



ES6

---





## ECMAScript6

ECMAScript 6（以下简称ES6）是JavaScript语言的下一代标准。因为当前版本的ES6是在2015年发布的，所以又称ECMAScript 2015。也就是说，ES6就是ES2015。

## Babel

Babel是一个广泛使用的转码器，可以将ES6代码转为ES5代码，从而在现有环境执行。这意味着，你可以现在就用ES6编写程序，而不用担心现有环境是否支持。下面是一个例子。

// 转码前

```
input.map(item => item + 1);
```

// 转码后

```
input.map(function (item) {  
  return item + 1;  
});
```

## 配置文件.babelrc

Babel的配置文件是.babelrc，存放在项目的根目录下。使用Babel的第一步，就是配置这个文件。  
该文件用来设置转码规则和插件

```
{  
  "presets": [],  
  "plugins": []  
}
```

presets字段设定转码规则，官方提供以下的规则集，你可以根据需要安装。

# ES2015转码规则

```
$ npm install --save-dev babel-preset-es2015
```

## 配置文件.babelrc

然后，将这些规则加入.babelrc

```
{  
  "presets": [  
    "es2015",  
    "stage-2"  
  ],  
  "plugins": []  
}
```

注意，以下所有Babel工具和模块的使用，都必须先写好.babelrc。

## 命令行转码babel-cli

Babel提供babel-cli工具，用于命令行转码。

命令：\$ npm install --global babel-cli

基本用法如下：

# 转码结果输出到标准输出

\$ babel example.js

## 命令行转码babel-cli

# 转码结果写入一个文件

# --out-file 或 -o 参数指定输出文件

```
$ babel example.js --out-file compiled.js
```

# 或者

```
$ babel example.js -o compiled.js
```

# 整个目录转码

# --out-dir 或 -d 参数指定输出目录

```
$ babel src --out-dir lib
```

# 或者

```
$ babel src -d lib
```

# -s 参数生成source map文件

```
$ babel src -d lib -s
```

## 项目依赖

```
$ npm install --save-dev babel-cli
```

改写package.json

```
{  
  "devDependencies": {  
    "babel-cli": "^6.24.1"  
  },  
  "scripts": {  
    "build": "babel src -d lib"  
  },  
}
```

转码的时候，就执行下面的命令

```
$ npm run build
```



## babel-node

babel-cli工具自带一个babel-node命令，提供一个支持ES6的REPL环境。它支持Node的REPL环境的所有功能，而且可以直接运行ES6代码

它不用单独安装，而是随babel-cli一起安装。然后，执行babel-node就进入REPL环境。

```
$ babel-node  
> (x => x * 2)(1)  
2
```

babel-node命令可以直接运行ES6脚本。将上面的代码放入脚本文件es6.js，然后直接运行。

```
$ babel-node es6.js  
2
```

## babel-node也可以安装在项目中。

```
$ npm install --save-dev babel-cli
```

然后，改写package.json。

```
{  
  "scripts": {  
    "script-name": "babel-node script.js"  
  }  
}
```

上面代码中，使用babel-node替代node，这样script.js本身就不用做任何转码处理。

## 在线转换

在线转换地址：<https://babeljs.io/repl/>



## 浏览器环境

在线转换地址：<https://babeljs.io/repl/>

Babel也可以用于浏览器环境。但是，从Babel 6.0开始，不再直接提供浏览器版本，而是要用构建工具构建出来。如果你没有或不想使用构建工具，可以通过安装5.x版本的babel-core模块获取。

```
$ npm install babel-core@5
```

运行上面的命令以后，就可以在当前目录的node\_modules/babel-core/子目录里面，找到babel的浏览器版本browser.js（未精简）和browser.min.js（已精简）。

然后，将下面的代码插入网页。

## 浏览器环境

```
<script src="node_modules/babel-core/browser.js"></script>  
<script type="text/babel">  
// Your ES6 code  
</script>
```

上面代码中，browser.js是Babel提供的转换器脚本，可以在浏览器运行。用户的ES6脚本放在script标签之中，但是要注明type="text/babel"。

下面是如何将代码打包成浏览器可以使用的脚本，以Babel配合Browserify为例。首先，安装babelify模块。

```
$ npm install --save-dev babelify babel-preset-es2015
```

在package.json设置下面的代码，就不用每次命令行都输入参数了。

```
{
  "browserify": {
    "transform": [["babelify", { "presets": ["es2015"] }]]
  }
}
```

执行：\$ browserify script.js -o bundle.js



## WebStorm配置转换ES6环境

博客地址：

<http://www.iwen.wiki/blog/2016/11/04/webstorm%E9%85%8D%E7%BD%AEes6%E4%BB%A3%E7%A0%81%E8%BD%AC%E6%8D%A2es5%E4%BB%A3%E7%A0%81/>

# ES6

参考地址: <http://es6.ruanyifeng.com/>



A close-up of a person's hand, palm facing up, reaching out towards the camera. The hand is well-lit, showing skin texture and fingernails. The background is a blurred office environment with desks, chairs, and computer monitors. Overlaid on the right side of the image is the text '谢谢观看！' in a large, white, bold font with a slight shadow.

谢谢观看！