UIButton 事件监听以及属性

1能够定义的button类型有以下6种。但是现在已经没有这么多形态了，因为苹果UI全部换成扁平化，现在只剩下四种，系统的，自定义的，加号，感叹号，最后那种圆可以画

UIButtonTypeCustom = 0, //自定义风格

UIButtonTypeRoundedRect, //圆角矩形

UIButtonTypeDetailDisclosure, //蓝色小箭头按钮，主要做详细说明用

UIButtonTypeInfoLight,// 亮色感叹号

UIButtonTypeInfoDark, //暗色感叹号

UIButtonTypeContactAdd,// 十字加号按钮

代码： **UIButton** \*btn2 = [UIButton buttonWithType:UIButtonTypeRoundedRect];

[view的frame和bounds之区别（位置和大小）](http://blog.csdn.net/mad1989/article/details/8711697)

先看到下面的代码你肯定就明白了一些：  
-(CGRect)frame{  
    return CGRectMake(self.frame.origin.x,self.frame.origin.y,self.frame.size.width,self.frame.size.height);  
}  
-(CGRect)bounds{  
    return CGRectMake(0,0,self.frame.size.width,self.frame.size.height);  
}  
很明显，bounds的原点是(0,0)点（就是view本身的坐标系统，默认永远都是0，0点，除非认为setbounds），而frame的原点却是任意的（相对于父视图中的坐标位置）。

//背景色

btn.backgroundColor = [UIColor cyanColor];

//设置button 的填充图片

[btn setImage:[UIImage imageNamed:@"share\_qq"] forState:UIControlStateNormal];

//选中状态下的图片

[btn setImage:[UIImage imageNamed:@"share\_pengyouquan"] forState:UIControlStateSelected];

//背静图片

[btn setBackgroundImage:[UIImage imageNamed:@"share\_weixin"] forState:UIControlStateNormal];

//设置tag值

btn.tag = 12345;

//添加事件

[btn addTarget:self action:@selector(btnClick:) forControlEvents:UIControlEventTouchUpInside];

//加载到view上

[self.view addSubview:btn];

//取消按钮已经添加的所有事件:(这个比较重要,若添加了两个事件 两个事件都会被触发)

[btn removeTarget:nil action:nil forControlEvents:UIControlEventTouchUpInside];

// 初始化按钮样式

UIButton \*button = [UIButton buttonWithType:UIButtonTypeCustom];

/\*

UIButtonTypeCustom 自定义 (默认 无样式)

UIButtonTypeRoundedRect 白色圆角矩形

UIButtonTypeDetailDisclosure 蓝色小箭头按钮，主要做详细说明用

UIButtonTypeInfoLight 亮色圆圈感叹号信息按钮

UIButtonTypeInfoDark 深色圆圈感叹号信息按钮

UIButtonTypeContactAdd 蓝色加号(+)按钮

\*/

// 设置位置和大小

button.frame = CGRectMake(20, 100, 250, 50);

// 设置裁剪半径

button.layer.cornerRadius = 20;

// 设置裁剪后边框颜色

button.layer.borderColor = [UIColor redColor].CGColor;

// 设置裁剪后边框宽度

button.layer.borderWidth = 1;

// 设置隐藏裁剪区域

button.layer.masksToBounds = YES;

// 设置标签 用来辨别点击的是哪个button

button.tag = 1000;

// 设置背景颜色

button.backgroundColor = [UIColor brownColor];

// 设置不同状态时的标题文字及颜色

[button setTitle:@"Normal Title" forState:UIControlStateNormal];

[button setTitle:@"Hlight Title" forState:UIControlStateHighlighted];

[button setTitleColor:[UIColor redColor] forState:UIControlStateNormal];

[button setTitleColor:[UIColor yellowColor] forState:UIControlStateHighlighted];

// 设置不同状态时的富文本标题 标题其他设置无效

//NSAttributedString \*attributedString = [[NSAttributedString alloc] initWithString:@"Attributed Title" attributes:@{NSFontAttributeName:[UIFont systemFontOfSize:14], NSForegroundColorAttributeName:[UIColor purpleColor]}];

//[button setAttributedTitle:attributedString forState:UIControlStateNormal];

// 设置不同状态时的标题图片

// 原比例的显示 当宽高不足时图片不足的一边被压缩

// 当设置了标题且按钮宽度不够时优先显示图片 图片原大小显示，当宽高不足时

[button setImage:[UIImage imageNamed:@"4"] forState:UIControlStateNormal];

// 设置不同状态时的背景图片

// 图片自动拉伸来适应button的大小 背景颜色被覆盖

[button setBackgroundImage:[UIImage imageNamed:@"pink"] forState:UIControlStateNormal];

// 设置不同状态时的标题阴影颜色.不同状态默认的标题阴影颜色是UIControlStateNormal时的值.若normal状态的值没有设定,默认是个系统值,所以你只要要设置normal状态

[button setTitleShadowColor:[UIColor blueColor] forState:UIControlStateNormal];

/\*

UIControlStateNormal 常规状态

UIControlStateHighlighted 高亮状态

UIControlStateDisabled 禁用状态

UIControlStateSelected 选中状态

UIControlStateFocused 当屏幕支持聚焦时可用 iOS 9 新属性

UIControlStateApplication 当应用程序标志使用时

UIControlStateReserved 为内部框架预留的

\*/

// 设置标题图片和文字偏移量 (默认UIEdgeInsetsZero)

// UIEdgeInsetsMake(CGFloat top, CGFloat left, CGFloat bottom, CGFloat right)

//button.contentEdgeInsets = UIEdgeInsetsMake(100, 0, 0, 0);

// 设置标题文字偏移量 (默认UIEdgeInsetsZero)

//button.titleEdgeInsets = UIEdgeInsetsMake(0, 0, 0, 100);

// 设置标题图片偏移量 (默认UIEdgeInsetsZero)

//button.imageEdgeInsets = UIEdgeInsetsMake(0, 100, 0, -100);

// 设置高亮状态改变阴影 阴影在雕刻与浮雕感之间变化(差不多就是去正常offset的相反数作为新的offset)

button.reversesTitleShadowWhenHighlighted = NO;

// 设置高亮状态图片变深色 (默认YES)

button.adjustsImageWhenHighlighted = YES;

// 设置禁用状态图片变浅色 (默认YES)

//button.enabled = NO;

button.adjustsImageWhenDisabled = YES;

// 设置高亮状态下按钮发光 (默认NO)

// 多用于信息按钮或重要按钮

// 若设置了标题图片则图片发光 若未设置则按钮中间发光

button.showsTouchWhenHighlighted = YES;

// 想深入了解TintColor请猛戳这里:http://www.cocoachina.com/ios/20150703/12363.html?utm\_medium=referral&utm\_source=pulsenews

//[button setTintColor:[UIColor blueColor]];

// 设置事件

[button addTarget:self action:@selector(buttonAction:) forControlEvents:UIControlEventTouchUpInside];

/\*

UIControlEventTouchDown 按下

UIControlEventTouchDownRepeat 多次按下

UIControlEventTouchDragInside 保持按下然后在按钮及其一定的外围拖动

UIControlEventTouchDragOutside 保持按下,在按钮外面拖动

UIControlEventTouchDragEnter DragOutside进入DragInside触发

UIControlEventTouchDragExit in到out触发

UIControlEventTouchUpInside 在按钮及其一定外围内松开

UIControlEventTouchUpOutside 按钮外面松开

UIControlEventTouchCancel 点击取消

\*/

// 获取按钮当前状态的属性 (只读)

/\*

button.currentTitle 当前按钮上显示的标题(只读).当按钮状态改变时值自动改变.值可以为nil

button.currentTitleColor 当前标题颜色(只读).此值要保证不为nil,默认是白色

button.currentTitleShadowColor 标题的阴影颜色(只读).默认是白色

button.currentImage 当前按钮上的图片(只读).可以是nil.

button.currentBackgroundImage 当前按钮背景图片(只读).可以是nil.

button.currentAttributedTitle 当前按钮上显示的富文本标题(只读).当按钮状态改变时值自动改变.值可以为nil

button.titleLabel 显示按钮当前标题的视图(只读).虽然它是只读的,但是它的属性是可读写的.它的属性在按钮还没有显示之前就有返回值.系统按钮这些值为nil

button.imageView 按钮上的图片视图(只读).虽然它是只读的,但是他的属性是可读写的.imageView的属性在按钮还没有显示前就有值了.系统按钮这些值是nil

\*/

[self.view addSubview:button];

}

- (void)buttonAction:(UIButton \*)button

{

}

/\*

– backgroundRectForBounds: 返回背景绘制区域.

– contentRectForBounds: 返回内容绘制区域.内容区域是显示图片和标题及他们特定对齐缩放等的范围.

– titleRectForContentRect: 返回标题的绘制区域.

– imageRectForContentRect: 返回图片的绘制区域.

\*/

对待