# 1. 关于view controller

View controller是app的数据和视觉界面之间的重要的连接。当一个iOS app显示用户界面的时候，显示的内容一般都是由一个或者几个 view controller协同管理的。

iOS内置了很多的view controller来支持一般的用户接口，比如导航(navigation), 标签栏(tab bar)。我们在开发app的时候通常会自定义view controller来展示用户界面。

# 2. At a Glance

view controller在经典的mvc模型中是controller部分，但实际上他的工作还不止这些。系统内置的view controller基类提供基础的功能，用户可以继承并扩展自己的接口，来自定义app的行为。

# 3. 管理一组view

view controller 管理应用程序的用户界面，一般他会提供一个view，这个view是一组复杂view层级的根view(root view)，上面的子view一般是iOS系统提供的button, label等控件元素。

# 4.容器视图控制器(controller view controller)管理其他的controller

容器(container view controller)会展示其他view controller管理的内容，容器与管理的controller之间以父子关系显式的连接在一起，形成单根的继承层级。

View controller可以分为两类：容器（container）controller和内容（content）controller

# 5. View Controller Basics

不能直接把view加到window上，应该把view controller设置给window(rootViewController)，window会自动把这个view controller的view加到上面。

每个view只由一个view controller管理（通过\_viewDelegate成员变量），一个view如果是view controller的view属性(property)，那么这个view就由这个controller管理，如果一个view是subview的话，那么他可能被当前的view controller（本view层级的根view的controlller）管理，也可能是其他的view controller管理（比如，把另外一个view controller的view加到当前view上，那么这些subview就由原先的view controller管理）。

一个界面尽量用一个view controller来管理，不要用多个view controller除非需要公共的控件——那样也可以将控件做成control，而不是一个单独的controller。

# 6. About Navigation Controllers

Navigation controller除了管理子controller，还管理一些界面元素，如navigation bar和button。被navigation controller管理的controller可以通过navigationController属性来访问自己所在的navigation controller，而navigation controller也有一个root view controller作为自己的栈底view controller。

# 7. About Split View Controllers

iPad only， 且横屏和竖屏的形式是不一样的——为了更好的利用iPad的大屏幕。

# 8. About Page View Controllers

以翻书的效果切换view。

# 9. A View Controller’s Content Can Be Displayed in Many Ways

presentViewController:animated:completion这种方法用在当一个view controller调用另外一个view controller，结束后再返回调用者的时候，很有用（iOS5.0以后提供的）。被显示的view controller还可以继续再显示其他的view controller，组成一个模态交互的chain。另外，还可以控制被调view controller的显示区域——通过presentation context，默认是整个window。

# 10. View Controllers Work Together to Create an App’s Interface

view controller的层级结构的根是window的root view controller。每一个view controller都被另外一个view controller展示，这2个controller之间形成了一个pair的关系，一般source controller设置destination controller的属性并显示destination controller，如果destination controller需要返回数据或者操作给source controller，一般会使用delegate。

# 11. Storyboards Help You Design Your User Interface

story board 是否可以成为一个很好的原型工具呢？

# 12. IMPORTANT [练习]

**Important:** Never install the view controller’s view into a view hierarchy directly. To present and manage views properly, the system makes a note of each view (and its associated view controller) that you display. It uses this information later to report view controller–related events to your app. For example, when the device orientation changes, a window uses this information to identify the front most view controller and notify it of the change. If you incorporate a view controller’s view into your hierarchy by other means, the system may handle these events incorrectly.

If you are implementing your own custom container controller, you add another view controller’s view to your own view hierarchy, but you also create a parent-child relationship first. This ensures that events are delivered correctly. See [“Creating Custom Container View Controllers.”](file:///Users/majie/Library/Developer/Shared/Documentation/DocSets/com.apple.adc.documentation.AppleiOS6.0.iOSLibrary.docset/Contents/Resources/Documents/featuredarticles/ViewControllerPGforiPhoneOS/CreatingCustomContainerViewControllers/CreatingCustomContainerViewControllers.html#//apple_ref/doc/uid/TP40007457-CH18-SW6)

永远不要直接把一个view controller的view加到一个view层级里面，因为系统会根据view和controller的关系来处理一些系统事件——比如说转屏。不过这个规则又和container controller相矛盾了，因此如果想设计自己的container controller，需要遵循一定的规则：创建parent-child关系。