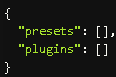
1. **查看部署进度**
2. [ruanyf.github.io/es-checker](http://ruanyf.github.io/es-checker) 查看正在使用的浏览器对ES6的支持程度。
3. 在node中npm install –g es-checker后运行es-checker可以看到正在使用的Node环境对ES6的支持程度。
4. **安装babel-cli以及部署Babel转码器**
5. Babel可以让你用ES6的方式编写程序，又能将它转换成ES5的语法来兼容现有环境。
6. 全局安装babel-cli工具，用于命令行转码：



1. 初始化项目：**npm init**
2. 在项目中安装babel-cli工具：

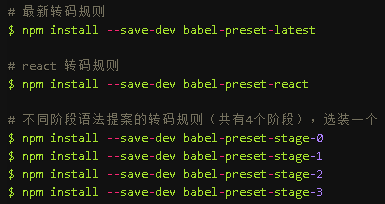


1. 第五步（虽然我这里放在第五，但是这是所有的前提。），配置.babelrc文件（可用type null>.babelrc命令创建），存放在项目的根目录下面，这个文件用来设置转码规则和插件，基本格式如下：



安装必备插件命令：**npm install –save-dev babel-preset-es2015,同样要在.babelrc文件的”presets”中添加”es2015”**

1. Presets字段设定的转码规则，有如下几个，按需安装：



1. 将装了的规则名称加入.babelrc文件：



现在Babel转码器就安装设置完毕了。

1. **使用Babel转码器**

基于babel-cli工具，进行命令行转码。

1. 基本用法：

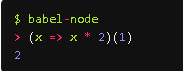


1. 若只在项目中安装了依赖，则就不能使用babel命令了，而是在package.json文件中进行改写：



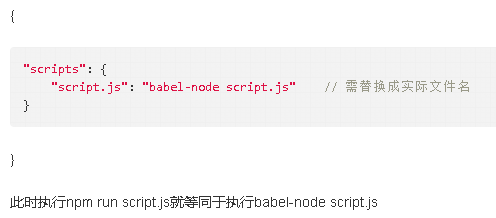
1. **babel-node的使用**

是babel-cli工具自带的一个命令，开启之后就能在node中直接运行ES6代码。



甚至可以这样**babel-node es6.js**来直接运行你的es6代码。

如只有项目依赖则可以改写package.json：



**其他一些东西及与webpack、gulp配合使用查看**<https://segmentfault.com/a/1190000008491089>