# 1.UIButton

## (1) 文字在左边，图片在右边

CGRect titleF = self.titleLabel.frame;

    CGRect imageF = self.imageView.frame;

    titleF.origin.x = imageF.origin.x;

    self.titleLabel.frame = titleF;

    imageF.origin.x = CGRectGetMaxX(titleF);

    self.imageView.frame = imageF;

## //获取当前系统时间

-(NSString \*)getSystemTime

{

NSDateFormatter\* formatter = [[NSDateFormatter alloc] init];

[formatter setDateFormat:@"yyyy-MM-dd HH:mm:ss"];

NSString\* dateTime= [formatter stringFromDate:[NSDate date]];

// DLog(@"UnixFormat = %@", [self timeStamp:dateTime]);

// DLog(@"TimeFormat = %@", [self formatTime: [self timeStamp:dateTime]]);

return dateTime;

}

## //标准时间转化为时间戳

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

{

NSDateFormatter\* formater = [[NSDateFormatter alloc] init];

[formater setDateFormat:@"yyyy-MM-dd HH:mm:ss"];

NSDate\* date = [formater dateFromString:str];

NSString \*timeSp = [NSString stringWithFormat:@"%.f", [date timeIntervalSince1970]];

return timeSp;

}

## //时间戳转换成标准时间

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

{

NSDateFormatter\* formatter = [[NSDateFormatter alloc]init];

[formatter setDateFormat:@"yyyy-MM-dd HH:mm:ss"];

NSDate \*date = [NSDate dateWithTimeIntervalSince1970:[str intValue]];

NSString \* dateTime = [formatter stringFromDate:date];

return dateTime;

}

## //时间戳转换成标准时间

- (NSString\*)formatTime:(NSString\*)str format:(NSString \*)format

{

NSDateFormatter\* formatter = [[NSDateFormatter alloc]init];

[formatter setDateFormat:format];

NSDate \*date = [NSDate dateWithTimeIntervalSince1970:[str intValue]];

NSString \* dateTime = [formatter stringFromDate:date];

return dateTime;

}

## // 求时差

- (NSString\*)timeDifference:(NSString\*)newTime ToOld:(NSString \*)oldTime

{

double dateTimeValue = newTime.doubleValue - oldTime.doubleValue;

NSString \*dateTime = [self timeStringByTimeValue:dateTimeValue];

return dateTime;

}

**//根据时间戳得到对应的时间字符串**

- (NSString \*)timeStringByTimeValue:(double)dateTimeValue

{

NSInteger h = (int)dateTimeValue / 3600;

NSInteger m = ((int)dateTimeValue - h \* 3600)/60;

NSInteger s = (int)dateTimeValue - 3600 \* h - m \* 60;

NSString \*h\_s = [self getTime:h];

NSString \*m\_s = [self getTime:m];

NSString \*s\_s = [self getTime:s];

NSString \*dateTime = [NSString stringWithFormat:@"%@:%@:%@",h\_s,m\_s,s\_s];

return dateTime;

}

## //计算一段字符串的长度，两个英文字符占一个长度。

- (int)countTheStrLength:(NSString\*)str

{

int strlength = 0;

char\* p = (char\*)[str cStringUsingEncoding:NSUnicodeStringEncoding];

for (int i=0 ; i<[str lengthOfBytesUsingEncoding:NSUnicodeStringEncoding] ;i++) {

if (\*p) {

p++;

strlength++;

}

else {

p++;

}

}

return (strlength+1)/2;

}

## //获取Document目录路径

-(NSString \*)getDocumetPath

{

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

NSString \*docDir = [paths objectAtIndex:0];

return docDir;

}

## //判断NSString字符串是否包含emoji表情

- (BOOL)stringContainsEmoji:(NSString \*)string

{

\_\_block BOOL returnValue =NO;

[string enumerateSubstringsInRange:NSMakeRange(0, [string length]) options:NSStringEnumerationByComposedCharacterSequences usingBlock:^(NSString \*substring, NSRange substringRange, NSRange enclosingRange, BOOL \*stop) {

const unichar hs = [substring characterAtIndex:0];

// surrogate pair

if (0xd800) {

if (0xd800 <= hs && hs <= 0xdbff) {

if (substring.length > 1) {

const unichar ls = [substring characterAtIndex:1];

const int uc = ((hs - 0xd800) \* 0x400) + (ls - 0xdc00) + 0x10000;

if (0x1d000 <= uc && uc <= 0x1f77f) {

returnValue =YES;

}

}

}else if (substring.length > 1) {

const unichar ls = [substring characterAtIndex:1];

if (ls == 0x20e3) {

returnValue =YES;

}

}else {

// non surrogate

if (0x2100 <= hs && hs <= 0x27ff) {

returnValue =YES;

}else if (0x2B05 <= hs && hs <= 0x2b07) {

returnValue =YES;

}else if (0x2934 <= hs && hs <= 0x2935) {

returnValue =YES;

}else if (0x3297 <= hs && hs <= 0x3299) {

returnValue =YES;

}else if (hs == 0xa9 || hs == 0xae || hs == 0x303d || hs == 0x3030 || hs == 0x2b55 || hs == 0x2b1c || hs == 0x2b1b || hs == 0x2b50) {

returnValue =YES;

}

}

}

}];

return returnValue;

}

# [修改UIPageControl的点点，可替换任意图片image](http://blog.csdn.net/a416863220/article/details/43407517)

[pageControl setValue:[UIImage imageWithName:@"selected"] forKeyPath:@"\_currentPageImage"];

[pageControl setValue:[UIImage imageWithName:@"normal"] forKeyPath:@"\_pageImage"];

[UITableView默认分割线左边补全方法](http://blog.csdn.net/a416863220/article/details/43794121)

需要重写viewDidLayoutSubviews方法

/\*\*

\* 显示完全分割线

\*/

-(void)viewDidLayoutSubviews

{

if ([self.tableView respondsToSelector:@selector(setSeparatorInset:)]) {

[self.tableView setSeparatorInset:UIEdgeInsetsZero];

}

if ([self.tableView respondsToSelector:@selector(setLayoutMargins:)]) {

[self.tableView setLayoutMargins:UIEdgeInsetsZero];

}

}

- (void)tableView:(UITableView \*)tableView willDisplayCell:(UITableViewCell \*)cell forRowAtIndexPath:(NSIndexPath \*)indexPath

{

if ([cell respondsToSelector:@selector(setSeparatorInset:)]) {

[cell setSeparatorInset:UIEdgeInsetsZero];

}

if ([cell respondsToSelector:@selector(setLayoutMargins:)]) {

[cell setLayoutMargins:UIEdgeInsetsZero];

}

}

# [UIView的四个边角单独设置为圆角弧度](http://blog.csdn.net/a416863220/article/details/43795509)

UIView \*view = [[UIView alloc] initWithFrame:CGRectMake(120, 100, 100, 100)];

view.backgroundColor = [UIColor blueColor];

[self.view addSubview:view];

UIBezierPath \*maskPath = [UIBezierPath bezierPathWithRoundedRect:view.bounds byRoundingCorners:UIRectCornerTopLeft | UIRectCornerBottomRight cornerRadii:CGSizeMake(15, 15)];

CAShapeLayer \*maskLayer = [[CAShapeLayer alloc] init];

maskLayer.frame = view.bounds;

maskLayer.path = maskPath.CGPath;

view.layer.mask = maskLayer;

# [UITableView头部ImageView下拉放大效果，导航栏透明渐变](http://blog.csdn.net/a416863220/article/details/50801164)

这里导航栏透明用的是这位大神的[http://tech.glowing.com/cn/change-uinavigationbar-backgroundcolor-dynamically/](http://tech.glowing.com/cn/change-uinavigationbar-backgroundcolor-dynamically/" \t "http://blog.csdn.net/a416863220/article/details/_blank)

#import "ViewController.h"

#import "UINavigationBar+Awesome.h"

#define ImageWidth [[UIScreen mainScreen] bounds].size.width

static CGFloat imageH = 200;

static CGFloat navH = 64;

@interface ViewController ()<UITableViewDelegate,UITableViewDataSource>

@property (strong, nonatomic) UITableView \*tableView;

@property (nonatomic, strong) UIImageView \*headerView;

@property (nonatomic, strong) UIImage \*shadowImage;

@end

@implementation ViewController

- (void)viewDidLoad {

[super viewDidLoad];

// Do any additional setup after loading the view, typically from a nib.

self.navigationItem.leftBarButtonItem = [[UIBarButtonItem alloc]initWithTitle:@"返回" style:UIBarButtonItemStyleDone target:self action:nil];

self.navigationItem.rightBarButtonItem = [[UIBarButtonItem alloc]initWithTitle:@"确定" style:UIBarButtonItemStyleDone target:self action:nil];

self.automaticallyAdjustsScrollViewInsets = NO;

self.tableView = [[UITableView alloc]initWithFrame:self.view.bounds style:UITableViewStylePlain];

self.tableView.delegate = self;

self.tableView.dataSource = self;

self.tableView.contentInset = UIEdgeInsetsMake(imageH, 0, 0, 0);

[self.view addSubview:self.tableView];

self.headerView = [[UIImageView alloc]init];

self.headerView.frame = CGRectMake(0, -imageH, ImageWidth, imageH);

self.headerView.image = [UIImage imageNamed:@"IMG\_0106.JPG"];

self.headerView.contentMode = UIViewContentModeScaleAspectFill;

[self.tableView addSubview:self.headerView];

[self.tableView insertSubview:self.headerView atIndex:0];

}

- (void)viewWillAppear:(BOOL)animated

{

[super viewWillAppear:animated];

self.shadowImage = self.navigationController.navigationBar.shadowImage;

[self.navigationController.navigationBar setShadowImage:[UIImage new]];

CGFloat offsetY = self.tableView.contentOffset.y;

[self changeNavAlphaWithConnentOffset:offsetY];

}

- (void)viewWillDisappear:(BOOL)animated

{

[super viewWillDisappear:animated];

[self.navigationController.navigationBar lt\_reset];

self.navigationController.navigationBar.shadowImage = self.shadowImage;

}

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

{

return 50;

}

-(CGFloat)tableView:(UITableView \*)tableView heightForRowAtIndexPath:(NSIndexPath \*)indexPath

{

return 50;

}

-(UITableViewCell \*)tableView:(UITableView \*)tableView cellForRowAtIndexPath:(NSIndexPath \*)indexPath

{

static NSString \*ID = @"XXXX";

UITableViewCell \*cell = [tableView dequeueReusableCellWithIdentifier:ID];

if (!cell) {

cell = [[UITableViewCell alloc]initWithStyle:UITableViewCellStyleDefault reuseIdentifier:ID];

}

cell.textLabel.text = [NSString stringWithFormat:@"%ld",indexPath.row];

return cell;

}

-(void)tableView:(UITableView \*)tableView didSelectRowAtIndexPath:(NSIndexPath \*)indexPath

{

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

vc.view.backgroundColor = [UIColor whiteColor];

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

}

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

{

CGFloat offsetY = scrollView.contentOffset.y;

NSLog(@"%f",offsetY);

if (offsetY < -imageH) {

NSLog(@"开始改变");

CGRect f = self.headerView.frame;

f.origin.y = offsetY;

f.size.height = -offsetY;

self.headerView.frame = f;

}

[self changeNavAlphaWithConnentOffset:offsetY];

}

-(void)changeNavAlphaWithConnentOffset:(CGFloat)offsetY

{

UIColor \*color = [UIColor colorWithRed:246/255.0 green:246/255.0 blue:246/255.0 alpha:1];

if (offsetY > -navH \* 2 ) {

NSLog(@"渐渐不透明");

CGFloat alpha = MIN(1, 1 - ((-navH \* 2 + navH - offsetY) / navH));

[self.navigationController.navigationBar lt\_setBackgroundColor:[color colorWithAlphaComponent:alpha]];

self.title = @"个人主页";

}

else {

NSLog(@"渐渐透明");

[self.navigationController.navigationBar lt\_setBackgroundColor:[color colorWithAlphaComponent:0]];

self.title = @"";

}

}

# React Native 中文版

[http://wiki.jikexueyuan.com/project/react-native/](http://wiki.jikexueyuan.com/project/react-native/" \t "http://blog.csdn.net/a416863220/article/details/_blank)

# Japatch库。

[http://www.ios122.com/2015/11/jspatch/](http://www.ios122.com/2015/11/jspatch/" \t "http://blog.csdn.net/a416863220/article/details/_blank)

# [IOS实现SpotLight搜索 让你的APP支持SpotLight搜索](http://blog.csdn.net/a416863220/article/details/51220375)

http://blog.csdn.net/a416863220/article/details/51220375