这个对象为webpack声明周期，不执行具体的任务，只是一些调度工作
4. 加载所有配置的插件plugin，监听webpack构建生命周期的事件节点做出相应的反应
5. 编译构建流程
6. 执行Compiler对象的run方法开始执行编译即make方法
7. 根据配置中的entry找出入口文件从入口文件出发开始解析文件即执行before-resolve和build module，构建AST语法树调用所有配置的Loader对模块进行编译（即执行normal-module-loader以及programe）再找出该模块依赖的模块再递归本步骤直到所有入口依赖文件都经过这一步处理
8. 根据入口和模块之间的依赖关系组装成一个个包含多个模块的Chunk再把每个 Chunk 转换成一个单独的文件加入到输出列表，执行seal方法并开始优化操作，在确定好输出内容后根据配置确定输出的路径和文件名把文件内容写到文件系统（即执行dist）
9. Webpack 会在特定的时间点广播出特定的事件，插件在监听到感兴趣的事件后会执行特定的逻辑，并且插件可以调用Webpack 提供的 API 改变 Webpack 的运行结果
10. webpack编译构建流程即Complier -> run方法主要流程
11. compile开始编译：构建Compilation对象执行模块创建、依赖收集、分块、打包等等
12. 然后根据入口文件分析模块和依赖的模块来创建这些模块对象
13. 根据入口依赖使用工厂函数创建一下空的module对象
14. 执行buildModule，调用使用的loader完成模块的转换使用acorn解析转换后的内容输出AST，从配置的入口分析AST，当遇到require语句后将其加入到依赖的模块列表同时对新找出的这个模块再次分析
15. seal方法做了那些优化
16. 优化打包块，像提取公共文件的CommonsChunkPlugin就是在这个钩子函数触发时候进行
17. 执行seal的主要流程