

# 更多JavaScript打包工具

---

## Rollup

---

专注JS的打包，也支持其他类型，但是不如Webpack通用  
如果项目只是打包JS可以考虑使用Rollup

Rollup不需要再项目内安装，直接全局安装即可

相比Webpack，并不添加额外的代码（Webpack会有代码注入）

**tree shaking 最早由Rollup实现**

### 可选的输出格式

- cjs (CommonJS)
- amd
- esm
- iife
- umd
- system

一个库往往需要支持多种模块形式，rollup可以把一份代码打包成多份

### Rollup构建JavaScript库

Rollup经常被用于打包一些库或框架（react/vue）

- 最低限度的附加代码
- ES6 Module 良好支持
- tree shaking去除开发环境
- 自定义插件实现React一些特殊的打包逻辑

没有loader, plugin, HMR等

## Parcel

---

比Webpack Rollup都要晚很多

### 打包速度

- 利用worker来并行执行任务
- 文件系统缓存
- 资源编译处理流程优化

资源编译处理流程优化  
webpack loader只能接收字符串，需要不断地做String-AST转换

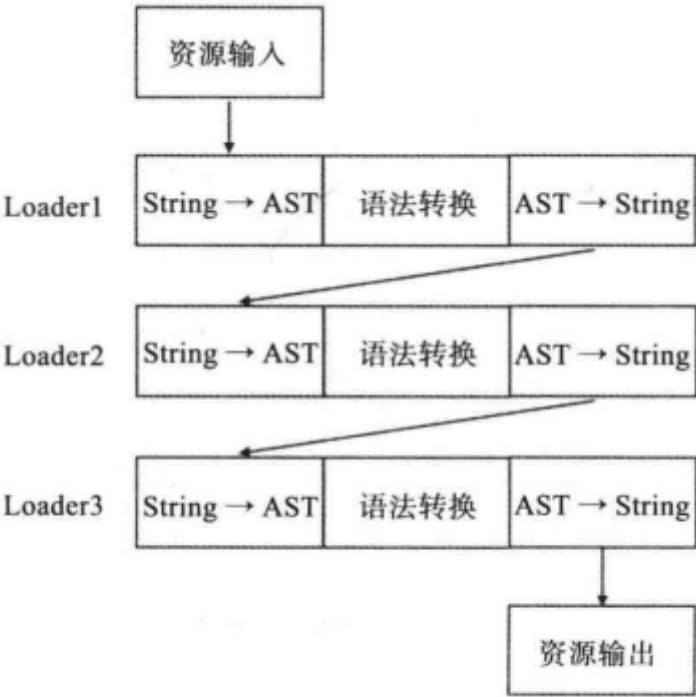


图 10-1 Webpack 多个 loader 资源处理流程

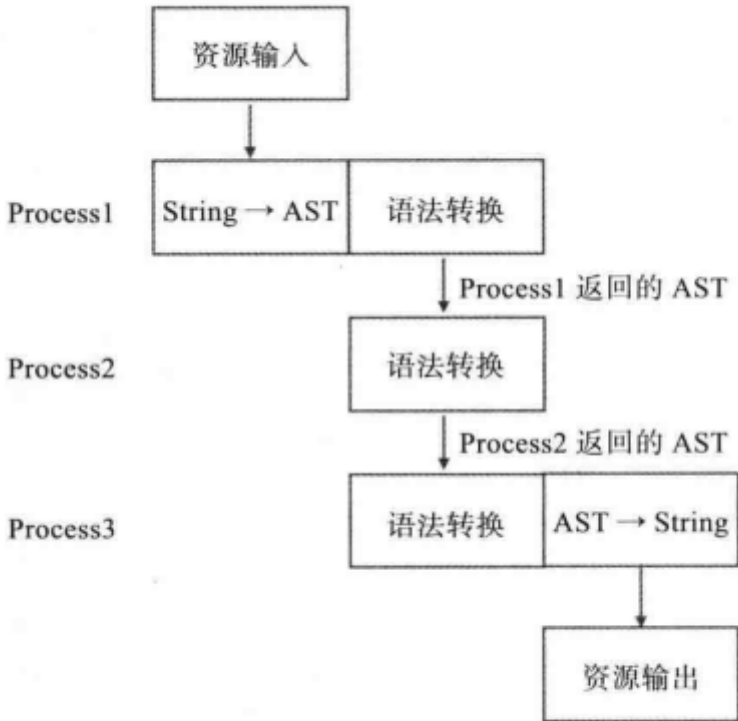


图 10-2 Parcel 资源处理流程

零配置

可以零配置，并且配置很简单  
快，灵巧，很适合小工程

# 打包工具发展趋势

---

## 性能与通用性

无法两全，得取舍

## 配置极小化与工程标准化

## WebAssembly

性能媲美原生

让c/java可以编译为WebAssembly在浏览器上运行

游戏、图像识别等计算密集型领域，WebAssembly由很广阔地前景

Webpack Rollup Parcel都支持

## 小结

---

- Rollup  
专注JS打包，附加代码少，tree shaking，输出多种形式地模块
- Parcel，资源处理流程优化，更快打包，零配置