

<<至理>>

图书基本信息

书名：<<至理>>

13位ISBN编号：9787121079863

10位ISBN编号：7121079860

出版时间：2009-2

出版时间：电子工业出版社

作者：张亚飞等

页数：782

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>

内容概要

本书完整系统地介绍Flex 3和Flex 4的开发技术，整合了Halo组件和Gumbo组件，使用最新的Flex Builder 3作为创作环境。

本书以严谨的逻辑，完整地介绍了Flex Builder 3创作环境、MXML语言、ActionScript 3语言、Flash Player创作环境、Flex SDK 3、Flex SDK 4。

按照从基础到进阶，从基本原理到实践应用的基本模式，将Flex的两大应用按照逻辑有机地分包含在两篇内容中：Flex组件的呈现和布局应用篇、Flex数据集成和数据呈现篇。

本书具有很强的实用性，包含的数百个范例几乎涵盖Flex开发各个相关方面。

本书每一章绝非泛泛而谈理论，而是围绕实际案例代码来组织的，让读者马上可以上手，并且加深理解。

“实用”是技术类书籍好坏的唯一衡量标准，而本书更是将这个词演绎到了极致。

本书适合RIA开发人员参考学习，也可作为高等院校相关专业师生的参考书。

书籍目录

第1篇 Flex开发基础入门篇	第1章 建立首要的、基础的正确认知	1.1 安装和配置Flex应用程序开发和运行环境	1.1.1 安装Flex Builder 3和Flash Player	1.1.2 创建第一个Flex应用程序	1.1.3 发布和部署Flex	1.2 正确认识Flex应用程序开发和Flash	1.2.1 正确认识Flex SDK、Flex Builder和Flash	1.2.2 正确认识Flex运行环境和开发语言	1.2.3 关于标准和实现	1.3 Flex开发的特点和RIA	第2章 Flex开发基础——使用Flex Builder 3创作环境和MXML标签语法基础	2.1 Flex Builder 3开发环境的基本组成	2.1.1 关闭和打开透视图	2.1.2 自定义透视图	2.2 使用开发透视图	2.2.1 可视化布局	2.2.2 使用代码提示功能	2.2.3 使用大纲视图	2.3 应用程序编译和运行设置	2.3.1 自动编译和SDK	2.3.2 优化编译和查看源代码	2.3.3 运行和测试应用程序	2.4 MXML语法基本常识	2.4.1 基本语法	2.4.2 MXML文件基本结构	2.4.3 MXML语法要求	2.4.4 MXML文件命名规则	2.4.5 定义ActionScript代码	2.4.6 注释	2.4.7 标签属性和数据绑定	2.4.8 嵌入应用程序资源	第3章 Flex开发基础——ActionScript 3.0语法入门	3.1 变量的定义	3.1.1 定义和命名变量	3.1.2 变量的数据类型	3.1.3 使用标签和标签自定义数据模型	3.1.4 常量	3.2 在程序中使用变量	3.2.1 为变量赋值和更改变量值	3.2.2 使用变量和获取变量值	3.3 创建和使用Object	3.4 创建和使用数组	3.4.1 创建数组的基本方法	3.4.2 创建和使用索引数组	3.4.3 实现和使用多维数组	3.4.4 创建和使用复合数组	3.5 ActionScript的语法规约定	3.5.1 空白和多行书写	3.5.2 点语法	3.5.3 花括号	3.5.4 分号	3.5.5 圆括号	3.5.6 字母的大小写	3.5.7 程序注释	3.5.8 关键字	第4章 Flex开发基础——ActionScript 3.0 数据运算和流程控制	4.1 数据类型和运算	4.1.1 表达式	4.1.2 算术运算符	4.1.3 字符串运算符	4.1.4 比较运算符和逻辑运算符	4.1.5 位运算符	4.1.6 赋值运算符	4.1.7 其他运算符	4.1.8 运算符的使用规则	4.2 流程控制	4.2.1 语句和语句块	4.2.2 ifelse条件语句	4.2.3 switch条件语句	4.2.4 dowhile循环	4.2.5 for循环	4.2.6 forin循环	4.2.7 for eachin循环	4.2.8 其他的流程控制	4.2.9 trycatchfinally语句	4.2.10 throw语句	4.2.11 流程控制总结	4.3 使用with语句	4.4 数据类型检查	4.4.1 编译时进行类型检查	4.4.2 运行时进行类型检查	第5章 Flex开发基础——ActionScript 3.0全局函数和自定义函数	5.1 使用预定义的全局函数	5.1.1 转义操作函数	5.1.2 转换函数	5.1.3 判断函数	5.2 自定义函数	5.2.1 函数的定义	5.2.2 函数的调用	5.2.3 定义参数初始值	5.2.4 (rest) 参数	5.2.5 匿名函数的定义	5.3 变量的作用范围和生命周期	5.3.1 了解变量的作用范围和生命周期	5.3.2 局部变量的声明	5.3.3 全局变量的声明	第6章 创建自定义类和进行面向对象的编程	6.1 第一个面向对象开发的范例	6.2 关于面向对象设计和类	6.2.1 面向对象和结构化程序设计的区别	6.2.2 什么是面向对象程序设计	6.2.3 面向对象程序设计的优点和基本特征	6.2.4 ActionScript面向对象开发的注意事项	6.3 创建和使用自定义类	6.3.1 创建类文件	6.3.2 创建类的实例和调用类的方法	6.3.3 自定义类的创建	6.3.4 方法的重载	6.3.5 使用包	6.3.6 访问类的方法和属性	6.3.7 控制成员访问	6.4 类的继承和创建子类	6.4.1 创建子类	6.4.2 重写方法和属性	6.4.3 在类中使用final修饰符	6.4.4 了解原型继承	6.5 区别实例成员和类成员	6.5.1 创建类成员	6.5.2 类成员和子类	6.5.3 类成员和实例成员的区别	6.6 创建动态类	6.7 创建和使用接口	6.7.1 创建接口	6.7.2 接口的实现	6.8 创建纯ActionScript工程	6.9 类路径	6.9.1 指定类路径	6.9.2 编译器如何解析类引用	6.10 扩展Error以创建自定义错误	6.11 隐式获取/设置属性	6.12 使用命名空间	6.12.1 创建和使用命名空间的基本步骤	6.12.2 定义命名空间	6.12.3 应用命名空间	6.12.4 引用命名空间	6.13 包的其他用法	6.13.1 在单个类文件内定义多个类	6.13.2 在包声明中定义的其他语句	第7章 事件处理和基本的交互控制	7.1 基本事件处理	7.1.1 了解怎样实现事件处理	7.1.2 处理事件	7.1.3 在MXML标签中注册事件处理程序	7.1.4 通过ActionScript注册事件监听器	7.1.5 使用事件属性定义行内的事件处理程序	7.1.6 了解事件流	7.1.7 可用的基本交互事件	7.2 响应键
-----------------	-------------------	--------------------------	-------------------------------------	---------------------	-----------------	--------------------------	---------------------------------------	-------------------------	---------------	-------------------	---	-----------------------------	----------------	--------------	-------------	-------------	----------------	--------------	-----------------	----------------	------------------	-----------------	----------------	------------	------------------	----------------	------------------	------------------------	----------	-----------------	----------------	------------------------------------	-----------	---------------	---------------	----------------------	----------	--------------	-------------------	------------------	-----------------	-------------	-----------------	-----------------	-----------------	-----------------	------------------------	---------------	-----------	-----------	----------	-----------	--------------	------------	-----------	--	-------------	-----------	-------------	--------------	-------------------	------------	-------------	-------------	----------------	----------	--------------	------------------	------------------	-----------------	-------------	---------------	--------------------	---------------	-------------------------	----------------	---------------	--------------	------------	-----------------	-----------------	--	----------------	--------------	------------	------------	-----------	-------------	-------------	---------------	-----------------	---------------	------------------	----------------------	---------------	---------------	----------------------	------------------	----------------	-----------------------	-------------------	------------------------	-------------------------------	---------------	-------------	---------------------	---------------	-------------	-----------	-----------------	--------------	---------------	------------	---------------	---------------------	--------------	----------------	-------------	--------------	-------------------	-----------	-------------	------------	-------------	-----------------------	---------	-------------	------------------	----------------------	----------------	-------------	-----------------------	---------------	---------------	---------------	-------------	---------------------	---------------------	------------------	------------	------------------	------------	------------------------	-----------------------------	-------------------------	-------------	-----------------	---------

- 盘事件 7.2.1 获取按键 7.2.2 响应组合键事件 7.3 响应鼠标事件 7.3.1 获取鼠标位置属性 7.3.2 控制鼠标指针 7.3.3 捕获鼠标滚轮事件 7.3.4 自定义的鼠标左键双击事件 7.3.5 鼠标与键盘的配合 7.4 自定义Flash Player右键菜单 7.4.1 使用ContextMenu对象 7.4.2 使用ContextMenuItem对象 7.5 使用Event对象第2篇 Flex组件的呈现和布局应用篇 第8章 使用Flex UI控件定义用户界面 8.1 控件的基本使用方法 8.1.1 基本的属性设置 8.1.2 使用Alert控件 8.1.3 使用基于按钮的控件 8.1.4 使用基于文本框的控件 8.1.5 动态使用组件 8.1.6 日历控件 8.2 使用工具提示 8.2.1 设置工具提示的样式 8.2.2 创建出错工具提示 8.3 使用基于菜单的控件 8.3.1 使用Menu控件 8.3.2 菜单项类型 8.3.3 菜单数据提供者 8.3.4 创建MenuBar控件 8.3.5 创建PopUpMenuButton控件 第9章 使用Flex容器组件——基于容器的布局规则 9.1 基于容器的定位 9.1.1 使用绝对定位 9.1.2 使用坐标系 9.1.3 使用基于限制的布局 9.1.4 了解容器BOX模型 9.2 组件幅面大小的计算方法 9.2.1 决定幅面大小的属性 9.2.2 Flex实施布局的过程 9.2.3 决定组件的幅面大小 9.2.4 使用基于百分比的尺寸 9.2.5 溢出和裁剪 9.2.6 改变Application容器的幅面大小的注意事项 第10章 使用Flex容器组件——基本操作和范例 10.1 使用布局容器 10.1.1 使用Grid容器实现类似HTML表格的布局 10.1.2 使用Tile容器实现类似浮动定位的布局 10.1.3 使用Panel容器和ControlBar容器 10.1.4 使用TitleWindow容器创建弹出对话框 10.1.5 使用ApplicationControlBar容器创建主菜单 10.1.6 使用DividedBox、HDividedBox和VDividedBox分隔容器 10.1.7 使用Spacer组件处理布局 10.2 使用导航容器 10.2.1 使用Accordion导航容器 10.2.2 使用TabNavigator导航容器 10.2.3 使用ViewStack导航容器 10.3 Application容器和组件的创建顺序 10.3.1 组件创建的顺序 10.3.2 设置语言特性属性 10.4 动态重复控件和容器 10.4.1 设置Repeater组件 10.4.2 引用创建的重复组件 10.4.3 引用Repeater组件 10.4.4 在事件中使用数据源元素 第11章 使用样式表呈现组件 11.1 在MXML文档中应用CSS 11.1.1 使用嵌入样式定义 11.1.2 外部引用式 11.1.3 行内样式 11.1.4 StyleManager类 11.1.5 使用getStyle()方法和setStyle()方法 11.1.6 规则集 11.1.7 注释 11.1.8 关键字和字符串 11.2 CSS的选择符语法 11.2.1 类型 (Type) 选择符 11.2.2 类 (class) 选择符 11.2.3 Application选择符和global选择符 11.3 属性值的数据类型和数据单位 11.3.1 长度 (Length) 11.3.2 颜色 11.3.3 时间格式 11.3.4 数组格式 11.3.5 在样式表中嵌入资源 11.4 样式表的继承和层叠规则 11.4.1 Flex和W3C CSS规范的不同 11.4.2 样式的叠加和选择符的优先级 11.5 使用Flex Builder创作和应用样式表 11.6 在运行时加载和卸载样式表 11.6.1 在运行时加载样式表 11.6.2 加载运行时样式表所触发的事件 11.6.3 在自定义组件中加载运行时样式表所要注意的事项 第12章 使用外观和主题改变组件的呈现 12.1 使用外观改变组件的呈现 12.1.1 使用Flash CS3创建组件外观 12.1.2 了解scale9Grid切片 12.1.3 使用Photoshop创建组件外观 12.1.4 在创作环境中使用样式应用和更改组件外观 12.2 创建和使用主题 12.2.1 应用主题 12.2.2 创建主题SWC文件 12.2.3 将主题SWC文件用做运行时样式表 第13章 使用文本框和动态呈现文本 13.1 使用TextField动态设置文本框 13.1.1 使用TextFormat动态设置文本格式 13.1.2 使用HTML标签动态设置文本格式 13.2 使用样式表动态设置文本格式 13.2.1 使用CSS呈现文本 13.2.2 styleSheet的方法和属性简表 第14章 为Flex应用程序创建自定义组件 14.1 使用MXML文件创建自定义组件 14.2 使用ActionScript类创建自定义组件 14.3 使用Flash CS3和Flex Component Kit for Flash CS3配合创建自定义组件 14.4 在程序中使用元数据标签的用法 第15章 为组件行为定义动画效果 15.1 使用行为、触发器和动画效果 15.1.1 使用MXML组件应用行为 15.1.2 使用ActionScript代码应用行为 15.1.3 可用的触发器属性 15.2 内建动画效果的基本使用 15.2.1 内建动画效果的基本使用方法 15.2.2 并发效果和顺序效果 15.2.3 为效果应用缓动 15.2.4 为数据应用效果 第16章 创建视图和为视图状态变化应用过渡效果动画 16.1 使用Flex视图 16.1.1 关于视图 16.1.2 创建和使用Flex视图 16.1.3 Flex视图状态是什么 16.1.4 在视图中添加和删除组件 16.1.5 视图状态和导航容器的比较 16.1.6 视图之间的逻辑关系 16.2 为视图状态变化创建和使用过渡 16.2.1 过渡和效果的

<<至理>>

- 对比 16.2.2 创建过渡 16.2.3 定义多个过渡 16.2.4 并发过渡和顺序过渡 16.2.5 定义效果的目标 16.2.6 定义效果的起始和结束值 16.2.7 在过渡中使用动作效果和布局
- 16.2.8 使用效果过滤器 16.3 响应视图状态事件 第17章 使用Flex 4和Gumbo组件——MXML
- 2009标签语法和视图状态语法 17.1 安装和配置开发环境 17.2 开发第一个Flex应用程序 17.3
- 基本语法 17.3.1 语言命名空间 17.3.2 组件命名空间 17.3.3 组件库清单 17.3.4
- 本地源和导入包 17.3.5 命名空间的合并 17.3.6 新增的编译器标签 17.3.7 兼容Flex 3
- 17.4 视图状态语法 17.4.1 在视图状态中指定组件实例 17.4.2 组件的重现 17.4.3
- 使用状态指定组件属性值 17.4.4 视图状态分组 17.4.5 自定义创建策略 第18章 使用Flex 4
- 和Gumbo组件——FXG文档和Gumbo组件应用 18.1 了解和使用FXG 1.0 18.1.1 FXG文档结构
- 18.1.2 元素类型 18.2 定义FXG图形 18.2.1 使用FXG路径定义图形 18.2.2 使用基本的FXG形状定义图形
- 18.2.3 位图 18.2.4 FXG填充、线条、可视性和混合模式 18.2.5 定义FXG文本 18.3 坐标空间转换 18.4 使用Gumbo组件 18.4.1 定义组件外观的基本方法
- 18.4.2 定义组件中子节点内容的外观 18.4.3 Gumbo组件的布局 18.4.4 增强的应用第3篇 Flex数据集成和数据呈现篇 第19章 Flex组件的数据集成基础——数据组件、数据绑定和集合
- 19.1 使用数据绑定 19.1.1 在ActionScript代码中实现数据绑定 19.1.2 数据绑定的原理 19.2 使用集合 19.2.1 了解Flex集合框架 19.2.2 使用方法处理集合 19.2.3 集合的过滤 19.2.4 集合的排序 19.2.5 重置集合视图 19.3 基于列表的控件和数据提供者
- 19.3.1 定义数据提供者 19.3.2 使用TileList和HorizontalList 19.3.3 使用DataGrid 组件和AdvancedDataGrid组件 19.3.4 使用OLAPDataGrid组件进行联机分析 19.3.5 响应事件和获取选中的项目行 19.3.6 列表控件单元格的呈现 19.3.7 列表控件的拖曳 19.4 使用图表组件 第20章 Flex组件的数据集成基础——数据格式化和数据验证 20.1 格式化数据
- 20.1.1 数字格式化 (NumberFormatter) 20.1.2 货币格式化 (CurrencyFormatter) 20.1.3 电话号码格式化 (PhoneFormatter) 20.1.4 邮政编码格式化 (ZipCodeFormatter) 20.1.5 日期格式化 (DateFormatter) 20.1.6 自定义格式化 20.2 验证数据 20.2.1 验证器的基本使用方法 20.2.2 数字验证器 20.2.3 货币验证器 20.2.4 电话号码验证器 20.2.5 字符串验证器 20.2.6 ZIP验证器 20.2.7 信用卡验证器 20.2.8 日期验证器 20.2.9 社保代码验证器 20.2.10 电子邮件地址验证器 20.2.11 正则表达式验证器 第21章 Flex组件的数据集成基础—— Flex和XML集成开发 21.1 使用W3C DOM处理XML文档 21.1.1 了解XML文档对象 21.1.2 DOM基于节点的浏览 21.1.3 创建XMLDocument对象、解析XML文档 21.1.4 加载和解析外部的XML文档 21.1.5 使用XMLNode对象处理XML文档 21.1.6 使用XMLDocument对象处理XML文档 21.2 使用XMLDocument对象进行密码验证 21.3 使用E4X处理XML 21.3.1 用于实现E4X的ActionScript类 21.3.2 创建XML对象和XMLList对象 21.3.3 访问XML结构 21.3.4 修改XML结构 21.3.5 使用XML对象的属性和方法 21.3.6 使用XMLList对象的方法 21.3.7 操控XML命名空间 21.3.8 将XML对象和XMLList对象转换为字符串 21.3.9 使用E4X进行密码验证 21.3.10 使用标签和标签 第22章 创建网络应用程序和部署网络应用程序 22.1 安装和设置网络应用程序的运行环境 22.1.1 安装IIS 22.1.2 通过服务器请求网页 22.2 浏览器怎样从服务器接收和发送数据 22.3 Flex应用程序接收和发送数据 22.3.1 创建和配置ASP网页 22.3.2 创建Flex应用程序 22.3.3 测试应用程序 22.3.4 使用HTTPService组件要点解析 22.3.5 如何操控数据库 22.4 Flex应用程序结合ASP.NET、JSP和PHP 22.4.1 ASP.NET网页与Flex结合——基本操作 22.4.2 ASP.NET网页与Flex结合——操控数据库 22.4.3 JSP网页与Flex应用程序的结合 22.4.4 JSP网页与Flex结合——操控数据库 22.4.5 PHP网页与Flex应用程序的结合 22.4.6 PHP网页与Flex结合——操控数据库 22.5 使用Form容器创建表单布局 22.5.1 表单布局 22.5.2 定义默认的提交按钮 22.5.3 指定必须填写的字段 22.5.4 存储和验证表单数据 22.6 部署应用程序 第23章 使用LCDS和BlazeDS——Java 平台的AMF-RPC实现 (AMF3) 23.1 安装和配置运行环境 23.1.1 安装、配置JRE和JDK 23.1.2 安装和配置Tomcat 23.1.3 安装和配置LCDS 23.2 第一个程序：嗨！

- AMF-RPC 23.3 操控数据库—雇员管理应用程序 23.4 了解LCDS工作原理 23.4.1 LCDS工作原理
 23.4.2 LCDS和数据类型 23.5 使用BlazeDS 第24章 使用LCDS和BlazeDS——消息服务和数据管理服务
 24.1 使用消息服务 24.1.1 使用RTMP消息服务 24.1.2 使用JMS消息服务
 24.2 数据管理服务 24.2.1 数据管理服务的服务端配置 24.2.2 创建Flex应用程序作为数据管理服务的客户端
 第25章 使用FluorineFx和Flash Remoting MX——.NET平台的AMF-RPC实现 (AMF3和AMF0)
 25.1 安装和配置.NET应用程序运行环境 25.2 使用FluorineFx 25.2.1 安装和配置FluorineFx
 25.2.2 定义一个新的.NET应用程序 25.2.3 第一个程序：嗨！
 AMF-RPC 25.3 操控数据库—雇员管理应用程序 25.4 了解FluorineFx工作原理 25.5 使用Flash Remoting MX for.NET (AMF0)
 25.5.1 安装和配置Flash Remoting MX for .NET
 25.5.2 定义一个新的.NET应用程序 25.5.3 第一个程序：嗨！
 AMF-RPC 25.5.4 操控数据库—雇员管理应用程序 25.5.5 数据类型转换 第26章 使用AMFPHP和SabreAMF——PHP平台的开源AMF-RPC实现 (AMF3和AMF0)
 26.1 安装和设置PHP网络应用程序的运行环境
 26.1.1 下载和安装PHP应用程序服务器 26.1.2 安装和配置Web服务器
 26.1.3 配置PHP使用PDO 26.2 使用AMFPHP 26.2.1 安装和配置AMFPHP
 开发环境 26.2.2 定义一个新的PHP应用程序 26.2.3 第一个程序：嗨！
 AMF-RPC 26.2.4 雇员数据管理系统—使用AMFPHP操控数据库 26.2.5 使用AMF0格式
 26.3 了解AMFPHP工作原理 26.4 使用SabreAMF—PHP平台的AMF-RPC实现 第27章 在Flex应用程序中使用Web服务
 27.1 使用WebService组件直接调用Web服务 27.1.1 查看Google搜索服务的接口、方法和属性
 27.1.2 使用Google搜索服务 27.2 使用代理操纵Web服务 27.2.1 使用LCDS和Blaze作为代理
 27.2.2 使用FluorineFx和Flash Remoting MX作为代理 27.2.3 使用AMFPHP作为代理
 第4篇 Flex开发高级应用主题 第28章 Flex调试原理与范例实践 28.1 错误和异常情况
 28.2 设置Flex调试环境 28.2.1 关于Debug Player的适用版本 28.2.2 卸载播放器控件
 28.3 Flex Builder调试透视图的使用方法 28.3.1 “Console”视图窗口和trace () 函数
 28.3.2 “Debug”视图窗口 28.4 使用调试器脚本调试工具范例 28.5 异常情况处理
 28.5.1 使用Error对象处理异常错误信息 28.5.2 使用内建的异常类型 第29章 配合FMS3开发Flash实时通信应用程序
 29.1 了解Flash Media Server 3服务器 29.1.1 了解通信应用程序的开发
 29.1.2 了解RTMP 29.1.3 了解流媒体服务器 29.2 安装和配置Flash Media Server 3服务器运行环境
 29.2.1 关于安装的文件 29.2.2 启动和停止服务器 29.3 安装开发和调试环境
 29.3.1 安装创作环境 29.3.2 安装、配置麦克风和照相机 29.4 创建第一个Flash通信应用程序
 29.4.1 创建Flash通信应用程序 29.4.2 创建发布视频的Flex应用程序 29.4.3 创建订阅视频的Flex应用程序
 29.4.4 测试Flash通信应用程序 29.5 Flash Player设置功能
 29.5.1 保密性和本地存储设置 29.5.2 麦克风和照相机侦测与设置 第30章 使用Flex开发AIR应用程序
 30.1 使用Flex Builder 3开发、打包第一个AIR应用程序 30.1.1 创建应用程序
 30.1.2 打包应用程序 30.1.3 安装应用程序 30.1.4 创建从网络安装应用程序 30.2 AIR应用程序的文件结构
 30.2.1 创建应用程序的描述文件 30.2.2 使用AIR组件 第31章 使用Flex Ajax Bridge (FABridge) 和JavaScript整合开发
 31.1 在网页中如何与JavaScript结合使用
 31.1.1 使用ExternalInterface类调用JavaScript函数 31.1.2 使用JavaScript调用ActionScript函数
 31.2 使用FABridge附录A 保留的关键字附录B AMF-RPC历史及重要概念附录C 键盘键和键控代码值对照表附录D 安装和配置数据库附录E 参考文献及资料来源

章节摘录

在浏览器中打开该文件，可以看到与图1~10相同的效果。

1. Flex与Flash的关系虽然最终形成的Flash影片都是在FlashPlayer，运行环境中运行的，但是，Flex开发与单纯的Flash开发有一些不同，因为Flex开发是基于一个被称为Flex框架的基础之上的，因此，基于这种框架开发的程序也被称为Flex应用程序，这是Flash影片应用程序的一个子集。

并且由于与之紧密相连的开发语言和创作环境的差异，二者的开发有很多不同。

1) 开发形式不同 单纯的Flash开发使用FlashCS3创作环境，代码是沿着时间轴组织的，不管是将代码直接写在时间轴的帧中还是以外部ActionScript类文件的形式被引用，它们最终都是围绕时间轴这个核心展开的，正是这个认识的原因，我建立了“基于时间轴完整系统的Flash开发的知识体系”，这个知识体系经历了时间的考验，现在已经根深叶茂。

牢牢记住“Flash是基于时间轴的应用程序”是你学好用好Flash的关键。

Flex开发则是基于一个我们大家都熟悉的Form，这个英文单词有很多中文译法。

在开发桌面应用程序时，Form被译做窗体，这来源于我们最常使用的视窗操作系统，这翻译非常贴切；在开发基于浏览器的网络应用程序时，Form被译做表单，网页表单是网络开发中用于与用户交互的主要形式，相信有基本动态网页开发经验的用户肯定都熟悉它。

在Flex中的Form开发形式最接近于桌面应用程序的窗体，但由于可以嵌入在浏览器中执行，所以很多人也会将其视做表单，不过这种区分并无多大意义。

在FlexBuilder3创作环境中，直接在单个Form中编写代码就可以完成所有功能，并且，也可以使用代码引用外部Actionscript类文件，类文件中的定义最终也被编译在同一个Form中。

2) 开发语言不同 单纯的Flash开发仅仅使用Actionscript语言就可以了，而Flex开发则需要MXML标签语言和Actionscript语言的结合。

MXML标签语言通过FlexSDK被转化成Actionscript代码，最终，它们都被编译为FlashPlayer可以识别的P代码。

编辑推荐

《至理：精通Flex网络开发技术：整合ActionScript/JavaScript/Ajax动态网站》一书揭开源于开发实践的技术经验，从基础到精通：Flex开发基础核心、组件的呈现和布局应用、数据集成、Flex调试原理与范例实践、配合FMS3开发Flash实时通信应用、AIR开发、使用Flex Ajax Bridge (FABridge) 和JavaScript整合开发。

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>