

# JavaScript设计模式



[JavaScript设计模式\\_下载链接1\\_](#)

著者:Addy Osmani

出版者:人民邮电出版社

出版时间:2013-6-1

装帧:平装

isbn:9787115314543

《JavaScript设计模式》是JavaScript设计模式的学习指南。全书分为14章。首先介绍了什么是模式、模式的结构、类别、模式的分类、如何编写模式等等；然后，集中介绍了很多流行的设计模式在JavaScript中的应用，包括Module（模块）模式、Observer（

观察者)模式、Facade (外观)模式和Mediator (中介者)模式;最后,还探讨了模块化的JavaScript模式、jQuery及其插件中的设计模式。

《JavaScript设计模式》适合专业的Web开发人员和前端工程师阅读。通过阅读本书,他们将能够提高对设计模式的认识,并学会如何将设计模式应用到JavaScript编程语言中。

作者介绍:

addy

osmani, 谷歌chrome团队的开发项目工程师,对javascript应用程序架构有着强烈的爱好。他创建比较流行的项目,如todomvc,并对modernizr和jquery等其他开源项目也做出很大贡献。作为一位高产的博主(<http://addyosmani.com/blog>),addy的文章经常出现在《javascript电子周刊》、《smashing杂志》及很多其他出版物上。

目录: 第1章 介绍	1
第2章 什么是模式	3
我们每天都在使用模式	4
第3章 模式状态测试、proto模式及三法则	6
第4章 设计模式的结构	8
第5章 编写设计模式	11
第6章 反模式	13
第7章 设计模式类别	15
第8章 设计模式分类	17
有关类(class)的要点	17
第9章 javascript设计模式	20
9.1 constructor(构造器)模式	21
9.1.1 对象创建	21
9.1.2 基本constructor(构造器)	23
9.1.3 带原型的constructor(构造器)	24
9.2 module(模块)模式	25
9.2.1 对象字面量	25
9.2.2 module(模块)模式	27
9.2.3 module模式变化	31
9.3 revealing module(揭示模块)模式	36
9.3.1 优点	38
9.3.2 缺点	38
9.4 singleton(单例)模式	38
9.5 observer(观察者)模式	42
9.5.1 observer(观察者)模式和publish/subscribe(发布/订阅)模式的区别	47
9.5.2 优点	49
9.5.3 缺点	49
9.5.4 publish/subscribe实现	49
9.6 mediator(中介者)模式	59
9.6.1 基本实现	60
9.6.2 高级实现	61
9.6.3 示例	67
9.6.4 优点和缺点	68
9.6.5 中介者(mediator)与观察者(observer)	69
9.6.6 中介者(mediator)与外观(facade)	69
9.7 prototype(原型)模式	70
9.8 command(命令)模式	73

- 9.9 facade(外观)模式 75
- 有关抽象的要点 78
- 9.10 factory(工厂)模式 78
  - 9.10.1 何时使用factory模式 81
  - 9.10.2 何时不应使用factory模式 81
  - 9.10.3 abstract factory(抽象工厂) 81
- 9.11 mixin模式 82
  - 9.11.1 子类化 83
  - 9.11.2 mixin(混入) 84
- 9.12 decorator(装饰者)模式 88
  - 9.12.1 伪经典decorator(装饰者) 91
  - 9.12.2 使用jquery的装饰者 96
  - 9.12.3 优点和缺点 97
- 9.13 flyweight(享元)模式 98
  - 9.13.1 使用flyweight模式 98
  - 9.13.2 flyweight和共享数据 99
  - 9.13.3 实现经典flyweight(享元) 99
  - 9.13.4 转换代码以使用flyweight(享元)模式 103
  - 9.13.5 基本工厂 105
  - 9.13.6 管理外部状态 106
  - 9.13.7 flyweight(享元)模式和dom 107
- 第10章 javascript mv\*模式 112
  - 10.1 mvc 112
    - smalltalk-80 mvc 113
  - 10.2 为javascript开发人员提供的mvc 114
    - 10.2.1 model(模型) 114
    - 10.2.2 view(视图) 116
    - 10.2.3 controller(控制器) 119
    - 10.2.4 spine.js与backbone.js 120
  - 10.3 mvc为我们提供了什么 122
  - 10.4 javascript中的smalltalk-80 mvc 122
    - 10.4.1 深入挖掘 123
    - 10.4.2 总结 123
  - 10.5 mvp 124
    - 10.5.1 model、view和presenter 124
    - 10.5.2 mvp或mvc? 125
    - 10.5.3 mvc、mvp和backbone.js 126
  - 10.6 mvvm 128
    - 10.6.1 历史 129
    - 10.6.2 model 129
    - 10.6.3 view 130
    - 10.6.4 viewmodel 133
    - 10.6.5 小结: view和viewmodel 135
    - 10.6.6 小结: viewmodel和model 135
  - 10.7 利与弊 135
    - 10.7.1 优点 135
    - 10.7.2 缺点 136
  - 10.8 使用更松散数据绑定的mvvm 136
  - 10.9 mvc、mvp与mvvm 141
  - 10.10 backbone.js与knockoutjs 142
- 第11章 模块化的javascript设计模式 144
  - 11.1 脚本加载器要点 145
  - 11.2 amd 145
    - 11.2.1 模块入门 146

- 11.2.2 使用dojo的amd模块 150
- 11.2.3 amd模块设计模式(dojo) 151
- 11.2.4 使用jquery的amd模块 152
- 11.2.5 amd总结 155
- 11.3 commonjs 155
  - 11.3.1 入门指南 156
  - 11.3.2 使用多个依赖 157
  - 11.3.3 支持commonjs的加载器和框架 158
  - 11.3.4 commonjs适用于浏览器吗? 158
  - 11.3.5 延伸阅读 159
- 11.4 amd和commonjs: 互相竞争, 标准同效 159
- umd: 用于插件的amd和commonjs兼容模块 160
- 11.5 es harmony 165
  - 11.5.1 具有imports和exports的模块 166
  - 11.5.2 从远程数据源加载的模块 167
  - 11.5.3 模块加载器api 167
  - 11.5.4 用于服务器的类commonjs模块 168
  - 11.5.5 具有构造函数、getter和setter的类 168
  - 11.5.6 es harmony总结 169
  - 11.5.7 延伸阅读 170
- 11.6 总结 170
- 第12章 jquery中的设计模式 171
  - 12.1 composite(组合)模式 171
  - 12.2 adapter(适配器)模式 173
  - 12.3 facade(外观)模式 174
  - 12.4 observer(观察者)模式 177
  - 12.5 iterator(迭代器)模式 180
  - 12.6 延迟初始化 181
  - 12.7 proxy(代理)模式 183
  - 12.8 builder(生成器)模式 184
- 第13章 jquery插件设计模式 187
  - 13.1 模式 188
  - 13.2 lightweight start模式 189
  - 延伸阅读 191
  - 13.3 完整的widget factory模式 191
  - 延伸阅读 194
  - 13.4 嵌套命名空间插件模式 194
  - 延伸阅读 196
  - 13.5 自定义事件插件模式(使用widget factory) 196
  - 延伸阅读 198
  - 13.6 使用dom-to-object bridge模式的原型继承 198
  - 延伸阅读 200
  - 13.7 jquery ui widget factory bridge模式 200
  - 延伸阅读 203
  - 13.8 使用widget factory的jquery mobile widget 203
  - 13.9 requirejs和jquery ui widget factory 206
    - 13.9.1 用法 208
    - 13.9.2 延伸阅读 209
  - 13.10 全局选项和单次调用可重写选项(最佳选项模式) 209
  - 延伸阅读 211
  - 13.11 高可配和高可变的插件模式 211
  - 延伸阅读 213
  - 13.12 是什么使插件超越模式 213
    - 13.12.1 质量 214

13.12.2 代码风格 214  
13.12.3 兼容性 214  
13.12.4 可靠性 214  
13.12.5 性能 214  
13.12.6 文档 215  
13.12.7 维护的可能性 215  
13.13 总结 215  
13.14 命名空间模式 215  
13.15 命名空间基础 216  
13.15.1 单一全局变量 216  
13.15.2 命名空间前缀 217  
13.15.3 对象字面量表示法 217  
13.15.4 嵌套命名空间 221  
13.15.5 立即调用的函数表达式(iife) 222  
13.15.6 命名空间注入 224  
13.16 高级命名空间模式 226  
13.16.1 自动嵌套的命名空间 227  
13.16.2 依赖声明模式 229  
13.16.3 深度对象扩展 229  
13.16.4 推荐 232  
第14章 总结 233  
附录 参考文献 235  
• • • • • ([收起](#))

[JavaScript设计模式\\_下载链接1](#)

## 标签

JavaScript

设计模式

前端开发

Web开发

Web前端开发

Web前端

前端

## 评论

记住你了，徐涛，你翻译的书绝不能看。

-----  
翻译好烂，读不下去了

-----  
前半段在講怎麼讓 JavaScript 模仿有 Class 的強 Type 語言，後半段在講怎麼利用 JavaScript Prototype Object 的弱 Type 特性來實現設計模式。  
不過現在的人已經都在寫 ES2015 或 TypeScript 了，所以有些章節可以省去，主要還是前面 JavaScript Prototype Object、閉包與函式的特性比較重要。

-----  
翻译不咋的，内容上讲了太多 jQuery 相关的东西了.....

-----  
翻译的是在太烂,中文我都看不懂

-----  
不能完全怪翻译，有些概念和用词找到对应的中文拼一块本来就不知所云...

-----  
没工作的时候，读读书。只是确实有点烂翻译，让人不想继续读，当然，自己的水平也是一方面。

-----  
内容不错，但是代码的缩进着实太随便了

-----  
开始看第五本！tom大叔翻译的很烂，代码也很多没有缩进。。。不推荐

-----  
翻译的人，人品有问题，根本就没有一个技术类书籍翻译的态度，谷歌翻译的？徐涛以后大家要注意他翻译的书！

-----  
内容还是值得肯定，例子很多。但是排版也太随心所欲了，代码缩进就不说了，还有很多错误，这是要虐死处女座的节奏·····

-----  
在作者网站有完全版。最好能先读一下reference里的基本经典。最后一部分需要对jQuery有很深的了解才能完全读懂。整体面向中高阶js开发者

-----  
还是看原文吧。看译者博客有水平，但显然态度太差了，比机器翻译好不了多少。

-----  
翻译实在太烂了，我这么对翻译不挑剔的人都忍不了了，比机翻的好不到哪儿去，直接看原版去了····

-----  
不懂的地方仍然不懂，懂的东西一下子豁然开朗。

-----  
有些难度啊，翻译还行，有些错误，排版很烂

-----  
如果是权威指南跟红宝石书是兵器谱，这本可以说是排兵布阵图

-----  
翻译和排版扣分，除去一些错误，还是挺不错的一本书，有辨识能力的话可以学到不少东西

非常棒的设计模式入门书。把由来，发展，原理。实现讲的简洁易懂。还薄的感人

## 26种设计模式那块略看

在讲解各种js框架和MVP\MVC等模式的时候还是可以。特别是js的模板、AMD等等这些的讲解 为什么要引入这些框架 在哪些情况引入这些框架 还是比较透彻的  
最主要这本书比较新 谈到的各种技术比较先进

[JavaScript设计模式\\_下载链接1](#)

书评

翻译太差劲了，随便找一句对比：原文：“While the Observer pattern is useful to be aware of, quite often in the JavaScript world, we’ ll find it commonly implemented using a variation known as the Publish/Subscribe pattern.”（虽然 Observer 模式非常有用，...

显然译者的态度有问题。看原文吧。随便翻几页就看到这些惨不忍睹的文字： p83: 对于...开发人员来说， 在...之前， 将阅读初学者内容； p6: This is a reason why patterns generally focus on addressing a visually identifiable structure -->...

一句话:好书被翻译搞的什么都不是 什么,还是太短? 我说了,翻译太烂了,去看原书吧  
还是短? 豆瓣你想怎样?

[illegible]

javascript中有一个名为prototype的属性? what's the fuck?

看这么烂的书简直是浪费时间!

太不专业，幸好书是首图借的，不然立马扔。幸好书是首图借的，不然立马扔。幸好书是首图借的，不然立马扔。幸好书是首图借的，不然立马扔。幸好书是首图借的，不然立马扔。



立马扔。幸好书是首...

-----  
这是一本通过Javascript语言来阐述设计模式的书籍。  
前面讲解的Javascript特性已经面向对象思想还是很有意思的。  
至于后面的设计模式，没什么意思。不具有实际应用的特色。  
纯粹为了设计模式的实现而撰写。案例没有连续性。  
三天时间翻完了，还是很不错的。对面向对象有了认...

-----  
原作者Andy Osmani是JavaScript大牛，这本书将GoF的Design  
Pattern用JavaScript实现。前八章是Design  
Pattern的简介，后面章节是各个模式的实例，以及MVC等。  
翻译的内容还行，但是代码的缩进真的太随便了!!! 如果可以，还是推荐读原文。

-----  
总体还行 是js设计模式入门学习，前八章基本可略过或扫读就行。  
译者水平是有，但不得不提的是，玷污原著最大的地方是，大部分实例代码的缩进被搞  
得太随意了!! 对读这本书的代码已有阴影。。。  
代码的缩进被搞得太随意了代码的缩进被搞得太随意了代码的缩进被搞得太随意了代...

-----  
[JavaScript设计模式\\_下载链接1](#)