七、浏览器环境

七、浏览器环境

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。

```
<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"` 。
另一种方法是使用babel-standalone模块提供的浏览器版本，将其插入网页。

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/babel-standalone/6.4.4/babel.min.js"></script>
<script type="text/babel">
// Your ES6 code
</script>
```

注意，网页中实时将ES6代码转为ES5，对性能会有影响。生产环境需要加载已经转码完成的脚本。
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
```

然后，再用命令行转换ES6脚本。

```
$  browserify script.js -o bundle.js \
  -t [ babelify --presets [ es2015 react ] ]
```

上面代码将ES6脚本 `script.js` ，转为 `bundle.js` ，浏览器直接加载后者就可以了。
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"] }]]
  }
}
```

上一篇：六、babel-polyfill

下一篇：八、在线转换