

iOS 阴影(shadow)

TIGER_XXXX

关注

6

2017.11.01 21:25:17

字数 178

阅读 29,033

1. 简单阴影

我们给layer设置了shadowOpacity后就能得到一个简单的阴影

```
1 | view.layer.shadowOpacity = 1;
```



shadowOpacity设置了阴影的不透明度,取值范围在0~1
这里shadow有一个默认值
shadowOffset = CGSizeMake(0, -3)
shadowRadius = 3.0
注意:如果view没有设置背景色阴影也是不会显示的

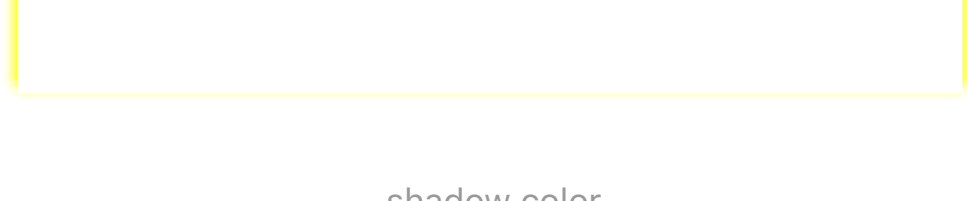
2. 阴影属性

layer中与阴影相关的属性有以下几个

```
1 | CGColorRef *) shadowColor//阴影颜色
2 | (Float) shadowOpacity//阴影透明度
3 | (CGSize) shadowOffset//阴影偏移量
4 | (CGFloat) shadowRadius//模糊计算的半径
5 | (CGPathRef *) shadowPath//阴影路径
```

3. shadowColor

```
1 | - (void)p_setupSubviews {
2 |     self.view.backgroundColor = [UIColor whiteColor];
3 |     [self p_setupViewWithY:100 shadowColor:[UIColor redColor]];
4 |     [self p_setupViewWithY:170 shadowColor:[UIColor blueColor]];
5 |     [self p_setupViewWithY:240 shadowColor:[UIColor yellowColor]];
6 | }
7 |
8 | - (void)p_setupViewWithY:(CGFloat)y shadowColor:(UIColor *)shadowColor {
9 |     UIView *view = [[UIView alloc] initWithFrame:CGRectMake(30, y, [UIScreen mainScreen
10 |     view.backgroundColor = [UIColor whiteColor];
11 |     [self.view addSubview:view];
12 |     view.layer.shadowOpacity = 1;
13 |     view.layer.shadowColor = shadowColor.CGColor;
14 | }
```



shadow color

4. shadowOpacity

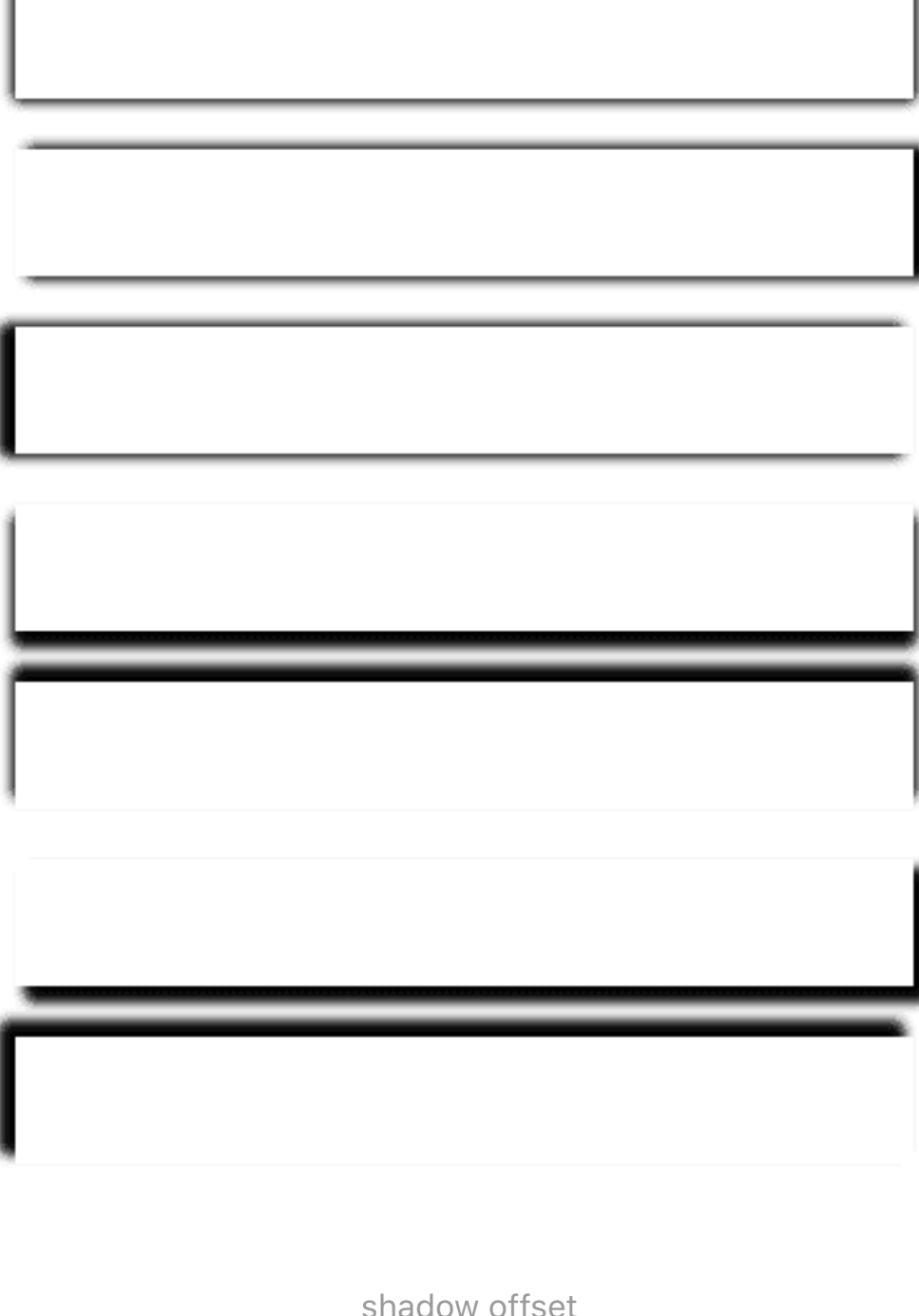
```
1 | - (void)p_setupSubviews {
2 |     self.view.backgroundColor = [UIColor whiteColor];
3 |     [self p_setupViewWithY:100 shadowOffset:CGSizeMake(0, 0)];
4 |     [self p_setupViewWithY:170 shadowOpacity:0.2];
5 |     [self p_setupViewWithY:240 shadowOpacity:0.6];
6 | }
7 |
8 | - (void)p_setupViewWithY:(CGFloat)y shadowOpacity:(Float)shadowOpacity {
9 |     UIView *view = [[UIView alloc] initWithFrame:CGRectMake(30, y, [UIScreen mainScreen
10 |     view.backgroundColor = [UIColor whiteColor];
11 |     [self.view addSubview:view];
12 |     view.layer.shadowOpacity = shadowOpacity;
13 | }
```



shadow opacity

5. shadowOffset

```
1 | - (void)p_setupSubviews {
2 |     self.view.backgroundColor = [UIColor whiteColor];
3 |     [self p_setupViewWithY:100 shadowOffset:CGSizeMake(0, 0)];
4 |     [self p_setupViewWithY:170 shadowOffset:CGSizeMake(5, 0)];
5 |     [self p_setupViewWithY:240 shadowOffset:CGSizeMake(-5, 0)];
6 |     [self p_setupViewWithY:310 shadowOffset:CGSizeMake(0, 5)];
7 |     [self p_setupViewWithY:380 shadowOffset:CGSizeMake(0, -5)];
8 |     [self p_setupViewWithY:450 shadowOffset:CGSizeMake(5, 5)];
9 |     [self p_setupViewWithY:520 shadowOffset:CGSizeMake(-5, -5)];
10 | }
11 |
12 | - (void)p_setupViewWithY:(CGFloat)y shadowOffset:(CGSize)shadowOffset {
13 |     UIView *view = [[UIView alloc] initWithFrame:CGRectMake(30, y, [UIScreen mainScreen
14 |     view.backgroundColor = [UIColor whiteColor];
15 |     [self.view addSubview:view];
16 |     view.layer.shadowOpacity = 1;
17 |     view.layer.shadowOffset = shadowOffset;
18 | }
```



shadow offset

6. shadowRadius

shadowRadius其实可以理解为阴影的宽度

```
1 | - (void)p_setupSubviews {
2 |     self.view.backgroundColor = [UIColor whiteColor];
3 |     [self p_setupViewWithY:100 shadowRadius:0];
4 |     [self p_setupViewWithY:170 shadowRadius:3.0];
5 |     [self p_setupViewWithY:240 shadowRadius:10.0];
6 | }
7 |
8 | - (void)p_setupViewWithY:(CGFloat)y shadowRadius:(CGFloat)shadowRadius {
9 |     UIView *view = [[UIView alloc] initWithFrame:CGRectMake(30, y, [UIScreen mainScreen
10 |     view.backgroundColor = [UIColor whiteColor];
11 |     [self.view addSubview:view];
12 |     view.layer.shadowOpacity = 1;
13 |     view.layer.shadowOffset = CGSizeMake(0, 0);
14 |     view.layer.shadowRadius = shadowRadius;
15 | }
16 | }
```



shadow radius

7. shadowPath

```
1 | - (void)p_setupSubviews {
2 |     self.view.backgroundColor = [UIColor whiteColor];
3 |     [self p_setupViewWithY1:100];
4 |     [self p_setupViewWithY2:170];
5 |     [self p_setupViewWithY3:240];//贝塞尔曲线未闭合
6 | }
7 |
8 | - (void)p_setupViewWithY1:(CGFloat)y {
9 |     UIView *view = [[UIView alloc] initWithFrame:CGRectMake(30, y, [UIScreen mainScreen
10 |     view.backgroundColor = [UIColor whiteColor];
11 |     [self.view addSubview:view];
12 |     view.layer.shadowOpacity = 1;
13 |     UIBezierPath *path = [UIBezierPath bezierPathWithRect:view.bounds];
14 |     view.layer.shadowPath = path.CGPath;
15 | }
16 |
17 | - (void)p_setupViewWithY2:(CGFloat)y {
18 |     UIView *view = [[UIView alloc] initWithFrame:CGRectMake(30, y, [UIScreen mainScreen
19 |     view.backgroundColor = [UIColor whiteColor];
20 |     [self.view addSubview:view];
21 |     view.layer.shadowOpacity = 1;
22 |     UIBezierPath *path = [UIBezierPath bezierPath];
23 |     [path moveToPoint:CGPointMake(0, 0)];
24 |     [path addLineToPoint:CGPointMake(0, view.frame.size.height + 10)];
25 |     [path addLineToPoint:CGPointMake(view.frame.size.width, view.frame.size.height + 10)];
26 |     [path addLineToPoint:CGPointMake(view.frame.size.width, view.frame.size.height)];
27 |     [path addLineToPoint:CGPointMake(0, view.frame.size.height)];
28 |     view.layer.shadowPath = path.CGPath;
29 | }
30 |
31 | - (void)p_setupViewWithY3:(CGFloat)y {
32 |     UIView *view = [[UIView alloc] initWithFrame:CGRectMake(30, y, [UIScreen mainScreen
33 |     view.backgroundColor = [UIColor whiteColor];
34 |     [self.view addSubview:view];
35 |     view.layer.shadowOpacity = 1;
36 |     view.layer.shadowOffset = CGSizeMake(0, 0);
37 |     UIBezierPath *path = [UIBezierPath bezierPath];
38 |     [path moveToPoint:CGPointMake(-5, 0)];
39 |     [path addLineToPoint:CGPointMake(-5, view.frame.size.height)];
40 |     [path addLineToPoint:CGPointMake(view.frame.size.width, view.frame.size.height)];
41 |     [path addLineToPoint:CGPointMake(view.frame.size.width, 0)];
42 |     view.layer.shadowPath = path.CGPath;
43 | }
```



shadow path

当用bounds设置path时,看起来的效果与只设置了shadowOpacity一样
但是添加了shadowPath后消除了离屏渲染问题

微信公众号TIGERXXXX后台回复636T7获取demo源码

62人点赞

iOS 知识点

...

"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下

TIGER_XXXX

菜鸡程序员，欢迎关注微信公众号TIGERXXXX留言交流

总资产 101 (约7.21元)

共写了 2.6W字

获得 167个赞

共 32个粉丝

关注

模型的制作 写字楼出租出租 全包的装修公司 淘气堡 学习做奶茶 动画设计创作

写下你的评论...

全部评论 2 只看作者 按时间倒序 按时间正序

金凤细雨 3楼 02.07 18:40

写的无比的👍,太赞

赞 回复

JerseyBro 2楼 2019.04.14 16:56

简洁清晰

赞 回复

被以下专题收入，发现更多相似内容

CAShape... iOS开发知识收录 征服iOS 开发文档 iOS

基础应用 iOS 深度好文 展开更多

推荐阅读

iOS中CALayer的使用

在iOS中随处都可以看到绚丽的动画效果，实现这些动画的过程并不复杂，今天将带大家一窥iOS动画全貌。在这里你可以看...

每天刷两次牙 阅读 4,906 评论 4 赞 22

CALayer的使用

在iOS中随处都可以看到绚丽的动画效果，实现这些动画的过程并不复杂，今天将带大家一窥iOS动画全貌。在这里你可以看...

F麦子 阅读 3,238 评论 3 赞 10

iOS shadow阴影 + 给整个tableView外框添加阴影

最近项目涉及到对tableView设置阴影效果，实现技术上不难，主要是思路。这里记录下来，也方便大家学习。阴影是...

Jin_先生 阅读 11,954 评论 9 赞 17

ios-layer

转载：http://www.jianshu.com/p/32fcadd12108 每个UIView有一个伙伴称为L...

F麦子 阅读 2,114 评论 0 赞 12

iOS原生网络请求-"连接"与"会话"

1.NSURLConnection 1.1 get 异步请求 1.2 post 异步请求 1.3 post 同步请求...

小苗晓雪 阅读 422 评论 0 赞 4

写下你的评论... 评论2 赞62 ...