### **1. webpack与grunt、gulp的不同？**

grunt和gulp是基于任务和流（Task、Stream）的。

对文件做一系列链式操作，更新流上的数据，整条链式操作构成了一个任务，多个任务就构成了整个web的构建流程。

webpack是基于入口的。webpack会自动地递归解析所有文件需要的所有资源文件，然后用不同的Loader来处理不同的文件，用Plugin来扩展webpack功能。

gulp和grunt需要开发者将整个前端构建过程拆分成多个任务，并控制他们的调用关系吧，webpack需要开发者找到入口，并需要清楚对于不同的资源应该使用什么Loader做何种解析和加工

webpack更适用于大型复杂的前端站点构建

### **Loader和Plugin的不同？**

**不同的作用**

* **Loader**直译为"加载器"。Webpack将一切文件视为模块，但是webpack原生是只能解析js文件，如果想将其他文件也打包的话，就会用到loader。 所以Loader的作用是让webpack拥有了加载和解析非JavaScript文件的能力。
* **Plugin**直译为"插件"。Plugin可以扩展webpack的功能，让webpack具有更多的灵活性。 在 Webpack 运行的生命周期中会广播出许多事件，Plugin 可以监听这些事件，在合适的时机通过 Webpack 提供的 API 改变输出结果。

**不同的用法**

* **Loader**在module.rules中配置，也就是说他作为模块的解析规则而存在。 类型为数组，每一项都是一个Object，里面描述了对于什么类型的文件（test），使用什么加载(loader)和使用的参数（options）
* **Plugin**在plugins中单独配置。 类型为数组，每一项是一个plugin的实例，参数都通过构造函数传入。