The background is a light beige or cream color with a subtle texture. On the left side, there are several black ink splatters of varying sizes, some with fine lines radiating from them, creating a dynamic, artistic effect.

# 渐变色进度条

主讲:蒋玉龙

指导:张乐

# Quartz2D

- Quartz2D 是一个 二维视图 引擎.

可以完成的工作:

- ①. 绘制图形: 三角形\矩形\圆形等.
- ②. 绘制文字
- ③. 裁剪图片
- ...

平时的开发中, 为了是界面更为的美观, 但是有时候UI会比较复杂, UIKit框架中并实现出来, 这个时候就可以通过Quartz2D 来自定义一个UI控件.



# 开始绘图

- 那么, 现在开始绘图:  
先简单的来描述一下绘图:
  - ①. 首先你要拿一张纸(你要画的东西都是在这张纸上体现)
  - ②. 开始画图(设置你画图的描述路径)
  - ③. 提交你的成果

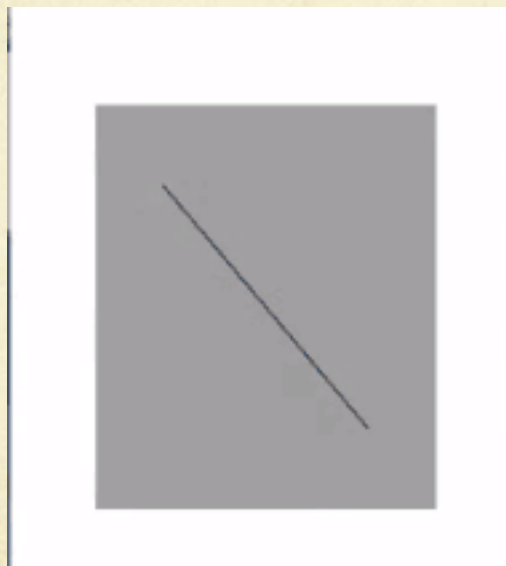
# 代码

```
- (void)drawRect:(CGRect)rect {  
    //1. 获取图形上下文  
    CGContextRef ctx = UIGraphicsGetCurrentContext();  
    //2. 描述路径  
    CGContextMoveToPoint(ctx, 50, 50);  
    CGContextAddLineToPoint(ctx, 100, 100);  
    //4. 渲染上下文  
    CGContextStrokePath(ctx);  
}
```



# 结果

- 简单的一条线



# 单色进度条

## ○ 思路：

从图形上来看，进度条也是一个线，只不过形状不一样。所以画进度条的画法步骤和画线的是差不多的。

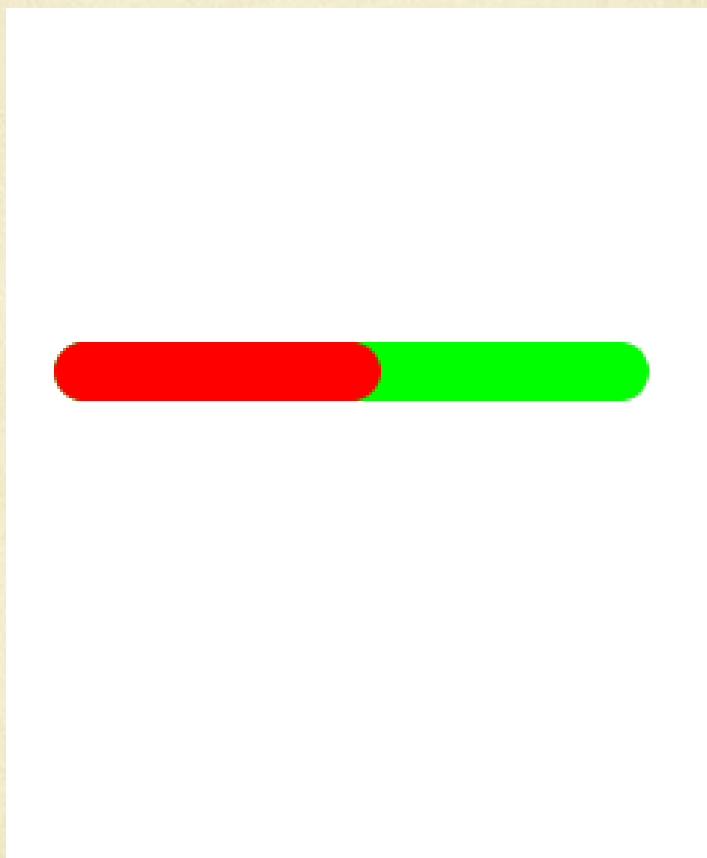
- ①. 先画一条线，设置线的大小和圆角
- ②. 设置进度条的填充色



# 代码

```
// 画纯色
- (void)drawSingleClrView:(CGContextRef)ref
    storeClr:(UIColor *)clr
    startPt:(CGPoint)startPoint
    endPt:(CGPoint)endPoint
    width:(CGFloat)width
{
    //设置起始位置
    CGContextMoveToPoint(ref, 5, 5);
    //设置结束位置
    CGContextAddLineToPoint(ref, 50, 5);
    //设置颜色
    [[UIColor greenColor] setStroke];
    //设置线的宽度
    CGContextSetLineWidth(ref, 10);
    //设置圆角
    CGContextSetLineCap(ref, kCGLineCapRound);
    CGContextDrawPath(ref, kCGPathStroke);
}
```

# 效果图

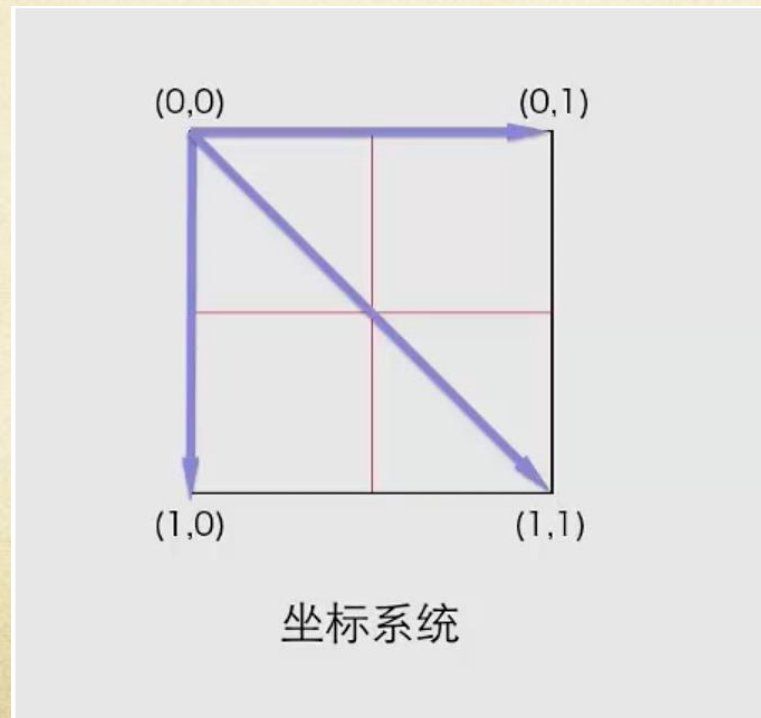




# 渐变梯度

## ○ CAGradientLayer:

- 1, CAGradientLayer的坐标系是从(0, 0)到(1, 1)绘制的矩形
- 2, CAGradientLayer的startPoint和endPoint会直接决定颜色的绘制方向
- 3, CAGradientLayer的颜色分割点时以0到1的比例来计算的



# 梯度代码

```
//初始化渐变层
self.gradientLayer = [CAGradientLayer layer];
self.gradientLayer.frame = imageView.bounds;
[imageView.layer addSublayer:self.gradientLayer];

//设置渐变颜色方向
self.gradientLayer.startPoint = CGPointMake(0, 0);
self.gradientLayer.endPoint = CGPointMake(0, 1);

//设定颜色组
self.gradientLayer.colors = @[(__bridge id)[UIColor clearColor].CGColor,
                              (__bridge id)[UIColor purpleColor].CGColor];

//设定颜色分割点
self.gradientLayer.locations = @[@(0.5f) ,@(1.0f)];
```



# 渐变色进度条

原理：画一个圆角矩形，设置填充颜色

- ①. 首先得创建一个颜色数组.
- ②. 用贝塞尔画出一个圆角矩形
- ③. 设置填充颜色

# 代码

```
NSArray *gradientColors = colors;
// 定义颜色渐变位置
CGFloat gradientLocations[] = {0.2, 0.5, 1};
// 定义色彩空间引用
CGColorSpaceRef colorSpace = CGColorSpaceCreateDeviceRGB();
// 定义渐变引用,并创建颜色渐变
CGGradientRef gradient = CGGradientCreateWithColors(colorSpace, (CFArrayRef)
    gradientColors, gradientLocations);
// 创建贝塞尔路径
UIBezierPath *path = [UIBezierPath bezierPathWithRoundedRect:CGRectMake(0, 0,
    progress, 10) cornerRadius:5];
CGContextSaveGState(ref);
// 添加颜色填充
[path fill];
// 添加剪切路径
[path addClip];
// 绘制线性渐变
CGContextDrawLinearGradient(ref, gradient, startPoint, endPoint, 0);
// 释放颜色空间
CGColorSpaceRelease(colorSpace);
// 释放渐变引用
CGGradientRelease(gradient);
```



# 效果图

