

窗口 UI window

视图 UIView

视图控制器 UIViewController

UIKit

显示数据的视图 (UITextView.UILabel.UITableView)

做出选择的视图 (UIAlertView.UIActionSheet)

其他 (UIButton.UISegmentControl.UITableView)

所有的对象都是继承自NSObject

在字符转义时可以使用%@

[[NSObject alloc] init] alloc init 相当于new关键字

输出对象就是输入对象的内存地址

其实是执行了NSObject中的description方法 此方法返回的是一个NSString

此方法是可以重写的

第一个文件夹里面大部分都是iOS的源码文件

storyboard文件 属于ios的界面设计文件

Supporting 文件夹里面是存放资源文件的 比如可以拖入图片

可以i 直接在main.storyboard文件中使用加入view Controller进行跳转

方法调用 可以直接食用c语言的方法命名以及方法调用

```
void method(){  
    }  
}
```

调用 method();

其中是可以引入c和c++语言文件的

TableView的处理

<http://blog.csdn.net/crazychickone/article/details/38704503>

NSUnknownKeyException有的代理已经失效 但是没有清处代理关系

实体的声明必须 要有实现

NSMutableArray 在使用前必须

Android 静态常量的生命周期不回在activity销毁时销毁

@property的各个属性值的含义

<http://blog.csdn.net/dqjyong/article/details/7668601>

两个字符串的 定义都为strong 给字符串1复制为@"str1" 将字符串1的内容复制给字符串2 然后将 字符串1清空 此时字符串2打印出来是str1

两个字符串的 定义一个为strong 一个定义为weak 那么它1的内容复制给2之后 清空1 2也会同事清空

一个定义为strong 一个定义为__unsafe_unretained 那么在上述操作之后 第二个字符串不为空 但是是野指针

strong,weak, unsafe_unretained往往都是用来声明属性的，如果想声明临时变量就得用__strong, __weak, __unsafe_unretained, __autoreleasing, 其用法与上面介绍的类似。

@property来标识属性

变量可分为只读变量，那么变量的属性为readonly

而copy与retain的具体区别为：**copy**其实是建立了一个相同的对象，而**retain**只是保存其对象，并且其计数值+1。

例如：一个NSString对象，地址为0x1000，内容为@"string"

copy到另外一个NSString之后，地址为0x2000，内容相同，新的对象retain为1，旧有对象没有变化

retain到另外一个NSString之后，地址相同（建立一个指针，指针拷贝），内容当然相同，但是这个新对象的retain值+1，并释放旧的对象。

也就是说，**retain**是指针拷贝，**copy**是内容拷贝。

Android就是一些view属性的集合 可以继承 也可以多继承

[http://115.29.111.103:8080/issues/?jqI=project%20%3D%20GC%20AND%20status%20in%20\(Open%2C%20%22In%20Progress%22%2C%20Reopened\)%20ORDER%20BY%20createdDate%20DESC](http://115.29.111.103:8080/issues/?jqI=project%20%3D%20GC%20AND%20status%20in%20(Open%2C%20%22In%20Progress%22%2C%20Reopened)%20ORDER%20BY%20createdDate%20DESC)

<svn://wycode.cn/projects/WyCode/ios>

属性前面加上 IBOutlet

联想字

Cy123456

18291489600

_codeTF.keyboardType=***UIKeyboardTypeNumberPad***;
弹出数字键盘

4.声明三方maven仓库:

可能你项目需要的一些库文件是在你们公司的私服上,这时候repositories中仅有jcenter就不行了,你还需要把私服地址配到里面来,注意,应该配到project的build.gradle中的allprojects结点下或者直接配到某个模块中如果仅有这个模块用到。配置方式:

```
1
2
3
4
5 repositories{
    maven{
        url="http://mvnrepo.xxx.com"
    }
}
```

8.排除依赖:

当出现依赖冲突的时候可以通过排除依赖解决,具体方式如下:

```
compile (group:'xxx',name:'xxx',version:'xxx'){
    exclude group:'xxx',module:'xxx'//module对应的就是artifactId
}
```

真实项目中 如果需要debugKey 将信息配置到gradle.properties中 然后在build.gradle中引用

数组可以给数组赋值

```
//忽略xlint异常
//allprojects {
//    gradle.projectsEvaluated {
//        tasks.withType(JavaCompile) {
//            options.compilerArgs << "-Xlint:unchecked" << "-Xlint:deprecation"
//        }
//    }
//}
```

<http://mikewang.blog.51cto.com/3826268/1003276>

```
Activity com.ourslook.changyu.activity.MessageCenterActivity has leaked window
com.android.internal.policy.impl.PhoneWindow$DecorView{42e5d858 V.E..... R.....D
0,0-720,1280} that was originally added here
                                android.view.WindowLeaked: Activity
com.ourslook.changyu.activity.MessageCenterActivity has leaked window
com.android.internal.policy.impl.PhoneWindow$DecorView{42e5d858 V.E..... R.....D
0,0-720,1280} that was originally added here
                                at
android.view.ViewRootImpl.<init>(ViewRootImpl.java:432)
```

如果activity以及结束 但是依靠他的dialog还没有dismiss 那么就会出现窗口泄漏 在这种情况下 可以在activitydestory之前将dialog结束

UIScrollView 一个能够滚动的视图 通过滚动查看内容

[self.navigationController pushViewController:detailVc animated:YES]; 跳转

通知

```
[[NSNotificationCenter defaultCenter]postNotificationName:@"qudiaojiekouba"
object:nil userInfo:@{@"key":@"value"}];
[[NSNotificationCenter defaultCenter]addObserver:self
selector:@selector(woqudiaojiekoule:) name:@"qudiaojiekouba" object:nil];
-(void)woqudiaojiekoule:(NSNotification *)notification {
    NSDictionary *dic=notification.userInfo;
}
```

删除分割线

```
- (void)setExtraCellLineHidden: (UITableView *)tableView
{
    UIView *view =[ [UIView alloc]init];
    view.backgroundColor = [UIColor clearColor];
    [tableView setTableFooterView:view];
}
```

clipboardManager

//颜色互转

```
long a=Long.parseLong("ffff00",16);
    int color= (int) (a&0x00000000ff000000);
//    textView1.setTextColor(Color.parseColor("#ff0000"));
    textView1.setTextColor(color);
```

maven center

```
if(ContextCompat.checkSelfPermission(mContext,
android.Manifest.permission.CAMERA)!
=PackageManager.PERMISSION_GRANTED||
    ContextCompat.checkSelfPermission(mContext,
android.Manifest.permission.WRITE_EXTERNAL_STORAGE)!
=PackageManager.PERMISSION_GRANTED
){

    ActivityCompat.requestPermissions(mContext, new String[]
{android.Manifest.permission.CAMERA,
    android.Manifest.permission.MOUNT_UNMOUNT_FILESYSTEMS,

android.Manifest.permission.WRITE_EXTERNAL_STORAGE},Constants.CODE_20
1);
}else{

    if(grantResults[0]==
PackageManager.PERMISSION_GRANTED&&grantResults[1]==
PackageManager.PERMISSION_GRANTED){
        try {
            //    String imagePath = Environment.getExternalStorageDirectory() +
Constants.IMAGE_PATH + "userIcon.jpg";
            File imageFile = new File(Constants.Chhoose_imagePath);
            Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
```

```

        intent.putExtra(MediaStore.EXTRA_OUTPUT,
Uri.fromFile(imageFile));
        if (intent.resolveActivity(mContext.getPackageManager()) != null) {
            startActivityForResult(intent,
Constants.CODE_REQUEST_CAMERA);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
} else {
    ToastUtil.showToastDefault(mContext, "权限被拒绝");
}

```

```

{"code":1,"message":"成功!","object":{"id":3,"createTime":"2016-03-02
11:44:10","mobile":"18392974435","photo":"/upload/user/
1458274458892.jpg","isPush":1,"account":2830003,"glodBean":191,"dayNumber":
5,"existPay":1,"jpushId":"o1md9qtin1lvssys"}}

```

<http://blog.csdn.net/congqingbin/article/details/7869730> 自定义属性的使用

```

if(!str.length){
    [SVProgressHUD showErrorWithStatus:@"内容不能为空"];
    return;
}
非0是1 1是YES

```

三种ViewController跳转的异同

- (void)presentViewController:(UIViewController *)viewControllerToPresent
 animated:(BOOL)flag completion:(void (^)(void))completion
 MainVC *mainVC = [[MainVC alloc] init];

[self presentViewController:mainVC animated:YES completion:nil];

这种方式一般出现在需要使用者完成某件事情，如输入密码、增加资料等操作后，才能(回到跳转前的控制器)继续。例如系统的WIFI连接输入密码提示。默认动画是从下至上。

- (void)pushViewController:(**UIViewController** *)viewController animated:
 (BOOL)animated

这种方式一般是使用者浏览资料，继而可以前进到下一个页面或回到上一个页面。默认动画是从右至左。

- (void)addChildViewController:(UIViewController *)childController

这个方法出现在iOS5以后，通过它即使不使用NavigationController也能够实现view hierarchy。有以下优点：

1. 页面逻辑很清晰，相应的View对应相应的ViewController。
2. 当某个子View没有显示时，将不会被Load，减少了内存的使用。
3. 当内存紧张时，没有Load的View将被首先释放，优化了程序的内存释放机制。

push方法时NavigationController的方法 present是viewController的方法

iOS的 dialog使用

```
UIAlertView *alter = [[UIAlertView alloc] initWithTitle:@"提示" message:@"你点击了  
导航栏右按钮" delegate:self cancelButtonTitle:@"确  
定" otherButtonTitles:nil, nil]; [alter show];
```

mac配置环境变量 cd~ touch .bash_profile

打开文件 open -e .bash_profile 在其中配置环境变量

source .bash_profile 刷新配置的环境变量

```
export PATH=${PATH}:/Users/huangyi/Library/Android/sdk/platform-tools  
export PATH=${PATH}:/Users/huangyi/Library/Android/sdk/tools
```

su 请求权限

- 1、打开文件夹的方法使用：cd 文件夹 -----进入文件夹，ll ----查看文件夹里的文件
- 2、打开文件的方法：编辑文件用 vi 文件名 或者 vim 文件名。退出按ESC，输入：q! 回车。

查看文件用 cat 文件名 或 less 文件名

如果tableView是plain类型，那么cell的背景颜色就是tableView的背景颜色，就算设置了cell.backgroundColor貌似也没用

如果tableView是group类型，cell的背景颜色跟tableView的背景无关，需要单独在设置cell.backgroundColor

```
@property(nonatomic,retain) UIView                      *backgroundView;
```

`[tableView deselectRowAtIndexPath:indexPath animated:YES];` cell点击后松开颜色恢复

设置图片圆角

```
[_image.layer setMasksToBounds:YES];  
//设置圆角半径  
_image.layer.cornerRadius=_image.frame.size.width/2.0f;
```

设置图片是否可以交互

```
_headerImageView.userInteractionEnabled = YES;
```

self和_的区别是 如果你想使用此对象的set get方法 那么 就要使用self

```
[_UIImage setUserInteractionEnabled:YES];
```

设置imageview可点击

cell上的label可交互 必须要在xib上设置可交互

```
nobanklable.textAlignment = NSTextAlignmentCenter;
```

//设置文字居中

Android属性allowBackup安全风险浅析

当在application里面设置 **allowBackup=true**时 此时程序可以回复可以备份 尤其是通讯录应用，一旦应用程序支持备份和恢复功能，攻击者即可通过adb backup和adb restore进行恢复新安装的同一个应用来查看聊天记录等信息；对于支付金融类应用，攻击者可通过此来进行恶意支付、盗取存款等；因此为了安全起见，开发者务必将allowBackup标志值设置为false来关闭应用程序的备份和恢复功能，以免造成信息泄露和财产损失。

颠倒布局 在application里面设置 **android:supportsRtl="true"** 在相应的布局标签中加入**android:layoutDirection="rtl"**

此布局下面的属性将会从右边开始

java环境变量

path环境变量的作用 让系统去识别java命令 让操作系统能够找到 **java.exe** 和**javac.exe**两个程序

classpath 比如在 **a.class**中使用了 **b.class**里面的内容 那么我们需要让系统首先能够找到 **b.class**的位置

javahome 这个可以让我们更方便的使用jdk的路径 比如在配置 **path**以及**classpath**时我们可以直接使用**javahome**

jdk java Development Kit **jdk**就是面向java开发人员使用的**sdk** 包括了 java开发环境以及java 运行环境

sdk software Develop **sdk**就是软件开发包 包括函数库和编译程序

jre是 java runtime Environment 是指java的运行环境 是对于java的使用者而言的 就像一台pc一样 可以运行java的程序 包括 java的pc虚拟机 所以基本上只要你的电脑上安装了java **jre**你的电脑上就可以运行java程序了

jvm java virtual machine **JVM** 指的是java虚拟机 他是整个java实现跨平台最核心的部分 所有的java文件都会被受限变异成.class文件 这种类文件可以直接与java虚拟机进行交互 也就是说他不直接与机器的操作系统交互 而是通过java虚拟机将其解释给操作系统

jre **jre**指的是java 运行环境 逛游**jvm**还不能完成class的执行，因为在解释class的时候**jvm**需要的类库lib 在**jdk**的安装目录里面可以找到**jre** 目录 里面有两个文件夹bin和lib 在这里可以认为bin里可以认为bin里就是**jvm** lib中则是**jvm** 工作所需要的类库 而**jvm**和lib合起来就成为**jre** 所以在你写完java程序编译成.class 之后 你可以把这个.class文件和**jre**一起打包发给朋友，这样你的朋友就可以运行你写的程序了 (**jre**里面有运行.class的 java.exe)

jdk在其目录下面有六个文件夹。一个src类库源码压缩包和其他几个声明文件 其中在运行java时起作用的是一个四个文件夹：bin,include,lib,jre

jdk包含**jre**,而包含**jvm** bin:最主要的是编译器 (javac.exe) include:java和**jvm**交互用的头文件lib:类库**jre**:java运行环境 (注意：这里的bin,lib文件和**jre**里的bin, lib是不同的) 总的来说**jdk**是用于java开发 而**jre**则只是运行.class 而没有编译的功能

lable的下划线

```
//CGContextAddLines(CGContextRef c, const CGPoint points[],size_t count)
//points[]坐标数组, 和count大小
// NSMutableAttributedString *content = [[NSMutableAttributedString
alloc]initWithString:[NSString stringWithFormat:@"《报名协议》"];
// NSRange contentRange = {0,[content length]};
// [content addAttribute:NSUnderlineStyleAttributeName value:[NSNumber
numberWithInteger:NSUnderlineStyleSingle] range:contentRange];

// self.footer.agreementLabel.attributedText = content;
```

输入框弹起键盘

```
// CGPoint point= scrollView.contentOffset;
// DLOG(@"标记%f",point.y);
// scrollView.contentOffset.y;
```

//#pragma mark - 屏幕上弹

```
//-(void)textFieldDidBeginEditing:(UITextField *)textField
//{
// //键盘高度216
//
// //滑动效果 (动画)
// NSTimeInterval animationDuration = 0.30f;
// [UIView beginAnimations:@"ResizeForKeyboard" context:nil];
// [UIView setAnimationDuration:animationDuration];
//
//
// //1.计算当前点击的输入框相对于屏幕的位置
// CGFloat llocal=textField.frame.size.height*(textField.tag+1)-scrollView.contentOffset.y+64;
// DLOG(@"屏幕高度%f点击高度%f",UIScreenHeight,llocal);
// //2.判断次textfield是否需要使view上移
// if(UIScreenHeight-llocal<252){//需要进行移动处理
// //2.判断需要上移多少
// CGFloat offset=252-(UIScreenHeight-llocal);
// DLOG(@"偏移%f",offset);
// //3.开始移动跟布局
// self.view.frame = CGRectMake(0.0f, -offset, self.view.frame.size.width,
self.view.frame.size.height);//64-216
```

```

//
// }
//
// [UIView commitAnimations];
//}
//
//-(void)touchesBegan:(NSSet<UITouch *> *)touches withEvent:(UIEvent *)event{
//  NSSet *allTouches = [event allTouches]; //返回与当前接收者有关的所有的触摸
//  对象
//  UITouch *touch = [allTouches anyObject]; //视图中的所有对象
//  CGPoint point = [touch locationInView:[touch view]]; //返回触摸点在视图中的当前
//  坐标
//  int x = point.x;
//  int y = point.y;
//  NSLog(@"touch (x, y) is (%d, %d)", x, y);
//}
//
//#pragma mark -屏幕恢复
//-(void)textFieldDidEndEditing:(UITextField *)textField
//{
//
//
//// 滑动效果
//  NSTimeInterval animationDuration = 0.30f;
//  [UIView beginAnimations:@"ResizeForKeyboard" context:nil];
//  [UIView setAnimationDuration:animationDuration];
//// 恢复屏幕
//  self.view.frame = CGRectMake(0.0f, -0.0f, self.view.frame.size.width,
//  self.view.frame.size.height);
//  [UIView commitAnimations];
//}

```

<http://114.215.84.189:8888/liuda/docs/#!/guide/saveGuide>

如果android studio中丢失了 subVersionControl 那么 在插件中搜索 使用即可
 Android Studio中的svn突然不见了，解决方法如下：File->Setting->搜索plugin->搜索
 Subversion Integration，然后选中，再apply一下、重启Android Studio即可

scroll方法 必须要在 所有的view加载完成后 才能使用

calendar.getActualMaximum(Calendar.**DATE**)//获取当前月份的天数

gradle.properties文件

此文件是用来进行一些特殊的gradle设置

ViewController的结束 present用 [self
dismissModalViewControllerAnimated:YES];//退出当前viewController

push用 [self.navigationController
popViewControllerAnimated:YES];

Looper.prepare()方法不在一个线程中被调用两次 因为一个线程只能有一个Loop

第二三个方法是用来截取字符串
canvas.drawText("狗蛋",0,1,50,50,paint);

webView.loadData(data, "text/html; charset=UTF-8", null);//这种写法可以正确解码

Integer的最大值 ((2<<30)-1)