未整理前原文见素材es6（提案都不整理）

前言

ECMAScript 6简介

（6）babel转码器

babel可以将es6转为es5。

项目根目录下有配置文件.babelrc，基本格式如下：

{

"presets": [], 用来设置转码规则

"plugins": [] 用来设置插件

}

babel-cli：命令行转码

$ babel 待转文件 --out-file或-o 转后文件

$ babel 待转目录 --out-dir或-d 转后目录

-s参数生成source map文件

$ npm install --global babel-cli 环境依赖

$ npm install --save-dev babel-cli 项目依赖

项目依赖可以改写package.json文件：

{

// ...

"devDependencies": {

"babel-cli": "^6.0.0"

},

"scripts": {

"build": "babel src -d lib"

},

}

执行$ npm run build进行转码

babel-node：babel-cli自带babel-node命令，支持ES6的REPL环境，可以直接运行ES6代码。

$ babel-node es6代码或文件

babel-register：为require命令加上钩子。此后，每当使用require加载.js、.jsx、.es和.es6后缀名的文件，就会先用Babel进行转码。

$ npm install --save-dev babel-register

当前文件（当前文件不会被转码。实时转码。只适合开发环境使用）：

require("babel-register");

require("需要被转码文件");

babel-core：代码调用 babel 的 api 进行转码

babel-polyfill：babel转换语法，babel-polyfill转换API

let 和 const 命令

for循环控制是父作用域，循环体是子作用域

es5有全局作用域和函数作用域，es6有全局作用域和函数作用域和块级作用域

var的环境最低保存在函数作用域function内，let的环境最低保存在块级作用域{}内

var可不声明，可先用再声明，可多次声明，let必须先声明再用，必须单次声明

const保证变量指向的内存地址不得改变。const声明的变量赋值后不能再改变，const声明的变量若为对象或数组，则对象或数组内的属性可以改变

顶层对象：浏览器环境指的是window对象，Node环境指的是global对象

变量的解构赋值

解构赋值允许指定默认值。===undefined时，默认值生效

（1）数组的解构赋值

等号两侧关于数组结构满足模式匹配即可，[数组第一个值的别名，[数组第二个值的别名],数组第三个值的别名， ，数组第五个值的别名,…数组剩下值的别名]

等号右侧是可遍历结构（实现了iterator接口）即可

（2）对象的解构赋值:次序可不相同

等号两侧关于对象结构满足模式匹配即可

若变量与属性同名，解构为{属性名}

若变量与属性不同名，解构为{属性名：变量名}

let node = {

a: {

b: {

c: 1,

d: 5

}

}

};

let { a, a: { b }, a: { b: { c }} } = node;

数组本质是特殊的对象：

let arr = [1, 2, 3];

let {0 : first, [arr.length - 1] : last} = arr;

first // 1

last // 3

（3）字符串的解构赋值

let [a, b] = 'hi'; //字符串被转换成类似数组的对象

a // "h"

b // "i"

let {length} = 'hello'; //类似数组的对象都有一个length属性

length // 5