

<<JavaScript设计模式>>

图书基本信息

书名：<<JavaScript设计模式>>

13位ISBN编号：9787115314543

10位ISBN编号：7115314543

出版时间：2013-6-1

出版时间：人民邮电出版社

作者：Addy Osmani

译者：徐涛

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<JavaScript设计模式>>

内容概要

《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。

<<JavaScript设计模式>>

作者简介

作者:(美)奥斯马尼 译者:徐涛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。

他创建了一些流行的项目, 如Todo MVC, 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《JavaScript电子周刊》、《Smashing杂志》及很多其他出版物上。

<<JavaScript设计模式>>

书籍目录

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 209.1 Constructor(构造器)模式 219.1.1 对象创建 219.1.2 基本Constructor(构造器) 239.1.3 带原型的Constructor(构造器) 249.2 Module(模块)模式 259.2.1 对象字面量 259.2.2 Module(模块)模式 279.2.3 Module模式变化 319.3 Revealing Module(揭示模块)模式 369.3.1 优点 389.3.2 缺点 389.4 Singleton(单例)模式 389.5 Observer(观察者)模式 429.5.1 Observer(观察者)模式和Publish/Subscribe(发布/订阅)模式的区别 479.5.2 优点 499.5.3 缺点 499.5.4 Publish/Subscribe实现 499.6 Mediator(中介者)模式 599.6.1 基本实现 609.6.2 高级实现 619.6.3 示例 679.6.4 优点和缺点 689.6.5 中介者(Mediator)与观察者(Observer) 699.6.6 中介者(Mediator)与外观(Facade) 699.7 Prototype(原型)模式 709.8 Command(命令)模式 739.9 Facade(外观)模式 75 有关抽象的要点 789.10 Factory(工厂)模式 789.10.1 何时使用Factory模式 819.10.2 何时不应使用Factory模式 819.10.3 Abstract Factory(抽象工厂) 819.11 Mixin模式 829.11.1 子类化 839.11.2 Mixin(混入) 849.12 Decorator(装饰者)模式 889.12.1 伪经典Decorator(装饰者) 919.12.2 使用jQuery的装饰者 969.12.3 优点和缺点 979.13 Flyweight(享元)模式 989.13.1 使用Flyweight模式 989.13.2 Flyweight和共享数据 999.13.3 实现经典Flyweight(享元) 999.13.4 转换代码以使用Flyweight(享元)模式 1039.13.5 基本工厂 1059.13.6 管理外部状态 1069.13.7 Flyweight(享元)模式和DOM 107第10章 JavaScript MV*模式 11210.1 MVC 112 Smalltalk-80 MVC 11310.2 为JavaScript开发人员提供的MVC 11410.2.1 Model(模型) 11410.2.2 View(视图) 11610.2.3 Controller(控制器) 11910.2.4 Spine.js与Backbone.js 12010.3 MVC为我们提供了什么 12210.4 JavaScript中的Smalltalk-80 MVC 12210.4.1 深入挖掘 12310.4.2 总结 12310.5 MVP 12410.5.1 Model、View和Presenter 12410.5.2 MVP或MVC? 12510.5.3 MVC、MVP和Backbone.js 12610.6 MVVM 12810.6.1 历史 12910.6.2 Model 12910.6.3 View 13010.6.4 ViewModel 13310.6.5 小结: View和ViewModel 13510.6.6 小结: ViewModel和Model 13510.7 利与弊 13510.7.1 优点 13510.7.2 缺点 13610.8 使用更松散数据绑定的MVVM 13610.9 MVC、MVP与MVVM 14110.10 Backbone.js与KnockoutJS 142第11章 模块化的JavaScript设计模式 14411.1 脚本加载器要点 14511.2 AMD 14511.2.1 模块入门 14611.2.2 使用Dojo的AMD模块 15011.2.3 AMD模块设计模式(Dojo) 15111.2.4 使用jQuery的AMD模块 15211.2.5 AMD总结 15511.3 CommonJS 15511.3.1 入门指南 15611.3.2 使用多个依赖 15711.3.3 支持CommonJS的加载器和框架 15811.3.4 CommonJS适用于浏览器吗? 15811.3.5 延伸阅读 15911.4 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 16011.5 ES Harmony 16511.5.1 具有Imports和Exports的模块 16611.5.2 从远程数据源加载的模块 16711.5.3 模块加载器API 16711.5.4 用于服务器的类CommonJS模块 16811.5.5 具有构造函数、getter和setter的类 16811.5.6 ES Harmony总结 16911.5.7 延伸阅读 17011.6 总结 170第12章 jQuery中的设计模式 17112.1 Composite(组合)模式 17112.2 Adapter(适配器)模式 17312.3 Facade(外观)模式 17412.4 Observer(观察者)模式 17712.5 Iterator(迭代器)模式 18012.6 延迟初始化 18112.7 Proxy(代理)模式 18312.8 Builder(生成器)模式 184第13章 jQuery插件设计模式 18713.1 模式 18813.2 Lightweight Start模式 189 延伸阅读 19113.3 完整的Widget Factory模式 191 延伸阅读 19413.4 嵌套命名空间插件模式 194 延伸阅读 19613.5 自定义事件插件模式(使用Widget Factory) 196 延伸阅读 19813.6 使用DOM-to-Object Bridge模式的原型继承 198 延伸阅读 20013.7 jQuery UI Widget Factory Bridge模式 200 延伸阅读 20313.8 使用Widget Factory的jQuery Mobile Widget 20313.9 RequireJS和jQuery UI Widget Factory 20613.9.1 用法 20813.9.2 延伸阅读 20913.10 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 21113.11 高可配和高可变的插件模式 211 延伸阅读 21313.12 是什么使插件超越模式 21313.12.1 质量 21413.12.2 代码风格 21413.12.3 兼容性 21413.12.4 可靠性 21413.12.5 性能 21413.12.6 文档 21513.12.7 维护的可能性 21513.13 总结 21513.14 命名空间模式 21513.15 命名

<<JavaScript设计模式>>

空间基础 21613.15.1 单一全局变量 21613.15.2 命名空间前缀 21713.15.3 对象字面量表示法
21713.15.4 嵌套命名空间 22113.15.5 立即调用的函数表达式(IIFE) 22213.15.6 命名空间注入
22413.16 高级命名空间模式 22613.16.1 自动嵌套的命名空间 22713.16.2 依赖声明模式 22913.16.3
深度对象扩展 22913.16.4 推荐 232第14章 总结 233附录 参考文献 235

<<JavaScript设计模式>>

编辑推荐

通过阅读奥斯马尼的《JavaScript设计模式》，你将学会如何把经典和现代设计模式应用到JavaScript语言中，来编写优美、结构化和可维护的代码。

如果能让代码保持高效、更易于管理，并且能同步最新的最佳实践，那么本书正是为你打造的。

本书介绍了很多流行的设计模式，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；帮助你从现代Web应用程序开发者的角度来了解MVC、MVP、MVVM等现代架构模式的实用性。

本书将带领你了解现代模块格式、有效为代码定义名称空间的方法和其他重要主题。

<<JavaScript设计模式>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>