**1、设置UILabel行间距**

NSMutableAttributedString\* attrString = [[NSMutableAttributedString  alloc] initWithString:label.text];

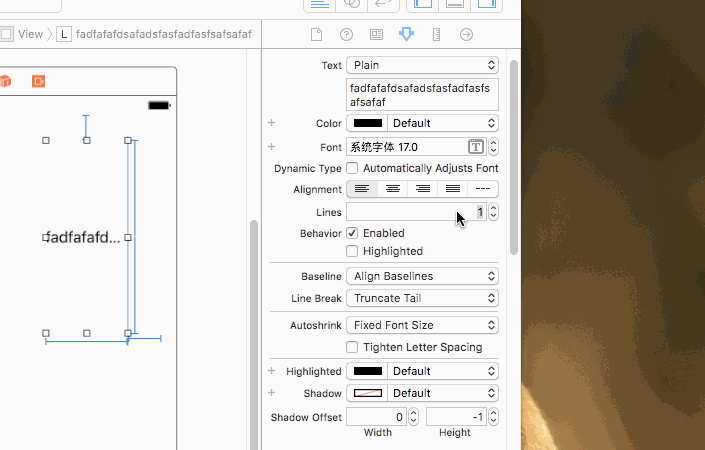
    NSMutableParagraphStyle \*style = [[NSMutableParagraphStyle alloc] init];

    [style setLineSpacing:20];

    [attrString addAttribute:NSParagraphStyleAttributeName value:style range:NSMakeRange(0, label.text.length)];

    label.attributedText = attrString;

// 或者使用xib，看下gif图



Untitled.gif

**2、当使用-performSelector:withObject:withObject:afterDelay:方法时，需要传入多参数问题**

// 方法一、

// 把参数放进一个数组／字典，直接把数组／字典当成一个参数传过去，具体方法实现的地方再解析这个数组／字典

NSArray \* array =

    [NSArray arrayWithObjects: @"first", @"second", nil];

[self performSelector:@selector(fooFirstInput:) withObject: array afterDelay:15.0];

// 方法二、

// 使用NSInvocation

SEL aSelector = NSSelectorFromString(@"doSoming:argument2:");

    NSInteger argument1 = 10;

    NSString \*argument2 = @"argument2";

    if([self respondsToSelector:aSelector]) {

        NSInvocation \*inv = [NSInvocation invocationWithMethodSignature:[self methodSignatureForSelector:aSelector]];

        [inv setSelector:aSelector];

        [inv setTarget:self];

        [inv setArgument:&(argument1) atIndex:2];

        [inv setArgument:&(argument2) atIndex:3];

        [inv performSelector:@selector(invoke) withObject:nil afterDelay:15.0];

    }

**3、UILabel显示不同颜色字体**

NSMutableAttributedString \* string = [[NSMutableAttributedString alloc] initWithString:label.text];

[string addAttribute:NSForegroundColorAttributeName value:[UIColor redColor] range:NSMakeRange(0,5)];

[string addAttribute:NSForegroundColorAttributeName value:[UIColor greenColor] range:NSMakeRange(5,6)];

[string addAttribute:NSForegroundColorAttributeName value:[UIColor blueColor] range:NSMakeRange(11,5)];

label.attributedText = string;

**4、比较两个CGRect/CGSize/CGPoint是否相等**

if (CGRectEqualToRect(rect1, rect2)) { // 两个区域相等

        // do some

    }

    if (CGPointEqualToPoint(point1, point2)) { // 两个点相等

        // do some

    }

    if (CGSizeEqualToSize(size1, size2)) { // 两个size相等

        // do some

    }

**5、比较两个NSDate相差多少小时**

 NSDate\* date1 = someDate;

 NSDate\* date2 = someOtherDate;

 NSTimeInterval distanceBetweenDates = [date1 timeIntervalSinceDate:date2];

 double secondsInAnHour = 3600;

// 除以3600是把秒化成小时，除以60得到结果为相差的分钟数

 NSInteger hoursBetweenDates = distanceBetweenDates / secondsInAnHour;

**6、每个cell之间增加间距**

// 方法一，每个分区只显示一行cell，分区头当作你想要的间距(注意，从数据源数组中取值的时候需要用indexPath.section而不是indexPath.row)

- (NSInteger)numberOfSectionsInTableView:(UITableView \*)tableView

{

    return yourArry.count;

}

- (NSInteger)tableView:(UITableView \*)tableView numberOfRowsInSection:(NSInteger)section

{

    return 1;

}

-(CGFloat)tableView:(UITableView \*)tableView heightForHeaderInSection:(NSInteger)section

{

    return cellSpacingHeight;

}

// 方法二，在cell的contentView上加个稍微低一点的view，cell上原本的内容放在你的view上，而不是contentView上，这样能伪造出一个间距来。

// 方法三，自定义cell，重写setFrame：方法

- (void)setFrame:(CGRect)frame

{

    frame.size.height -= 20;

    [super setFrame:frame];

}

**7、播放一张张连续的图片**

// 加入现在有三张图片分别为animate\_1、animate\_2、animate\_3

// 方法一

    imageView.animationImages = @[[UIImage imageNamed:@"animate\_1"], [UIImage imageNamed:@"animate\_2"], [UIImage imageNamed:@"animate\_3"]];

imageView.animationDuration = 1.0;

// 方法二

    imageView.image = [UIImage animatedImageNamed:@"animate\_" duration:1.0];

// 方法二解释下，这个方法会加载animate\_为前缀的，后边0-1024，也就是animate\_0、animate\_1一直到animate\_1024

**8、加载gif图片**

推荐使用这个框架 [FLAnimatedImage](https://github.com/Flipboard/FLAnimatedImage%5C%22+target=%5C%22_blank%5C%22+style=%5C%22margin:+0px;+padding:+0px;+color:+rgb%28235,+97,+0%29;+text-decoration:+none;)

**9、防止离屏渲染为image添加圆角**

// image分类

- (UIImage \*)circleImage

{

// NO代表透明

UIGraphicsBeginImageContextWithOptions(self.size, NO, 1);

// 获得上下文

CGContextRef ctx = UIGraphicsGetCurrentContext();

// 添加一个圆

CGRect rect = CGRectMake(0, 0, self.size.width, self.size.height);

// 方形变圆形

CGContextAddEllipseInRect(ctx, rect);

// 裁剪

CGContextClip(ctx);

// 将图片画上去

[self drawInRect:rect];

UIImage \*image = UIGraphicsGetImageFromCurrentImageContext();

UIGraphicsEndImageContext();

return image;

}

**10、查看系统所有字体**

// 打印字体

for (id familyName in [UIFont familyNames]) {

    NSLog(@"%@", familyName);

    for (id fontName in [UIFont fontNamesForFamilyName:familyName]) NSLog(@"  %@", fontName);

}

// 也可以进入这个网址查看 http://iosfonts.com/

**11、获取随机数**

NSInteger i = arc4random();

**12、获取随机数小数(0-1之间)**

#define ARC4RANDOM\_MAX      0x100000000

double val = ((double)arc4random() / ARC4RANDOM\_MAX);

**13、AVPlayer视频播放完成的通知监听**

[[NSNotificationCenter defaultCenter]

      addObserver:self

      selector:@selector(videoPlayEnd)

      name:AVPlayerItemDidPlayToEndTimeNotification

      object:nil];

**14、判断两个rect是否有交叉**

 if (CGRectIntersectsRect(rect1, rect2)) {

}

**15、判断一个字符串是否为数字**

NSCharacterSet \*notDigits = [[NSCharacterSet decimalDigitCharacterSet] invertedSet];

    if ([str rangeOfCharacterFromSet:notDigits].location == NSNotFound)

    {

      // 是数字

    } else

    {

      // 不是数字

    }

**16、将一个view保存为pdf格式**

- (void)createPDFfromUIView:(UIView\*)aView saveToDocumentsWithFileName:(NSString\*)aFilename

{

    NSMutableData \*pdfData = [NSMutableData data];

    UIGraphicsBeginPDFContextToData(pdfData, aView.bounds, nil);

    UIGraphicsBeginPDFPage();

    CGContextRef pdfContext = UIGraphicsGetCurrentContext();

    [aView.layer renderInContext:pdfContext];

    UIGraphicsEndPDFContext();

    NSArray\* documentDirectories = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask,YES);

    NSString\* documentDirectory = [documentDirectories objectAtIndex:0];

    NSString\* documentDirectoryFilename = [documentDirectory stringByAppendingPathComponent:aFilename];

    [pdfData writeToFile:documentDirectoryFilename atomically:YES];

    NSLog(@"documentDirectoryFileName: %@",documentDirectoryFilename);

}

**17、让一个view在父视图中心**

child.center = [parent convertPoint:parent.center fromView:parent.superview];

**18、获取当前导航控制器下前一个控制器**

- (UIViewController \*)backViewController

{

    NSInteger myIndex = [self.navigationController.viewControllers indexOfObject:self];

    if ( myIndex != 0 && myIndex != NSNotFound ) {

        return [self.navigationController.viewControllers objectAtIndex:myIndex-1];

    } else {

        return nil;

    }

}

**19、保存UIImage到本地**

NSArray \*paths = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask, YES);

NSString \*filePath = [[paths objectAtIndex:0] stringByAppendingPathComponent:@"Image.png"];

[UIImagePNGRepresentation(image) writeToFile:filePath atomically:YES];

**20、键盘上方增加工具栏**

UIToolbar \*keyboardDoneButtonView = [[UIToolbar alloc] init];

[keyboardDoneButtonView sizeToFit];

UIBarButtonItem \*doneButton = [[UIBarButtonItem alloc] initWithTitle:@"Done"

                                                               style:UIBarButtonItemStyleBordered target:self

                                                              action:@selector(doneClicked:)];

[keyboardDoneButtonView setItems:[NSArray arrayWithObjects:doneButton, nil]];

txtField.inputAccessoryView = keyboardDoneButtonView;

**21、copy一个view**

因为UIView没有实现copy协议，因此找不到copyWithZone方法，使用copy的时候导致崩溃

但是我们可以通过归档再解档实现copy，这相当于对视图进行了一次深拷贝，代码如下

id copyOfView =

[NSKeyedUnarchiver unarchiveObjectWithData:[NSKeyedArchiver archivedDataWithRootObject:originalView]];

**22、在image上绘制文字并生成新的image**

UIFont \*font = [UIFont boldSystemFontOfSize:12];

    UIGraphicsBeginImageContext(image.size);

    [image drawInRect:CGRectMake(0,0,image.size.width,image.size.height)];

    CGRect rect = CGRectMake(point.x, point.y, image.size.width, image.size.height);

    [[UIColor whiteColor] set];

    [text drawInRect:CGRectIntegral(rect) withFont:font];

    UIImage \*newImage = UIGraphicsGetImageFromCurrentImageContext();

    UIGraphicsEndImageContext();

**23、判断一个view是否为另一个view的子视图**

// 如果myView是self.view本身，也会返回yes

BOOL isSubView = [myView isDescendantOfView:self.view];

**24、判断一个字符串是否包含另一个字符串**

// 方法一、这种方法只适用于iOS8之后，如果是配iOS8之前用方法二

if ([str containsString:otherStr]) NSLog(@"包含");

// 方法二

NSRange range = [str rangeOfString:otherStr];

if (range.location != NSNotFound) NSLog(@"包含");

**25、UICollectionView自动滚动到某行**

// 重写viewDidLayoutSubviews方法

-(void)viewDidLayoutSubviews {

   [super viewDidLayoutSubviews];

   [self.collectionView scrollToItemAtIndexPath:indexPath atScrollPosition:UICollectionViewScrollPositionCenteredVertically animated:NO];

}

**26、修改系统UIAlertController**

// 但是据说这种方法会被App Store拒绝(慎用！)

UIAlertController \*alertVC = [UIAlertController alertControllerWithTitle:@"" message:@"" preferredStyle:UIAlertControllerStyleActionSheet];

    NSMutableAttributedString \*hogan = [[NSMutableAttributedString alloc] initWithString:@"我是一个大文本"];

    [hogan addAttribute:NSFontAttributeName

                  value:[UIFont systemFontOfSize:30]

                  range:NSMakeRange(4, 1)];

    [hogan addAttribute:NSForegroundColorAttributeName

                  value:[UIColor redColor]

                  range:NSMakeRange(4, 1)];

    [alertVC setValue:hogan forKey:@"attributedTitle"];

    UIAlertAction \*button = [UIAlertAction actionWithTitle:@"Label text" style:UIAlertActionStyleDefault handler:^(UIAlertAction \*action){ }];

    UIImage \*accessoryImage = [UIImage imageNamed:@"1"];

    [button setValue:accessoryImage forKey:@"image"];

    [alertVC addAction:button];

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

**27、判断某一行的cell是否已经显示**

CGRect cellRect = [tableView rectForRowAtIndexPath:indexPath];

BOOL completelyVisible = CGRectContainsRect(tableView.bounds, cellRect);

**28、让导航控制器pop回指定的控制器**

NSMutableArray \*allViewControllers = [NSMutableArray arrayWithArray:[self.navigationController viewControllers]];

for (UIViewController \*aViewController in allViewControllers) {

    if ([aViewController isKindOfClass:[RequiredViewController class]]) {

        [self.navigationController popToViewController:aViewController animated:NO];

    }

}

**29、动画修改label上的文字**

// 方法一

CATransition \*animation = [CATransition animation];

    animation.timingFunction = [CAMediaTimingFunction functionWithName:kCAMediaTimingFunctionEaseInEaseOut];

    animation.type = kCATransitionFade;

    animation.duration = 0.75;

    [self.label.layer addAnimation:animation forKey:@"kCATransitionFade"];

    self.label.text = @"New";

// 方法二

[UIView transitionWithView:self.label

                      duration:0.25f

                       options:UIViewAnimationOptionTransitionCrossDissolve

                    animations:^{

                        self.label.text = @"Well done!";

                    } completion:nil];

// 方法三

[UIView animateWithDuration:1.0

                     animations:^{

                         self.label.alpha = 0.0f;

                         self.label.text = @"newText";

                         self.label.alpha = 1.0f;

                     }];

**30、判断字典中是否包含某个key值**

if ([dic objectForKey:@"yourKey"]) {

    NSLog(@"有这个值");

} else {

    NSLog(@"没有这个值");

}

**31、获取屏幕方向**

UIInterfaceOrientation orientation = [UIApplication sharedApplication].statusBarOrientation;

if(orientation == 0) //Default orientation

    //默认

else if(orientation == UIInterfaceOrientationPortrait)

    //竖屏

else if(orientation == UIInterfaceOrientationLandscapeLeft)

    // 左横屏

else if(orientation == UIInterfaceOrientationLandscapeRight)

    //右横屏

**32、设置UIImage的透明度**

// 方法一、添加UIImage分类

- (UIImage \*)imageByApplyingAlpha:(CGFloat) alpha {

    UIGraphicsBeginImageContextWithOptions(self.size, NO, 0.0f);

    CGContextRef ctx = UIGraphicsGetCurrentContext();

    CGRect area = CGRectMake(0, 0, self.size.width, self.size.height);

    CGContextScaleCTM(ctx, 1, -1);

    CGContextTranslateCTM(ctx, 0, -area.size.height);

    CGContextSetBlendMode(ctx, kCGBlendModeMultiply);

    CGContextSetAlpha(ctx, alpha);

    CGContextDrawImage(ctx, area, self.CGImage);

    UIImage \*newImage = UIGraphicsGetImageFromCurrentImageContext();

    UIGraphicsEndImageContext();

    return newImage;

}

// 方法二、如果没有奇葩需求，干脆用UIImageView设置透明度

UIImageView \*imageView = [[UIImageView alloc] initWithImage:[UIImage imageWithName:@"yourImage"]];

imageView.alpha = 0.5;

**33、Attempt to mutate immutable object with insertString:atIndex:**

这个错是因为你拿字符串调用insertString:atIndex:方法的时候，调用对象不是NSMutableString，应该先转成这个类型再调用

**34、UIWebView添加单击手势不响应**

UITapGestureRecognizer \*tap = [[UITapGestureRecognizer alloc] initWithTarget:self action:@selector(webViewClick)];

        tap.delegate = self;

        [\_webView addGestureRecognizer:tap];

// 因为webView本身有一个单击手势，所以再添加会造成手势冲突，从而不响应。需要绑定手势代理，并实现下边的代理方法

- (BOOL)gestureRecognizer:(UIGestureRecognizer \*)gestureRecognizer shouldRecognizeSimultaneouslyWithGestureRecognizer:(UIGestureRecognizer \*)otherGestureRecognizer{

    return YES;

}

**35、获取手机RAM容量**

// 需要导入#import

mach\_port\_t host\_port;

    mach\_msg\_type\_number\_t host\_size;

    vm\_size\_t pagesize;

    host\_port = mach\_host\_self();

    host\_size = sizeof(vm\_statistics\_data\_t) / sizeof(integer\_t);

    host\_page\_size(host\_port, &pagesize);

    vm\_statistics\_data\_t vm\_stat;

    if (host\_statistics(host\_port, HOST\_VM\_INFO, (host\_info\_t)&vm\_stat, &host\_size) != KERN\_SUCCESS) {

        NSLog(@"Failed to fetch vm statistics");

    }

    /\* Stats in bytes \*/

    natural\_t mem\_used = (vm\_stat.active\_count +

                          vm\_stat.inactive\_count +

                          vm\_stat.wire\_count) \* pagesize;

    natural\_t mem\_free = vm\_stat.free\_count \* pagesize;

    natural\_t mem\_total = mem\_used + mem\_free;

    NSLog(@"已用: %u 可用: %u 总共: %u", mem\_used, mem\_free, mem\_total);

**36、地图上两个点之间的实际距离**

// 需要导入#import

CLLocation \*locA = [[CLLocation alloc] initWithLatitude:34 longitude:113];

    CLLocation \*locB = [[CLLocation alloc] initWithLatitude:31.05 longitude:121.76];

// CLLocationDistance求出的单位为米

    CLLocationDistance distance = [locA distanceFromLocation:locB];

**37、在应用中打开设置的某个界面**

// 打开设置->通用

[[UIApplication sharedApplication] openURL:[NSURL URLWithString:@"prefs:root=General"]];

// 以下是设置其他界面

prefs:root=General&path=About

prefs:root=General&path=ACCESSIBILITY

prefs:root=AIRPLANE\_MODE

prefs:root=General&path=AUTOLOCK

prefs:root=General&path=USAGE/CELLULAR\_USAGE

prefs:root=Brightness

prefs:root=Bluetooth

prefs:root=General&path=DATE\_AND\_TIME

prefs:root=FACETIME

prefs:root=General

prefs:root=General&path=Keyboard

prefs:root=CASTLE

prefs:root=CASTLE&path=STORAGE\_AND\_BACKUP

prefs:root=General&path=INTERNATIONAL

prefs:root=LOCATION\_SERVICES

prefs:root=ACCOUNT\_SETTINGS

prefs:root=MUSIC

prefs:root=MUSIC&path=EQ

prefs:root=MUSIC&path=VolumeLimit

prefs:root=General&path=Network

prefs:root=NIKE\_PLUS\_IPOD

prefs:root=NOTES

prefs:root=NOTIFICATIONS\_ID

prefs:root=Phone

prefs:root=Photos

prefs:root=General&path=ManagedConfigurationList

prefs:root=General&path=Reset

prefs:root=Sounds&path=Ringtone

prefs:root=Safari

prefs:root=General&path=Assistant

prefs:root=Sounds

prefs:root=General&path=SOFTWARE\_UPDATE\_LINK

prefs:root=STORE

prefs:root=TWITTER

prefs:root=FACEBOOK

prefs:root=General&path=USAGE prefs:root=VIDEO

prefs:root=General&path=Network/VPN

prefs:root=Wallpaper

prefs:root=WIFI

prefs:root=INTERNET\_TETHERING

prefs:root=Phone&path=Blocked

prefs:root=DO\_NOT\_DISTURB

**38、在UITextView中显示html文本**

    UITextView \*textView = [[UITextView alloc] initWithFrame:CGRectMake(20, 30, 100, 199)];

    textView.backgroundColor = [UIColor redColor];

    [self.view addSubview:textView];

    NSString \*htmlString = @"

![](http://blogs.babble.com/famecrawler/files/2010/11/mickey\_mouse-1097.jpg)";

    NSAttributedString \*attributedString = [[NSAttributedString alloc] initWithData: [htmlString dataUsingEncoding:NSUnicodeStringEncoding] options: @{ NSDocumentTypeDocumentAttribute: NSHTMLTextDocumentType } documentAttributes: nil error: nil];

    textView.attributedText = attributedString;

**39、监听scrollView是否滚动到了顶部／底部**

-(void)scrollViewDidScroll: (UIScrollView\*)scrollView

{

    float scrollViewHeight = scrollView.frame.size.height;

    float scrollContentSizeHeight = scrollView.contentSize.height;

    float scrollOffset = scrollView.contentOffset.y;

    if (scrollOffset == 0)

    {

        // 滚动到了顶部

    }

    else if (scrollOffset + scrollViewHeight == scrollContentSizeHeight)

    {

        // 滚动到了底部

    }

}

**40、UISlider增量／减量为固定值(假如为5)**

- (void)setupSlider

{

    UISlider \*slider = [[UISlider alloc] init];

    [self.view addSubview:slider];

    [slider addTarget:self action:@selector(sliderAction:) forControlEvents:UIControlEventValueChanged];

    slider.maximumValue = 100;

    slider.minimumValue = 0;

    slider.frame = CGRectMake(200, 20, 100, 30);

}

- (void)sliderAction:(UISlider \*)slider

{

    [slider setValue:((int)((slider.value + 2.5) / 5) \* 5) animated:NO];

}

**41、选中textField或者textView所有文本(我这里以textView为例)**

[self.textView setSelectedTextRange:[self.textView textRangeFromPosition:self.textView.beginningOfDocument toPosition:self.textView.endOfDocument]]

**42、从导航控制器中删除某个控制器**

// 方法一、知道这个控制器所处的导航控制器下标

NSMutableArray \*navigationArray = [[NSMutableArray alloc] initWithArray: self.navigationController.viewControllers];

[navigationArray removeObjectAtIndex: 2];

self.navigationController.viewControllers = navigationArray;

// 方法二、知道具体是哪个控制器

NSArray\* tempVCA = [self.navigationController viewControllers];

for(UIViewController \*tempVC in tempVCA)

{

    if([tempVC isKindOfClass:[urViewControllerClass class]])

    {

        [tempVC removeFromParentViewController];

    }

}

**43、隐藏UITextView/UITextField光标**

textField.tintColor = [UIColor clearColor];

**44、当UITextView/UITextField中没有文字时，禁用回车键**

textField.enablesReturnKeyAutomatically = YES;

**45、字符串encode编码(编码url字符串不成功的问题)**

// 我们一般用这个方法处理stringByAddingPercentEscapesUsingEncoding但是这个方法好想不会处理／和&这种特殊符号，这种情况就需要用下边这个方法处理

@implementation NSString (NSString\_Extended)

- (NSString \*)urlencode {

    NSMutableString \*output = [NSMutableString string];

    const unsigned char \*source = (const unsigned char \*)[self UTF8String];

    int sourceLen = strlen((const char \*)source);

    for (int i = 0; i < sourceLen; ++i) {

        const unsigned char thisChar = source[i];

        if (thisChar == ' '){

            [output appendString:@"+"];

        } else if (thisChar == '.' || thisChar == '-' || thisChar == '\_' || thisChar == '~' ||

                   (thisChar >= 'a' && thisChar = 'A' && thisChar = '0' && thisChar <= '9')) {

            [output appendFormat:@"%c", thisChar];

        } else {

            [output appendFormat:@"%%X", thisChar];

        }

    }

    return output;

}

**46、计算UILabel上某段文字的frame**

@implementation UILabel (TextRect)

- (CGRect)boundingRectForCharacterRange:(NSRange)range

{

    NSTextStorage \*textStorage = [[NSTextStorage alloc] initWithAttributedString:[self attributedText]];

    NSLayoutManager \*layoutManager = [[NSLayoutManager alloc] init];

    [textStorage addLayoutManager:layoutManager];

    NSTextContainer \*textContainer = [[NSTextContainer alloc] initWithSize:[self bounds].size];

    textContainer.lineFragmentPadding = 0;

    [layoutManager addTextContainer:textContainer];

    NSRange glyphRange;

    [layoutManager characterRangeForGlyphRange:range actualGlyphRange:&glyphRange];

    return [layoutManager boundingRectForGlyphRange:glyphRange inTextContainer:textContainer];

}

**47、获取随机UUID**

NSString \*result;

    if([[[UIDevice currentDevice] systemVersion] floatValue] > 6.0)

    {

       result = [[NSUUID UUID] UUIDString];

    }

    else

    {

        CFUUIDRef uuidRef = CFUUIDCreate(NULL);

        CFStringRef uuid = CFUUIDCreateString(NULL, uuidRef);

        CFRelease(uuidRef);

        result = (\_\_bridge\_transfer NSString \*)uuid;

    }

**48、仿苹果抖动动画**

#define RADIANS(degrees) (((degrees) \* M\_PI) / 180.0)

- (void)startAnimate {

    view.transform = CGAffineTransformRotate(CGAffineTransformIdentity, RADIANS(-5));

    [UIView animateWithDuration:0.25 delay:0.0 options:(UIViewAnimationOptionAllowUserInteraction | UIViewAnimationOptionRepeat | UIViewAnimationOptionAutoreverse) animations:^ {

                         view.transform = CGAffineTransformRotate(CGAffineTransformIdentity, RADIANS(5));

                     } completion:nil];

}

- (void)stopAnimate {

    [UIView animateWithDuration:0.25 delay:0.0 options:(UIViewAnimationOptionAllowUserInteraction | UIViewAnimationOptionBeginFromCurrentState | UIViewAnimationOptionCurveLinear) animations:^ {

                         view.transform = CGAffineTransformIdentity;

                     } completion:nil];

}

**49、修改UISearBar内部背景颜色**

UITextField \*textField = [\_searchBar valueForKey:@"\_searchField"];

textField.backgroundColor = [UIColor redColor];

**50、UITextView滚动到顶部**

    // 方法一

    [self.textView scrollRangeToVisible:NSMakeRange(0, 0)];

    // 方法二

    [self.textView setContentOffset:CGPointZero animated:YES];

**51、通知监听APP生命周期**

UIApplicationDidEnterBackgroundNotification 应用程序进入后台

UIApplicationWillEnterForegroundNotification 应用程序将要进入前台

UIApplicationDidFinishLaunchingNotification 应用程序完成启动

UIApplicationDidFinishLaunchingNotification 应用程序由挂起变的活跃

UIApplicationWillResignActiveNotification 应用程序挂起(有电话进来或者锁屏)

UIApplicationDidReceiveMemoryWarningNotification 应用程序收到内存警告

UIApplicationDidReceiveMemoryWarningNotification 应用程序终止(后台杀死、手机关机等)

UIApplicationSignificantTimeChangeNotification 当有重大时间改变(凌晨0点，设备时间被修改，时区改变等)

UIApplicationWillChangeStatusBarOrientationNotification 设备方向将要改变

UIApplicationDidChangeStatusBarOrientationNotification 设备方向改变

UIApplicationWillChangeStatusBarFrameNotification 设备状态栏frame将要改变

UIApplicationDidChangeStatusBarFrameNotification 设备状态栏frame改变

UIApplicationBackgroundRefreshStatusDidChangeNotification 应用程序在后台下载内容的状态发生变化

UIApplicationProtectedDataWillBecomeUnavailable 本地受保护的文件被锁定,无法访问

UIApplicationProtectedDataWillBecomeUnavailable 本地受保护的文件可用了

**52、触摸事件类型**

UIControlEventTouchCancel 取消控件当前触发的事件

UIControlEventTouchDown 点按下去的事件

UIControlEventTouchDownRepeat 重复的触动事件

UIControlEventTouchDragEnter 手指被拖动到控件的边界的事件

UIControlEventTouchDragExit 一个手指从控件内拖到外界的事件

UIControlEventTouchDragInside 手指在控件的边界内拖动的事件

UIControlEventTouchDragOutside 手指在控件边界之外被拖动的事件

UIControlEventTouchUpInside 手指处于控制范围内的触摸事件

UIControlEventTouchUpOutside 手指超出控制范围的控制中的触摸事件

**53、UITextField文字周围增加边距**

    // 子类化UITextField，增加insert属性

@interface WZBTextField : UITextField

@property (nonatomic, assign) UIEdgeInsets insets;

@end

// 在.m文件重写下列方法

- (CGRect)textRectForBounds:(CGRect)bounds {

    CGRect paddedRect = UIEdgeInsetsInsetRect(bounds, self.insets);

    if (self.rightViewMode == UITextFieldViewModeAlways || self.rightViewMode == UITextFieldViewModeUnlessEditing) {

        return [self adjustRectWithWidthRightView:paddedRect];

    }

    return paddedRect;

}

- (CGRect)placeholderRectForBounds:(CGRect)bounds {

    CGRect paddedRect = UIEdgeInsetsInsetRect(bounds, self.insets);

    if (self.rightViewMode == UITextFieldViewModeAlways || self.rightViewMode == UITextFieldViewModeUnlessEditing) {

        return [self adjustRectWithWidthRightView:paddedRect];

    }

    return paddedRect;

}

- (CGRect)editingRectForBounds:(CGRect)bounds {

    CGRect paddedRect = UIEdgeInsetsInsetRect(bounds, self.insets);

    if (self.rightViewMode == UITextFieldViewModeAlways || self.rightViewMode == UITextFieldViewModeWhileEditing) {

        return [self adjustRectWithWidthRightView:paddedRect];

    }

    return paddedRect;

}

- (CGRect)adjustRectWithWidthRightView:(CGRect)bounds {

    CGRect paddedRect = bounds;

    paddedRect.size.width -= CGRectGetWidth(self.rightView.frame);

    return paddedRect;

}

**54、监听UISlider拖动状态**

// 添加事件

[slider addTarget:self action:@selector(sliderValurChanged:forEvent:) forControlEvents:UIControlEventValueChanged];

// 实现方法

- (void)sliderValurChanged:(UISlider\*)slider forEvent:(UIEvent\*)event {

    UITouch \*touchEvent = [[event allTouches] anyObject];

    switch (touchEvent.phase) {

        case UITouchPhaseBegan:

            NSLog(@"开始拖动");

            break;

        case UITouchPhaseMoved:

            NSLog(@"正在拖动");

            break;

        case UITouchPhaseEnded:

            NSLog(@"结束拖动");

            break;

        default:

            break;

    }

}

**55、设置UITextField光标位置**

// textField需要设置的textField，index要设置的光标位置

- (void)cursorLocation:(UITextField \*)textField index:(NSInteger)index

{

    NSRange range = NSMakeRange(index, 0);

    UITextPosition \*start = [textField positionFromPosition:[textField beginningOfDocument] offset:range.location];

    UITextPosition \*end = [textField positionFromPosition:start offset:range.length];

    [textField setSelectedTextRange:[textField textRangeFromPosition:start toPosition:end]];

}

**56、去除webView底部黑色**

    [webView setBackgroundColor:[UIColor clearColor]];

    [webView setOpaque:NO];

    for (UIView \*v1 in [webView subviews])

    {

        if ([v1 isKindOfClass:[UIScrollView class]])

        {

            for (UIView \*v2 in v1.subviews)

            {

                if ([v2 isKindOfClass:[UIImageView class]])

                {

                    v2.hidden = YES;

                }

            }

        }

    }

**57、获取collectionViewCell在屏幕中的frame**

UICollectionViewLayoutAttributes \*attributes = [collectionView layoutAttributesForItemAtIndexPath:indexPath];

CGRect cellRect = attributes.frame;

CGRect cellFrameInSuperview = [collectionView convertRect:cellRect toView:[cv superview]];

**58、比较两个UIImage是否相等**

- (BOOL)image:(UIImage \*)image1 isEqualTo:(UIImage \*)image2

{

    NSData \*data1 = UIImagePNGRepresentation(image1);

    NSData \*data2 = UIImagePNGRepresentation(image2);

    return [data1 isEqual:data2];

}

**59、解决当UIScrollView上有UIButton的时候，触摸到button滑动不了的问题**

// 子类化UIScrollView，并重写以下方法

- (instancetype)initWithFrame:(CGRect)frame {

    if (self = [super initWithFrame:frame]) {

        self.delaysContentTouches = NO;

    }

    return self;

}

- (BOOL)touchesShouldCancelInContentView:(UIView \*)view {

    if ([view isKindOfClass:UIButton.class]) {

        return YES;

    }

    return [super touchesShouldCancelInContentView:view];

}

**60、UITextView中的文字添加阴影效果**

- (void)setTextLayer:(UITextView \*)textView color:(UIColor \*)color

{

    CALayer \*textLayer = ((CALayer \*)[textView.layer.sublayers objectAtIndex:0]);

    textLayer.shadowColor = color.CGColor;

    textLayer.shadowOffset = CGSizeMake(0.0f, 1.0f);

    textLayer.shadowOpacity = 1.0f;

    textLayer.shadowRadius = 1.0f;

}

**61、MD5加密**

+ (NSString \*)md5:(NSString \*)str

{

    const char \*concat\_str = [str UTF8String];

    unsigned char result[CC\_MD5\_DIGEST\_LENGTH];

    CC\_MD5(concat\_str, (unsigned int)strlen(concat\_str), result);

    NSMutableString \*hash = [NSMutableString string];

    for (int i =0; i<16; i++){

        [hash appendFormat:@"X", result[i]];

    }

    return [hash uppercaseString];

}

**62、base64加密**

@interface NSData (Base64)

/\*\*

 \*  @brief  字符串base64后转data

 \*/

+ (NSData \*)dataWithBase64EncodedString:(NSString \*)string

{

    if (![string length]) return nil;

    NSData \*decoded = nil;

#if \_\_MAC\_OS\_X\_VERSION\_MIN\_REQUIRED < \_\_MAC\_10\_9 || \_\_IPHONE\_OS\_VERSION\_MIN\_REQUIRED < \_\_IPHONE\_7\_0

    if (![NSData instancesRespondToSelector:@selector(initWithBase64EncodedString:options:)])

    {

#pragma clang diagnostic push

#pragma clang diagnostic ignored "-Wdeprecated-declarations"

        decoded = [[self alloc] initWithBase64Encoding:[string stringByReplacingOccurrencesOfString:@"[^A-Za-z0-9+/=]" withString:@"" options:NSRegularExpressionSearch range:NSMakeRange(0, [string length])]];

#pragma clang diagnostic pop

    }

    else

#endif

    {

        decoded = [[self alloc] initWithBase64EncodedString:string options:NSDataBase64DecodingIgnoreUnknownCharacters];

    }

    return [decoded length]? decoded: nil;

}

/\*\*

 \*  @brief  NSData转string

 \*  @param wrapWidth 换行长度  76  64

 \*/

- (NSString \*)base64EncodedStringWithWrapWidth:(NSUInteger)wrapWidth

{

    if (![self length]) return nil;

    NSString \*encoded = nil;

#if \_\_MAC\_OS\_X\_VERSION\_MIN\_REQUIRED < \_\_MAC\_10\_9 || \_\_IPHONE\_OS\_VERSION\_MIN\_REQUIRED < \_\_IPHONE\_7\_0

    if (![NSData instancesRespondToSelector:@selector(base64EncodedStringWithOptions:)])

    {

#pragma clang diagnostic push

#pragma clang diagnostic ignored "-Wdeprecated-declarations"

        encoded = [self base64Encoding];

#pragma clang diagnostic pop

    }

    else

#endif

    {

        switch (wrapWidth)

        {

            case 64:

            {

                return [self base64EncodedStringWithOptions:NSDataBase64Encoding64CharacterLineLength];

            }

            case 76:

            {

                return [self base64EncodedStringWithOptions:NSDataBase64Encoding76CharacterLineLength];

            }

            default:

            {

                encoded = [self base64EncodedStringWithOptions:(NSDataBase64EncodingOptions)0];

            }

        }

    }

    if (!wrapWidth || wrapWidth >= [encoded length])

    {

        return encoded;

    }

    wrapWidth = (wrapWidth / 4) \* 4;

    NSMutableString \*result = [NSMutableString string];

    for (NSUInteger i = 0; i < [encoded length]; i+= wrapWidth)

    {

        if (i + wrapWidth >= [encoded length])

        {

            [result appendString:[encoded substringFromIndex:i]];

            break;

        }

        [result appendString:[encoded substringWithRange:NSMakeRange(i, wrapWidth)]];

        [result appendString:@"\r\n"];

    }

    return result;

}

/\*\*

 \*  @brief  NSData转string 换行长度默认64

 \*/

- (NSString \*)base64EncodedString

{

    return [self base64EncodedStringWithWrapWidth:0];

}

**63、AES加密**

#import

@interface NSData (AES)

/\*\*

 \*  利用AES加密数据

 \*/

- (NSData\*)encryptedWithAESUsingKey:(NSString\*)key andIV:(NSData\*)iv {

    NSData \*keyData = [key dataUsingEncoding:NSUTF8StringEncoding];

    size\_t dataMoved;

    NSMutableData \*encryptedData = [NSMutableData dataWithLength:self.length + kCCBlockSizeAES128];

    CCCryptorStatus status = CCCrypt(kCCEncrypt,kCCAlgorithmAES128,kCCOptionPKCS7Padding,keyData.bytes,keyData.length,iv.bytes,self.bytes,self.length,encryptedData.mutableBytes, encryptedData.length,&dataMoved);

    if (status == kCCSuccess) {

        encryptedData.length = dataMoved;

        return encryptedData;

    }

    return nil;

}

/\*\*

 \*  @brief  利用AES解密据

 \*/

- (NSData\*)decryptedWithAESUsingKey:(NSString\*)key andIV:(NSData\*)iv {

    NSData \*keyData = [key dataUsingEncoding:NSUTF8StringEncoding];

    size\_t dataMoved;

    NSMutableData \*decryptedData = [NSMutableData dataWithLength:self.length + kCCBlockSizeAES128];

    CCCryptorStatus result = CCCrypt(kCCDecrypt,kCCAlgorithmAES128,kCCOptionPKCS7Padding,keyData.bytes,keyData.length,iv.bytes,self.bytes,self.length,decryptedData.mutableBytes, decryptedData.length,&dataMoved);

    if (result == kCCSuccess) {

        decryptedData.length = dataMoved;

        return decryptedData;

    }

    return nil;

}

**64、3DES加密**

#import

@interface NSData (3DES)

/\*\*

 \*  利用3DES加密数据

 \*/

- (NSData\*)encryptedWith3DESUsingKey:(NSString\*)key andIV:(NSData\*)iv {

    NSData \*keyData = [key dataUsingEncoding:NSUTF8StringEncoding];

    size\_t dataMoved;

    NSMutableData \*encryptedData = [NSMutableData dataWithLength:self.length + kCCBlockSize3DES];

    CCCryptorStatus result = CCCrypt(kCCEncrypt,kCCAlgorithm3DES,kCCOptionPKCS7Padding,keyData.bytes,keyData.length,iv.bytes,self.bytes,self.length,encryptedData.mutableBytes,encryptedData.length,&dataMoved);

    if (result == kCCSuccess) {

        encryptedData.length = dataMoved;

        return encryptedData;

    }

    return nil;

}

/\*\*

 \*  @brief   利用3DES解密数据

 \*/

- (NSData\*)decryptedWith3DESUsingKey:(NSString\*)key andIV:(NSData\*)iv {

    NSData \*keyData = [key dataUsingEncoding:NSUTF8StringEncoding];

    size\_t dataMoved;

    NSMutableData \*decryptedData = [NSMutableData dataWithLength:self.length + kCCBlockSize3DES];

    CCCryptorStatus result = CCCrypt(kCCDecrypt,kCCAlgorithm3DES,kCCOptionPKCS7Padding,keyData.bytes,keyData.length,iv.bytes,self.bytes,self.length,decryptedData.mutableBytes,decryptedData.length,&dataMoved);

    if (result == kCCSuccess) {

        decryptedData.length = dataMoved;

        return decryptedData;

    }

    return nil;

}

**65、单个页面多个网络请求的情况，需要监听所有网络请求结束后刷新UI**

dispatch\_group\_t group = dispatch\_group\_create();

    dispatch\_queue\_t serialQueue = dispatch\_queue\_create("com.wzb.test.www", DISPATCH\_QUEUE\_SERIAL);

    dispatch\_group\_enter(group);

    dispatch\_group\_async(group, serialQueue, ^{

        // 网络请求一

        [WebClick getDataSuccess:^(ResponseModel \*model) {

            dispatch\_group\_leave(group);

        } failure:^(NSString \*err) {

            dispatch\_group\_leave(group);

        }];

    });

    dispatch\_group\_enter(group);

    dispatch\_group\_async(group, serialQueue, ^{

        // 网络请求二

        [WebClick getDataSuccess:getBigTypeRM onSuccess:^(ResponseModel \*model) {

            dispatch\_group\_leave(group);

        }                                  failure:^(NSString \*errorString) {

            dispatch\_group\_leave(group);

        }];

    });

    dispatch\_group\_enter(group);

    dispatch\_group\_async(group, serialQueue, ^{

        // 网络请求三

        [WebClick getDataSuccess:^{

            dispatch\_group\_leave(group);

        } failure:^(NSString \*errorString) {

            dispatch\_group\_leave(group);

        }];

    });

    // 所有网络请求结束后会来到这个方法

    dispatch\_group\_notify(group, serialQueue, ^{

        dispatch\_async(dispatch\_get\_global\_queue(0, 0), ^{

            dispatch\_async(dispatch\_get\_main\_queue(), ^{

                // 刷新UI

            });

        });

    });

**66、解决openUrl延时问题**

// 方法一

dispatch\_async(dispatch\_get\_main\_queue(), ^{

    UIApplication \*application = [UIApplication sharedApplication];

    if ([application respondsToSelector:@selector(openURL:options:completionHandler:)]) {

        [application openURL:URL options:@{}

           completionHandler:nil];

    } else {

        [application openURL:URL];

    }

    });

// 方法二

[self performSelector:@selector(redirectToURL:) withObject:url afterDelay:0.1];

- (void) redirectToURL

{

UIApplication \*application = [UIApplication sharedApplication];

    if ([application respondsToSelector:@selector(openURL:options:completionHandler:)]) {

        [application openURL:URL options:@{}

           completionHandler:nil];

    } else {

        [application openURL:URL];

    }

}

**67、页面跳转实现翻转动画**

// modal方式

    TestViewController \*vc = [[TestViewController alloc] init];

    vc.view.backgroundColor = [UIColor redColor];

    vc.modalTransitionStyle = UIModalTransitionStyleCoverVertical;

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

// push方式

    TestViewController \*vc = [[TestViewController alloc] init];

    vc.view.backgroundColor = [UIColor redColor];

    [UIView beginAnimations:@"View Flip" context:nil];

    [UIView setAnimationDuration:0.80];

    [UIView setAnimationCurve:UIViewAnimationCurveEaseInOut];

    [UIView setAnimationTransition:UIViewAnimationTransitionFlipFromRight forView:self.navigationController.view cache:NO];

    [self.navigationController pushViewController:vc animated:YES];

    [UIView commitAnimations];

**68、tableView实现无限滚动**

- (void)scrollViewDidScroll:(UIScrollView \*)scrollView

{

    CGFloat actualPosition = scrollView.contentOffset.y;

    CGFloat contentHeight = scrollView.contentSize.height - scrollView.frame.size.height;

    if (actualPosition >= contentHeight) {

        [self.dataArr addObjectsFromArray:self.dataArr];

        [self.tableView reloadData];

    }

}

**69、代码方式调整屏幕亮度**

// brightness属性值在0-1之间，0代表最小亮度，1代表最大亮度

[[UIScreen mainScreen] setBrightness:0.5];

**70、获取当前应用CUP用量**

float cpu\_usage()

{

    kern\_return\_t kr;

    task\_info\_data\_t tinfo;

    mach\_msg\_type\_number\_t task\_info\_count;

    task\_info\_count = TASK\_INFO\_MAX;

    kr = task\_info(mach\_task\_self(), TASK\_BASIC\_INFO, (task\_info\_t)tinfo, &task\_info\_count);

    if (kr != KERN\_SUCCESS) {

        return -1;

    }

    task\_basic\_info\_t      basic\_info;

    thread\_array\_t         thread\_list;

    mach\_msg\_type\_number\_t thread\_count;

    thread\_info\_data\_t     thinfo;

    mach\_msg\_type\_number\_t thread\_info\_count;

    thread\_basic\_info\_t basic\_info\_th;

    uint32\_t stat\_thread = 0; // Mach threads

    basic\_info = (task\_basic\_info\_t)tinfo;

    // get threads in the task

    kr = task\_threads(mach\_task\_self(), &thread\_list, &thread\_count);

    if (kr != KERN\_SUCCESS) {

        return -1;

    }

    if (thread\_count > 0)

        stat\_thread += thread\_count;

    long tot\_sec = 0;

    long tot\_usec = 0;

    float tot\_cpu = 0;

    int j;

    for (j = 0; j < (int)thread\_count; j++)

    {

        thread\_info\_count = THREAD\_INFO\_MAX;

        kr = thread\_info(thread\_list[j], THREAD\_BASIC\_INFO,

                         (thread\_info\_t)thinfo, &thread\_info\_count);

        if (kr != KERN\_SUCCESS) {

            return -1;

        }

        basic\_info\_th = (thread\_basic\_info\_t)thinfo;

        if (!(basic\_info\_th->flags & TH\_FLAGS\_IDLE)) {

            tot\_sec = tot\_sec + basic\_info\_th->user\_time.seconds + basic\_info\_th->system\_time.seconds;

            tot\_usec = tot\_usec + basic\_info\_th->user\_time.microseconds + basic\_info\_th->system\_time.microseconds;

            tot\_cpu = tot\_cpu + basic\_info\_th->cpu\_usage / (float)TH\_USAGE\_SCALE \* 100.0;

        }

    } // for each thread

    kr = vm\_deallocate(mach\_task\_self(), (vm\_offset\_t)thread\_list, thread\_count \* sizeof(thread\_t));

    assert(kr == KERN\_SUCCESS);

    return tot\_cpu;

}

**71、float数据取整四舍五入**

    CGFloat f = 4.65;

    NSLog(@"%d", (int)f);    // 打印结果4

    CGFloat f = 4.65;

    NSLog(@"%d", (int)round(f));    // 打印结果5

**72、删除UISearchBar系统默认边框**

    // 方法一

    searchBar.searchBarStyle = UISearchBarStyleMinimal;

    // 方法二

    [searchBar setBackgroundImage:[[UIImage alloc]init]];

    // 方法三

    searchBar.barTintColor = [UIColor whiteColor];

**73、为UICollectionViewCell设置圆角和阴影**

cell.contentView.layer.cornerRadius = 2.0f;

cell.contentView.layer.borderWidth = 1.0f;

cell.contentView.layer.borderColor = [UIColor clearColor].CGColor;

cell.contentView.layer.masksToBounds = YES;

cell.layer.shadowColor = [UIColor lightGrayColor].CGColor;

cell.layer.shadowOffset = CGSizeMake(0, 2.0f);

cell.layer.shadowRadius = 2.0f;

cell.layer.shadowOpacity = 1.0f;

cell.layer.masksToBounds = NO;

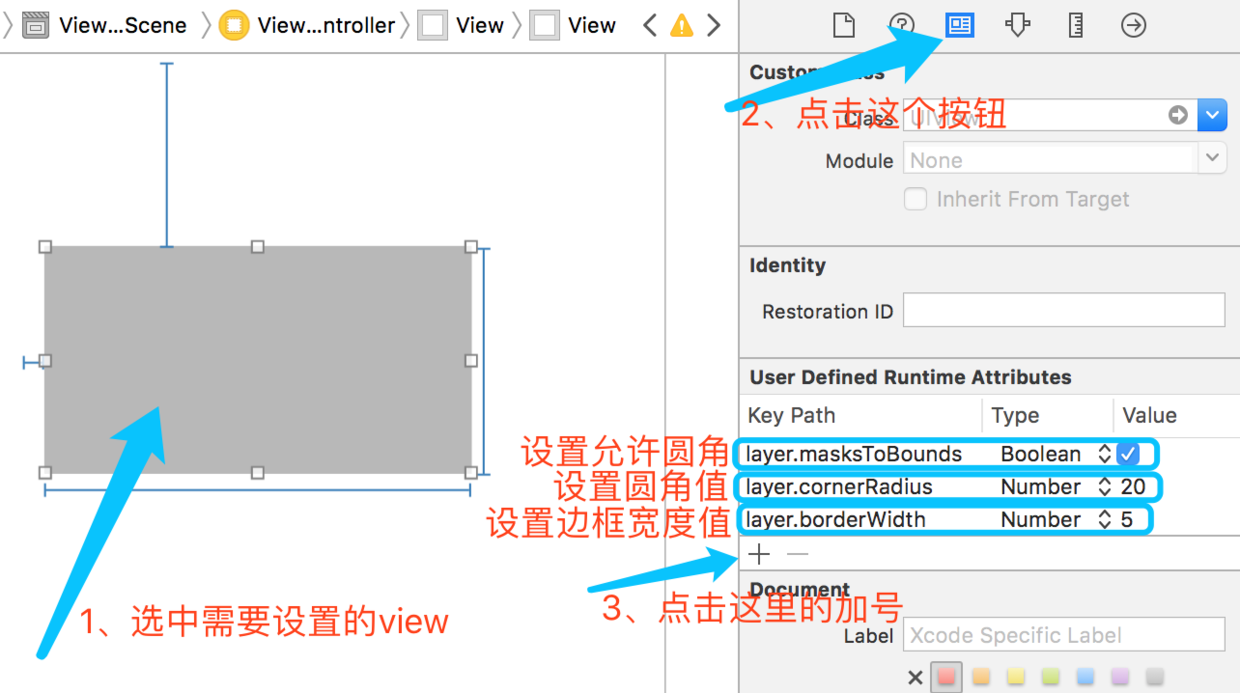
cell.layer.shadowPath = [UIBezierPath bezierPathWithRoundedRect:cell.bounds cornerRadius:cell.contentView.layer.cornerRadius].CGPath;

**74、让正在滑动的scrollView停止滚动(不是禁止，而是暂时停止滚动)**

[scrollView setContentOffset:scrollView.contentOffset animated:NO];

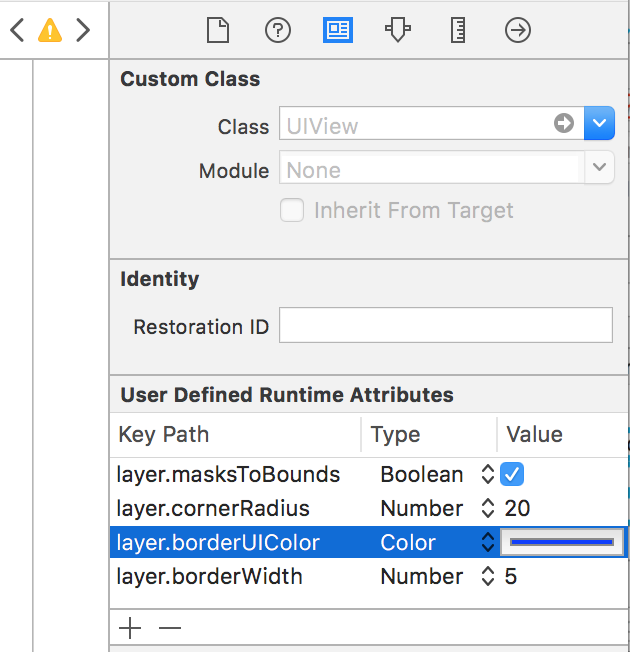
**75、使用xib设置UIView的边框、圆角**

圆角和边框看下图即可设置



xib设置圆角边框.png

但是增加layer.borderColor的keyPath设置边框颜色并不能起作用，后来查了资料，这里应该用layer.borderUIColor，但是这里设置的颜色不起作用，无论设置什么颜色显示出来的都是黑色的。后来又去查了下，有种解决方案是给CALayer添加一个分类，提供一个 - (void)setBorderUIColor:(UIColor \*)color;方法就可以解决了，实现如下：



xib设置边框颜色.png

#import "CALayer+BorderColor.h"

@implementation CALayer (BorderColor)

- (void)setBorderUIColor:(UIColor \*)color

{

    self.borderColor = color.CGColor;

}

**76、根据经纬度获取城市等信息**

// 创建经纬度

    CLLocation \*location = [[CLLocation alloc] initWithLatitude:latitude longitude:longitude];

    //创建一个译码器

    CLGeocoder \*cLGeocoder = [[CLGeocoder alloc] init];

    [cLGeocoder reverseGeocodeLocation:userLocation completionHandler:^(NSArray \*placemarks, NSError \*error) {

        CLPlacemark \*place = [placemarks objectAtIndex:0];

        // 位置名

    　　NSLog(@"name,%@",place.name);

    　　// 街道

    　　NSLog(@"thoroughfare,%@",place.thoroughfare);

    　　// 子街道

    　　NSLog(@"subThoroughfare,%@",place.subThoroughfare);

    　　// 市

    　　NSLog(@"locality,%@",place.locality);

    　　// 区

    　　NSLog(@"subLocality,%@",place.subLocality);

    　　// 国家

    　　NSLog(@"country,%@",place.country);

        }

    }];

/\*  CLPlacemark中属性含义

name                    地名

thoroughfare            街道

subThoroughfare        街道相关信息，例如门牌等

locality                城市

subLocality            城市相关信息，例如标志性建筑

administrativeArea      直辖市

subAdministrativeArea  其他行政区域信息（自治区等）

postalCode              邮编

ISOcountryCode          国家编码

country                国家

inlandWater            水源，湖泊

ocean                  海洋

areasOfInterest        关联的或利益相关的地标

\*/

**77、如何防止添加多个NSNotification观察者？**

// 解决方案就是添加观察者之前先移除下这个观察者

[[NSNotificationCenter defaultCenter] removeObserver:observer name:name object:object];

        [[NSNotificationCenter defaultCenter] addObserver:observer selector:selector name:name object:object];

**78、将一个xib添加到另外一个xib上**

// 假设你的自定义view名字为CustomView，你需要在CustomView.m中重写 `- (instancetype)initWithCoder:(NSCoder \*)aDecoder` 方法，代码如下：

- (instancetype)initWithCoder:(NSCoder \*)aDecoder {

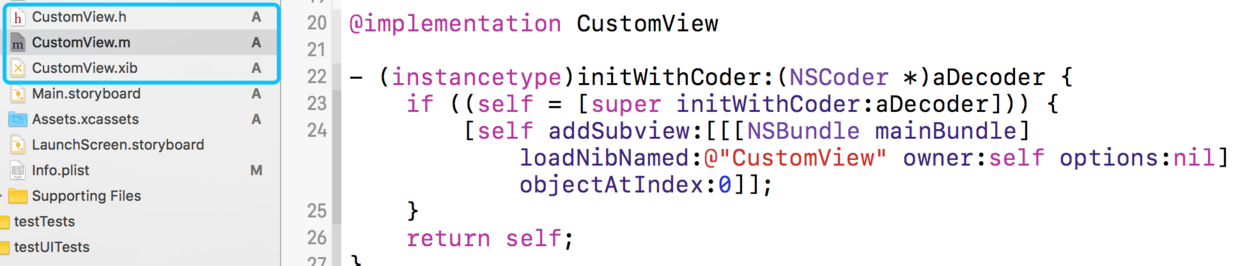
    if ((self = [super initWithCoder:aDecoder])) {

        [self addSubview:[[[NSBundle mainBundle] loadNibNamed:@"CustomView" owner:self options:nil] objectAtIndex:0]];

    }

    return self;

}



将一个xib添加到另外一个xib上.png

**79、处理字符串，使其首字母大写**

    NSString \*str = @"abcdefghijklmn";

    NSString \*resultStr;

    if (str && str.length > 0) {

        resultStr = [str stringByReplacingCharactersInRange:NSMakeRange(0,1) withString:[[str substringToIndex:1] capitalizedString]];

    }

    NSLog(@"%@", resultStr);

**80、判断一个UIAlertView/UIAlertController是否显示**

// UIAlertView自带属性

if (alert.visible)

{

      NSLog(@"显示了");

} else {

      NSLog(@"未显示");

}

// UIAlertController没有visible属性，需要自己判断，添加一个全局变量 BOOL visible

UIAlertController \*alertController = [UIAlertController alertControllerWithTitle:@"Title" message:@"message" preferredStyle:UIAlertControllerStyleAlert];

    UIAlertAction \*alertAction = [UIAlertAction actionWithTitle:@"ActionTitle" style:UIAlertActionStyleDefault handler:^(UIAlertAction \* \_Nonnull action) {

        self.visible = NO;

    }];

    UIAlertAction \*calcelAction = [UIAlertAction actionWithTitle:@"calcelTitle" style:UIAlertActionStyleCancel handler:^(UIAlertAction \* \_Nonnull action) {

        self.visible = NO;

    }];

    [alertController addAction:alertAction];

    [alertController addAction:calcelAction];

    [self presentViewController:alertController animated:YES completion:^{

        self.visible = YES;

    }];

**81、获取字符串中的数字**

- (NSString \*)getNumberFromStr:(NSString \*)str

{

    NSCharacterSet \*nonDigitCharacterSet = [[NSCharacterSet decimalDigitCharacterSet] invertedSet];

    return [[str componentsSeparatedByCharactersInSet:nonDigitCharacterSet] componentsJoinedByString:@""];

}

6

    NSLog(@"%@", [self getNumberFromStr:@"a0b0c1d2e3f4fda8fa8fad9fsad23"]); // 00123488923

**82、为UIView的某个方向添加边框**

// 添加UIView分类

// UIView+WZB.h

#import

/\*\*

 边框方向

 - WZBBorderDirectionTop: 顶部

 - WZBBorderDirectionLeft: 左边

 - WZBBorderDirectionBottom: 底部

 - WZBBorderDirectionRight: 右边

 \*/

typedef NS\_ENUM(NSInteger, WZBBorderDirectionType) {

    WZBBorderDirectionTop = 0,

    WZBBorderDirectionLeft,

    WZBBorderDirectionBottom,

    WZBBorderDirectionRight

};

@interface UIView (WZB)

/\*\*

 为UIView的某个方向添加边框

 @param direction 边框方向

 @param color 边框颜色

 @param width 边框宽度

 \*/

- (void)wzb\_addBorder:(WZBBorderDirectionType)direction color:(UIColor \*)color width:(CGFloat)width;

@end

// UIView+WZB.m

#import "UIView+WZB.h"

@implementation UIView (WZB)

- (void)wzb\_addBorder:(WZBBorderDirectionType)direction color:(UIColor \*)color width:(CGFloat)width

{

    CALayer \*border = [CALayer layer];

    border.backgroundColor = color.CGColor;

    switch (direction) {

        case WZBBorderDirectionTop:

        {

            border.frame = CGRectMake(0.0f, 0.0f, self.bounds.size.width, width);

        }

            break;

        case WZBBorderDirectionLeft:

        {

            border.frame = CGRectMake(0.0f, 0.0f, width, self.bounds.size.height);

        }

            break;

        case WZBBorderDirectionBottom:

        {

            border.frame = CGRectMake(0.0f, self.bounds.size.height - width, self.bounds.size.width, width);

        }

            break;

        case WZBBorderDirectionRight:

        {

            border.frame = CGRectMake(self.bounds.size.width - width, 0, width, self.bounds.size.height);

        }

            break;

        default:

            break;

    }

    [self.layer addSublayer:border];

}

**83、通过属性设置UISwitch、UIProgressView等控件的宽高**

mySwitch.transform = CGAffineTransformMakeScale(5.0f, 5.0f);

progressView.transform = CGAffineTransformMakeScale(5.0f, 5.0f);

**84、自动搜索功能，用户连续输入的时候不搜索，用户停止输入的时候自动搜索(我这里设置的是0.5s，可根据需求更改)**

// 输入框文字改变的时候调用

-(void)searchBar:(UISearchBar \*)searchBar textDidChange:(NSString \*)searchText{

    // 先取消调用搜索方法

    [NSObject cancelPreviousPerformRequestsWithTarget:self selector:@selector(searchNewResult) object:nil];

    // 0.5秒后调用搜索方法

    [self performSelector:@selector(searchNewResult) withObject:nil afterDelay:0.5];

}

**85、修改UISearchBar的占位文字颜色**

    // 方法一（推荐使用）

    UITextField \*searchField = [searchBar valueForKey:@"\_searchField"];

    [searchField setValue:[UIColor blueColor] forKeyPath:@"\_placeholderLabel.textColor"];

    // 方法二（已过期）

    [[UILabel appearanceWhenContainedIn:[UISearchBar class], nil] setTextColor:[UIColor redColor]];

    // 方法三（已过期）

    NSDictionary \*placeholderAttributes = @{NSForegroundColorAttributeName : [UIColor redColor], NSFontAttributeName : [UIFont fontWithName:@"HelveticaNeue" size:15],};

    NSAttributedString \*attributedPlaceholder = [[NSAttributedString alloc] initWithString:searchBar.placeholder attributes:placeholderAttributes];

    [[UITextField appearanceWhenContainedIn:[UISearchBar class], nil] setAttributedPlaceholder:attributedPlaceholder];

**86、某个界面多个事件同时响应引起的问题(比如，两个button同时按push到新界面，两个都会响应，可能导致push重叠)**

// UIView有个属性叫做exclusiveTouch，设置为YES后，其响应事件会和其他view互斥(有其他view事件响应的时候点击它不起作用)

view.exclusiveTouch = YES;

// 一个一个设置太麻烦了，可以全局设置

[[UIView appearance] setExclusiveTouch:YES];

// 或者只设置button

[[UIButton appearance] setExclusiveTouch:YES];

**87、修改tabBar的frame**

// 子类化UITabBarViewController，我这里以修改tabBar高度为例，重写viewWillLayoutSubviews方法

#import "WZBTabBarViewController.h"

@interface WZBTabBarViewController ()

@end

@implementation WZBTabBarViewController

- (void)viewWillLayoutSubviews {

    CGRect tabFrame = self.tabBar.frame;

    tabFrame.size.height = 100;

    tabFrame.origin.y = self.view.frame.size.height - 100;

    self.tabBar.frame = tabFrame;

}

@end

**88、修改键盘背景颜色**

// 设置某个键盘颜色

    textField.keyboardAppearance = UIKeyboardAppearanceAlert;

// 设置工程中所有键盘颜色

[[UITextField appearance] setKeyboardAppearance:UIKeyboardAppearanceAlert];

**89、修改image颜色**

UIImage \*image = [UIImage imageNamed:@"test"];

    imageView.image = [image imageWithRenderingMode:UIImageRenderingModeAlwaysTemplate];

    CGRect rect = CGRectMake(0, 0, image.size.width, image.size.height);

    UIGraphicsBeginImageContext(rect.size);

    CGContextRef context = UIGraphicsGetCurrentContext();

    CGContextClipToMask(context, rect, image.CGImage);

    CGContextSetFillColorWithColor(context, [[UIColor redColor] CGColor]);

    CGContextFillRect(context, rect);

    UIImage \*img = UIGraphicsGetImageFromCurrentImageContext();

    UIGraphicsEndImageContext();

    UIImage \*flippedImage = [UIImage imageWithCGImage:img.CGImage scale:1.0 orientation: UIImageOrientationDownMirrored];

    imageView.image = flippedImage;

**90、动画执行removeFromSuperview**

    [UIView animateWithDuration:0.2

                     animations:^{

                         view.alpha = 0.0f;

                     } completion:^(BOOL finished){

                         [view removeFromSuperview];

                     }];

**91、设置UIButton高亮背景颜色**

    [UIView animateWithDuration:0.2

                     animations:^{

                         view.alpha = 0.0f;

                     } completion:^(BOOL finished){

                         [view removeFromSuperview];

                     }];

**92、设置UIButton高亮时的背景颜色**

// 方法一、子类化UIButton，重写setHighlighted:方法，代码如下

#import "WZBButton.h"

@implementation WZBButton

- (void)setHighlighted:(BOOL)highlighted {

    [super setHighlighted:highlighted];

    UIColor \*normalColor = [UIColor greenColor];

    UIColor \*highlightedColor = [UIColor redColor];

    self.backgroundColor = highlighted ? highlightedColor : normalColor;

}

// 方法二、利用setBackgroundImage:forState:方法

[button setBackgroundImage:[self imageWithColor:[UIColor blueColor]] forState:UIControlStateHighlighted];

- (UIImage \*)imageWithColor:(UIColor \*)color {

    CGRect rect = CGRectMake(0.0f, 0.0f, 1.0f, 1.0f);

    UIGraphicsBeginImageContext(rect.size);

    CGContextRef context = UIGraphicsGetCurrentContext();

    CGContextSetFillColorWithColor(context, [color CGColor]);

    CGContextFillRect(context, rect);

    UIImage \*image = UIGraphicsGetImageFromCurrentImageContext();

    UIGraphicsEndImageContext();

    return image;

}

**93、关于图片拉伸**

推荐看这个博客，讲的很详细：[http://blog.csdn.net/q199109106q/article/details/8615661](http://blog.csdn.net/q199109106q/article/details/8615661%5C%22+target=%5C%22_blank%5C%22+style=%5C%22margin:+0px;+padding:+0px;+color:+rgb%28235,+97,+0%29;+text-decoration:+none;)

**94、利用runtime获取一个类所有属性**

- (NSArray \*)allPropertyNames:(Class)aClass

{

    unsigned count;

    objc\_property\_t \*properties = class\_copyPropertyList(aClass, &count);

    NSMutableArray \*rv = [NSMutableArray array];

    unsigned i;

    for (i = 0; i < count; i++)

    {

        objc\_property\_t property = properties[i];

        NSString \*name = [NSString stringWithUTF8String:property\_getName(property)];

        [rv addObject:name];

    }

    free(properties);

    return rv;

}

**95、设置textView的某段文字变成其他颜色**

- (void)setupTextView:(UITextView \*)textView text:(NSString \*)text color:(UIColor \*)color {

    NSMutableAttributedString \*string = [[NSMutableAttributedString alloc]initWithString:textView.text];

    [string addAttribute:NSForegroundColorAttributeName value:color range:[textView.text rangeOfString:text]];

    [textView setAttributedText:string];

}

**96、让push跳转动画像modal跳转动画那样效果(从下往上推上来)**

- (void)push

{

TestViewController \*vc = [[TestViewController alloc] init];

    vc.view.backgroundColor = [UIColor redColor];

    CATransition\* transition = [CATransition animation];

    transition.duration = 0.4f;

    transition.type = kCATransitionMoveIn;

    transition.subtype = kCATransitionFromTop;

    [self.navigationController.view.layer addAnimation:transition forKey:kCATransition];

    [self.navigationController pushViewController:vc animated:NO];

}

- (void)pop

{

CATransition\* transition = [CATransition animation];

    transition.duration = 0.4f;

    transition.type = kCATransitionReveal;

    transition.subtype = kCATransitionFromBottom;

    [self.navigationController.view.layer addAnimation:transition forKey:kCATransition];

    [self.navigationController popViewControllerAnimated:NO];

}

**97、上传图片太大，压缩图片**

-(UIImage \*)resizeImage:(UIImage \*)image

{

    float actualHeight = image.size.height;

    float actualWidth = image.size.width;

    float maxHeight = 300.0;

    float maxWidth = 400.0;

    float imgRatio = actualWidth/actualHeight;

    float maxRatio = maxWidth/maxHeight;

    float compressionQuality = 0.5;//50 percent compression

    if (actualHeight > maxHeight || actualWidth > maxWidth)

    {

        if(imgRatio < maxRatio)

        {

            //adjust width according to maxHeight

            imgRatio = maxHeight / actualHeight;

            actualWidth = imgRatio \* actualWidth;

            actualHeight = maxHeight;

        }

        else if(imgRatio > maxRatio)

        {

            //adjust height according to maxWidth

            imgRatio = maxWidth / actualWidth;

            actualHeight = imgRatio \* actualHeight;

            actualWidth = maxWidth;

        }

        else

        {

            actualHeight = maxHeight;

            actualWidth = maxWidth;

        }

    }

    CGRect rect = CGRectMake(0.0, 0.0, actualWidth, actualHeight);

    UIGraphicsBeginImageContext(rect.size);

    [image drawInRect:rect];

    UIImage \*img = UIGraphicsGetImageFromCurrentImageContext();

    NSData \*imageData = UIImageJPEGRepresentation(img, compressionQuality);

    UIGraphicsEndImageContext();

    return [UIImage imageWithData:imageData];

}