

# iOS 人机界面指南 ISUX 原创翻译（上）

## 目录

为 iOS7 而设计 .....	2
以内容为核心 .....	3
保证清晰度 .....	6
用深度来体现层次 .....	9
iOS 应用解析 .....	13
起始与停止 .....	16
即时启动 .....	16
时刻准备好停止 .....	19
布局 .....	20
导航 .....	21
模态对话 .....	21
交互性和反馈 .....	23
反馈能帮助用户理解 .....	26
输入信息过程需要简易 .....	27
术语和措辞 .....	28
动画 .....	29
排版和颜色 .....	30
排版必须保持清晰 .....	30
颜色可以增强沟通性 .....	31

图标和图形.....	32
App 图标 .....	32
其他图标.....	32
图形 .....	33

## 为 iOS7 而设计

iOS7 的革新关键词如下：

- **遵从：**新 UI 更好的帮助用户理解内容并与之互动，但却不会分散用户对内容本身的注意力
- **清晰：**各种大小的文字易读，图标醒目，去除了多余的修饰，重点突出，很好地突显了设计理念
- **深度：**视觉层次和生动的交互动作赋予 UI 新的活力，不但帮助用户更好的理解新 UI 的操作并让用户在使用过程中感到惊喜



无论你是重新设计一个现有的 app 或是重新开发一个,尝试一下苹果重新设计系统内置 app 的方式:

- 首先,去除了 UI 元素让 app 的核心功能呈现的更加直接并强调其相关性。
- 其次,直接使用 iOS7 的系统主题让其成为 app 的 UI,这样能给用户统一的视觉感受。
- 纵观全局,以内容和功能为核心来指导设计,从前的设计模式可以先放到一边。

## 以内容为核心

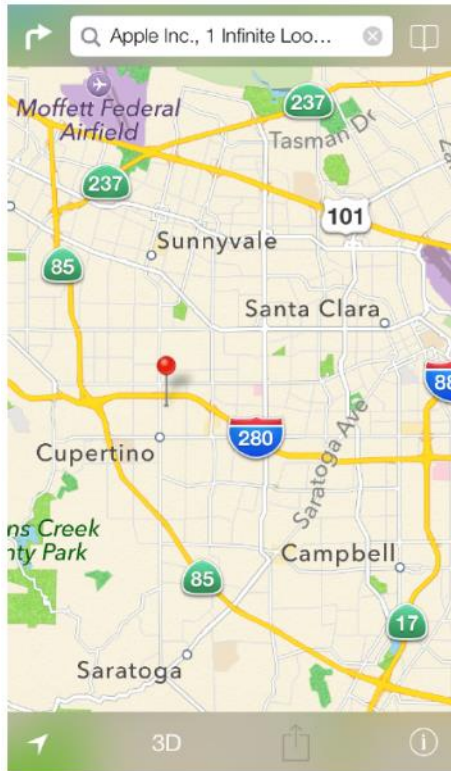
虽然明快美观的 UI 和流畅的动态效果是 iOS7 体验的亮点,但内容始终是 iOS7 的核心。

这里有一些方法,以确保您的设计能够提升您的 app 功能体验并关注内容本身。



**充分利用整个屏幕。**无需使用边框、底图等等多余的 UI 元素，让内容扩展到屏幕边缘。

天气 app 是最好的例子：漂亮的天气图片充满全屏，告知用户天气情况，同时也很好的呈现了如每个时段气温等等的其他重要信息。



**尽量减少视觉修饰和拟物化设计的使用。**UI 面板、渐变和阴影有时会让 UI 元素显得很厚重，致使抢了内容的风头。应该以内容为核心，让 UI 成为内容的支撑。

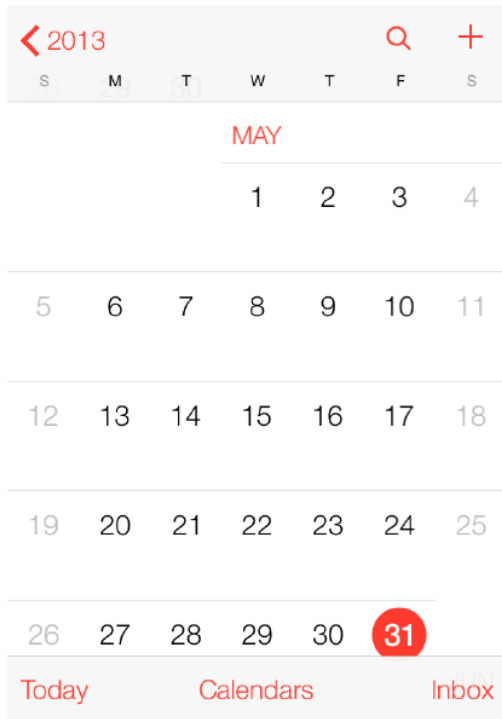


**尝试使用半透明底板。**半透明的底板可以让用户看到后面的内容，在某些场景下起到了上下文提示的作用，另一个角度上来说，也让用户（比以前）看到了更多内容。

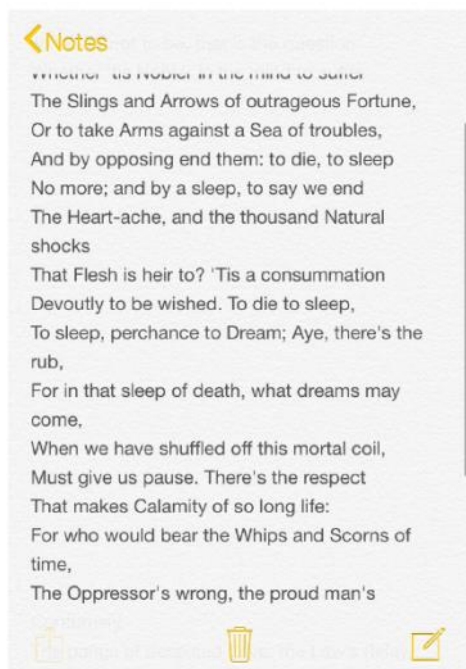
## 保证清晰度

保证清晰度是另一个方法，以确保你的 app 中内容始终是核心。

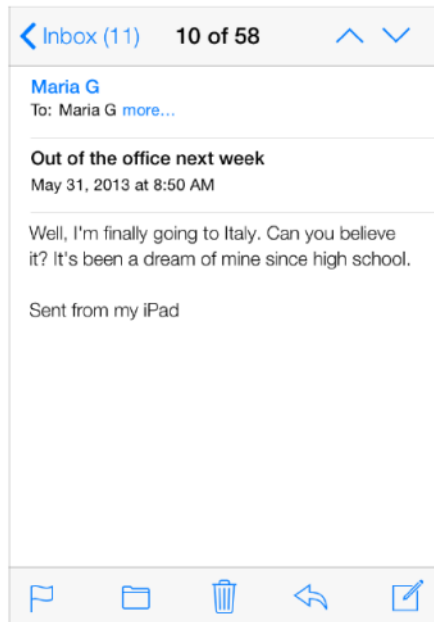
以下是几种方法，让最重要的内容和功能清晰，易于交互。



**大量留白。**空白让重要内容和功能显得更加醒目。此外，空白可以传达一种平静和安宁的视觉感受，它可以使一个 app 看起来更加聚焦和高效。

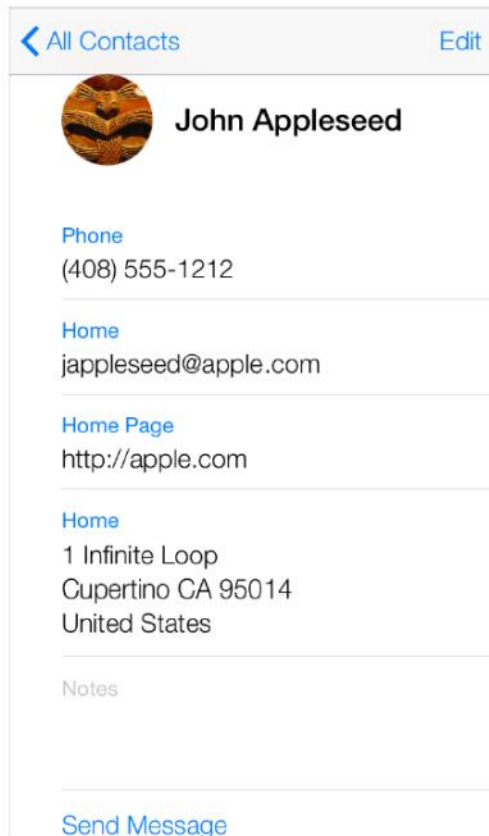


**让颜色简化 UI。**一个主题色——比如在记事本中使用的黄色——让重要区域更加醒目并巧妙地表示交互性。这同时也给了一个 app 一个统一的视觉主题。



**通过使用系统字体确保易读性。**iOS7 的系统字体自动调整行间距和行的高度，使阅读时文本清晰易读，无论用户选择何种大小的字号都表现良好。

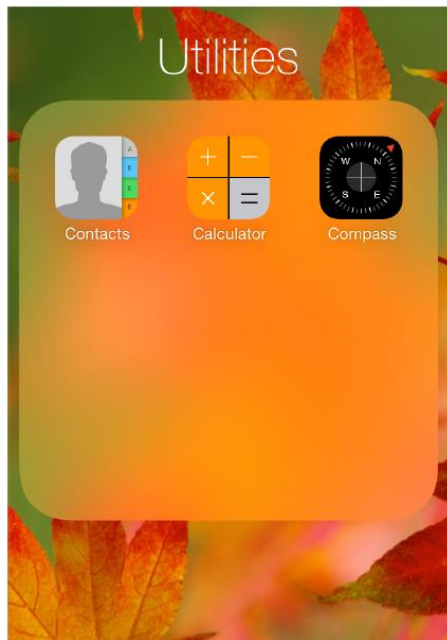




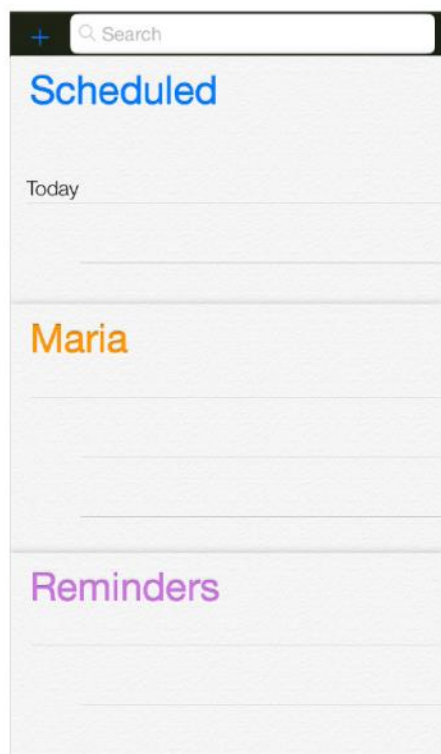
**使用无边框的按钮。**按钮名称、箭头以及系统颜色向用户展示了交互元素，这些内容替代了原先的带有形状的按钮。联系人界面使用了系统色蓝色箭头文字展示了按钮的导航性。

## 用深度来体现层次

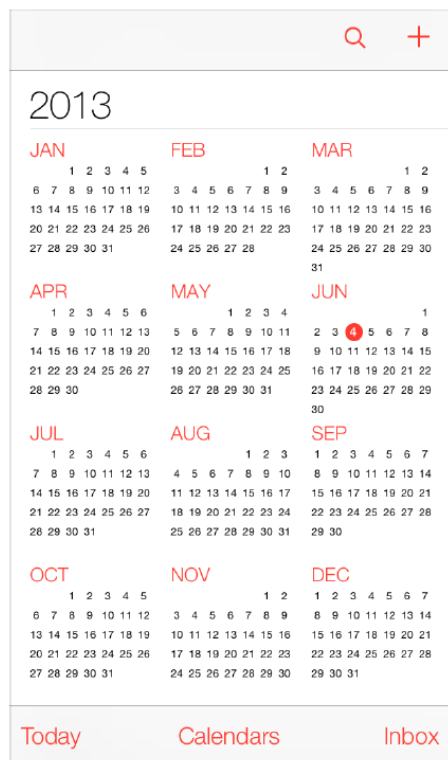
iOS7 经常在不同的层级上展现内容，用以表达分组和位置，并帮助用户了解在屏幕上的对象之间的关系。



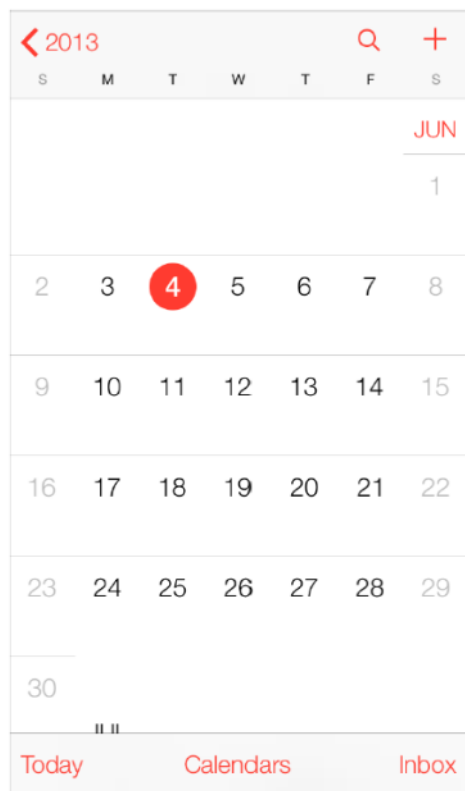
通过使用一个在主屏幕上方的半透明背景浮层来区分文件夹和其余部分的内容。



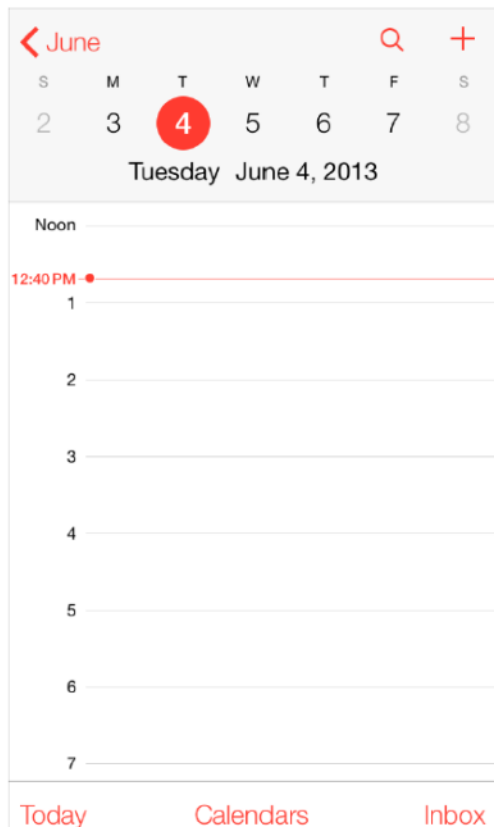
用户在使用备忘录里的某个条目时，其他的条目呈现在屏幕底部的其他分层上，这样用户可以通过滑动展开所有分类条目（译者按：算是一种提示手段）。



日历有较深的层级，，当他们在翻阅年、月、日的时候，以及增强的交互动画给用户一种层级纵深感（循序切换的层次，从年到月到日）。在滚动年份视图时，用户可以即时看到今天的日期以及其他日历任务。



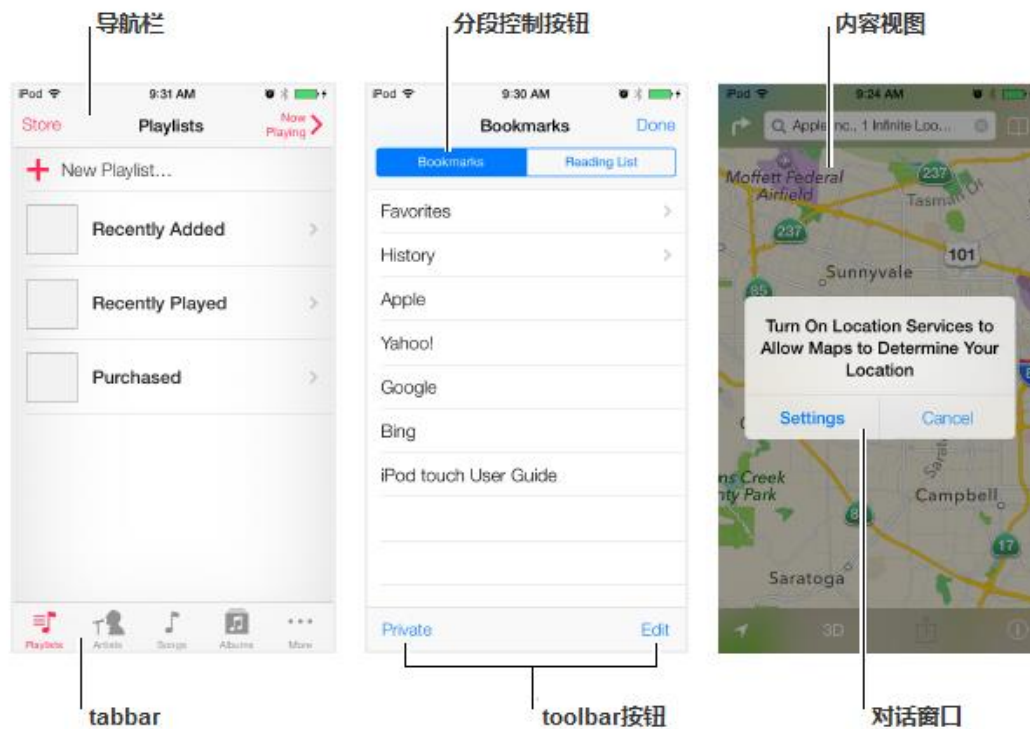
当用户处于月份视图时，点击年份视图按钮，月份会缩小至年份视图中的所处位置。



类似的过度出现在月份和日期视图的切换时，当用户选择某个日期时，月份视图向外扩展，显示出日期视图。

## iOS 应用解析

几乎所有的 iOS app 都应用了 UIKit framework 中定义的组件。了解这些组件的名字，作用和构成能够帮助你设计 app 过程中做出更好的决定。



UI 组件大致分成以下 4 种大类：

**Bars:** 包含了导航信息，告诉用户他们所在的位置并包含了一些能帮助用户浏览或启动某些操作的控制按钮。

**内容视图:** 包含了 app 的主体内容以及某些操作行为，比如滚动、插入、删除、排序等等。

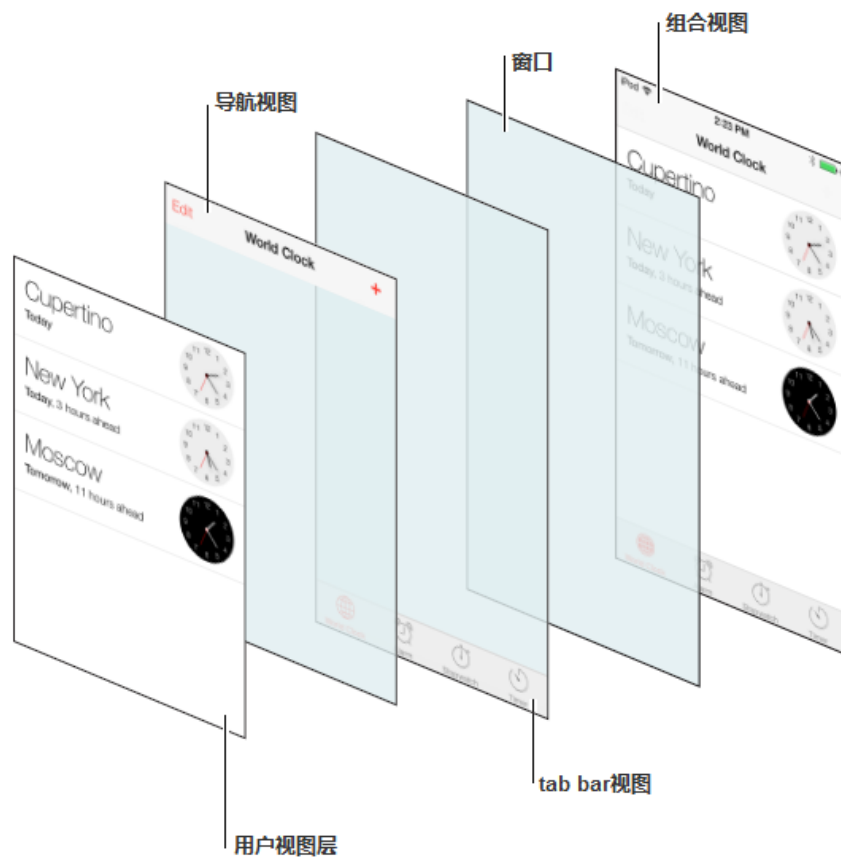
**控制按钮:** 展示信息或者控制动作。

**临时视图（对话框）:** 短时间出现，给用户重要信息或者额外的选择或者其他功能。

除了定义 UI 组件，UIKit 也定义对象实现的功能，例如手势识别，绘图，辅助功能，打印支持。

从编程的角度说，UI 组件被认为是不同类别的视图，因为他们从 UIView 得到继承。视图能绘制屏幕内容并且知道用户何时触摸了屏幕。要在 app 中管理一组或者一系列的视图，通常需要使用一个视图控制器，它能协调视图的显示内容，实现与用户交互的功能并能在不同屏幕内容之间切换。

下面是一个例子，关于视图与视图控制器如何结合并呈现 iOS app 的 UI。



虽然开发者认为真正起作用的是视图和视图控制器，但一般用户感知到的 iOS app 是不同屏幕内容的集合。从这个角度来看，在 app 里，屏幕内容一般对应于一个独特的视觉状

态或者模式。

# 起始与停止

## 即时启动

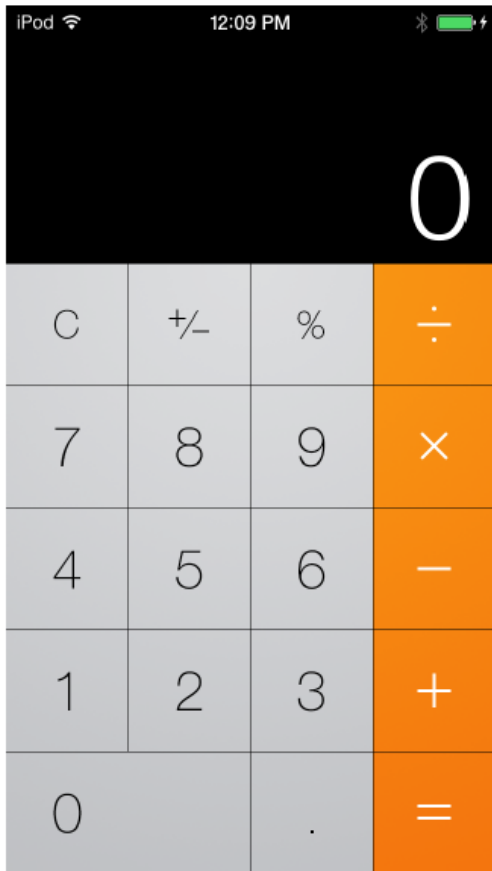
有种说法是用户往往不会花超过一两分钟去审视一个新应用,当你将软件从打开到启动这段时间压缩得很短,并且同时在载入过程中呈现一些对用户有帮助的内容,你会激发用户的兴趣并给所有用户一个惊喜。

**重要:** 不要在安装过程结束后告诉用户需要重启设备之类的。重启需要时间并且会让人觉得你的应用看上去不可靠而且很难使用。如果你的应用将使用存储空间,或者不重启机器就无法流畅运行,你必须声明这些问题。

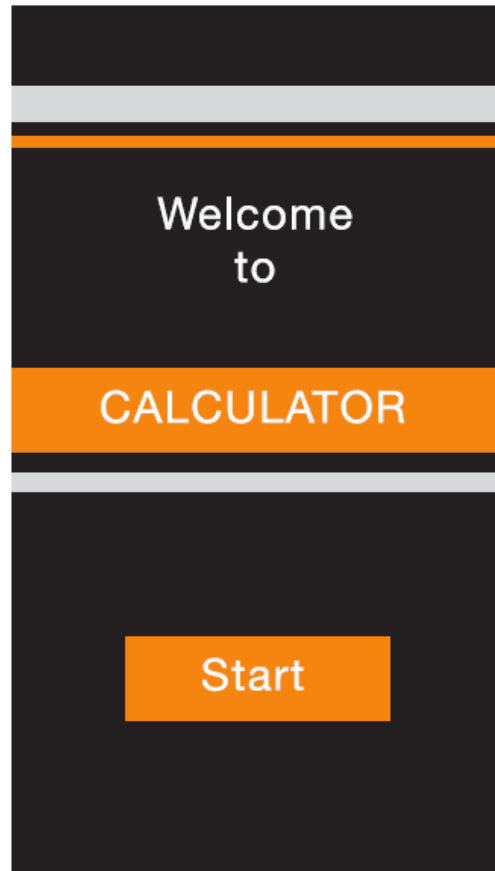
**尽可能避免使用闪屏或者其他启动体验。**用户能够在启动后立即开始使用软件是最好不过的。



推荐（直接启动）



不推荐（使用欢迎屏幕）



避免让用户做过多设置。而应该如此：

- **聚焦在满 80%的用户需求上。**这样主体用户群就无需设置各种选项，因为你的 app 已默认处于他们想要的状态。如果有些功能有少部分用户想要，换句话说大部分人不需要的话，就别管它了。
- **尽可能用其他方式获取更多（用户）信息。**如果你能得到用户在内置软件或硬件设置中提供的信息，直接从系统中获取它们，而不需要再次让用户输入。
- **如果你必须获取设置信息，在你的应用中直接向用户询问，然后尽快保存这些设定（这**

段讲的是权限许可，如能否访问照片或者日历或地理位置信息等等)。这样用户就无需强制跳出 app 进入系统设置页面了。如果用户需要更改设置，他们可以在任何时候进入 app 的设置选项进行修改。

**尽可能让用户晚一些再登录。**让用户在无需登录的情况下就能尽量多的浏览内容并使用部分功能是最理想的状态，。如果用户在熟悉你的 app 之前就被强迫需要登录，那么启动流程就会变得拖沓繁琐。

**一般来说，按照屏幕默认的定向方式启动你的 app。**对于 iPhone，竖向是设备默认定向，而 iPad 则是设备当前所处的方向。如果你的 app 只能在横向模式运行，那么就始终以横向状态启动，让用户在他们自己需要时改变设备方向。

最好让横屏 app 支持两种模式的横屏，即 home 键处于左右两侧的状态。如果设备当前已经处于横向状态，那么就按照当前状态启动 app，除非你有充分的理由不这么做。其他情况时，可以考虑按 home 键处于右侧的方式启动 app（译者按：大部分人习惯使用右手）。

**可以准备一张与 app 首页看上去一样的闪屏，**iOS 会在启动 app 时调用这张图，这样可以让用户觉得启动速度很快，降低对等待时间的感知度。

**如果可能，不要让用户在初次启动应用时阅读免责声明或者确认用户协议。**你可以直接在 app store 展示这些内容，使用户在下载前就有所了解；虽然这个办法能最大地减少麻烦，但也不是一直可行。如果在某些情况下你必须展示这些内容，要确保它们与 UI 保持统一并

在产品功能与用户体验之间达成平衡。

在应用重启后，需要恢复到用户退出使用时的状态，让他们可以从中断之处继续使用。无需让用户记住是如何达到此种退出状态的。

## 时刻准备好停止

iOS app 无需关闭或退出选项。当用户切换 app 或回到主屏幕或者将他们的设备调至睡眠模式的时候，其实就是停止了当前 app 的使用。

当用户切换 app 时，iOS 的多任务系统将其放置到后台并将新 app 的 UI 替换上来。在这种情况下，你必须做到以下几点：

**随时并尽快保存用户信息**，因为在后台的应用随时有可能被终止或退出。

**当程序停止的时候尽可能在细节层面保存当前状态**，使用户可以在回到应用时能从中断之处继续使用。例如，在使用可滚动的数据列表时，退出后保存列表所在的位置。

**不要强制让 app 退出**，因为这样会让用户误以为是 crash。如果有问题产生，需要告诉用户具体状况以及如何解决。以下有两个建议，取决于出现的问题有多严重而酌情使用：

- **使用吸引注意的屏幕内容描述出现的问题并给出建议的方案。**如此可让用户了解到 app 本身没有问题，并将主动权交给用户，让他们决定是解决问题并继续使用还是切换到其他应用。
- **如果只是某些 app 功能无法使用**，可以在用户使用这些功能时弹出一个对话框告诉

他们当前的情况。只有在用户使用的功能确实无法工作时再继续弹出警告提示。

## 布局

**布局远比 UI 组件的样式重要。**布局能让你向用户展示什么是最重要的,他们是如何选择的,内容是如何相关的。取决于 app 运行的设备——以及设备的方向——布局可能会有所不同。

**让用户尽可能容易地跟内容交互并控制好每个控件的间距。**需要点击的控件大小至少要有 44x44 像素。

**通过平衡重要内容或者功能,让用户专注于主要任务之上。**将重要的组件放置于屏幕上半部分是最常用的方法之一——同时也要遵循从左到右的原则——放置在靠左侧的屏幕上。

**利用视觉重心和平衡向用户展示屏幕元素之间的相对重要性。**大型部件——以及那些看起来比较重的——更加吸引眼球并且让人感觉比小的部件重要些。

**一般来说,避免你的 UI 不一致。**尽可能地让有相似功能的组件有相似的外观。人们经常认为不一致一定有某些原因,并尝试花时间去搞清楚(这样其实是浪费用户的时间)。

**确保默认大小的内容(文字、图片)用户能够看清楚。**比方说,不要让用户滚动屏幕来阅读(屏幕以外的部分)文本或者无需通过放大操作来看清图像。

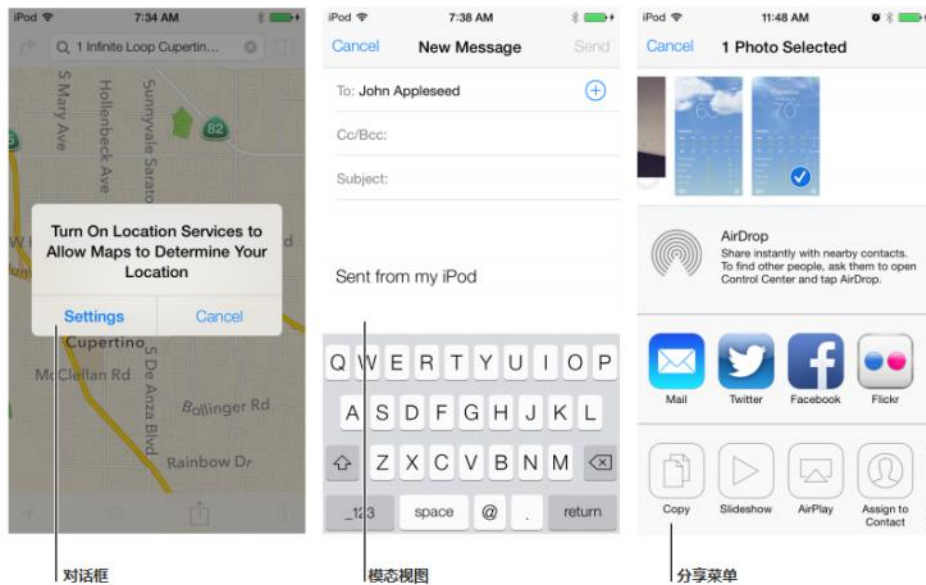
## 导航

用户很少察觉到一个程序中的导航体验除非它不符合他们的期望。放置导航到一个能够支撑你的 app 整体结构和目的却又不过分引起用户注意的状态。

广义来说，有三种主要类型的导航，每种导航都有其适应的 app 结构：分层、扁平、内容或经验驱动。在某些情况下，几种导航形式可以出现在同一个 app 里并且各司其职。无论你的 app 适合使用哪一种导航结构，最重要的是用户所体验的内容必须被有逻辑的、可预测并易于遵循的呈现在你的 app 中。用户需要始终很清楚他们在哪儿，并且如何到他们想去的地方。

## 模态对话

模态视图是一个优缺点并存的模式，承载某些连贯操作或内容，可以给用户在不脱离主任务的情况下完成某些任务或获取某些信息；但是这些操作都是临时的，以防止用户此时与应用程序其余的功能发生交互。



在理想状态下，用户可以与 iOSApp 发生非线性的交互，所以模态视图下最好的做法是最大程度的减小模态操作的数量。大体上说，当以下情况出现的时候，考虑使用模态对话：

- 必须引起用户关注的时候
- 某个任务必须被完成，或者明确被放弃，以避免在模棱两可的状态下遗漏用户信息（操作）。

**保持模态任务简单，简短并且高度聚焦。**你肯定不希望用户像使用一个 mini app 那样使用一个模态视图。如果一个模态对话中的子任务太复杂，用户会从暂停的主任务上迷失。创造一个包含一系列视图的模态任务时要特别注意这一点。如果一个模态任务必须在独立视图包含子任务，务必给用户一个独立、清晰的导航路径，并避免迂回。

**总是提供一个显眼并安全的方法用以退出模态任务（cancel 按钮）。**

一个包含一系列视图的任务，必须让用户明白不同步骤中“完成”按钮的作用。

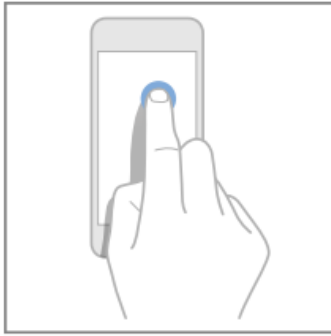
**保证对话框提供的信息都是必要的并是可操作的。**对话框是打断用户预期的，并且需要点击才会消失，所以让对话框所提示的信息必须是值得中断用户操作的，这对于用户体验来说很重要。

**尊重用户关于接收推送通知的选择。**在设置界面，用户可以设置 app 是否接收推送。必须遵循用户的设置，否则会触怒用户导致关闭所有推送通知。

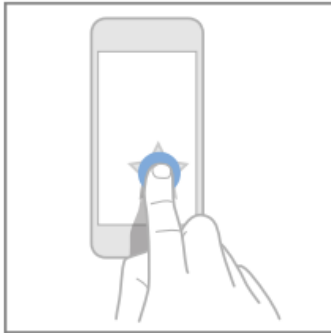
## 交互性和反馈

标准手势让用户感到舒适

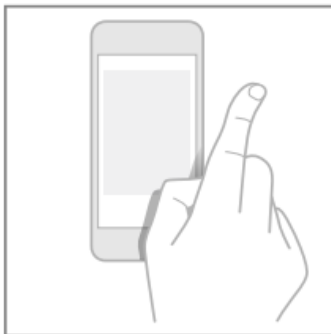
手势操作拉近了设备与用户之间的距离并提高他们的直接操纵感。在 app 中经常使用的统一手势操作如下：



**点击** 按压或选择一个控制按钮或物件

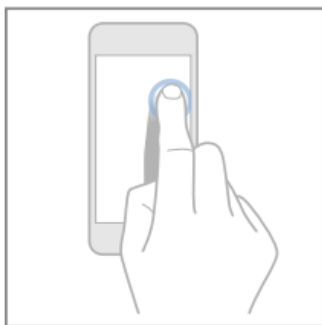


**拖动** 滚动或平移 从一边移到另一边 拖动某个控件

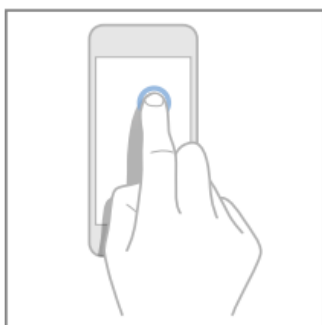


**滑动** 快速滚动或平移

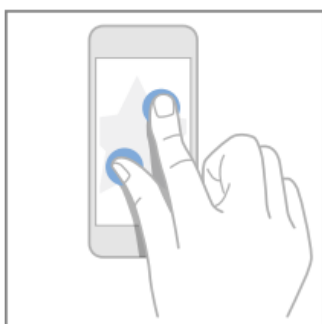




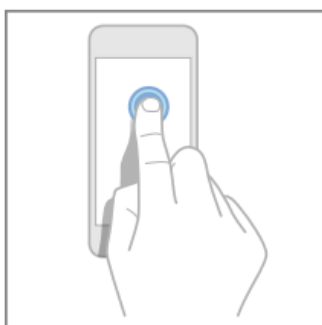
**轻扫** 用一个手指反向滑动呼出删除按钮 四指切换任务等



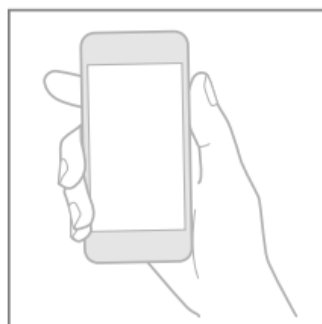
**双击** 放大缩小，图像中心定位等



**捏抬** 放大缩小，图像中心定位等



**长按** 呼出编辑状态或隐藏菜单



**摇晃** 撤销或重做

避免使用与常规手势操作含义不同的动作。

避免创造与常规手势功能雷同的新手势。

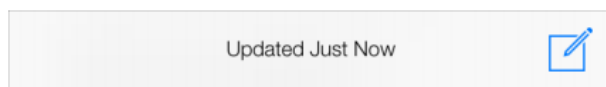
避免让用户用复杂手势完成某种任务。手势操作务必保持简单，直接。

避免创造新的手势，除了在游戏中。

对于 iPad 来说，尝试使用多指手势。iPad 较大的屏幕给多指操作带来空间。虽然并不是每款 app 都需要复杂手势操作，但复杂手势可以丰富用户体验，比如多人同乐的游戏等等。

## 反馈能帮助用户理解

iOS 用户习惯于得到反馈，帮助他们知道 app 正在做什么，下一步可以做什么，并了解他们的操作结果。尽可能地将状态或其他相关的反馈信息集成到 UI 上。例如，将邮件的更新状态显示在工具栏上：



避免不必要的对话框。对话框属于强提示机制，但应仅用于传递重要和有预期的操作信息。

如果用户看到太多的对话框，没有包含太重要的信息，那么用户很快就学会忽略所有对话框提示。

## 输入信息过程需要简易

（在手持设备上）用户利用触摸软键盘输入信息很花时间和精力。如果你的 app 因为在输入前出现一堆冗余操作，用户会感到崩溃。

**让用户更容易的做出选择。**使用选择器或表格视图替代输入操作可以让用户的使用体验更好。

The screenshot shows a mobile app interface for sending a birthday present. At the top, the name "Maria" is displayed in orange, and a "Done" button is in the top right corner. Below this, there is a section for "Send birthday present to Sarah" with a radio button and a "Clear" button. The date and time are set to "Today, 9:00 AM". Below this, there is a "When?" section with a "More..." button. The "When?" section contains a table of dates and times:

Date	Hour	Minute	Period
Fri May 31	6	45	
Sat Jun 1	7	50	
Sun Jun 2	8	55	
Today	9	00	AM
Tue Jun 4	10	05	PM
Wed Jun 5	11	10	
Thu Jun 6	12	15	

At the bottom, there are three buttons: "Later Today", "Tomorrow", and "Next Week".

**尽可能利用 iOS 获取信息。**用户储存了很多信息在他们的设备中。开发者可以自由使用用户已经在系统中输入的信息，比如联系人，日历信息等（当然要获得用户的许可）。

**在输入和回报上做出平衡。**（在输入后）给出反馈或回报，让用户感觉到他们的行为有价值。

## 术语和措辞

App 中呈现的每一个词都是与用户的一次对话，利用这个机会让用户在使用过程中感到舒适。

**使用术语时确保用户能理解。**针对用户群确定使用何种短语，例如一些技术术语某些高端用户能理解，但是普通用户就不太熟悉。

**可以使用非正式的友好语气，避免太正式又不能太虚假或低三下四。**请记住，用户在使用过程中会反复阅读文本，所以有些起初看上去很乖巧的语句多看几次就有可能让人厌烦。

**当你的 UI 文本简短直接，用户可以快速轻松地理解。**像新闻编辑一般遣词造句，确定最重要的信息，并强调显示，这样人们就不用看大段文字就知道下一步该怎么做。

**给按钮短标签或者易于理解的图标，用户可以一目了然的知道该做什么。**

**描述时间的时候注意要准确。**“今天”、“明天”这样的词听上去是比较友好，但是有的时候会让人混淆，比如你不清楚用户所在的环境（时区不同）。

潜在用户在逛 APP Store 时，app 描述就是最好的沟通机会。除了描述 app 的品质，你还需要做以下事情：

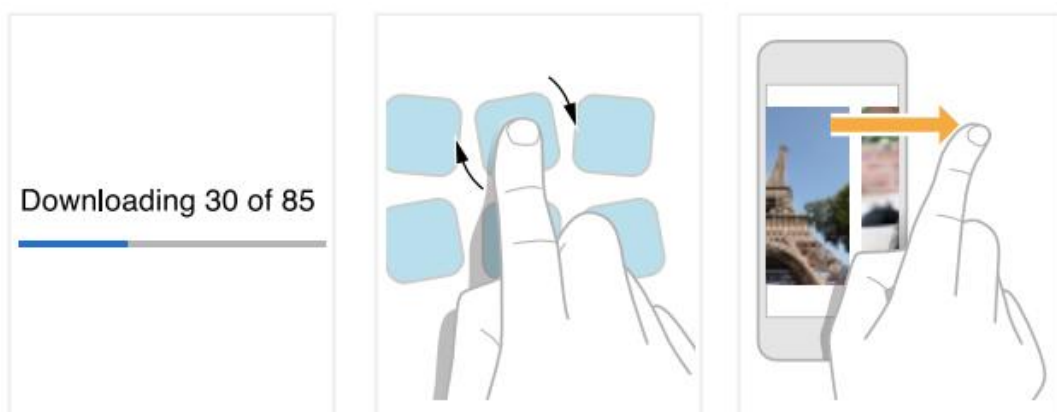
- 修正所有的拼写、语法和标点符号错误。虽然这些小错误不影响使用，但是会给人带来负面印象。
- 尽量少地使用全大写的词汇。虽然有时大写单词可以吸引人们注意，但是全大写的段落不适合阅读，而且有一直朝用户扯着嗓子吼叫的感觉。
- 可以描述 bug 修复情况。如果您的 app 新版包含用户一直期待的 bug 修复，那在你的

软件描述中提到这一点就是个很好的做法。

## 动画

美妙优雅的动画贯穿于 iOS 的用户界面，让 app 使用体验更加动感和引人注目。微妙和恰当的动画可以：

- 表示状态
- 增强直接操作的意图
- 帮助人们可视化操作的结果。



添加动画的时候需要谨慎，尤其是在 app 不提供沉浸式体验的时候。app 使用过程中，如果在执行主要任务时，过分的、无明显作用的动画往往会分散用户注意力，也会影响 app 的性能表现。

使用与系统内置 app 一致的动态表现。用户比较熟悉内置应用的动画表现，恰到好处。事实上，用户往往把视图之间的切换、方向变化时的响应、物理感应的滚动等这些效果看做 iOS 带来的使用体验，除了那些沉浸式 app——比如游戏——某些动画可以与系统内置动画相媲美。（译者按：其实就是建议开发者使用系统原生动画效果）

在 app 中使用的动画效果要统一。如同其他类型的订制，使用统一的自定义动画很重要，这可以让用户在使用不同的 app 时累计经验。

在大多数情况下，将自定义动画尽量做的真实一点是没问题的。人们往往愿意接受外观上的艺术创意，但违反物理定律的动画体验也会让人崩溃。

## 排版和颜色

### 排版必须保持清晰

清晰是排版第一要务。如果用户无法阅读文字内容，再漂亮的文字设计都没有意义。

在你的 app 中只使用一种字体。不同字体混合使用会让你的 app 看起来零散拖沓。



## 颜色可以增强沟通性

尝试定义 key color。内置软件使用了 key color——比如备忘录中的黄色——用来表明交互和元素状态。

颜色有代表性但人们对颜色的感知并不都相同。不同文化和个体对颜色都有不同理解，这值得花时间去钻研，以免使用的颜色在不同国家和文化中产生禁忌。

在大部分情况下，不要让颜色分散用户的注意力。除非颜色在你的 app 里扮演着必不可少的角色，颜色常常在不经意间带来增强效果。

## 图标和图形

### App 图标

每个 app 都需要一个漂亮的图标。每个人对图标都有先入为主的印象,关系到 app 的品质、目的和可靠性。



有几点关于图标的指引务必记住：

- app 图标是 app 品牌形象很重要的部分。让图标的设计成为一个机会，给用户讲设计背后的故事，并建立情感链接。
- 最好的应用程序图标应该是独一无二的，整洁的，打动人心的。
- 一个好的 app 图标在每种尺寸和不同背景下看起来都应该合适。细节设计在大尺寸下能丰富图标，但可能在小尺寸时会让图标显得浑浊（译者按：暗示图标图形设计需要简洁）。

### 其他图标

iOS 提供了许多代表了常见任务和操作的小图标，常被用在分页栏、工具栏和导航栏上。



建议使用容易被用户理解的内置图标。当然，可以使用自定义图标，如果需要表达自定义的操作或者内容。设计这些小的线型图标与 app 图标有很大区别。

## 图形

iOS 应用大多是图形丰富的。无论显示用户照片还是提供自定义的图片，都有一些应该遵循的准则：

**支持 Retina 显示屏。**请确保提供两种规格的图片资源。

**显示照片或图片时不要超过原始尺寸的 100%，**如果你不想在 app 中看到拉伸变形的图形的话。让用户来选择他们是否要放大或者缩小图片。

**不要使用带有苹果符号和版权的图片。**这些图形和版权产品的设计经常被修改。

（未完待续，控件和视图解析部分将在“iOS 人机界面指南翻译下”中呈现）

## 品牌

品牌塑造不止是展示一个品牌的颜色或者 logo。理想情况下，为你的 app 创造独特的外观和感觉并带给用户难忘的体验，进而打造出特有的品牌形象。

当你需要在应用中展示已有的品牌形象时，请记住下面的要点：

**巧妙自然（非强迫式）地融入品牌的颜色或图形。**人们使用你的 app 解决问题或者娱乐；他们并不想感觉像被迫看一个广告似的看到你的品牌宣传。最好的体验是，让你的 app 自己说话：比如 iOS7 通过品牌关键色来表现 app 的互动性和选择状态。

**避免侵占主要内容空间（用于展示品牌信息）。**例如，在屏幕上方展示一个用于承载品牌形象的信息栏，这样做反而占用了内容显示的空间。考虑使用其他低干扰的方式来宣传品牌，例如巧妙地定制背景图片（译者按：常见的方式是将品牌 logo 以淡淡的水印形式呈现在背景上）。

**重要：**对于以上的要点来说，app 图标是例外，它应该完全聚焦在品牌塑造上。因为用户经常看到 app 图标，更应该花时间来设计它，从而让其在具备品牌辨识度的基础上更加吸引眼球。

## 与 iOS 的整合

### 使用标准 UI 元素

尽可能用 UIKit 提供的标准 UI 元素。当你使用标准而非自定义元素时，你和你的用户都将受益：

- 标准 UI 元素会自动更新，如果 iOS 有了重新设计——而自定义元素就不会被升级。
- 使用标准元素对于用户来说没有学习成本。
- 为了充分利用标准 UI 元素的有点，以下几点比较关键：

- 遵循每个 UI 元素的设计规范。
- 大体来说，请避免创造自定义 UI 元素用于表现标准交互行为。
- 不要用系统自带的按钮和图标表达其他含义。
- 如果你的 app 是沉浸式体验，那么创造完全自定义的 UI 才有足够的意义。因为你在创造一个统一的体验环境，让用户在其中能够有所期待并探索如何控制 app。

## 对切换设备方向的响应

人们通常希望在各种方向都能使用 iOS 设备，所以在转换方向时设备应该有合适的响应。

不管设备处于什么方向，请聚焦于主要内容，这是最重要的。人们需要在使用 app 的过程中与其关心的内容交互。如果主体内容随着设备方向改变而丢失聚焦，那么用户就会感到迷茫并觉得丢失了对 app 的控制权。

通常，要让 app 能够在不同屏幕方向下正常运作。人们期望设备在不同方向时都能正常使用 app，能满足这一点是最好的。iPad 用户常常期望在当前把持设备的方向（正常）使用 app，但某些 app 只能在横屏下使用。如果确实是这样，请注意以下几点：

- 按默认支持的方向启动 app，忽略设备当前朝向。
- 避免在 UI 中告知用户需要调转设备方向。
- 横屏或竖屏模式时，支持水平调转设备。例如：在横屏 app 中，无论 home 键在左或右，app 都能正常使用，即支持设备调转 180 度，app 会自行响应。

如果你的 app 将方向变化当做一种交互的手段，那你可以将方向响应针对 app 进行特殊处理。比如某些方向响应的游戏通过改变设备方向来移动游戏中的物件，那此时 app 就无法响应改变设备方向原本应有的变化。这种情况下，可以让用户在进入主线任务前选择改变设

备方向；一旦开始主线任务，则按用户此前选择的方向为基准进行响应。

在 iPhone 上，预测用户什么时候会需要旋转屏幕方向。比如在用户浏览时，旋转方向是为了看到更多内容。如果此时 app 仅仅放大内容尺寸，你无法达到用户的预期，应该要重新调整内容布局，行间距等等直到更多内容能够很好的被呈现在屏幕上。

在 iPad 上，尽量支持所有方向来达到用户期望。iPad 的大屏幕（能呈现更多内容）减轻了用户在“看到更多”这方面的需求。iPad 也很少被用户认为有默认方向，如果可以，尽量满足在任何把持方向下都能正常让用户与你的 app 进行互动。遵循以下几点规范：

- **考虑改变显示辅助信息或功能的方式。**以 iPad 内置的邮件为例，账户和邮箱属于次要信息（选中的邮件是主要内容）。在横屏时，账户和邮箱被放在左侧面板中，而竖屏时出现在弹出面板上。

在某些游戏中，不同方向下的 UI 也许需要重绘，从而在边界上留下额外空间，这种情况下可以在这些地方展示游戏中的辅助信息或对象。

- **避免无意义的布局变化。**尽可能在不同方向下提供一致的体验，从而让用户在旋转屏幕时维持他们的操作习惯。如果你的 iPad app 在横屏时以网格形式展现图片，在竖屏时就完全没必要改成列表式。

- **避免重新定义信息或文本的方向。**对于文字内容来说，尽量保持相同的格式。这样可以避免用户在旋转屏幕时丢失文章定位。

如果某些样式必须发生改变，使用动画来帮助用户感受到变化过程。

- **为不同方向都准备一张独立的启动图片。**无论用户在何种方向启动应用，体验到的是平滑的启动过程。与 iPhone 的主屏幕不同，iPad 的主屏支持所有方向，以使用户退

出并继续打开同一个 app。

## 淡化文件和文档处理

iOS 可以帮助用户创建并管理文件，但并不代表用户必须考虑 iOS 设备的文件系统如何运作。

在 iOS 中没有类似于 OS X 系统中的 Finder( 管理软件 ), 用户无法像在电脑上那样操作。

特别是不应该让用户考虑文件所在的位置之类的东西，比如：

- 打开或保存文件的对话框
- 文件许可状态的信息

**尽可能允许用户无需在电脑上打开 iTunes 就能管理文档。**考虑使用 iCloud 帮助用户访问不同设备上的内容。

如果你的 app 能帮助用户创建并编辑文档，可以提供文件选择器让用户打开已有文件或者创建新文件。

**给用户信心，他们的工作成果会被随时保存除非主动取消或者删除。**iOS 应用应该承担起帮助用户保存输入内容的责任，无论是打开另一个文档或切换应用的时候。

如果你的 app 主要功能不是创造内容，但又允许用户查看或编辑信息，这种情况下你需要询问用户是否需要保存修改。提供“编辑”按钮点击后进入编辑状态，同时编辑按钮变成“保存”和“取消”按钮，这种变化可以提示用户处于编辑模式。“保存”可以 保留改变，“取消”可以退出编辑模式。

## 必要时提供可设置选项

某些应用需要安装或设置选项，但是大部分应用不需要这么做。一个成功的 app 可以让大部分用户上手迅速并通过主界面给用户调整体验的方式。

**避免让用户去（系统）设置中（寻找解决方法）。**请记住，用户必须关闭你的 app 才能进入系统设置，相信你也不希望用户这么做。

当你的 app（的默认状态）满足大部分用户的期望，（用户）对设置项的需求就减少了。

**如果有必要，让用户在你的 app 内进行设置。**提供设置选项可让你的 app 的变化直接体现，并让用户看到，也无需离开你的 app 再去设置。

尽可能在主界面提供设置选项。用户在执行主线任务时如果想频繁改变设置，放置在主界面的设置项就很有意义。如果用户只是偶尔用到设置项，就将其放在独立的视图中。



ISUX

腾讯社交用户体验设计  
[isux.tencent.com](http://isux.tencent.com)



官方博客，火热上线

感谢你的阅读，本文由 [腾讯 ISUX](#) 版权所有，转载时请注明出处，违者必究，谢谢你的合作。注明出处格式：腾讯 ISUX (<http://isux.tencent.com/?p=6668>)