

# day02

## 1. 绘制几何图形

### 1.1 绘制几何图形基本操作

```
"""
    绘制几何图形
    1. 线
        画布(图像) 两点 颜色 大小
    2. 矩形
        画布(图像) 左上角的点 右下角的点 颜色 大小
    3. 圆
        画布(图像) 圆心 半径 颜色 大小

    4. 多边形
        多个点

    """"
import cv2
import numpy as np

# 创建一个图像
img = cv2.imread('./media/sy.png')
# 参数1:画布
# 参数2: 起始点
# 参数3: 终点
# 参数4: 颜色
# 参数5: 线的大小
# cv2.line(img, (100, 100), (200, 200), color=(255, 0, 0), thickness=2)
# 绘制矩形
# 参数2: 左上角的点
# 参数3: 右下角的点
# 参数4: 颜色
# cv2.rectangle(img, (100, 100), (200, 200), color=(0, 255, 0), thickness=-1)
# cv2.rectangle(img, (100, 100), (200, 200), color=(0, 255, 0), thickness=2)

# 绘制圆
# cv2.circle(img, center=(150, 150), radius=50, color=(0, 0, 255), thickness=2)
# cv2.circle(img, center=(150, 150), radius=50, color=(0, 0, 255), thickness=-1)

# 绘制多边形
# pts = np.array([(50, 100), (80, 30), (150, 60), (200, 200)])
# cv2.polylines(img, [pts], isClosed=True, color=(0, 255, 255), thickness=2)
# cv2.fillPoly(img, [pts], color=(0, 255, 255))
# 绘制文字
# img, text, org, fontFace, fontScale, color, thickness=None
# 参数1: 绘制的地方(画布)
# 参数2: text 需要绘制的文字是字符串。
# 参数3: 绘制文字的位置。
# 参数4: 文字的字体
# 参数5: 文字的字体大小
# 参数6: 字体的颜色
# 参数7: 字体的粗细
```

```

cv2.putText(image, str(100), (500, 200), cv2.FONT_HERSHEY_SIMPLEX, 2, color=(0, 255, 255), thickness=2)

cv2.imshow('img', img)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

## 1.2 绘制几何图形案例

作业1：绘制线、矩形、圆

扩展

作业1：扩展作业能够看见绘制的过程

作业2：按下不同的按键 绘制不同的图形 l(线) r(矩形) c(圆)

```

"""
    简易画板
    鼠标
    位置 事件(按下去 弹起来 移动)

"""

import cv2

# 回调
st_p = (-1, -1)
end_p = (-1, -1)
img = cv2.imread('./media/sy.png')
WINDOWNAME = "board"

"""
    绘制直线
    确定起始点、终点
    鼠标弹起来之后进行显示(绘制)

    作业1：绘制线、矩形、圆
    扩展
    作业2：扩展作业能够看见绘制的过程
"""

def drawLine(event, x, y, flags, param):
    global st_p, end_p
    if event == cv2.EVENT_LBUTTONDOWN:
        st_p = (x, y)
        print(f'DOWN-event:{event} - x:{x} - y:{y}')

    if event == cv2.EVENT_LBUTTONUP:
        end_p = (x, y)
        print(f'UP-event:{event} - x:{x} - y:{y}')
        cv2.line(img, st_p, end_p, color=(0, 0, 255), thickness=2)
        cv2.imshow(WINDOWNAME, img)

def callback(event, x, y, flags, param):
    drawLine(event, x, y, flags, param)

```

```

# 简易画板
def mouseDrawing():
    cv2.namedWindow(WINDOWNAME, cv2.WINDOW_NORMAL)
    # cv2.resizeWindow('board', 1000, 600)
    # img = cv2.imread('./media/sy.png')
    # 参数1: windowName
    # 参数2: onMouse 函数
    # 参数3: param 参数
    # 将函数callback注册在窗口(board)上 我想要得到那些信息 event x y flags params
    cv2.setMouseCallback(WINDOWNAME, callback, 'hello')
    while True:
        key = cv2.waitKey(0)
        ...获取到鼠标按键 判断对应的按键

    cv2.imshow(WINDOWNAME, img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
    pass

mouseDrawing()

```

## 2.绘制人脸的矩形区域

```

"""
人脸识别

怎样才算一张脸：
眼睛、眉毛、鼻子、嘴巴...

级联分类器

"""
import cv2

class FaceDetect:
    def __init__(self):
        self.faceImg = cv2.imread('./media/sy.png')
        # 创建一个级联分类器 用来识别人脸使用
        self.classifier = cv2.CascadeClassifier()
        # 加载特征文件
        self.classifier.load('./model/haarcascade_frontalface_alt.xml')
        pass

    def faceDetect(self):
        faceImg = self.faceImg
        classifier = self.classifier
        # 识别图像中的人脸 返回矩形区域 使用list存储
        faceRects = classifier.detectMultiScale(faceImg)
        # [[365 187 276 276]] x y w h
        print(faceRects)
        # for rect in faceRects:
        #     pass
        for x, y, w, h in faceRects:

```

```
        cv2.rectangle(faceImg, (x, y), (x+w, y+h), color=(0, 0, 255),
thickness=2)
        pass
    cv2.imshow('faceImg', faceImg)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
    pass

detect = FaceDetect()
detect.faceDetect()
```