1. PIL模块（图片处理）：
   1. Image对象：
      1. im = Image.open(path).convert(‘RGBA’) 打开图片并转换为RGBA颜色空间,返回image对象
         1. im.format 图片来源，若不是从文件读取，则为None
         2. im.size 宽高二元组 im.mode 图片所使用的的像素格式
         3. im.show() 显示图片
         4. im.save(outpath, ‘jpeg’) 将im以jpeg格式保存在指定路径
         5. im.resize(size, resample=Image.ANTIALIAS) 重设图片大小
         6. im.thumbnail((width, height)) 缩略图生成，只能缩小不能放大，比例缩小到满足
         7. im.putalpha(im\_alpha) 把im的A像素空间用L模式的image对象im\_alpha替代
         8. im\_list = im.split() 此方法进行像素空间分离，返回值为各颜色空间image对象的列表
         9. im = im.point(lambda x:x\*0.5) 对image对象所有像素的所有值进行lambda运算
         10. im.paste(im2, position[, im2\_alpha]) 将im2[按指定透明度]画在im上
         11. new\_im = im.filter(滤镜参数) 对图形进行指定滤镜处理，返回image对象
             1. 滤镜参数：ImageFilter.xxx

.CONTOUR 轮廓获取

.EMBOSS 浮雕

.EDGE\_ENHANCE\_MORE 边界增强

.BLUR 模糊效果

* 1. Image类：
     1. Image.save(im[, ‘png’]) 将im[以png格式]保存在当前目录下
     2. im = Image.alpha\_composite(im1, im2) 将im2画在im1上面，考虑透明度，返回image对象，尺寸需相同
  2. ImageFont类：
     1. font = ImageFont.truetype(‘path’, int) 打开字体并设置大小为int，返回imagefont对象
     2. font.getsize(text) 获得指定文本的宽高度，返回宽高二元组
  3. ImageDraw类：
     1. draw = ImageDraw.Draw(im) 生成im的imagedraw类
     2. draw.text(position, text, font=font, fill=’red’) 在im上指定位置画出文字，fill为文字填充颜色
  4. ImageEnhance类：（亮度即为不透明度，0表示完全透明）
     1. brightness = ImageEnhance.Brightness(im\_alpha) 将L模式Image对象转换为Brightness对象，返回Brightness对象
     2. im\_alpha = brightness.enhance(float) 将brightness对象的亮度乘以float，float为0-1之间的1位小数，返回image对象
  5. 添加水印：
     1. watermark\_image = Image.new(‘RGBA’, im.size, (255, 255, 255, 0))
     2. imagedraw = ImageDraw.Draw(watermark\_image)
     3. position = (im.size[0]/2, im.size[1]/2)
     4. imagedraw.text(position, text, font = font, fill = ‘red’)
     5. alpha\_image = watermark\_image.split()[3] 此方法进行像素空间分离
     6. alpha\_image = ImageEnhance.Brightness(alpha\_image).enhance(0.5) 修改alpha图层
     7. watermark\_image.putalpha(alpha\_image) 将alpha图层替换
     8. Image.alpha\_composite(im, watermark\_image).save(img, ‘png’) 输出

1. MyQR模块：（个性二维码生成）：
   1. from MyQR import myqr
   2. myqr.run(words=’xxx’，picture=’xxx.png’，colorized=True，save\_name=’out.png’) 文本为xxx，底图为xxx.png，保留色彩输出
2. qrcode模块：（二维码生成）
   1. qr = qrcode.QRCode(version=1, error\_correction=qrcode.constants.ERROR\_CORRECT\_L, box\_size=10, border=4) 生成类
      1. version：1-40的整数，控制二维码模块数，1为21\*21，每一阶多4
      2. error\_correction：设置错误纠正能力
      3. box\_size：每个模块的边长占多少像素
      4. border：二维码与图片边界的距离（模块数）
   2. qr.clear() 清除数据
   3. qr.add\_data(data=’xx’) 加入指定数据
   4. qr.make(fit=True) 生成二维码
   5. img = qr.make\_image(fill\_color = ‘red’, back\_color=’white’) 生成二维码图片
   6. img.save(‘xxx.png’) 保存