PEARSON

移动开发经典丛书

iOS Auto Layout Demystified
Second Edition

iOS Auto Layout 开发秘籍

(第2版)

[美] Erica Sadun 著
孟立标 译

PEARSON

清华大学出版社

移动开发经典丛书

iOS Auto Layout 开发秘籍

(第 2 版)

[美] Erica Sadun 著

孟立标 译

清华大学出版社

北 京

图书在版编目(CIP)数据

iOS Auto Layout 开发秘籍(第 2 版)/(美) 撒敦(Sadun, E.) 著；孟立标 译. —北京：清华大学出版社，2015
(移动开发经典丛书)

书名原文：iOS Auto Layout Demystified, Second Edition

ISBN 978-7-302-38306-2

Ⅰ. ①i… Ⅱ. ①撒… ②孟… Ⅲ. ①移动终端—应用程序—程序设计 Ⅳ. ①TN929.53

中国版本图书馆 CIP 数据核字(2014)第 241433 号

责任编辑：王　军　于　平
装帧设计：孔祥峰
责任校对：曹　阳
责任印制：宋　林

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 刷 者：北京鑫丰华彩印有限公司
装 订 者：三河市溧源装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　印　张：15　　字　数：365 千字
版　　次：2015 年 1 月第 1 版　　印　次：2015 年 1 月第1 次印刷
印　　数：1～3000
定　　价：49.80 元

产品编号：059185-01

译 者 序

iOS 已陪伴我们走过了7年，搭载着 iOS 系统的设备，例如 iPhone、iPod touch、iPad，以其丰富的功能和出色的用户体验，改变了人们日常的生活、工作和娱乐方式。iOS 7的发布，表明 iOS 进入了新的纪元，它呈现给用户一个扁平、简约而又明朗的界面。

如同一对初次邂逅的男女，往往第一眼便决定了是否会一见钟情。推而广之，开发者往往会花费大量时间去雕琢出一张充满魅力的“脸”，让它无论在浅浅微笑时，还是在扮鬼脸时都富有吸引力。但这确实不是一件容易的事情。在 iOS 界面方面，Apple 做了许多卓有成效的努力。无论是从 OS X 平台引入了 Autosizing 技术，还是到 iOS 6 的时候，引入了自动布局(Auto Layout)技术。自动布局可以实现早前布局技术所无法实现的布局要求。

但是相对而言，自动布局还是一种较新的技术，目前市面上的书籍，也没有对该部分内容进行比较细致的讲解，让一些初次接触自动布局的开发者感觉无从下手。《iOS Auto Layout 开发秘籍(第 2 版)》可以说是弥补了这个空白，难能可贵的是，这本书不仅涵盖了 iOS 和 OS X 平台，而且几乎深入阐述了自动布局技术的方方面面，既有深度又有广度。

该译本出版的时候，可能 iOS 8 已经发布正式版。在 iOS 8 中，自动布局也会有显著的改变。不过，本书仍是学习自动布局的不二选择。一方面，因为它是比较基础而又系统的，可以帮助读者掌握自动布局的一些基本原理和实现；另一方面，了解一种技术最初的形态，可以帮助改善应用的兼容性。最后，诚如作者所言，“我的意图是使你在放下这本书之后，便能拥有 Auto Layout 方面的坚实基础”。

本书由孟立标翻译，参与翻译活动的还有孔祥亮、陈跃华、杜思明、熊晓磊、曹汉鸣、陶晓云、王通、方峻、李小凤、曹晓松。限于时间、精力以及本人的专业水平能力，书中难免会有疏漏错误之处，敬请读者批评指正。

译 者

作 者 简 介

Erica Sadun 是数十本畅销书的作者、合著者和供稿者，这些书涉及编程、数字视频、数字摄影和 Web 设计，包括广受欢迎的 *The Core iOS 6 Developer's Cookbook* 的第 4 版。她最近在 TUAW.com 上发表博文，过去在 O'Reilly 的 Mac Devcenter、Lifehacke 和 ArsTechnica 上发表博文。除了是数十款 iOS 原生应用的作者外，Sadun 拥有佐治亚理工学院图形可视化和可用性研究中心计算机科学专业的博士学位。她是一名极客、一名程序员和一名著作者。写作之余，她和她的极客丈夫共同养育三个“尚在训练中”的小极客。

致　　谢

单凭一己之力是无法完成一本书的。在此，我想要感谢我的团队，是他们使本书得以顺利出版。可爱的 Trina MacDonald 为本书的书名开了绿灯，因此最终提供给你一个读到本书的机会。Chris Zahn 是我出色的开发编辑，还有 Olivia Basegio 使任何事情得以顺利进行，即使当事情变得一团糟时也是如此。

我向全体 Addison-Wesley/Pearson 产品团队表示感谢，特别是 Kristy Hart、Betsy Gratner、Kitty Wilson、Nonie Ratcliff 和 Chuti Prasertsith。

同样要感谢 Neil Salkind，我这么多年的代理人；Stacey Czarnowski，我的新 Neil。感谢 Rich Wardwell，本书第 1 版的技术编辑；Mike Shields 和 Ashley Ward，本书的技术编辑。还要感谢在 TUAW 和其他我工作过的博客的现在和以前的同事。

我深深受惠于广大的 iOS 开发者社区，他们在 IRC 上支持我，帮助我阅读本书的草稿，提供反馈。在此，特别感谢 Oliver Drobnik、Aaron Basil (of Ethervision)、 Harsh Trivedi、Alfonso Urdaneta、Michael Prenez-Isbell、Alex Hertzog、Neil Taylor、Maurice Sharp、Mike Greiner、Rod Strougo、Chris Samuels、Hamish Allan、Jeremy Tregunna、Lutz Bendlin、DiederikHoogenboom、Matt Yohe、MahipalRaythattha、Neil Ticktin、Robert Jen、 Greg Hartstein、Jonathan Thompson、Ajay Gautam，Shane Zatezalo、Wil Macaulay、Douglas Drumond、Bill DeMuro、Evan Stone、Alex Mault、David Smith、Duncan Champney、Jeremy Sinclair、 August Joki、 Mike Vosseller、 Remy “psy” Demarest、 Joshua Weinburg、EmanueleVulcano 和 Charles Choi。他们的技术、建议和反馈使得本书的顺利付梓成为可能。如果我遗漏了任何帮助过我的人，请原谅我的疏忽。

特别感谢我的丈夫和孩子，你们太好了！

前　言

Auto Layout 重新构思了开发者创建界面的方式。它创建了一个灵活、强大的系统，来描述视图和它们的内容是如何相互关联的，它们和它们占据的窗口和父视图是如何关联的。与旧的设计方法相比，这种技术为布局提供了难以置信的控制，比 frame、spring、strut 所允许的范围更广阔。也有被一些激怒的开发者中伤的，使得 Auto Layout 获得了难以使用、令人沮丧的名声，特别是通过 Interface Builder(IB)使用时。

这就是本书存在的原因。你将通过一些包含大量解释和提示的示例来揭示 Auto Layout 的优势。本书并不过多介绍类文档，而是通过简单的步骤来学习该系统的工作原理，以及它为什么比你初次所想的更强大。你将看到一些常见的设计场景，并发现使用 Auto Layout 是一种乐趣，是最佳实践，而不是一项累人的工作。

你也会探索很多 Auto Layout 的优点。它是一项非常有用的技术：

- **Auto Layout 是声明性的**。表达界面时不必担心这些规则是如何实现的。只要描述这个布局即可，可以让 Auto Layout 计算 frame。
- **Auto Layout 是描述性的和相关性的**。你需要描述项在屏幕上是如何相互关联的。可以忘掉尺寸和位置。重要的只是关系。
- **Auto Layout 是集中的**。无论在 IB 还是在你自己代码里的布局区域，Auto Layout 规则倾向于迁移到一个简单的关系，使它更易于检查和调试。
- **Auto Layout 是动态的**。你的界面会在需要响应用户和源自应用的改变时而更新。
- **Auto Layout 是可本地化的**。使用 Auto Layout 可以征服世界。它在维护界面完整性时，适应不同的单词和词组长度。
- **Auto Layout 是表达性的**。你可以描述比能在旧的 spring-strut 系统中更多的关系。不只是“吸附这条边”或者“沿着这个坐标轴改变尺寸大小”，它可以表示一个视图关联到另一个视图的方式，而不仅仅是它的父视图。
- **Auto Layout 是增量式的**。可以根据你自己的时间表使用它。可以添加它，将它作为应用和界面的一部分，或者将其作为一个完整的 Auto Layout 经历。Auto Layout 提供向后兼容，使你可以使用所有 spring 和 strut、所有约束或者两者的混合，来创建自己的界面。

本书旨在为你提供灵感。我试着去演示使用 Auto Layout 来创建可交互元素、动画和其他超越你可能在 IB 中遇到的特性的示例。这些章节为 Auto Layout 工作提供了一个启动平板，引入了一些可以拓展设计可能性的不常见特性。

正如书名所提示的，本书基本的目标读者是 iOS 开发者。我在可能覆盖的地方引入了 OS X。因此，如果你是一个 OS X 开发者，不会被冷落。本书的内容主要还是针对 iOS。当你阅读时请记住这一点。

Auto Layout 已经对我的日常开发产生了深远的影响。我撰写本书，希望 Auto Layout 也能对你的开发工作带来深远影响。我的意图是使你在放下这本书之后，便能拥有 Auto Layout 方面的坚实基础。如果我幸运的话，这本书会给你“我找到了！”的惊喜一刻，本书将引导你前进。

——Erica Sadun

这本书的内容安排

这本书提供了实际的 Auto Layout 教程。以下是本书内容介绍。

- 第 1 章，“Auto Layout 介绍”——准备好了吗？本章解释 Auto Layout 背后的基本概念。你将学习为什么应当在应用里使用 Auto Layout，以及为什么它必须是一个满足约束的系统。
- 第 2 章，“约束”——在 Auto Layout 中，通过声明关于视图的规则来创建界面。你添加的每个布局规则创建一个关于界面的某部分如何布局的要求。这些规则根据你提供给系统的一个数值优先级来排定等级，相应地，Auto Layout 创建你的界面的可视化表现。本章介绍约束以及布局规则，并且解释了为什么你的规则必须是无歧义的、可满足的。
- 第 3 章，“Interface Builder 布局”——在 Interface Builder 中使用基于约束的设计有时对于 Auto Layout 开发新手来说，可能是一个令人沮丧的经历。iOS 7 和 Xcode 5 完全更新后，本章教你一些你需要的窍门，使 IB 精确地创建你想要的界面。
- 第 4 章，“可视化格式”——本章探索可视化约束看起来如何，你如何创建它们，以及如何在项目中使用它们。你将学到度量字典和约束选项是如何拓展可视化格式来获取更多的灵活性。本章介绍了大量示例，用于演示这些格式以及探索它们产生的结果。
- 第 5 章，“调试约束”——约束有时比较晦涩。你创建约束时使用的代码和界面文件并不易于细读。它只提供一些“有用的”Xcode 日志消息，这让一些开发者十分纠结。本章专注并聚焦于底层约束并帮助调试你的工作。
- 第 6 章，“使用 Auto Layout 创建”——对 Auto Layout 的设计改变了你创建界面的方式。它是一个描述性的系统，远离了准确的度量，例如 frame 和 center，差别比较大。你将注意力放在视图间的关系上，它描述了屏幕上的某项是如何跟随另一项的。通过基于约束的规则，在设计中揭示了这种自然关系，并详细描述了它们。本章介绍 Auto Layout 设计的表达，聚焦于它的基本原则，并提供了一些展示其特性的示例。
- 第 7 章，“布局解决方案”——本书前面章节关注于基础知识和原理。本章介绍解决方案。你已经学习了各种现实世界的挑战，以及 Auto Layout 是如何为日常开发工作提供切实可行的答案。这些主题就像一个摸彩袋，展示开发者通常会提出的请求。
- 附录 A，“练习参考答案”——该附录提供了所有章节后的练习题的答案。

关于示例代码

本书沿用了 *iOS Developer's Cookbook* 系列的风格。书中的 iOS 示例代码总是以单个 main.m 文件开始，你会在其中发现支撑该示例的应用的核心部分。人们一般在开发 iOS 或

者 Cocoa 应用时不采用这种方式，但是它提供了一种展现单一想法的绝佳方式。当读者需要在许多文件中搜寻，并试图找出哪些文件是相关的，哪些是无关的时候，要讲清楚这一过程就很难了。提供单个启动点浓缩了这个过程，使得在单个代码块中便能获悉整个想法。

本书所提供的代码并非遵循标准的日常最佳实践方式。书中提供了精确的解决方案，你可以根据需要将它们纳入到你的工作中。书中的示例大都使用一个应用标识：com.sadun.helloworld。这使你的 iOS 设备避免同时被许多示例阻塞。每个示例替换前面一个，确保你的主屏幕保持相对整洁。如果想要同时安装若干示例程序，只需要编辑标识，添加一个独一无二的前缀即可，例如 com.sadun.helloworld.table-edits。

你也可以编辑自定义的显示名称，使应用在视觉上看起来截然不同。你的 iOS Team Provisioning Profile 匹配任何应用标识，包括 com.sadun.helloworld。这允许将编译后的代码安装到设备上，而无须更改该标识，只需要确保在每个项目的构建设置(build settings)中更新你的签名标识(signing identifier)。

本书中还有一些浅显易懂的 OS X 代码。这不是一本以 OS X 为中心的书籍(你可以从书名中猜到这一点)，但是在必要的地方，覆盖到了 OS X 主题。本书的篇幅主要花费在 iOS 上，因此请原谅任何在 OS X 方面的失误，请务必写邮件帮助纠正任何错误。

获得示例代码

你可以在开放源码托管站点 GitHub 上的 http://github.com/erica/Auto-Layout-Demystified 页面上，找到本书的示例代码。其中可以找到按章节划分的源码，这些源码提供了本书涉及的示例文件。

正如后面解释的，你可以通过直接使用 git 或者单击 GitHub 的下载按钮来获取示例代码。在我撰写本书时，它位于页面的右边。它使你能够以一个 zip 或者 tarball(.tar)压缩文件的形式获取整个代码库。

获取 Git

可以通过使用 git 版本控制系统来下载本书的源码。git 的一个 OS X 实现可以通过 http://code.google.com/p/git-osx-installer 获取。OS X git 实现包括了命令行和 GUI 解决方案，这样，你可以去寻找一个最适合自己开发需求的版本。

获取 Github

Github(http://github.com)是最大的 git 托管的站点，拥有超过 150 000 个公共代码库。它既为公有项目提供了免费的托管，也为私有项目提供了付费选项。Github 拥有一个定制的 Web 界面，包括了 wiki 托管、问题跟踪以及对项目开发者社交网络的强调，使得它成为找寻新代码或者在现有库上展开合作的绝佳地方。可以在 Github 网站上注册一个免费账户，这使得你可以复制并修改这个代码库，或者创建自己的 iOS 开发源码项目与他人分享。

贡献！

示例代码永远不会是最终版本。它会随着 Apple 更新它的 SDK 和 Cocoa Touch 库而持续演进。加入我们吧！你可以通过建议需要修复的 bug、提出修复 bug 的方式以及扩展提

供的代码参与进来。Github 允许你创建代码库的分支，使用你自己的微调和特性来扩展它们，然后将它们分享回主代码库。如果你提出一个新的想法或者方法，请告诉我。我的团队和我非常乐于将好的建议纳入到代码库和本书的下一个版本中。

联系作者

如果你有关于本书的任何评论或者疑问，请给我发送邮件(erica@ericasadun.com)或者访问 Github 库并联系我。

编者按：我们想要听到你的声音！

作为本书的读者，你是我们最重要的评论家和评论员。我们非常重视你的观点，并希望知道什么是我们做得好的，什么是我们可以做得更好的，什么领域的书籍是你希望我们出版的，以及任何其他你愿意传达给我们的想法。

你可以发送 Email 或者直接给我写信让我知道你喜欢还是不喜欢本书——以及我们该做些什么来使我们的书更具价值。

但请注意，我无法给予你任何与本书主题相关的技术问题的帮助，由于我收到的邮件数量较多，因此可能无法回复每一封邮件。

当你来信时，请确保包含本书的书名、作者以及你的名字和电话号码或者邮件地址。我会仔细地阅读你的评论，并将它分享给本书的作者和编辑人员。

E-mail: trina.macdonald@pearson.com

Mail: Trina MacDonald

Senior Acquisitions Editor

Addison-Wesley/Pearson Education, Inc. 75 Arlington St., Ste. 300

Boston, MA 02116

目　录

第 1 章

Auto Layout 介绍

Auto Layout(自动布局)颠覆了开发人员创建用户界面的方式。它提供了一种灵活而强大的系统，该系统可以描述视图与其内容相互之间的关系，也可以描述视图及其内容与父视图之间的关系。与原来的设计方法相比，这一技术具有令人难以置信的布局控制能力，其自定义范围比采用 frame、spring 和 strut 所能获得的范围更广。

Auto Layout 既有忠实的用户群，也有激烈的反对者。有人说它不好用、挫败感强，尤其是通过 Interface Builder(IB)使用时更是如此。这种说法有时候并不是空穴来风。虽然 Xcode 5 大大改善了这种情况(去掉了一些让人一头雾水、晕头转向的功能)，但这一技术仍然有待不断发展，直至完全成熟。

Auto Layout 是一种神奇的工具，它做到了使用先前技术的人做梦都想不到的事情。从处理边界情况(edge case)到创建视图之间的相互关系，Auto Layout 无所不能。此外，Auto Layout 与苹果公司的许多优秀的应用编程接口(API)兼容，包括动画(animation)、动画效果(motion effect)和精灵(sprite)。

基于此，本书应运而生。本书将通过示例来介绍 Auto Layout 的使用技巧，这些示例中会包含大量解释和提示。你不必再让类文档折磨得死去活来。相反，本书只用了几个简单的步骤，就讲明白了这一系统的工作原理、如何对它进行调整使其发挥更大的作用，以及为什么 Auto Layout 的功能强大得超乎想象。本书将提供一些常用的设计方案和最佳实践经验，使 Auto Layout 的使用成为一种享受，而不是苦差事。

1.1 Auto Layout 的由来

2012 年，Auto Layout 作为 iOS 6 版本的一部分，首次在 iOS 中露面。大约一年前的 OS X 10.7 Lion 中也有它的身影。Auto Layout 旨在取代原来基于 spring 和 strut 的 Autosizing 系统，它是一种全新的系统，用来构建视图之间的关系，指定视图与其父视图之间以及视

图与视图之间的关系。

Auto Layout 的前身是 Cassowary 约束解析工具包。Cassowary 是华盛顿大学的 Greg J. Badros 和 Alan Borning 为解决用户界面布局问题而开发的。Cassowary SourceForge 项目的网页(http://sourceforge.net/p/cassowary/wiki/Home/)上对它的解释如下：

Cassowary是一种递增式约束解析工具包，它能有效地解析线性等式系统和线性不等式系统。约束可能是需求，也可能是偏好。该工具包能快速地重新解析系统，并且支持UI应用程序。

Cassowary 是围绕这样一个重要的界面现象开发的：用户界面中生来就会出现不等关系和相等关系。Cassowary 开发了一种基于规则的系统，使开发人员能够描述视图之间的关系。这些关系是通过约束来描述的。约束是指一些规则，这些规则指出了一个视图相对于另一个视图的位置。例如，有些约束要求一个视图仅占据屏幕的左半边，也有些约束要求两个视图的底部始终对齐。

Cassowary提供了一种自动解析工具，将基于其约束系统的布局规则(对于数学达人来说，这些规则本质上就是一个联立线性方程组)转换成表达那些规则的视图几何特征。Cassowary 的约束系统功能强大而微妙。自从它首次亮相以来，已经移植到了 JavaScript、.NET/Java、Python、Smalltalk和C++中，并且通过Auto Layout移植到了Cocoa和Cocoa Touch中。

在 iOS 和 OS X 中，带约束功能的 Auto Layout 能够将视图在界面中进行有效的摆放。无论是通过 IB 还是通过代码提供规则，Auto Layout 系统都会将这些规则转换成视图框架。

1.2 使用 Auto Layout 的好处

有些开发人员不愿意使用 Auto Layout，原因有很多。有人觉得它太新奇、太古怪，有人觉得用它更新界面还需要做一些额外的工作，嫌麻烦。但是我认为这个工具值得使用。Auto Layout 在视图布局方面取得了重大突破，该系统有一些奇妙、新鲜和新颖之处。苹果公司的布局功能会使你的生活更轻松，界面更统一。此外，苹果公司还在 Auto Layout 中免费添加了与分辨率无关的布局。不管设备的几何形状、方向以及窗口大小如何，都能实现这一切布局功能。

Auto Layout 的工作原理是通过创建屏幕上的对象之间的关系来实现布局。它指定运行系统如何自动摆放视图，其结果是产生一组适合屏幕和窗口几何形状的健壮规则。使用 Auto Layout，可以描述一些约束，用来指定视图之间的关系；也可以设置一些视图属性，用来描述视图与其内容之间的关系。通过 Auto Layout 可以进行类似如下的请求：

- 将一个视图的尺寸与另一个视图的尺寸匹配，使两个视图始终保持相同的宽度。
- 无论父视图的形状如何改变，都将一个视图(或者一组视图)相对于父视图居中。
- 摆放一行视图时将几个视图的底部对齐。

- 将两个视图偏移一定的距离(例如，在两个视图之间添加标准的 8 点补白空间)。
- 将一个视图的底部与另一个视图的顶部绑定，使得移动一个视图时，两个视图一起移动。
- 防止图像视图在按自然大小看不到完整内容时收缩成一个点(即不压缩或剪切视图的内容)。
- 显示按钮时文本四周不要有太多的补白。

上述列表中的前 5 项描述了定义视图几何特征与布局的约束，建立了视图之间的可视化关系。后两项建立了视图与其呈现的内容之间的关系。使用 Auto Layout 时，这两种情况都会涉及。

下面是使用 Auto Layout 进行开发的一些优势。

1.2.1　几何关系

建立关系是 Auto Layout 的强项。图 1-1 显示了一个完全用 Auto Layout 构建的自定义 iOS 控件。用户可以用这个选择器选择颜色。每支画笔由一个尺寸固定的笔尖视图直接放在一个可伸缩的笔杆底视图上方构成。当用户进行选择时，两个视图一起上下移动来指示其当前选择。Auto Layout 约束确保每个笔尖恰好位于其笔杆的上方，每支画笔与其他画笔大小一致，并且每一对笔尖视图和笔杆底视图组成的画笔排成一行，底部对齐。

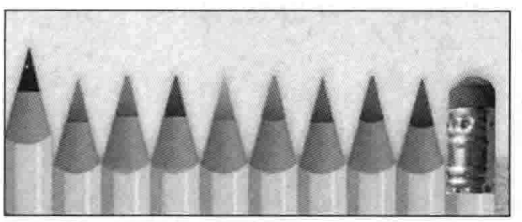

图 1-1　此画笔选择器自定义控件完全是用 Auto Layout 构建的

这里的画笔选择器是通过代码构建的。也就是说，有一个数据源提供了画笔的支数和每个笔尖的艺术特征。Auto Layout 通过描述画笔之间的关系，简化了扩展该控件的过程。你只需要指出“将每支新画笔放在右边，宽度与现有画笔一样，底部对齐”，即可将此选择器由 10 支画笔增加到 11 支、12 支，或者更多支。最重要的是，约束的变化可以制作成动画。当改变约束偏移量，使笔杆的长度发生变化时，笔尖会随之上下活动。

下面的代码说明了我在自己的项目中摆放这些画笔的方式：

```
// This sample extensively uses custom macros to minimize the
// repetition and wordiness of this code, while giving a sense of the
// design choices and layout vocabulary offered by Auto Layout.
// Read more about similar custom macros in Chapter 6.

- (void) layoutPicker
{
    for (int i = 0; i < segmentCount; i++)
    {
        // Add base
```