

devtool

区别:

- **source-map** 源码映射: 增加映射文件, 可以帮我们调试源代码, 会单独生成一个sourceMpa文件, 出错会标识报错的列和行

在控制台, 打包的日志中, 可以看到 XX.js.map 文件

优点: 大而全

- **eval-source-map** 不会产生单独的文件, 可以显示行和列
- **inline-source-map**
- **cheap-source-map** 不会产生列, 但是是一个单独的映射文件, 产生后, 可以保留起来
- **cheap-module-eval-source-map** 不会产生map文件, 但是会集成在打包后的文件中, 不会产生列

watch 实时打包

[watch-webpack官网](#)

webpack开启watch 选项, 如果每次修改文件, 都要npm run build略显繁琐, 就开始 watch 监控

webpack.config.js

```
module.exports = {
  entry: "",
  output: {},
  .... 其他配置
  watch: true,
  watchOptions: {
    aggregateTimeout: 1000, // 1秒钟
    ignored: "/node_modules/",
    poll: 1000
  }
}
```

watchOptions

- **aggregateTimeout** (毫秒, 默认300毫秒)

在你重新构建的, 打包的时候, 增加多少的延迟再去打包构建, 会有一个等待时间, 这个时间就是我们指定的, 然后在这段时间内做的任何文件更改, 都会在这个指定时间 1000 (1秒钟) 以后, 将更改打包到构建结果中

其实就是防抖

- **ignored** 忽略对哪些 【文件夹】 的监控

我们是不希望去监控 `node_modules`的

- poll

多久轮询检查一次文件的变动，poll单位是毫秒。poll: 2000 意思是，2秒去轮询查看一次，文件是否有变动，有则重新构建；这个时间越小，则更改以后构建的文件真实性就越高，但同时也比较耗CPU，因为总在轮询