

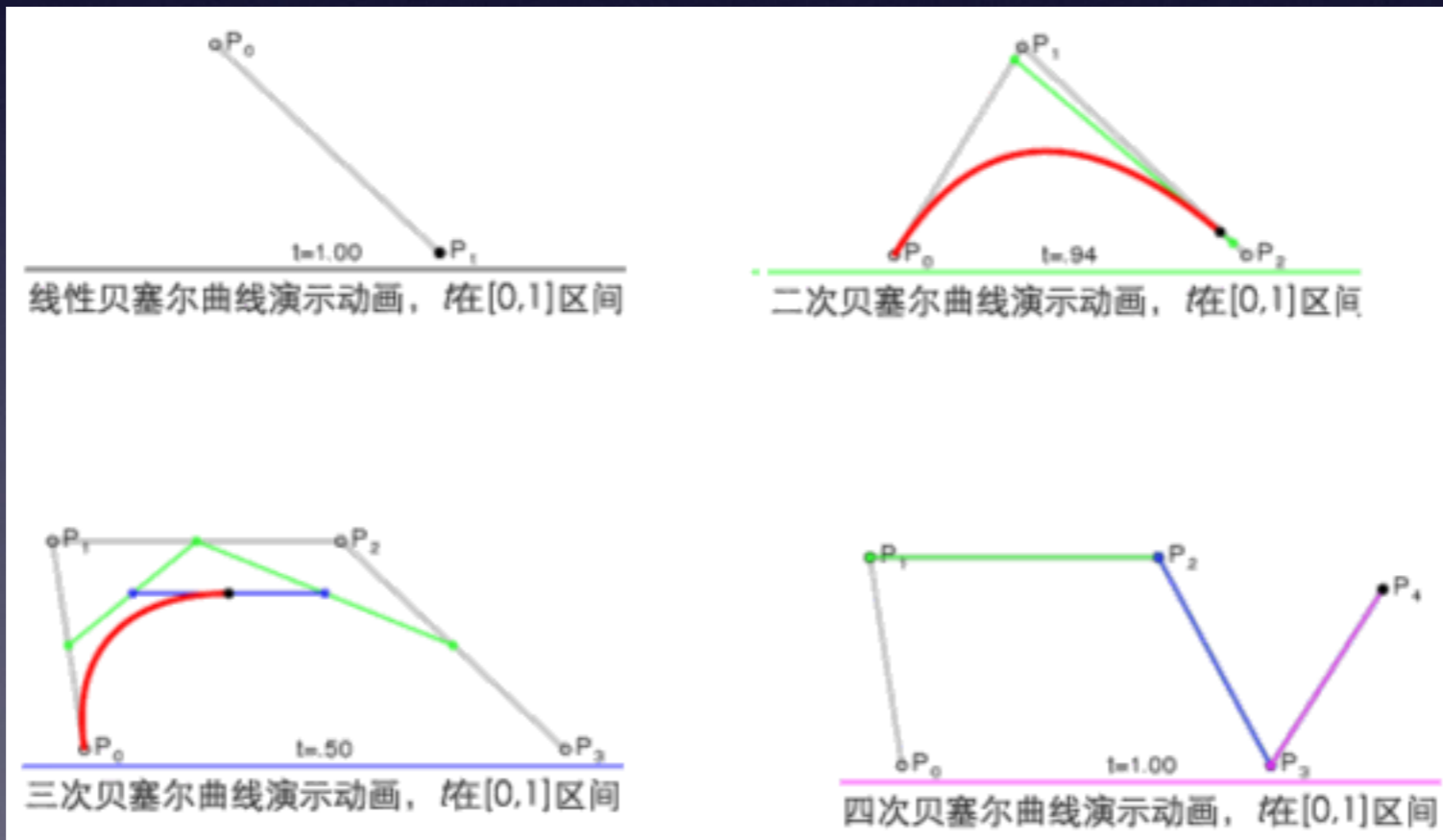
# 图形的绘制

# UIView的drawRect方法

```
override func drawRect(rect: CGRect) {  
    //绘制代码  
}
```

# Bézier Curve (贝赛尔曲线)

1. 1962年法国工程师Pierre Bézier发表;
2. 计算机图形图像造形基本工具, 运用最多的基本线条之一;



## 画一根线段

```
let linePath = UIBezierPath()  
linePath.move(to: CGPoint(x: 30, y: 50))  
linePath.addLine(to: CGPoint(x: 190, y: 50))  
UIColor.black.setStroke() //设置边框颜色  
linePath.lineWidth = 2 //设置线宽  
linePath.stroke() //绘制边框
```



# 画一根虚线段

//MARK:画一根虚线段

```
let dashLinePath = UIBezierPath()
dashLinePath.move(to: CGPoint(x: 30, y: 50))
dashLinePath.addLine(to: CGPoint(x: 190, y: 50))
UIColor.black.setStroke()
dashLinePath.lineWidth = 2
```

```
guard let context = UIGraphicsGetCurrentContext() as
CGContext? else{ return }
context.saveGState()
context.setLineDash(phase: 0, lengths: [10,5]) //从0开始,循环地画:
dashLinePath.stroke() //10个点跳5个点
context.restoreGState()
```

-----

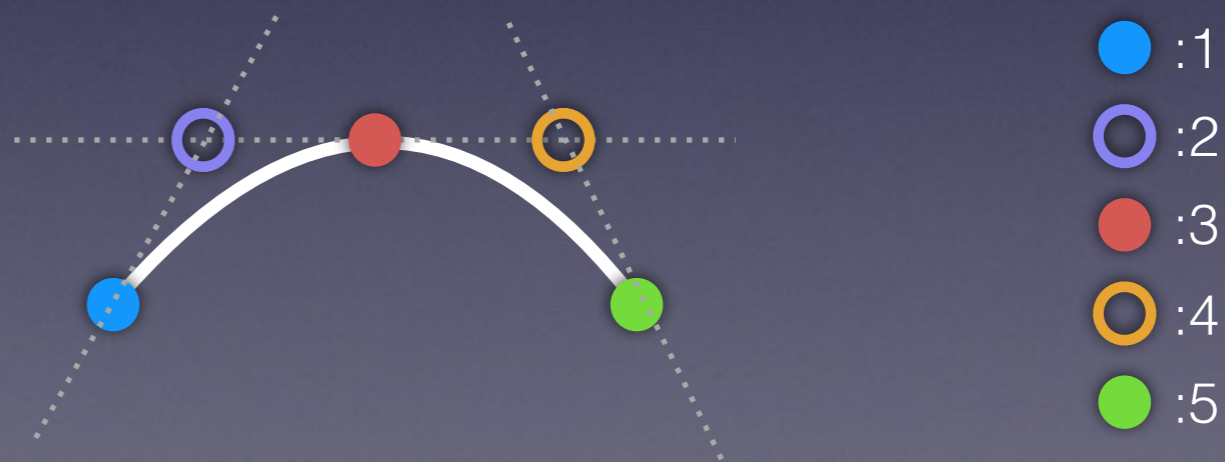
## 画一根折线

```
let brokenLinPath = UIBezierPath()  
brokenLinPath.move(to: CGPoint(x: 30, y: 120))  
brokenLinPath.addLine(to: CGPoint(x: 109.5, y: 90))  
brokenLinPath.addLine(to: CGPoint(x: 190, y: 120))  
UIColor.black.setStroke()  
brokenLinPath.lineWidth = 2  
brokenLinPath.stroke()
```



## 画一根曲线

```
let curvePath = UIBezierPath()
curvePath.move(to: CGPoint(x: 30, y: 180))
curvePath.addCurve(to: CGPoint(x: 110, y: 140), controlPoint1:
CGPoint(x: 30, y: 180), controlPoint2: CGPoint(x: 70, y: 140))
curvePath.addCurve(to: CGPoint(x: 190, y: 180), controlPoint1:
CGPoint(x: 150, y: 140), controlPoint2: CGPoint(x: 190, y: 180))
UIColor.black.setStroke()
curvePath.lineWidth = 2
curvePath.stroke()
```



## 绘制一个矩形

```
let rectangleRect = UIEdgeInsetsInsetRect(self.bounds,  
UIEdgeInsetsMake(20, 20, 20, 20))  
let rectanglePath = UIBezierPath(rect: rectangleRect)  
//填充颜色  
UIColor.gray.setFill()  
rectanglePath.fill()
```





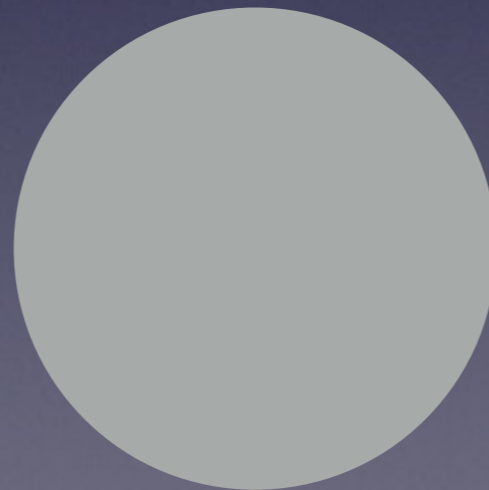
## 绘制三角形

```
let polygonRect = UIEdgeInsetsInsetRect(self.bounds,  
UIEdgeInsetsMake(20, 20, 20, 20))  
let polygon3Path = UIBezierPath()  
polygon3Path.move(to: CGPoint(x: polygonRect.minX + 100, y:  
polygonRect.minY + 40.5))  
polygon3Path.addLine(to: CGPoint(x: polygonRect.minX +  
151.53, y: polygonRect.minY + 129.75))  
polygon3Path.addLine(to: CGPoint(x: polygonRect.minX +  
48.47, y: polygonRect.minY + 129.75))  
polygon3Path.close()  
UIColor.green.setFill()
```



## 绘制一个圆形

```
let ovalRect = UIEdgeInsetsInsetRect(self.bounds,  
UIEdgeInsetsMake(20, 20, 20, 20))  
let ovalPath = UIBezierPath(ovalIn: ovalRect)  
    //MARK:填充颜色  
UIColor.darkGray.setFill()  
ovalPath.fill()
```



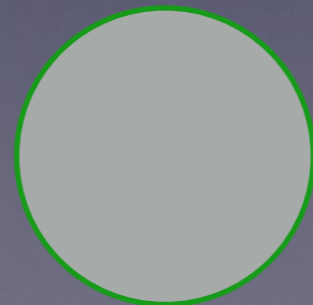
## 填充图片&加边框

//填充图片

```
guard let context = UIGraphicsGetCurrentContext() as  
CGContext?,  
    let image = UIImage(named: "image.png") as UIImage? else {  
        return  
    }  
context.saveGState()  
ovalPath.addClip()  
image.draw(in: ovalRect)  
context.restoreGState()
```

//加边框

```
UIColor.green.setStroke()  
ovalPath.lineWidth = 1
```



# 绘制一段文字

```
let rectangleStyle = NSMutableParagraphStyle() //段落样式集
rectangleStyle.alignment = .center

let rectangleFontAttributes = [NSFontAttributeName: UIFont(name:
"HelveticaNeue", size: 12)!, NSForegroundColorAttributeName:
UIColor.blue, NSParagraphStyleAttributeName: rectangleStyle]
//文字属性集

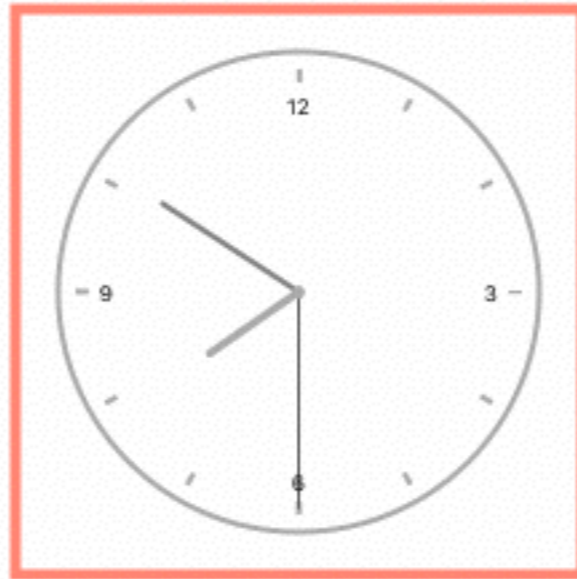
let textRect = UIEdgeInsetsInsetRect(self.bounds,
UIEdgeInsetsMake(90, 20, 90, 20))

"Hello\nWorld".draw(in:textRect, withAttributes: //绘制到矩形框内
rectangleFontAttributes)
```

Hello  
World

Demo

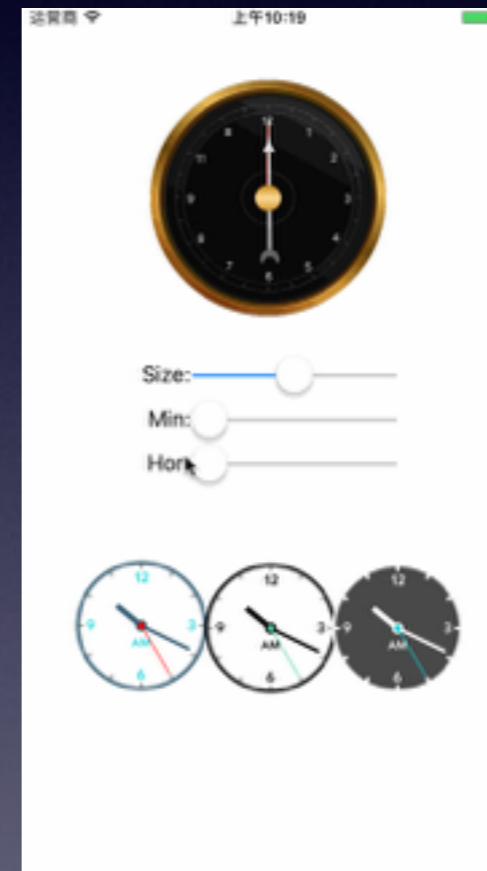
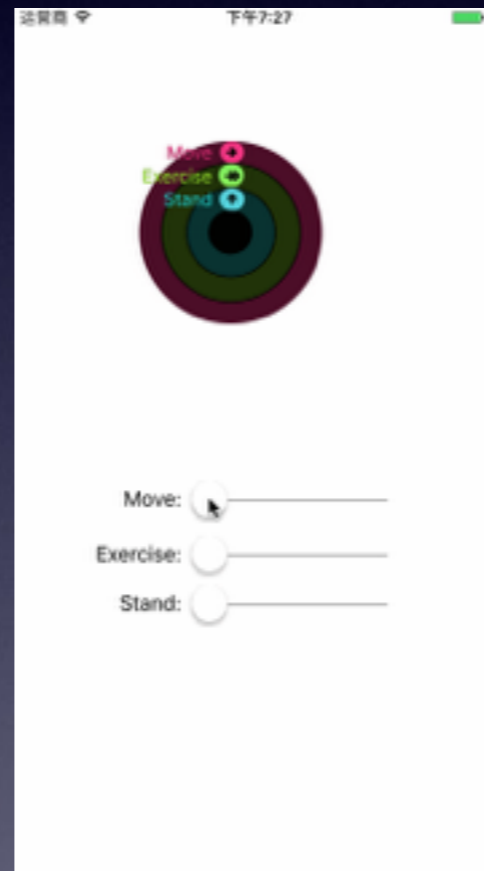
# 图形的绘制



1000个

# PaintCode & Sketch

sketchappsources & designcode



Thank You