

02

OPEN ORIENTED

凹凸实验室

# 前端实现滤镜效果

beidan

demo展示

# 一、filter

兼容性：

Current aligned	Usage relative	Date relative	Show all						
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49						
			56						
			57			9.2		4.4	
	414	52	58			10.2		4.4.4	
11	415	53	59	10.1	45	10.3	all	56	59
	416	54	60	11	46	11			
		55	61	TP	47				
		56	62						

# 一、filter

属性方法：

属性名	属性介绍	填写方式
none	无效果	默认值
grayscale	灰度	百分比
sepia	褐色	百分比
saturate	饱和度	百分比
hue-rotate	色相旋转	deg
invert	反色	百分比
opacity	透明度	百分比
brightness	亮度	百分比
contrast	对比度	百分比
blur	模糊	px



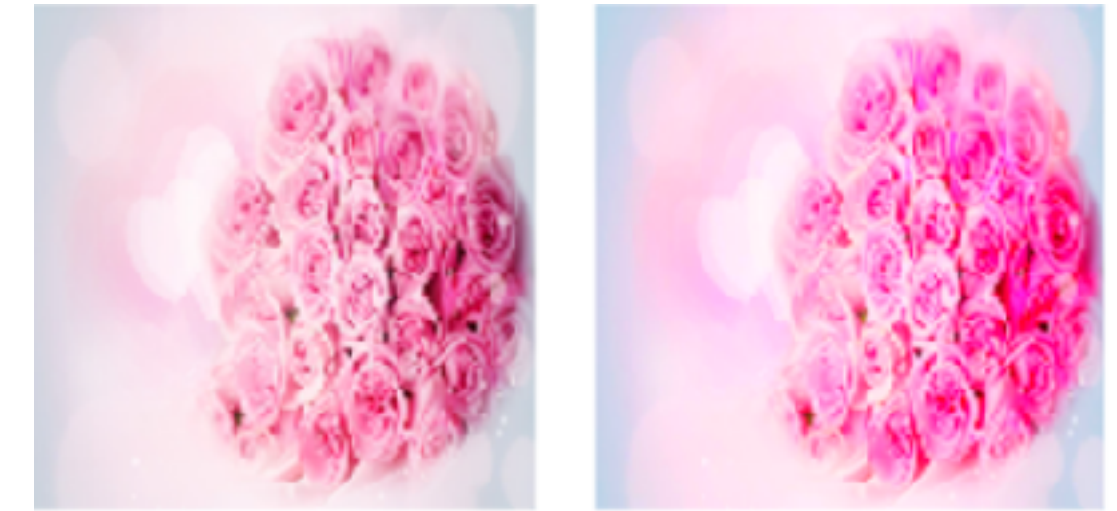
# —、filter



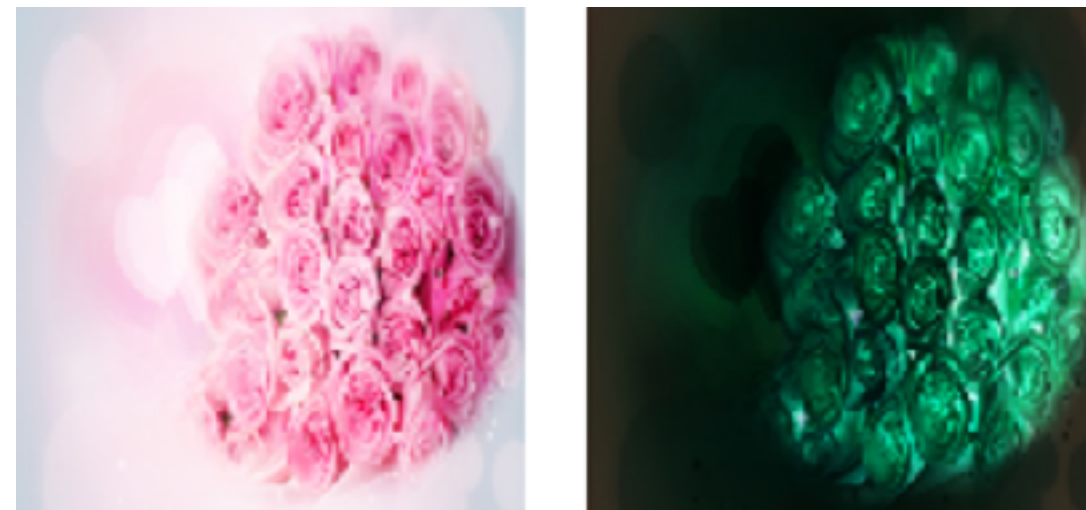
`-webkit-filter: grayscale(1);`



`-webkit-filter: sepia(1);`



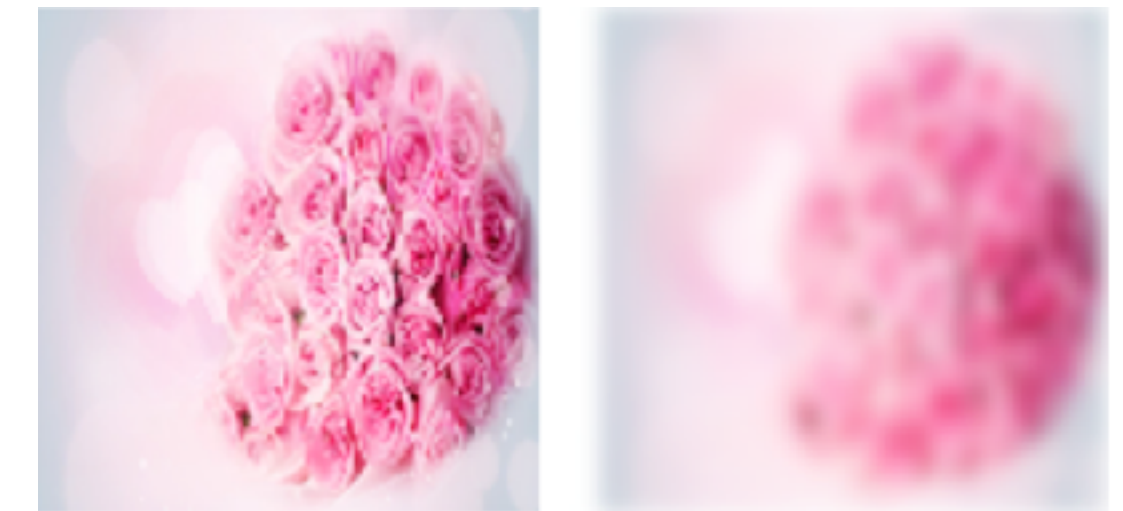
`--webkit-filter: saturate(5);`



`-webkit-filter: invert(1);`



`-webkit-filter: brightness(.5);`



`-webkit-filter: blur(2px);`

## 二、canvas

### 绘制图片

```
var canvas = document.getElementById("canvas"),
    context = canvas.getContext("2d"),
    image = new Image();
image.src = "timg.jpg";
image.onload = function () {
    context.drawImage(image, 0, 0, image.width,
        image.height, 0, 0, context.canvas.width,
        context.canvas.height);}
```

```
drawImage(image, sx, sy, sWidth, sHeight, dx, dy, dWidth, dHeight)
```

第1个参数image: 图片对象。

2,3,4,5个参数: 一个整体。代表了这张图片的(sx, sy)坐标开始, 取长为sWidth, sHeight的一部分图片。

6,7,8,9个参数: 一个整体。代表了将图片渲染到画布的(dx, dy)坐标, 渲染的实际长度为dWidth, 实际高度为dHeight。

## 二、canvas

### 图片的像素信息

```
var canvasData = context.getImageData(0, 0, 500, 300),  
    data = canvasData.data; // 获取 Canvas 画布上图片的像素信息。
```

																					0..19
																					20..39
																					40..59
																					60..79
																					80..99
																					100..119
																					120..139
																					140..159

图像像素是按照从左至右，从上到下的顺序依次排列的。

每一个像素信息包含4个值：红R(0~255)、绿G(0~255)、蓝B(0~255)、A透明度(0~255)。

[r,g,b,a,r,g,b,a]这样的按rgba顺序排列的单个的数组。

## 二、canvas

### 处理像素

```
for (i = 0; i <= data.length - 4; i += 4) {  
    average = (data[i] + data[i + 1] + data[i + 2]) / 3;  
    data[i] = average;  
    data[i + 1] = average;  
    data[i + 2] = average;  
}  
context.putImageData(canvasData, 0, 0);    //使用像素信息在 Canvas 画布上绘制图片。
```



## 二、canvas

### 底片效果

颜色反转的算法就是三原色求反，即  $255 - \text{原色}$

```
for (i = 0; i <= data.length - 4; i += 4) {  
    data[i] = 255 - data[i];  
    data[i + 1] = 255 - data[i + 1];  
    data[i + 2] = 255 - data[i + 2];  
}
```

## 二、canvas

### 黑白效果

求RGB平均值 $Avg = (R + G + B) / 3$ ，如果 $Avg \geq 100$ ，则新的颜色值为 $R = G = B = 255$ ；如果 $Avg < 100$ ，则新的颜色值为 $R = G = B = 0$ ；255就是白色，0就是黑色；

```
for (i = 0; i <= data.length - 4; i += 4) {  
    average = (data[i] + data[i + 1] + data[i + 2]) / 3;  
  
    (average >= 128) && (average = 255);  
    (average < 128) && (average = 0);  
  
    data[i] = average;  
    data[i + 1] = average;  
    data[i + 2] = average;  
}
```

## 二、canvas

### 浮雕效果

原理：当前点的值乘以2，加上128减去相邻点的值，然后减去下一行对应点的值；

由于图片中相邻点的颜色值是比较接近的，因此这样的算法处理之后，只有颜色的边沿区域，也就是相邻颜色差异较大的部分的结果才会比较明显，而其他平滑区域则值都接近128左右，也就是灰色，这样就具有了浮雕效果。

```
for (i = 0; i < length; i++) {
    if (i <= length - width * 4) {
        //跳过Alpha属性
        if ((i + 1) % 4 !== 0) {
            //判断是否在每行的最后一个像素
            if ((i + 4) % (width * 4) == 0) {
                data[i] = data[i - 4];
                data[i + 1] = data[i - 3];
                data[i + 2] = data[i - 2];
                data[i + 3] = data[i - 1];
                i += 4;
            }
            else {
                data[i] = 255 / 2 + 2 * data[i] - data[i + 4] - data[i + width * 4];
            }
        }
        else {
            //最后一行的情况
            if ((i + 1) % 4 !== 0) {
                data[i] = data[i - width * 4];
            }
        }
    }
}
```

## 二、canvas

### 灰度效果

计算公式:  $.299 * r + .587 * g + .114 * b$ ;

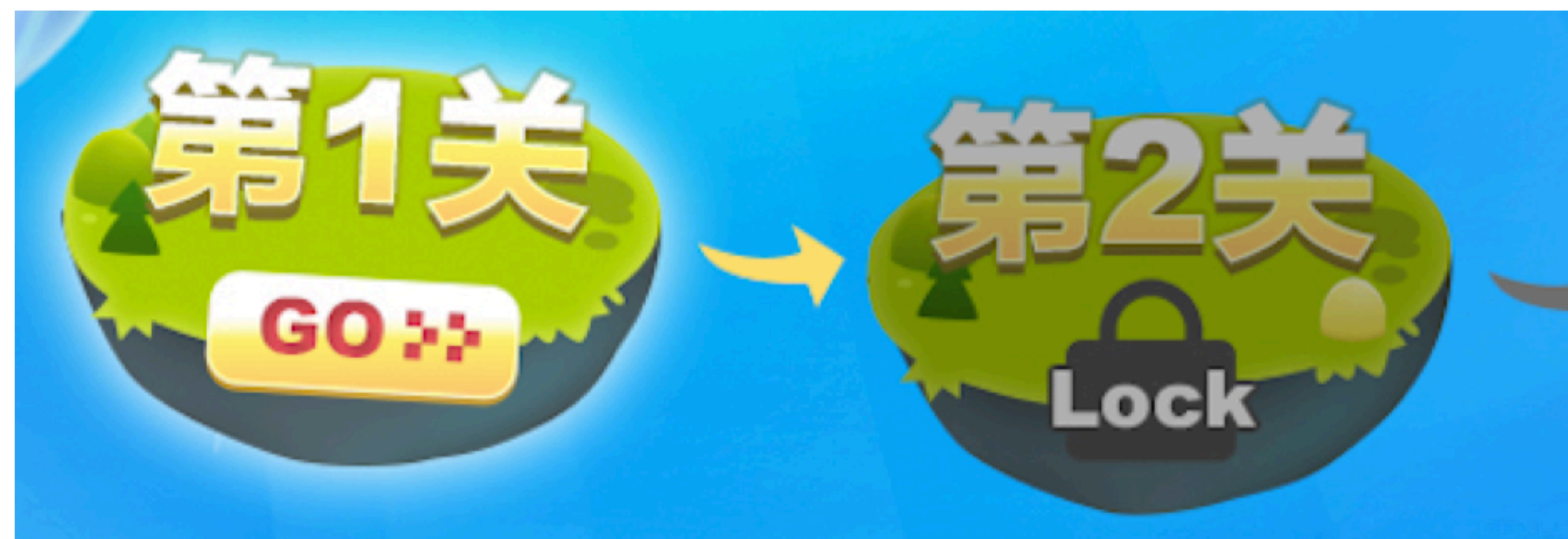
在RGB中,  $R = G = B$ 时便是灰色

```
for (i = 0; i <= data.length - 4; i += 4) {  
    average = data[i] * .299 + data[i + 1] * .587 + data[i + 2] * .114;  
    data[i] = average;  
    data[i + 1] = average;  
    data[i + 2] = average;  
}
```

### 三、实际应用



图片添加滤镜功能





## 四、更多图片像素处理

<https://wenku.baidu.com/view/275f9c4769eae009581bec56.html>

**T H A N K S**  
**FOR YOUR WATCHING**