Es6学习手记

**作　者　：韩　　涛**

**2017年3月3日**



[第一章 Let和const命令 3](#_Toc1134296635)

[1. Let命令 3](#_Toc462009338)

[2. 块级作用域 3](#_Toc2101813308)

[3. Const命令 3](#_Toc362406017)

[4. 顶层对象的属性 3](#_Toc1092911220)

[5. Global对象 3](#_Toc961191010)

[第二章 变量的解构和赋值 3](#_Toc1349411734)

[第三章 字符串的扩展 3](#_Toc1462421231)

[第四章 正则的扩展 4](#_Toc922920440)

[第五章 数值的扩展 4](#_Toc296766678)

[第六章 数组的扩展 4](#_Toc949473184)

[第七章 函数的扩展 4](#_Toc1978118182)

[第八章 对象的扩展 4](#_Toc1530868412)

[第九章 symbol 4](#_Toc1496898339)

[第十章 Set和map数据解构 4](#_Toc484273606)

[第十一章 proxy 4](#_Toc1292682082)

[第十二章 Reflect 4](#_Toc2060846231)

[第十三章 Promise对象 4](#_Toc1171417096)

[第十四章 Iterator和for...of循环 4](#_Toc1205743439)

[第十五章 Generator函数的语法 4](#_Toc1833076766)

[第十六章 Generator函数的异步应用 4](#_Toc522026889)

[第十七章 Async函数 5](#_Toc2120029888)

[第十八章 Class 5](#_Toc1722752969)

[第十九章 Decorator 5](#_Toc2089784149)

[第二十章 Module的语法 5](#_Toc2114321298)

[第二十一章 Module的加载实现 5](#_Toc1682895516)

[第二十二章 编程风格 5](#_Toc433895134)

# Let和const命令

## Let命令

## 块级作用域

## Const命令

## 顶层对象的属性

## Global对象

# 变量的解构和赋值

## [数组的解构赋值](http://es6.ruanyifeng.com/" \l "docs/destructuring#数组的解构赋值)

## [对象的解构赋值](http://es6.ruanyifeng.com/" \l "docs/destructuring#对象的解构赋值)

## [字符串的解构赋值](http://es6.ruanyifeng.com/" \l "docs/destructuring#字符串的解构赋值)

## [数值和布尔值的解构赋值](http://es6.ruanyifeng.com/" \l "docs/destructuring#数值和布尔值的解构赋值)

## [函数参数的解构赋值](http://es6.ruanyifeng.com/" \l "docs/destructuring#函数参数的解构赋值)

## [圆括号问题](http://es6.ruanyifeng.com/" \l "docs/destructuring#圆括号问题)

## [用途](http://es6.ruanyifeng.com/" \l "docs/destructuring#用途)

# 字符串的扩展

## [字符的Unicode表示法](http://es6.ruanyifeng.com/" \l "docs/string#字符的Unicode表示法)

## [codePointAt()](http://es6.ruanyifeng.com/" \l "docs/string#codePointAt())

## [String.fromCodePoint()](http://es6.ruanyifeng.com/" \l "docs/string#String.fromCodePoint())

## [字符串的遍历器接口](http://es6.ruanyifeng.com/" \l "docs/string#字符串的遍历器接口)

## [at()](http://es6.ruanyifeng.com/" \l "docs/string#at())

## [normalize()](http://es6.ruanyifeng.com/" \l "docs/string#normalize())

## [includes(), startsWith(), endsWith()](http://es6.ruanyifeng.com/" \l "docs/string#includes(), startsWith(), endsWith())

## [repeat()](http://es6.ruanyifeng.com/" \l "docs/string#repeat())

## [padStart()，padEnd()](http://es6.ruanyifeng.com/" \l "docs/string#padStart()，padEnd())

## [模板字符串](http://es6.ruanyifeng.com/" \l "docs/string#模板字符串)

## [实例：模板编译](http://es6.ruanyifeng.com/" \l "docs/string#实例：模板编译)

## [标签模板](http://es6.ruanyifeng.com/" \l "docs/string#标签模板)

## [String.raw()](http://es6.ruanyifeng.com/" \l "docs/string#String.raw())

## [模板字符串的限制](http://es6.ruanyifeng.com/" \l "docs/string#模板字符串的限制)

# 正则的扩展

## [RegExp构造函数](http://es6.ruanyifeng.com/" \l "docs/regex#RegExp构造函数)

## [字符串的正则方法](http://es6.ruanyifeng.com/" \l "docs/regex#字符串的正则方法)

## [u修饰符](http://es6.ruanyifeng.com/" \l "docs/regex#u修饰符)

## [y 修饰符](http://es6.ruanyifeng.com/" \l "docs/regex#y 修饰符)

## [sticky属性](http://es6.ruanyifeng.com/" \l "docs/regex#sticky属性)

## [flags属性](http://es6.ruanyifeng.com/" \l "docs/regex#flags属性)

## [RegExp.escape()](http://es6.ruanyifeng.com/" \l "docs/regex#RegExp.escape())

## [s 修饰符：dotAll 模式](http://es6.ruanyifeng.com/" \l "docs/regex#s 修饰符：dotAll 模式)

## [后行断言](http://es6.ruanyifeng.com/" \l "docs/regex#后行断言)

## [Unicode属性类](http://es6.ruanyifeng.com/" \l "docs/regex#Unicode属性类)

# 数值的扩展

## [二进制和八进制表示法](http://es6.ruanyifeng.com/" \l "docs/number#二进制和八进制表示法)

## [Number.isFinite(), Number.isNaN()](http://es6.ruanyifeng.com/" \l "docs/number#Number.isFinite(), Number.isNaN())

## [Number.parseInt(), Number.parseFloat()](http://es6.ruanyifeng.com/" \l "docs/number#Number.parseInt(), Number.parseFloat())

## [Number.isInteger()](http://es6.ruanyifeng.com/" \l "docs/number#Number.isInteger())

## [Number.EPSILON](http://es6.ruanyifeng.com/" \l "docs/number#Number.EPSILON)

## [安全整数和Number.isSafeInteger()](http://es6.ruanyifeng.com/" \l "docs/number#安全整数和Number.isSafeInteger())

## [Math对象的扩展](http://es6.ruanyifeng.com/" \l "docs/number#Math对象的扩展)

## [Math.signbit()](http://es6.ruanyifeng.com/" \l "docs/number#Math.signbit())

## [指数运算符](http://es6.ruanyifeng.com/" \l "docs/number#指数运算符)

# 数组的扩展

## [Array.from()](http://es6.ruanyifeng.com/" \l "docs/array#Array.from())

## [Array.of()](http://es6.ruanyifeng.com/" \l "docs/array#Array.of())

## [数组实例的copyWithin()](http://es6.ruanyifeng.com/" \l "docs/array#数组实例的copyWithin())

## [数组实例的find()和findIndex()](http://es6.ruanyifeng.com/" \l "docs/array#数组实例的find()和findIndex())

## [数组实例的fill()](http://es6.ruanyifeng.com/" \l "docs/array#数组实例的fill())

## [数组实例的entries()，keys()和values()](http://es6.ruanyifeng.com/" \l "docs/array#数组实例的entries()，keys()和values())

## [数组实例的includes()](http://es6.ruanyifeng.com/" \l "docs/array#数组实例的includes())

## [数组的空位](http://es6.ruanyifeng.com/" \l "docs/array#数组的空位)

# 函数的扩展

## [函数参数的默认值](http://es6.ruanyifeng.com/" \l "docs/function#函数参数的默认值)

## [rest参数](http://es6.ruanyifeng.com/" \l "docs/function#rest参数)

## [扩展运算符](http://es6.ruanyifeng.com/" \l "docs/function#扩展运算符)

## [严格模式](http://es6.ruanyifeng.com/" \l "docs/function#严格模式)

## [name 属性](http://es6.ruanyifeng.com/" \l "docs/function#name 属性)

## [箭头函数](http://es6.ruanyifeng.com/" \l "docs/function#箭头函数)

## [绑定 this](http://es6.ruanyifeng.com/" \l "docs/function#绑定 this)

## [尾调用优化](http://es6.ruanyifeng.com/" \l "docs/function#尾调用优化)

## [函数参数的尾逗号](http://es6.ruanyifeng.com/" \l "docs/function#函数参数的尾逗号)

# 对象的扩展

## [属性的简洁表示法](http://es6.ruanyifeng.com/" \l "docs/object#属性的简洁表示法)

## [属性名表达式](http://es6.ruanyifeng.com/" \l "docs/object#属性名表达式)

## [方法的 name 属性](http://es6.ruanyifeng.com/" \l "docs/object#方法的 name 属性)

## [Object.is()](http://es6.ruanyifeng.com/" \l "docs/object#Object.is())

## [Object.assign()](http://es6.ruanyifeng.com/" \l "docs/object#Object.assign())

## [属性的可枚举性](http://es6.ruanyifeng.com/" \l "docs/object#属性的可枚举性)

## [属性的遍历](http://es6.ruanyifeng.com/" \l "docs/object#属性的遍历)

## [\_\_proto\_\_Object.setPrototypeOf()，Object.getPrototypeOf()](http://es6.ruanyifeng.com/" \l "docs/object#__proto__属性，Object.setPrototypeOf()，Object.getPrototypeOf())

## [Object.key(),Object.values(),Object.entries()](http://es6.ruanyifeng.com/" \l "docs/object#Object.keys()，Object.values()，Object.entries())

## [对象的扩展运算符](http://es6.ruanyifeng.com/" \l "docs/object#对象的扩展运算符)

## [Object.getOwnPropertyDescriptors()](http://es6.ruanyifeng.com/" \l "docs/object#Object.getOwnPropertyDescriptors())

## [Null 传导运算符](http://es6.ruanyifeng.com/" \l "docs/object#Null 传导运算符)

# Symbol

## [概述](http://es6.ruanyifeng.com/" \l "docs/symbol#概述)

## [作为属性名的Symbol](http://es6.ruanyifeng.com/" \l "docs/symbol#作为属性名的Symbol)

## [实例：消除魔术字符串](http://es6.ruanyifeng.com/" \l "docs/symbol#实例：消除魔术字符串)

## [属性名的遍历](http://es6.ruanyifeng.com/" \l "docs/symbol#属性名的遍历)

## [Symbol.for()，Symbol.keyFor()](http://es6.ruanyifeng.com/" \l "docs/symbol#Symbol.for()，Symbol.keyFor())

## [实例：模块的 Singleton 模式](http://es6.ruanyifeng.com/" \l "docs/symbol#实例：模块的 Singleton 模式)

## [内置的Symbol值](http://es6.ruanyifeng.com/" \l "docs/symbol#内置的Symbol值)

# Set和map数据解构

## [Set](http://es6.ruanyifeng.com/" \l "docs/set-map#Set)

## [WeakSet](http://es6.ruanyifeng.com/" \l "docs/set-map#WeakSet)

## [Map](http://es6.ruanyifeng.com/" \l "docs/set-map#Map)

## [WeakMap](http://es6.ruanyifeng.com/" \l "docs/set-map#WeakMap)

# Proxy

## [概述](http://es6.ruanyifeng.com/" \l "docs/proxy#概述)

## [Proxy 实例的方法](http://es6.ruanyifeng.com/" \l "docs/proxy#Proxy 实例的方法)

## [Proxy.revocable()](http://es6.ruanyifeng.com/" \l "docs/proxy#Proxy.revocable())

## [this 问题](http://es6.ruanyifeng.com/" \l "docs/proxy#this 问题)

## [实例：Web 服务的客户端](http://es6.ruanyifeng.com/" \l "docs/proxy#实例：Web 服务的客户端)

# Reflect

## [概述](http://es6.ruanyifeng.com/" \l "docs/reflect#概述)

## [静态方法](http://es6.ruanyifeng.com/" \l "docs/reflect#静态方法)

## [实例：使用 Proxy 实现观察者模式](http://es6.ruanyifeng.com/" \l "docs/reflect#实例：使用 Proxy 实现观察者模式)

# Promise对象

## [Promise 的含义](http://es6.ruanyifeng.com/" \l "docs/promise#Promise 的含义)

## [基本用法](http://es6.ruanyifeng.com/" \l "docs/promise#基本用法)

## [Promise.prototype.then()](http://es6.ruanyifeng.com/" \l "docs/promise#Promise.prototype.then())

## [Promise.prototype.catch()](http://es6.ruanyifeng.com/" \l "docs/promise#Promise.prototype.catch())

## [Promise.all()](http://es6.ruanyifeng.com/" \l "docs/promise#Promise.all())

## [Promise.race()](http://es6.ruanyifeng.com/" \l "docs/promise#Promise.race())

## [Promise.resolve()](http://es6.ruanyifeng.com/" \l "docs/promise#Promise.resolve())

## [Promise.reject()](http://es6.ruanyifeng.com/" \l "docs/promise#Promise.reject())

## [两个有用的附加方法](http://es6.ruanyifeng.com/" \l "docs/promise#两个有用的附加方法)

## [应用](http://es6.ruanyifeng.com/" \l "docs/promise#应用)

## [Promise.try()](http://es6.ruanyifeng.com/" \l "docs/promise#Promise.try())

# Iterator和for...of循环

## [Iterator（遍历器）的概念](http://es6.ruanyifeng.com/" \l "docs/iterator#Iterator（遍历器）的概念)

## [数据结构的默认Iterator接口](http://es6.ruanyifeng.com/" \l "docs/iterator#数据结构的默认Iterator接口)

## [调用Iterator接口的场合](http://es6.ruanyifeng.com/" \l "docs/iterator#调用Iterator接口的场合)

## [字符串的Iterator接口](http://es6.ruanyifeng.com/" \l "docs/iterator#字符串的Iterator接口)

## [Iterator接口与Generator函数](http://es6.ruanyifeng.com/" \l "docs/iterator#Iterator接口与Generator函数)

## [遍历器对象的return()，throw()](http://es6.ruanyifeng.com/" \l "docs/iterator#遍历器对象的return()，throw())

## [for...of循环](http://es6.ruanyifeng.com/" \l "docs/iterator#for...of循环)

# Generator函数的语法

## [简介](http://es6.ruanyifeng.com/" \l "docs/generator#简介)

## [next方法的参数](http://es6.ruanyifeng.com/" \l "docs/generator#next方法的参数)

## [for...of循环](http://es6.ruanyifeng.com/" \l "docs/generator#for...of循环)

## [Generator.prototype.throw()](http://es6.ruanyifeng.com/" \l "docs/generator#Generator.prototype.throw())

## [Generator.prototype.return()](http://es6.ruanyifeng.com/" \l "docs/generator#Generator.prototype.return())

## [yield\* 语句](http://es6.ruanyifeng.com/" \l "docs/generator#yield* 语句)

## [作为对象属性的Generator函数](http://es6.ruanyifeng.com/" \l "docs/generator#作为对象属性的Generator函数)

## [Generator函数的this](http://es6.ruanyifeng.com/" \l "docs/generator#Generator函数的this)

## [含义](http://es6.ruanyifeng.com/" \l "docs/generator#含义)

## [应用](http://es6.ruanyifeng.com/" \l "docs/generator#应用)

# Generator函数的异步应用

## [传统方法](http://es6.ruanyifeng.com/" \l "docs/generator-async#传统方法)

## [基本概念](http://es6.ruanyifeng.com/" \l "docs/generator-async#基本概念)

## [Generator 函数](http://es6.ruanyifeng.com/" \l "docs/generator-async#Generator 函数)

## [Thunk 函数](http://es6.ruanyifeng.com/" \l "docs/generator-async#Thunk 函数)

## [co 模块](http://es6.ruanyifeng.com/" \l "docs/generator-async#co 模块)

# Async函数

## [含义](http://es6.ruanyifeng.com/" \l "docs/async#含义)

## [用法](http://es6.ruanyifeng.com/" \l "docs/async#用法)

## [语法](http://es6.ruanyifeng.com/" \l "docs/async#语法)

## [async 函数的实现原理](http://es6.ruanyifeng.com/" \l "docs/async#async 函数的实现原理)

## [与其他异步处理方法的比较](http://es6.ruanyifeng.com/" \l "docs/async#与其他异步处理方法的比较)

## [实例：按顺序完成异步操作](http://es6.ruanyifeng.com/" \l "docs/async#实例：按顺序完成异步操作)

## [异步遍历器](http://es6.ruanyifeng.com/" \l "docs/async#异步遍历器)

# Class

## [Class基本语法](http://es6.ruanyifeng.com/" \l "docs/class#Class基本语法)

## [Class的继承](http://es6.ruanyifeng.com/" \l "docs/class#Class的继承)

## [原生构造函数的继承](http://es6.ruanyifeng.com/" \l "docs/class#原生构造函数的继承)

## [Class的取值函数（getter）和存值函数（setter）](http://es6.ruanyifeng.com/" \l "docs/class#Class的取值函数（getter）和存值函数（setter）)

## [Class 的 Generator 方法](http://es6.ruanyifeng.com/" \l "docs/class#Class 的 Generator 方法)

## [Class 的静态方法](http://es6.ruanyifeng.com/" \l "docs/class#Class 的静态方法)

## [Class的静态属性和实例属性](http://es6.ruanyifeng.com/" \l "docs/class#Class的静态属性和实例属性)

## [类的私有属性](http://es6.ruanyifeng.com/" \l "docs/class#类的私有属性)

## [new.target属性](http://es6.ruanyifeng.com/" \l "docs/class#new.target属性)

## [Mixin模式的实现](http://es6.ruanyifeng.com/" \l "docs/class#Mixin模式的实现)

# Decorator

## [类的修饰](http://es6.ruanyifeng.com/" \l "docs/decorator#类的修饰)

## [方法的修饰](http://es6.ruanyifeng.com/" \l "docs/decorator#方法的修饰)

## [为什么修饰器不能用于函数？](http://es6.ruanyifeng.com/" \l "docs/decorator#为什么修饰器不能用于函数？)

## [core-decorators.js](http://es6.ruanyifeng.com/" \l "docs/decorator#core-decorators.js)

## [使用修饰器实现自动发布事件](http://es6.ruanyifeng.com/" \l "docs/decorator#使用修饰器实现自动发布事件)

## [Mixin](http://es6.ruanyifeng.com/" \l "docs/decorator#Mixin)

## [Trait](http://es6.ruanyifeng.com/" \l "docs/decorator#Trait)

## [Babel转码器的支持](http://es6.ruanyifeng.com/" \l "docs/decorator#Babel转码器的支持)

# Module的语法

## [概述](http://es6.ruanyifeng.com/" \l "docs/module#概述)

## [严格模式](http://es6.ruanyifeng.com/" \l "docs/module#严格模式)

## [export 命令](http://es6.ruanyifeng.com/" \l "docs/module#export 命令)

## [import 命令](http://es6.ruanyifeng.com/" \l "docs/module#import 命令)

## [模块的整体加载](http://es6.ruanyifeng.com/" \l "docs/module#模块的整体加载)

## [export default 命令](http://es6.ruanyifeng.com/" \l "docs/module#export default 命令)

## [export 与 import 的复合写法](http://es6.ruanyifeng.com/" \l "docs/module#export 与 import 的复合写法)

## [模块的继承](http://es6.ruanyifeng.com/" \l "docs/module#模块的继承)

## [跨模块常量](http://es6.ruanyifeng.com/" \l "docs/module#跨模块常量)

## [import()](http://es6.ruanyifeng.com/" \l "docs/module#import())

# Module的加载实现

## [浏览器加载](http://es6.ruanyifeng.com/" \l "docs/module-loader#浏览器加载)

## [ES6 模块与 CommonJS 模块的差异](http://es6.ruanyifeng.com/" \l "docs/module-loader#ES6 模块与 CommonJS 模块的差异)

## [Node 加载](http://es6.ruanyifeng.com/" \l "docs/module-loader#Node 加载)

## [循环加载](http://es6.ruanyifeng.com/" \l "docs/module-loader#循环加载)

## [ES6模块的转码](http://es6.ruanyifeng.com/" \l "docs/module-loader#ES6模块的转码)

# 编程风格

## [块级作用域](http://es6.ruanyifeng.com/" \l "docs/style#块级作用域)

## [字符串](http://es6.ruanyifeng.com/" \l "docs/style#字符串)

## [解构赋值](http://es6.ruanyifeng.com/" \l "docs/style#解构赋值)

## [对象](http://es6.ruanyifeng.com/" \l "docs/style#对象)

## [数组](http://es6.ruanyifeng.com/" \l "docs/style#数组)

## [函数](http://es6.ruanyifeng.com/" \l "docs/style#函数)

## [Map结构](http://es6.ruanyifeng.com/" \l "docs/style#Map结构)

## [Class](http://es6.ruanyifeng.com/" \l "docs/style#Class)

## [模块](http://es6.ruanyifeng.com/" \l "docs/style#模块)

## [ESLint的使用](http://es6.ruanyifeng.com/" \l "docs/style#ESLint的使用)