

huifeidexin_1的专栏

目录视图

摘要视图

RSS 订阅

个人资料



huifeidexin_1

访问： 551716次

积分： 4562

等级： 

排名： 第3081名

原创： 15篇

转载： 102篇

译文： 0篇

评论： 66条

文章搜索

文章分类

object-c (16)

ios开发 (99)

C/C++ (2)

文章存档

2015年06月 (1)

2015年01月 (2)

2014年12月 (4)

2014年09月 (3)

2014年08月 (7)

展开

阅读排行

ios中关于delegate（委托

(68351)

UIViewController的生命周期

(42064)

iOS 动画总结---UIView

(40491)

UIApplication sharedApplication

(34615)

UIPanGestureRecognizer

(28892)

iOS 基本编码格式转化

(23389)

如何获取iphone的硬件版

(21047)

iOS开发之异步加载图片

(20309)

写博客，送money、送书、送C币啦

7-8月博乐推荐文章

砸BUG 得大奖 100%中奖率

微信开发学习路线高级篇上线

UIViewController的生命周期及iOS程序执行顺序

分类： [iOS开发](#)

2012-05-14 20:07

42091人阅读

[评论\(4\)](#)

[收藏](#)

[举报](#)

[ios](#)

[uiview](#)

[hierarchy](#)

[accessor](#)

[interface](#)

[apple](#)

当一个视图控制器被创建，并在屏幕上显示的时候。 代码的执行顺序

- 1、 alloc 创建对象，分配空间
- 2、 init (initWithNibName) 初始化对象，初始化数据
- 3、 loadView 从nib载入视图， 通常这一步不需要去干涉。除非你没有使用xib文件创建视图
- 4、 viewDidLoad 载入完成， 可以进行自定义数据以及动态创建其他控件
- 5、 viewWillAppear 视图将出现在屏幕之前， 马上这个视图就会被展现在屏幕上了
- 6、 viewDidAppear 视图已在屏幕上渲染完成

当一个视图被移除屏幕并且销毁的时候的执行顺序， 这个顺序差不多和上面的相反

- 1、 viewWillDisappear 视图将被从屏幕上移除之前执行
- 2、 viewDidDisappear 视图已经被从屏幕上移除， 用户看不到这个视图了
- 3、 dealloc 视图被销毁， 此处需要对你在init和viewDidLoad中创建的对象进行释放

关于viewDidUnload： 在发生内存警告的时候如果本视图不是当前屏幕上正在显示的视图的话， viewDidUnload将会被执行， 本视图的所有子视图将被销毁， 以释放内存,此时开发者需要手动对viewLoad、viewDidLoad中创建的对象释放内存。 因为当这个视图再次显示在屏幕上的时候， viewLoad、viewDidLoad 再次被调用， 以便再次构造视图。

当我们创建一个UIViewController类的对象时， 通常系统会生成几个默认的方法， 这些方法大多与视图的调用有关， 但是在视图调用时， 这些方法的调用顺序如何， 需要整理下。

通常上述方法包括如下几种， 这些方法都是UIViewController类的方法：

- (void)viewDidLoad;

- (void)viewDidUnload;

- (void)viewWillAppear:(BOOL)animated;

- (void)viewDidAppear:(BOOL)animated;

- (void)viewWillDisappear:(BOOL)animated;

- (void)viewDidDisappear:(BOOL)animated;

下面介绍下APP在运行时的调用顺序。

- 1) - (void)viewDidLoad;

一个APP在载入时会先通过调用loadView方法或者载入IB中创建的初始界面的方法， 将视图载入到内存中。 然后会调用viewDidLoad方法来进行进一步的设置。 通常， 我们对于各种初始数据的载入， 初始设定等很多内容， 都会在这个方法中实现， 所以这个方法是一个很常用， 很重要的方法。

但是要注意， 这个方法只会在APP开始加载的时候调用一次， 以后都不会再调用它了， 所以只能用来做初始

http://blog.csdn.net/huifeidexin_1/article/details/7566226

1/5

iOS的相对路径和绝对路径 (18037)
<原>关键帧动画CAKeyframeAnimation (16014)

评论排行

- AFNetwork 2.0在请求时报错code 351 (14)
- ios中关于delegate（委托）的使用 (11)
- iOS 基本编码格式转化 (11)
- UIApplication sharedApplication (6)
- UIViewController的生命周期 (4)
- ios小项目——新浪微博登录 (2)
- IOS 分享新浪微博开发通 (2)
- IOS UITableView中行的动画 (2)
- iOS开发之异步加载图片 (2)
- Objective C定义私有方法 (2)

推荐文章

最新评论

- UIApplication sharedApplication: 水平不够，没有看懂，用法大致知道了
- AFNetwork 2.0在请求时报错code 351: 楼主可以的，mark
- AFNetwork 2.0在请求时报错code 351: 看来不是只有我会这么写
- iOS 基本编码格式转化
zy851473663: +(NSString *)utf8ToUnicode:(NSString *)string{ ...
- iOS 基本编码格式转化
zy851473663: 对于再网上搜资料搜到的都是同一个内容表示痛恨，对于你经实践就好评的表示鄙视。下边是改进后的代码。希...
- iOS 基本编码格式转化
zy851473663: NSString *test = @"http://www.baidu.com?a=b&c=d"; ...
- AFNetwork 2.0在请求时报错code 351: AFHTTPResponseSerializer serializer中间要加空格 我相信我不是一个...
- AFNetwork 2.0在请求时报错code 351: 嗯 没错 就是这个为题！感谢！！
- Objective C定义私有方法
u012986410: 这是扩展吧
- ios中关于delegate（委托）的使用
AGFFGH: 不错的文章，学习了

设置。

2) - (void)viewDidUnload

在内存足够的情况下，视图对象通常会一直保存在内存中，但是如果内存不够，一些没有正在显示的viewController就会收到内存不够的警告，然后就会释放自己拥有的视图，以达到释放内存的目的。但是系统只会释放内存，并不会释放对象的所有权，所以通常我们需要在这里将不需要在内存中保留的对象释放所有权，也就是将其指针置为nil。

这个方法通常并不会在视图变换的时候被调用，而只会在系统退出或者收到内存警告的时候才会被调用。但是由于我们需要保证在收到内存警告的时候能够对其作出反应，所以这个方法通常我们都需要去实现。

另外，即使在设备上按了Home键之后，系统也不一定会调用这个方法，因为IOS4之后，系统允许将APP在后台挂起，并将其继续滞留在内存中，因此，viewController并不会调用这个方法来清除内存。

3) - (void)viewWillAppear:(BOOL)animated;

系统在载入所有数据后，将会在屏幕上显示视图，这时会先调用这个方法。通常我们会利用这个方法，对即将显示的视图做进一步的设置。例如，我们可以利用这个方法来自设置设备不同方向时该如何显示。

另外一方面，当APP有多个视图时，在视图间切换时，并不会再次载入viewDidLoad方法，所以如果在调入视图时，需要对数据做更新，就只能在这个方法内实现了。所以这个方法也非常常用。

4) - (void)viewDidAppear:(BOOL)animated;

有时候，由于一些特殊的原因，我们不能在viewWillAppear方法里，对视图进行更新。那么可以重写这个方法，在这里对正在显示的视图进行进一步的设置。

5) - (void)viewWillDisappear:(BOOL)animated;

在视图变换时，当前视图在即将被移除、或者被覆盖时，会调用这个方法进行一些善后的处理和设置。

由于在IOS4之后，系统允许将APP在后台挂起，所以在按了Home键之后，系统并不会调用这个方法，因为这个APP本身而言，APP显示的view，仍是挂起时候的view，所以并不会调用这个方法。

6) - (void)viewDidDisappear:(BOOL)animated;

我们可以重写这个方法，对已经消失，或者被覆盖，或者已经隐藏了的视图做一些其他操作。

上述方法的流程图可以简单用如下表示：

运行APP —> 载入视图 —> 调用viewDidLoad方法 —> 调用viewWillAppear方法 —> 调用viewDidAppear方法 —> 正常运行



释放对象所有权 <— 调用viewDidUnload <— 收到内存警告 <— 调用viewDidDisappear <— 调用

viewWillDisappear <— APP需要调用另一个view

IOS程序启动执行顺序

<http://www.yifeiyang.net/iphone-developer-advanced-3-iphone-application-startup-process/>

IOS 开发 loadView 和 viewDidLoad 的区别

iPhone开发必不可少的要用到这两个方法。他们都可以用来在视图载入的时候，初始化一些内容。但是他们有什么区别呢？

viewDidLoad 此方法只有当view从nib文件初始化的时候才被调用。

loadView 此方法在控制器的view为nil的时候被调用。此方法用于以编程的方式创建view的时候用到。如：

```
1.
2. - ( void ) loadView {
3.     UIView *view = [ [ UIView alloc] initWithFrame:[ UIScreen
4.     mainScreen] .applicationFrame] ;
5.     [ view setBackgroundColor:_color] ;
6.     self.view = view;
7.     [ view release] ;
8. }
9.
```

你在控制器中实现了loadView方法，那么你可能会在应用运行的某个时候被内存管理控制调用。如果设备内存不足的时候， view 控制器会收到didReceiveMemoryWarning的消息。默认的实现是检查当前控制器的view是否在使用。如果它的view不在当前正在使用的view hierarchy里面，且你的控制器实现了loadView方法，那么这个view将被release, loadView方法将被再次调用来创建一个新的view。

Don't read self.view in -loadView. Only set it, don't get it.

The self.view property accessor *calls* -loadView if the view isn't currently loaded. There's your infinite recursion.

The usual way to build the view programmatically in -loadView, as demonstrated in Apple's pre-Interface-Builder examples, is more like this:

```
UIView    * view    =    [[ UIView alloc ] init ...];
...
[ view addSubview : whatever ];
[ view addSubview : whatever2 ];
...
self . view    = view ;
[ view release ];
```

And I don't blame you for not using IB. I've stuck with this method for all of Instapaper and find myself much more comfortable with it than dealing with IB's complexities, interface quirks, and unexpected behind-the-scenes behavior.

[上一篇](#) iOS Notification

[下一篇](#)
[iPhone http Post数据](#)

[主题推荐](#)
[ios](#)
[uiviewcontroller](#)
[生命周期](#)
[代码](#)
[class](#)

猜你在找

Apple Watch开发入门	UIViewController的生命周期及iOS程序执行顺序
iOS8开发技术（Swift版）：iOS基础知识	UIViewController的生命周期及iOS程序执行顺序
iOS开发教程之OC语言	UIViewController的生命周期及iOS程序执行顺序
iOS8开发技术（Swift版）：常用控件	UIViewController的生命周期及iOS程序执行顺序
APP内置IM 系统——从入门到千万级在线	UIViewController的生命周期及iOS程序执行顺序

[准备好了么？跳 吧！](#)
[更多职位尽在 CSDN JOB](#)

JAVA程序员	我要跳槽	C++客户端程序	我要跳槽
德州甘者糖业有限公司	5-10K/月	上海游唐网络技术有限公司	7-14K/月
PHP程序	我要跳槽	安卓程序猿	我要跳槽
重庆果动科技有限公司	4-8K/月	广州市牧车网络科技有限公司	4-8K/月



阿里云数据库

28.8元/月起

立即购买 >

- 1. 全SSD盘，完全兼容MySQL、SQL Server、PostgreSQL
- 2. 采取双机热备主从结构，故障秒级切换
- 3. 提供专业的备份恢复、安全监控和性能优化方案
- 4. 强大的源码控制力（亚太地区唯一的WebScaleSQL成员）
- 5. 中国最专业的DBA团队24小时护航服务

查看评论

4楼
[shanchelicun](#)
2015-06-11 14:09发表



蛮好的

3楼
[qiaoamyangpu](#)
2015-02-22 10:04发表



感谢分享

2楼
[fsssssss](#)
2014-10-09 15:01发表



写（转）的蛮好，

1楼
[zingos](#)
2012-10-23 14:55发表



系统会在内存不足时调用viewDidUnload，再次展示该页面时，还会调用viewDidLoad

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题	Hadoop	AWS	移动游戏	Java	Android	iOS	Swift	智能硬件	Docker	OpenStack
VPN	Spark	ERP	IE10	Eclipse	CRM	JavaScript	数据库	Ubuntu	NFC	WAP
BI	HTML5	Spring	Apache	.NET	API	HTML	SDK	IIS	Fedora	XML
Splashtop	UML	components	Windows Mobile	Rails	QEMU	KDE	Cassandra	CloudStack	FTC	
coremail	OPhone	CouchBase	云计算	iOS6	Rackspace	Web App	SpringSide	Maemo		
Compuware	大数据	aptch	Perl	Tornado	Ruby	Hibernate	ThinkPHP	HBase	Pure	Solr
Angular	Cloud Foundry	Redis	Scala	Django	Bootstrap					