

《多媒体技术基础》2017 秋

作业 5 – 图像工具

实验目标：

- ✓ 掌握图像的绘制及其绘制过程中的位置、尺寸、裁剪效果的设定
- ✓ 掌握针对图像像素的获取、处理、赋值
- ✓ 掌握剪切区域 clipping region 的应用

提交截止：

根据作业发布页面提示

提交要求：

- ✓ 提交 2 个文件：1 实验报告（word 或 pdf 文档） + 2 源代码（文件夹打包上传）
- ✓ 实验报告命名：：姓名_作业 5
- ✓ 实验报告思路清晰、步骤清楚、要求有明确的分步结果；
- ✓ 源代码书写规范，可读性佳
- ✓ 严禁抄袭

代码要求：

根据 [HW5_模板.js](#) 文件做脚本的代码填充。代码中应包括的部分如下图，这些注释内容请在提交的.js 文件中保留；

```
// .....变量声明.....

// .....函数定义.....
// .....借用已有函数.....

// .....自定义函数.....

// .....事件响应：控件.....

// .....事件响应：鼠标.....

// .....事件响应：键盘.....

// .....初始化.....
```

大部分所需定义的函数已列在模板文件中，其余请根据需要自行定义。

实验任务：

根据课堂上的演示效果，完成如下实验任务：

任务一、实现如下的基础界面效果

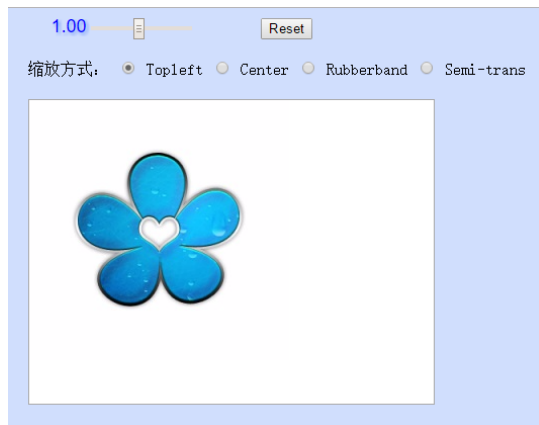


图 1

要求包括：

1. 1 个 canvas 元素：用于图像的绘制；设定任意区别于页面（body）的背景色；
2. 1 个滑动控制条（range）：用于图像缩放比例的设定；所设缩放值以文本显示在滑动条旁边；
3. 1 组缩放方式的单选按钮（radio）：用于图像的缩放方式的选择
单选按钮，意味着同组按钮中，只能有一个按钮被选中。示例用法如下，注意同组的 name 相同，value 值不同。

```
<div id='scaleTypeRadios'>
  缩放方式：
  <input type='radio' name='scaleType' value='topleft' checked/> Topleft
  <input type='radio' name='scaleType' value='center' /> Center
  <input type='radio' name='scaleType' value='rubberband' /> Rubberband
</div>
```

可利用 getElementById 获取整组按钮，当前所选中的按钮的值可由

`e.target.value` 返回；

可利用 getElementsByName 获取该组按钮的列表，其中某一按钮的选中与否可用如下语句设定：`scaleType[0].checked = true;`

4. 1 个重置按键（button）：按下后图像的绘制恢复初始状态（如图 1）

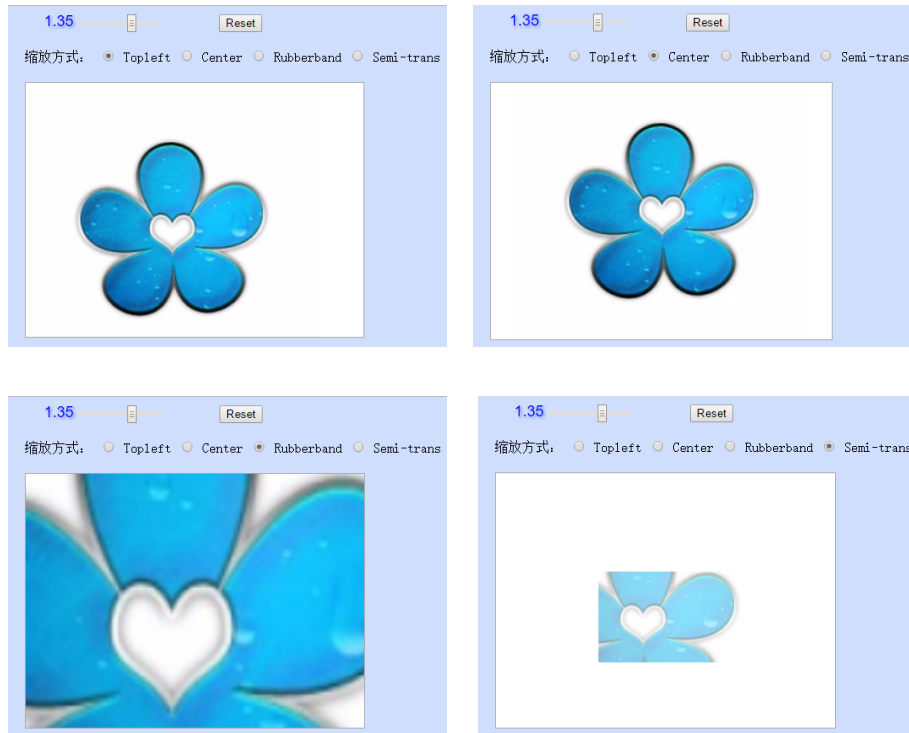
任务二、实现不同的图像操作效果

其中：

- a) Topleft 表示图像以 canvas 左上角为缩放中心；
- b) Center 表示图像以图像的中心点为缩放中心；

- c) Rubberband 表示将鼠标框选部分的图像拉伸到整个 canvas ;
d) Semi-trans 表示裁剪鼠标框选部分的图像 , 并做半透明效果。

不同模式下的操作效果如下图 :

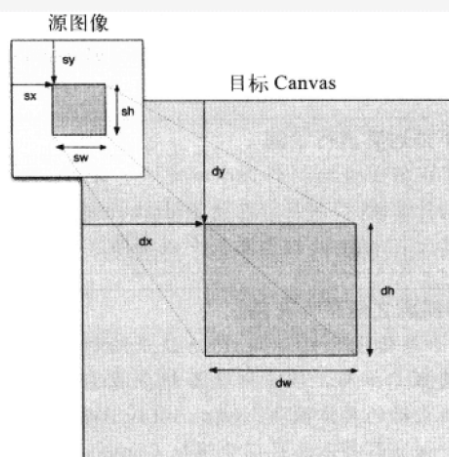


提示 :

- 1) 滑动条 + Topleft 模式 = 例 4-1
- 2) 滑动条 + Center 模式 = 例 4-1 , 但须修改缩放中心

图像的缩放效果可以利用 9 参数的.drawImage 方法实现 :

```
context.drawImage(img,sx,sy,width,height,x,y,width,height);
```



已知 Canvas 的宽和高分别是 W 和 H ,以缩放比例 S 进行缩放后变成 S*W 和 S*H。
 设缩放后图像的左上角位于坐标值 (x0, y0) 。以某一点(x, y)为缩放中心进行图像的缩

放，意味着该坐标点在目标 Canvas 和缩放后的图像中的相对位置是不变的，比如都是左上角，或者都是中心点之类。那么就可以确定所需的参数 $x_0, y_0, S*W, S*H$ 。

$$\frac{x - 0}{W} = \frac{x - x_0}{S * W}$$

说明：代码中所需的参数要求用通用公式表示，这样后期可以设定任意的坐标作为缩放中心，可用于实现“放大镜”例子的效果。

3) 鼠标操作 + Rubberband 模式 = 例 4-2，图像缩放可借助 offscreen canvas，或是剪辑区域实现。

说明 利用剪辑区域的方式实现，可用于实现“放大镜”例子中的局部放大效果。

4) 鼠标操作 + Semi-trans 模式 = 例 4-2，裁剪的效果可利用 getImageData 和 putImageData 实现；半透明效果的呈现可基于像素的修改。

任务三、（可选，加分任务）

在任务一、二基础上，可针对图像做任意功能及效果的扩展，从而实现一个更为丰富的“图像工具”。