









非官方的iOS设计指南

有时候为iOS设计app并不是一件简单的事，但是如果你能找到正确的最新的苹果设备信息，并按照正确的方向，那么为iOS设计app或许会变得简单容易些。

关于这些指南

这些指南描述了如何遵守苹果的iOS 人机交互指南来设计app，而不是讲用自定义控件可以做成什么样的设计，有时候打破规则也很重要。该文档的目的并不是为一些复杂的设计问题提供解决方案。该文档是非官方的，将会定期更新和扩充内容，最近一次更新是2014年11月11日。

分辨率和显示屏规格 (Resolutions和Display Specifications)

Device	Retina	Portrait (px)	Landscape (px)
iPhone 6+		1080 x 1920	1920 x1080
iPhone 6		750 x 1334	1334 x 750
iPhone 5 5, 5S, 5C		640 x 1136	1136 x 640
iPhone 4 4, 4S		640 x 960	960 x 640
iPhone 1st, 2nd & 3rd Generation		320 x 480	480 x 320
iPad Air / Retina iPad 1st & 2nd Generation / 3rd & 4th		1536 x 2048	2048 x 1536
iPad Mini 2nd & 3rd Generation		1536 x 2048	2048 x 1536
iPad Mini, 1st & 2nd Generation		768 x 1024	1024 x 768

Points 和Pixels的区别

Pixels (像素) 是数字显示屏上我们可控制的最小物理元素，在一个特定屏幕尺寸中可以有多个像素，PPI (pixels-per-inch) 越高，则渲染的内容会越清晰。

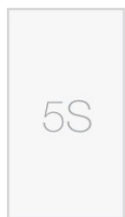
Points用以衡量分辨率。根据屏幕的像素密度，一个point可以包含多个像素 (比如在常规的retina屏上，1pt包含2 x 2的像素) 。

当你针对多个显示屏类型进行设计时，你应该以points进行思考，但以pixels进行设计。这意味着你仍需要以3种不同的分辨率来输出设计资产，不管你针对哪个分辨率设计应用程序。

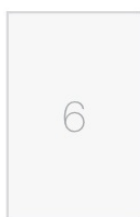
Device	Asset Resolution	PPI	Display Size
iPhone 6+	@3x	401	5.5"
iPhone 6	@2x	326	4.7"
iPhone 5 5, 5S, 5C	@2x	326	4.0"
iPhone 4 4, 4S	@2x	326	3.5"
iPhone 1st, 2nd & 3rd Generation	@1x	163	3.5"
iPad Air / Retina iPad 1st & 2nd Generation/ 3rd & 4th	@2x	264	9.7"
iPad Mini 2nd & 3rd Generation	@2x	326	7.9"
iPad Mini 1st Generation	@1x	163	7.9"
iPad 1st & 2nd Generation	@1x	132	9.7"

iPhone 6+ 缩减像素取样

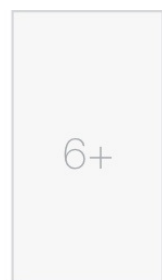
在iOS上渲染像素和物理像素（physical pixels）是等同的，只有一个例外：iPhone 6 Plus的Retina HD显示屏。由于它屏幕的像素分辨率要低于一个常规的@3x分辨率，所以被渲染内容会自动调整为原始尺寸的87%（从2208*1242像素来适应为1920*1082像素的显示屏分辨率）



320 x 568
Points

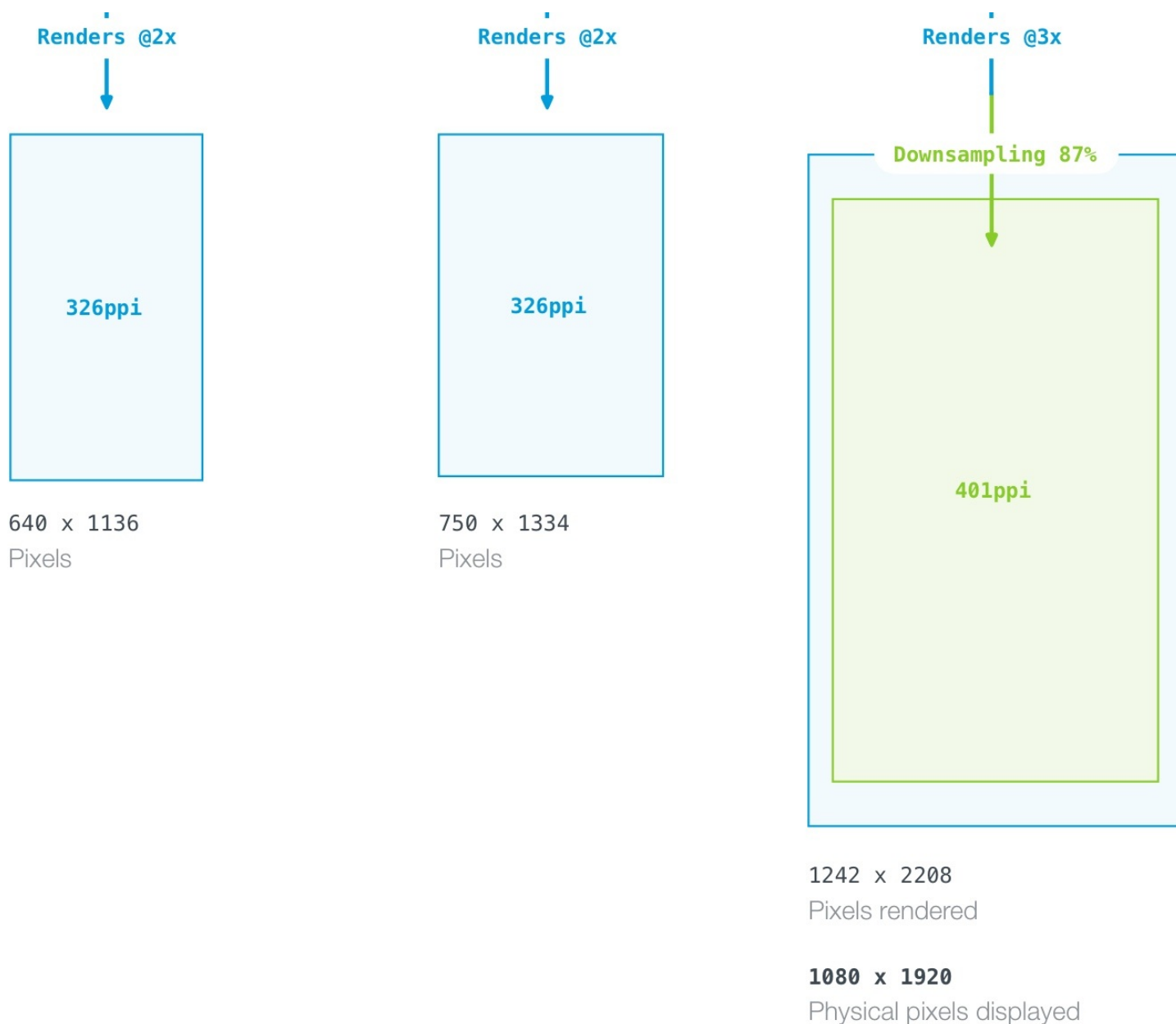


375 x 667
Points



414 x 736
Points





iPhone 5S, 6 以及6+显示屏区别的详细信息可参看：[The Ultimate Guide To iPhone Resolutions](#)

App Icons

Device	App Icon	AppStore Icon	Spotlight	Settings
iPhone 6+	180x180 px	1024x1024 px	120x120 px	87x87 px
iPhone 6, 5S, 5, 5C, 4S, 4	120x120 px	1024x1024 px	80x80 px	58x58 px
Old iPhones 1st, 2nd, 3rd Generation	57x57 px	1024x1024 px	29x29 px	29x29 px
Retina iPads Mini 2 & 3, Air, 3 & 4	152x152 px	1024x1024 px	80x80 px	58x58 px
Old iPads 1, 2, Mini 1	76x76 px	1024x1024 px	40x40 px	29x29 px

自动应用效果

多尺寸的应用程序icon通常被添加到应用程序包中，当在设备上渲染时，iOS会将效果应用在应用程序icon上。

(1) 圆角

圆角半径值已经不存在了。从iOS 7开始，app icon已经使用了超椭圆的形状。由于苹果没有发布该形状的官方模板，所以你得精确地使用[非官方的模板](#)。



圆角的图形不应该包含在最终的输出资产中，但如果你想要添加和应用程序icon拐角对齐的描边和阴影效果，那你可能还会用到圆角图形。

提醒：因为你想将应用效果和icon拐角对齐，所以如果你正使用超椭圆的形状对icon资产进行蒙版（遮罩），那要确保在遮罩外的区域不能使用任何透明的图形。应用程序icon不支持透明度，相反作为纯黑色进行渲染。如果你的遮罩不是百分百精确，那用户将会在圆角边缘看到黑色的锯齿。推荐将canvas的背景设置成和app icon一样的背景。

(2) 边框描边(某些情况下)

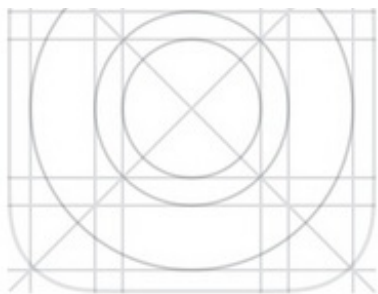
如果你使用的app icon有白色的背景，那么将会应用1pixel的灰色边框，以便更容易地识别icon的边缘。这只能在设置app和App Store中进行。

(3) 后续问题（iOS 6 和之前的版本）

在旧的iOS版本中，这些效果会自动应用：可以被禁用的圆角（和iOS 7+中使用的icon不同）、主屏幕上的阴影效果以及光泽效果等。

栅格系统





苹果开发了具有黄金分割比例的栅格系统，可用以正确地调整和对齐icon上的元素。不过，甚至是苹果设计师的原生app icon也没有完全严格地遵守栅格系统。所以如果你的icon上的元素在没有严格遵守栅格系统的情况下能更好地展示，那你可以考虑下打破一些固有的规则。

字体排版

所有的iOS版本中默认字体都是Helvetica Neue。从iOS 7开始，苹果使用了稍作修改的字体，但是在你的设计过程中使用原始的Helvetica Neue是极好的。除了默认字体外，你还可以使用很多可选的字体，你可以在[此查看完整的预置字体列表](#)。

自定义字体

从技术角度讲，True Type Font (.ttf)可以被用在iOS app中，但要注意许可问题。一般来说，使用完全免费或者商业化的字体是安全的。[MyFonts](#)最大数量地包含了可用在app中的许可字体。

调色板



iOS 7以后，苹果在操作系统和预装app中使用了更有生机和活力的调色板。虽然你可以使用上边默认的iOS调色板，但你也可以使用自己的颜色（如果你想与众不同，当然要使用了）。

图标

在iOS app中，icon一个很好的用法是使用视觉化的关系来支持文本标签，从而执行一些操作或者完全取代文本（最常用的比如“New”、“Delete”等）。通常，我们使用icon来区分导航栏、工具栏以及标签栏。

各种“栏”的按钮icon

各种“栏”中的按钮icon应该有两种状态：默认状态下的1或者1.5pt笔画宽度的轮廓样式，以及纯色填充的活跃状态。





不要在按钮icon上添加任何额外的效果，比如下拉阴影或者内阴影，因为这些是iOS 7之前版本中的用法。按钮icon应该在一个透明背景上以一种纯色进行绘制—icon的形状作为遮罩，颜色将会以编程形式应用。

活动视图图标

活动视图（通常指分享弹出视图）中的icon以轮廓样式设计，但苹果在iOS 8以后回归到白色背景上的实体填充icon风格。

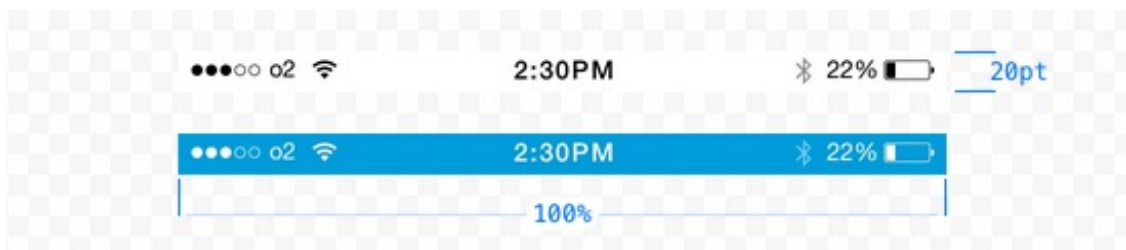


常用设计元素

iOS提供了很多不错的现成的视图和控件，可帮开发者快速构建页面。开发者可以将一些元素自定义到某个级别，但是也有一些元素不能或者不应该进行自定义。当为iOS设计应用程序时，你应该知道一些工具集的使用，只要是可能，就应该坚持下去。但在一些情况下，可能需要设计一些自定义控件，因为你需要一个更加定制化的界面或者想要改变现有控件的功能（有点危险）。几乎任何一件事情都是有可能的，而有时候你需要打破既有的规则，不过需要三思。

状态栏

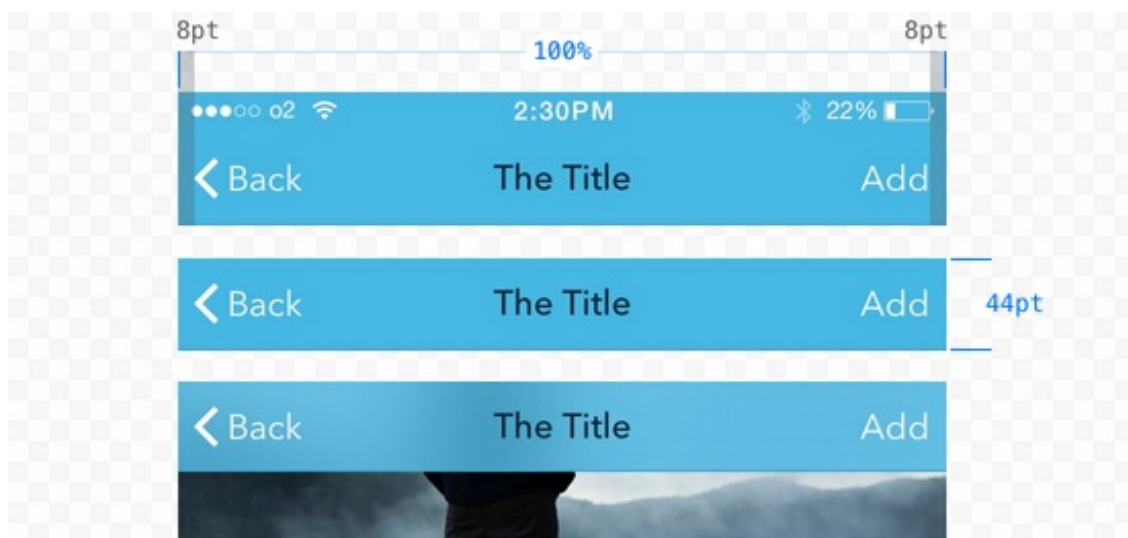
状态栏包含了基本的系统信息，比如运营商、时间、电池状态以及其他等，它在视觉上通常与导航栏联系在一起，并且使用相同的背景填充。为了匹配你的app的风格，并且易于阅读，状态栏的内容一般有两种不同的风格：深色（黑色）和浅色（白色）。



你可以隐藏导航栏，但要思考清楚。比如在app下载web内容时，用户可能对设备是否连接上WiFi网络比较感兴趣，在app要求蓝牙连接第三方硬件时，用户可能会想知道应用是否启用了蓝牙。一个令人信服的隐藏状态栏的理由是你想移除对的那个元素的所有干扰信息，比如全屏展示内容，比如图片。

导航栏

导航栏包含在app多个视图间进行导航的控件，以及在当前视图中管理内容的选项。导航栏通常展示在屏幕的顶部，状态栏的底部。默认情况下，背景是半透明的，在导航栏下方还有模糊的内容。导航栏的背景可以是纯色的，渐变的或者是自定义的位图模式。



竖屏模式下的iPhone 6导航栏。



横屏模式下的iPhone 4S导航栏。导航栏的高度减了12pt，除了iPad。这也是常见的横屏模式下隐藏状态栏的方法。

元素应当遵循特定的对齐模式：

- 1.返回按钮通常居左对齐。
- 2.当前视图的标题应当居中展示在bar中。
- 3.Action按钮通常居右对齐。如果可能的话action按钮应当限制在一个主要操作行文，以避免错误点击，并维持其简洁性。

工具栏

工具栏包含用于管理或者操作当前视图中内容的一些操作。在iPhone上，它通常出现在屏幕的底部，但在iPad上也能出现在屏幕的顶部。

和导航栏类似，工具栏的背景填充也能调整，默认情况下工具栏本身是半透明的，在其下方还有模糊的视图内容。





当一个特定视图要求三个以上主要活动，但放在导航栏上又显得凌乱时，你可以使用工具栏。

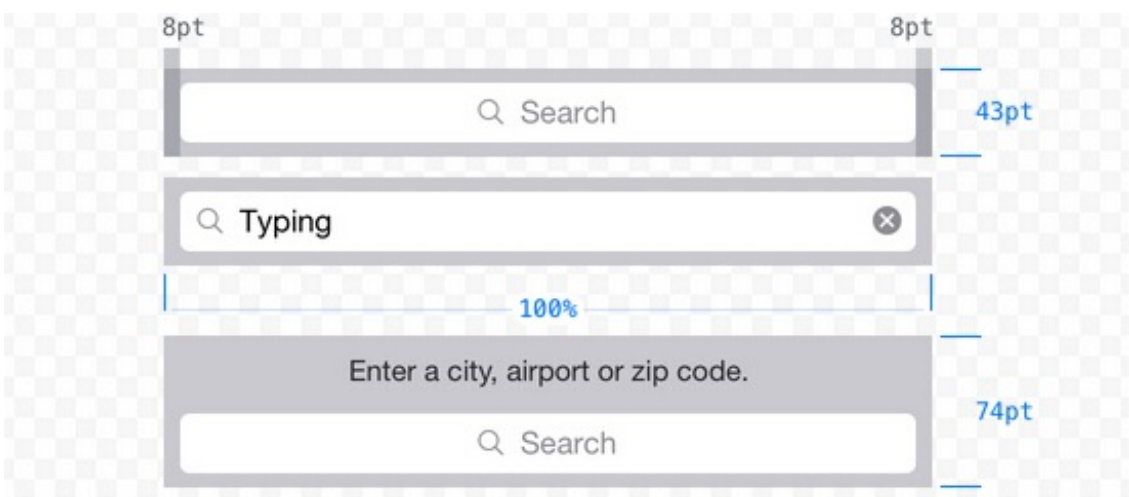
搜索栏

搜索栏默认有两种风格：突出的和最小化的。两种类型的搜索栏在功能上是一样的。

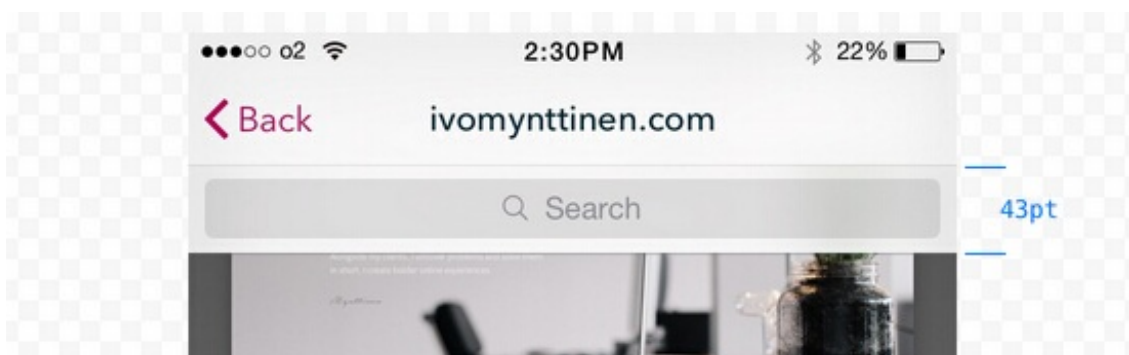
1.只要用户没有输入文本，搜索栏中会展示占位符文本，而书签icon则用来访问最近或者保存的搜索。

2.键入搜索项目后，占位符消失，一个清晰的删除按钮会出现在搜索栏的右侧。

搜索栏可以利用一个提示—一个短句来介绍搜索的上下文环境。比如”键入某个城市、邮政编码或者机场”



不使用提示和使用提示两种风格

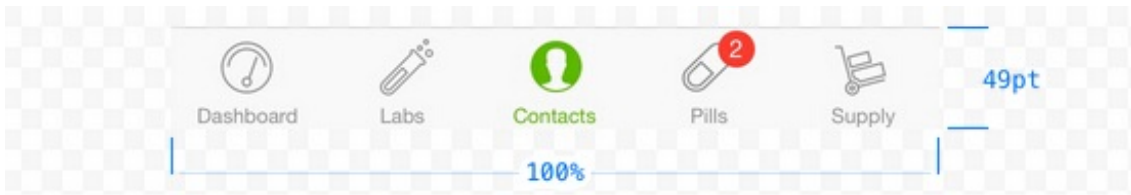


最小化搜索栏类型

想要提供对检索词条的更多控制，可用scope bar（范围栏）限制搜索栏，scope bar会使用与搜索栏一样的风格，当搜索结果有清晰的定义类别时，这种方法比较有用。比如，在一款音乐类app中，搜索结果可以按照专辑和歌曲再次过滤。

标签栏

用户可使用标签栏在app的单个视图间快速导航，并且标签栏也只能用于这个目的。标签栏通常出现在屏幕的底部。默认情况下，标签栏是略透明的，并且向导航栏一样使用系统的模糊效果。



标签栏包含固定的最大数量的tabs，一旦标签数量超过其可容纳的最大数量，后边的标签将会展示在隐藏的”More-tab”列表中，并且有一个选项可重排标签顺序。

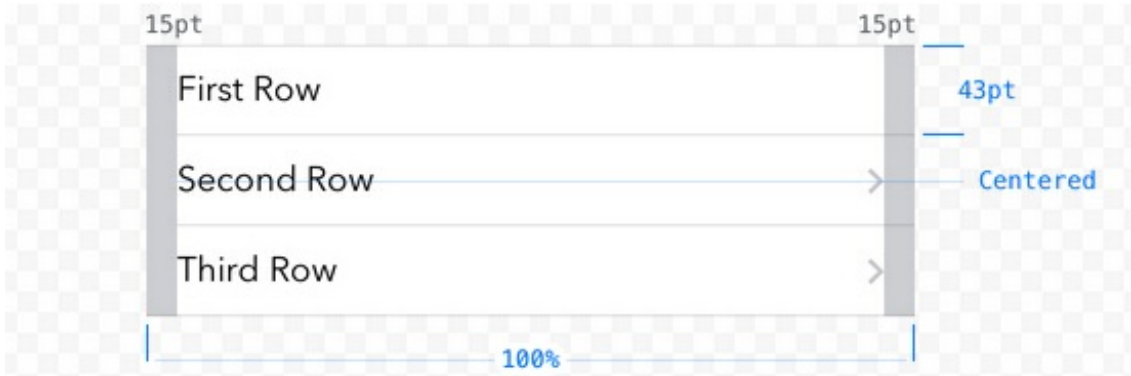
虽然iPhone上最多可展示5个标签，但是在iPad上最多可展示7个标签。

为了提醒用户视图上的新信息，有时候需要在标签栏按钮上使用标记数量。如果一个视图被临时禁用，那么相关的标签按钮不应当完全被隐藏，相反应当淡出视觉范围以表示其禁用状态。

表视图 (Table View)

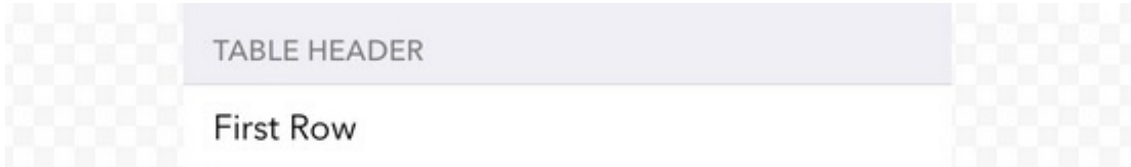
表视图以单列或者多列形式展示少数或者多个列表风格的信息，并有能将内容分组的选项。根据你展示的数据类型，通常可使用两种基本的表视图风格：

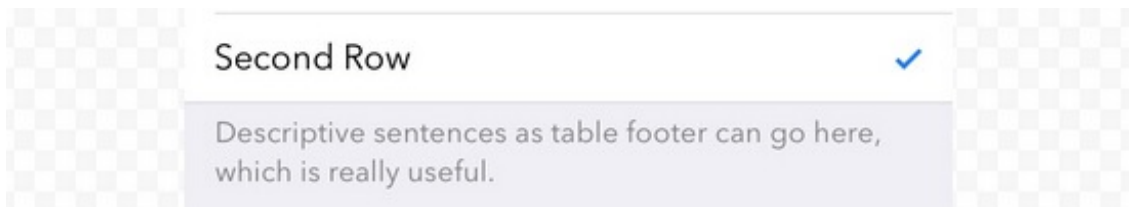
无格式的



无格式表视图包含的几行内容的顶部可以有页眉，最后一行后边可以有页脚。可以在屏幕右边缘展示垂直导航，以便在表中进行导航，这种情况适合展示以某种方式储存的大数据集的时候，比如按照字母降序排列。

分组样式

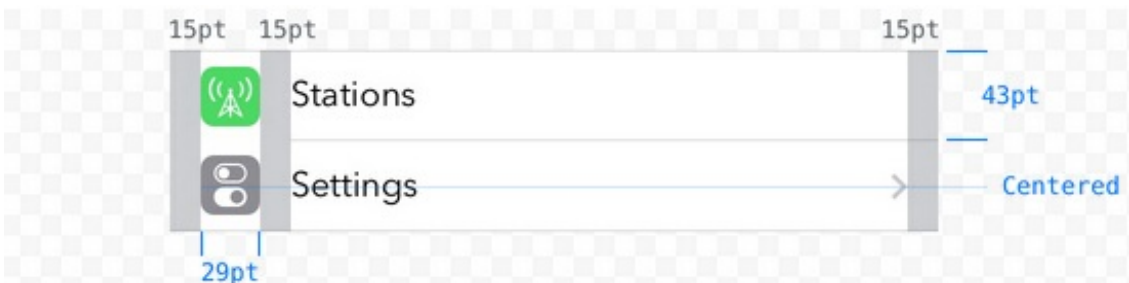




分组表视图允许用户对内容进行分组。每个分组可以有页眉（最佳用法是描述类组的上下文环境）和页脚（适用于帮助文本等）。一个分组的表视图至少需要包含一个类组，并且每个类组至少要包含一行内容。

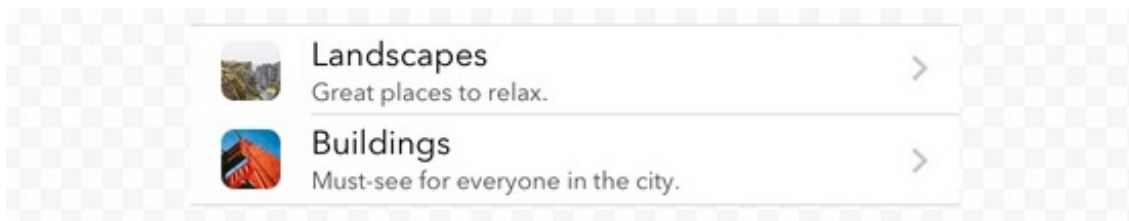
对于以上两种表视图类型，可用几种风格来展示数据，以方便用户快速扫描、阅读和适当调整内容。

默认



默认的表视图有一个居左对齐的可选图片和标题。

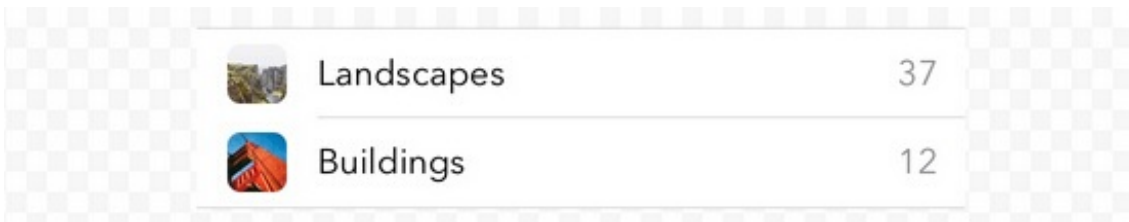
带有副标题



在每行标题下展示小字号的副标题，适用于进一步的解释说明或者简短描述。

带有数值

带数值表类型可展示与行标题相关的特定数值。类似默认的类型，每行都有一个居左对齐的图片和标题。在该类型中，数值居右对齐，通常使用比标题淡一点的文本颜色。

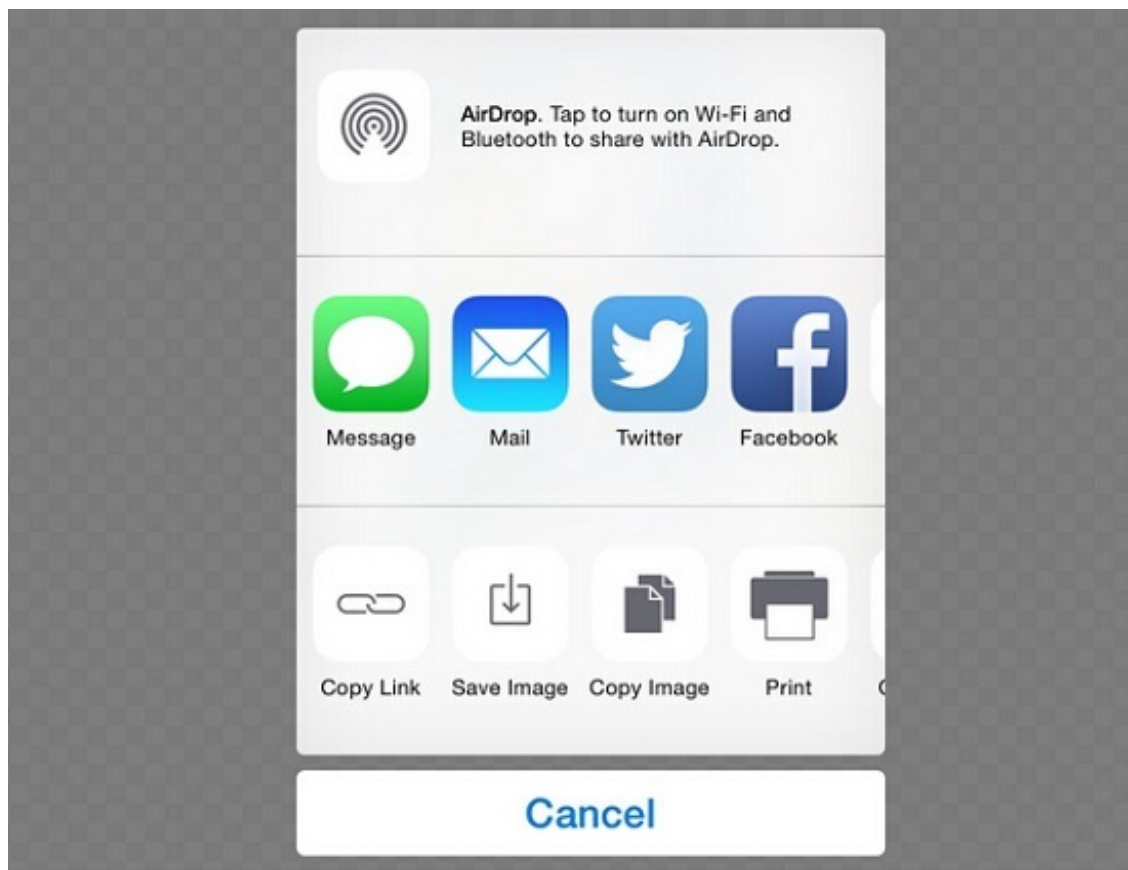


模态视图、弹出视图以及提醒（警示）视图

iOS提供了多种风格的临时视图，可以某种方式在既定的情况下展示、编辑或者操作数据。虽然每种临时视图因某个非常特定的目的而存在，但外观上却大有不同，不过所有临时视图都有一个相同的地方：在展示时，它就是当前视图上最上方的图层，下方的内容被一个黑色的背景所覆盖。

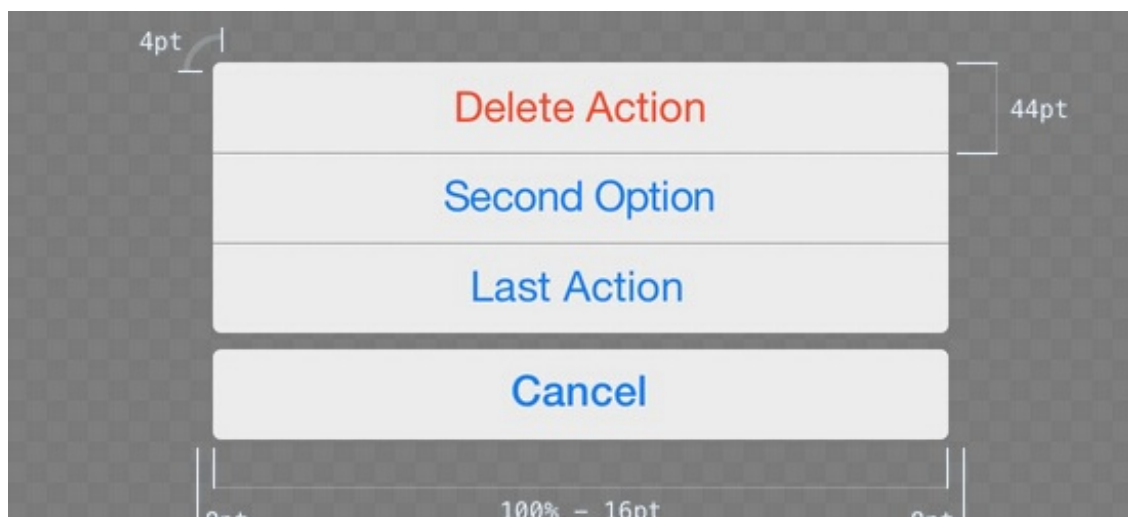
活动视图 (ACTIVITY VIEW)

活动视图用以展示特定的任务。这些任务可以是系统默认的任务，比如通过可用选项分享内容，或者是完全自定义的活动。当为自定义任务按钮设计icon时，你应当遵从活跃状态和栏按钮icon的一些设计指南—纯色填充、无任何效果、以及在一个透明背景上。



活动 (ACTIONS)

活动页面用来执行可用操作列表中的单项操作，并且强迫用户确认或者取消某个活动。



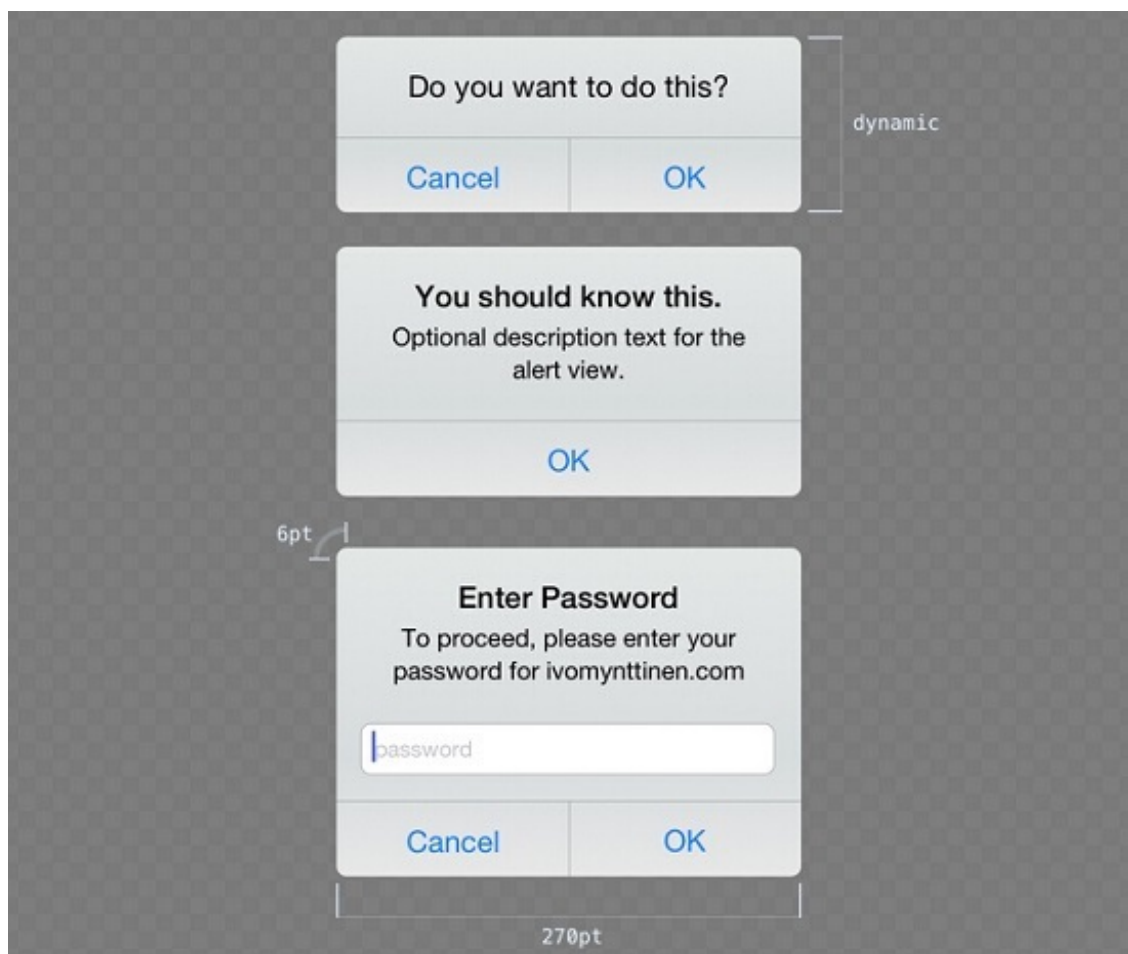
在竖屏模式下（以及尺寸比较小的横屏模式下），actions以按钮列表形式滑入，并呆在屏幕的底部。这种情况下，活动列表应该有一个取消按钮来关闭视图和执行任何列表中的action。

当有足够可用的空间时（比如iPad上），活动列表可在视觉上转为弹出视图。不过此时不一定非得有一个关闭按钮，用户点击弹出视图外的任何地方都能关闭弹出视图。

提醒视图

提醒视图的目的是用来通知用户一些关键性的信息，并有选择地迫使用户做出选择。

提醒视图通常包含一个标题文本（最好不要超过一行）、一个（纯信息提醒，比如“OK”）或者两个按钮（要求用户做出决定，比如“Send”或“Cancel”）。

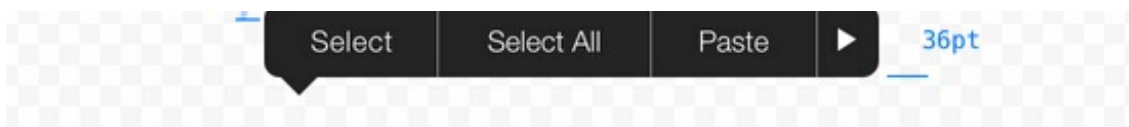


你可以在提醒视图中添加消息文本，如果需要的话则可添加两个文本域，其中之一可以是蒙版的输入区，适合密码或者PINs之类的敏感信息。

编辑菜单（EDIT MENU）

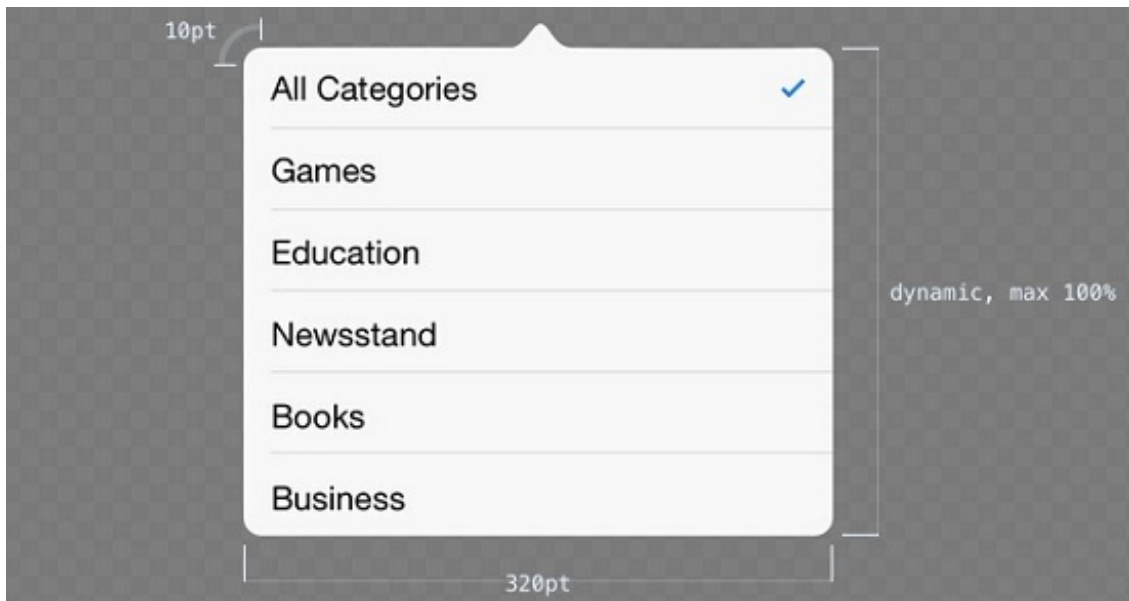
用户可使用编辑菜单执行复制、粘贴以及剪切等操作。虽然你能控制用户可以选择哪个操作，但是编辑菜单的视觉外观是设定好的，不能重新配置，除非你设计一个完全自定义的编辑菜单。





弹出视图 (Popovers)

当某项特定操作要求多个用户输入才能继续进行时弹出视图就非常有用。在水平方向上，弹出视图可通过箭头指向展示下方相关的控件（比如按钮）。弹出控件的背景透明度稍有降低，可展示其下方的模糊内容，像iOS 7以后其他很多UI元素一样。



弹出视图是一种非常强大的临时视图，可包含类似导航栏、表视图、地图或者web视图等对象。随着弹出视图所包含内容和元素的增加，其窗口也能滚动展示。

模态视图

模态视图适用于需要多个命令和用户输入的情况，位于屏幕上所有内容的最上方。典型的模态视图通常提供：

- 1.描述任务的标题。
- 2.关闭模态视图的按钮，没有保存或执行任何其他操作。
- 3.保存或者提交任何已输入内容的按钮。
- 4.模态窗口主体中提供大量用户可输入的元素。

有三种可用的模态视图类型：

- 1.全屏模式：覆盖整个屏幕
- 2.页表模式：竖屏模式下，模态视图覆盖屏幕上的部分内容，仅在半透明的暗色背景上展示部分父

视图的内容。横屏模式下，模态视图会像全屏模态视图那样展示。

3.表格页面模式：竖屏模式下，模态视图出现在屏幕的中间。模态视图范围之外，父视图内容展示在半透明背景之下。当需要展示键盘时，模态视图的位置会自动调整。横屏模式下类似全屏模态视图。

控件 (Controls)

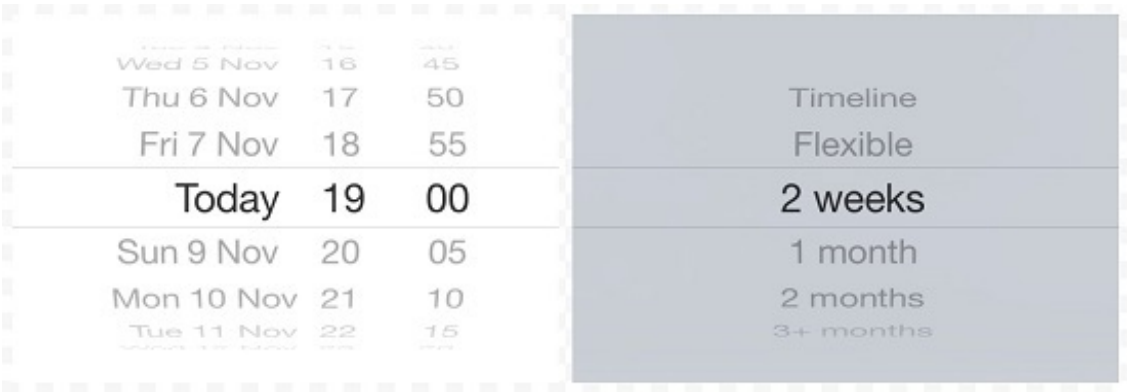
iOS为基本上任何类型的输入需求提供了各种各样的控件。以下列出的是最常用的控件，但想要看详细的完整的可用控件列表，请在[iOS Developer Library](#)中查看。

按钮

最常用的控件可能是按钮。iOS 7以来，默认的按钮设计看起来更像一个纯文本链接。按钮控件支持高度自定义。按钮可以有几种不同的状态，可以使用视觉语言传达：默认、突出、选择以及禁用等。

选择器 (PICKERS)

选择器用来在一个可用值列表中选择某个值，和Web上的下拉选框功能类似。选择器的扩展版本是日期选择器，允许用户滚动日期和时间列表来选择一个月、日以及具体时间。

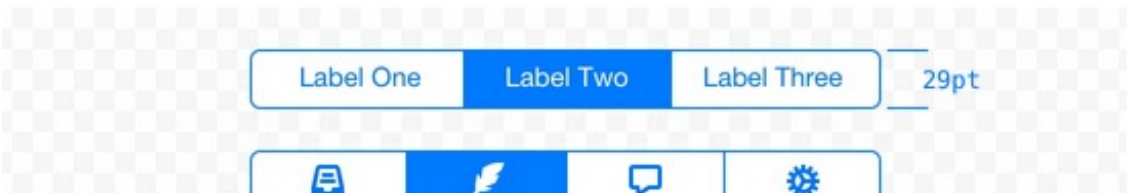


左：表视图中的日期选择器，右：选择器作为键盘

除了背景色外，不可能更改其视觉风格或者尺寸。很多时候，选择器位于屏幕的底部，像键盘一样展示，但不能用在其他地方。

分段控件 (SEGMENT CONTROLS)

分段控件包含一组至少两个以上的分段，可用来过滤内容或者为清晰分类的内容创建标签。



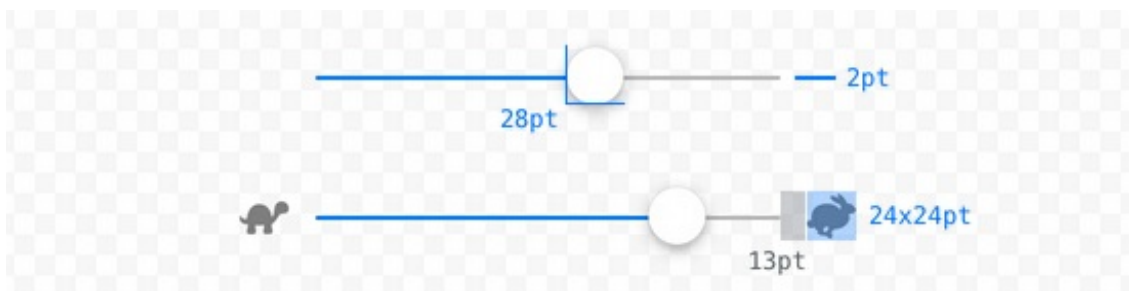


不带icon与带icon的分段控件

每个分段可包含一个文本标签或者一个图片（icon），但不能同时包含两者。另外，不建议在一个分段控件中混合使用不同的分段风格，比如文本和图片。每个分段的宽度会基于分段的数量自动调整（两个分段各占50%，5个分段各占20%）。

滑杆（SLIDERS）

滑杆控件允许用户从允许值范围中选择一个特定的值。由于使用滑杆选择一个值的操作非常流畅，并且无需额外的步骤，所以建议在选择预估值的时候使用滑杆，而不是选择精确的值。比如滑杆可以很好地用来设置音量，用户可以听到和看到音量大小的不同，而通过输入文本来设置精确的dB值是不切实际的。

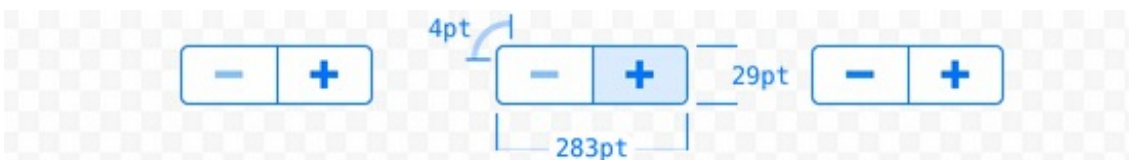


不带icon与带icon的滑杆控件

可以为最小值和最大值设置icon，icon会展示在滑杆控件的开始和末尾两端，从而在视觉上加强滑杆的目的。

进步器（STEPPER）

当用户从一个有限的值范围中（比如1-10）键入准确的值的时候，可使用进步器。进步器通常包含两个分段按钮，一个是降低当前值，一个是增加当前值。



进步器的视觉外观支持高度自定义：

- 1.可使用自己的icon作为进步器按钮；
- 2.当想维持iOS原生的外观时，你可以自定义进步器边框、背景以及icon的颜色。
- 3.如果你更进一步使用自定义，那你可以为进步器的按钮和分割符使用完全自定义的图片。

开关 (SWITCH)



用户可使用开关在”ON”和”OFF”两种状态间切换。设计师可自定义两种状态的颜色，但是开关按钮的外观和尺寸是设定好的不能更改。

键盘 (KEYBOARDS)

有多重键盘类型可为特定的文本输入提供最好的键盘。虽然你可以使用自己完全自定义的键盘，但默认的键盘无需调整风格和尺寸，很多时候更加方便一些。

来源：[CocoaChina](#)翻译 编辑：[suiling](#)

人人都是产品经理 (woshipm.com) 中国最大最活跃的产品经理学习、交流、分享平台