

用户使用手册

1. 介绍

这是一个我们自己实现的简单 CoffeeScript 编译器,能够完成部分 coffee 文件的编译工作:coffeescript → javascript。使用了外部开源工具 Jison。

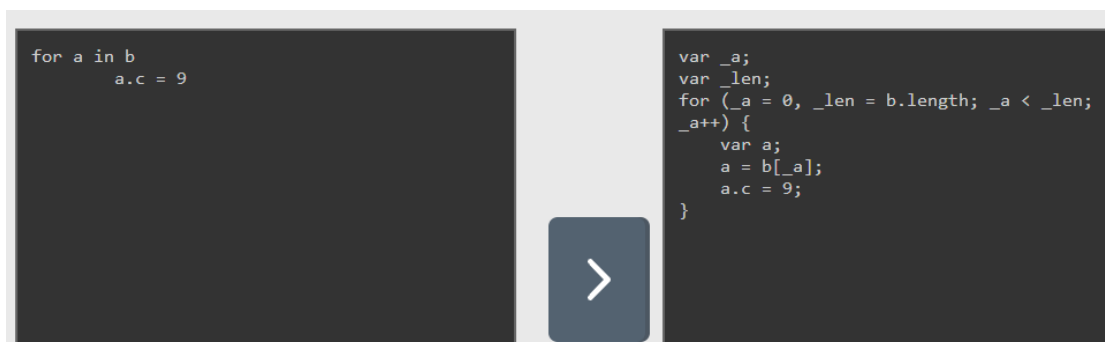
根目录中 parser.jison 是我们的核心文件,可以通过 Jison 工具将 parser.jison 编译成我们期待的 parser.js 文件,其中的 parser 对象负责对输入的 coffeescript 代码编译。

```
var parser = (function(){
var parser = {trace: function trace() { },
yy: {},
symbols_: {"error":2,"coffee":3,"S":4,"EOF":5,"Block":6,
terminals_: {2:"error",5:"EOF",18:"=",20:"VARIABLE",21:"
productions_: [0,[3,2],[4,2],[4,0],[6,1],[6,1],[6,1],[6,
performAction: function anonymous(yytext, yyleng, yyline
/* this == yyval */
```

2. 使用

您可以在 windows 上直接看 coffeescript 的编译结果:

打开 example/index.html, 在左边的 textarea 中输入代码, 点击转化按钮查看输出结果



或者, 您可以在 linux 中实时看到 coffee 文件编译成 js 文件:

首先安装 watchdog (负责检测文件的变化): `pip install watchdog`

然后进入 realTime/js 目录, 运行命令: `./watch.sh`; 保持 terminal 不关闭, 对 test.coffee 的任意修改都能立即被编译成 test.js, 结果输出到 ../test.html 中。(注意不要改变 test.coffee 的文件名)

3. 重新编译 jison 文件

如果语法文件 `parser.jison` 有改动，需要重新编译成 `parser.js`，请在当前目录下运行：`npm install jison -g` 和 `jison parser.jison` 并将新生成的 `parser.js` 文件替换掉 `example/js/` 和 `realTime/js/` 目录下的 `parser.js`。