canvas 像素级操作

李伟

课堂目标

1. 理解ImageData() 对象的获取方式。
2. 可以灵活操作像素。

知识点综述

1. ImageData() 对象
2. 在canvas 中显示ImageData
3. 操作像素

ImageData 是什么？

ImageData 是图片的数据化，它具备以下属性：

- data：Uint8ClampedArray [r,g,b,a, r,g,b,a, r,g,b,a, r,g,b,a]
- width：整数
- heidth：整数

注：Uint8ClampedArray 翻译过来是 **8位无符号整型固定数组**，其取值范围是[0,255]。若小于0，则为0，大于255，则为255。若为小数，则取整，取整的方法是银行家舍入。

怎么拿到 ImageData() 对象？

1. 直接建立ImageData() 对象（相当于自己新建了一张图片）。

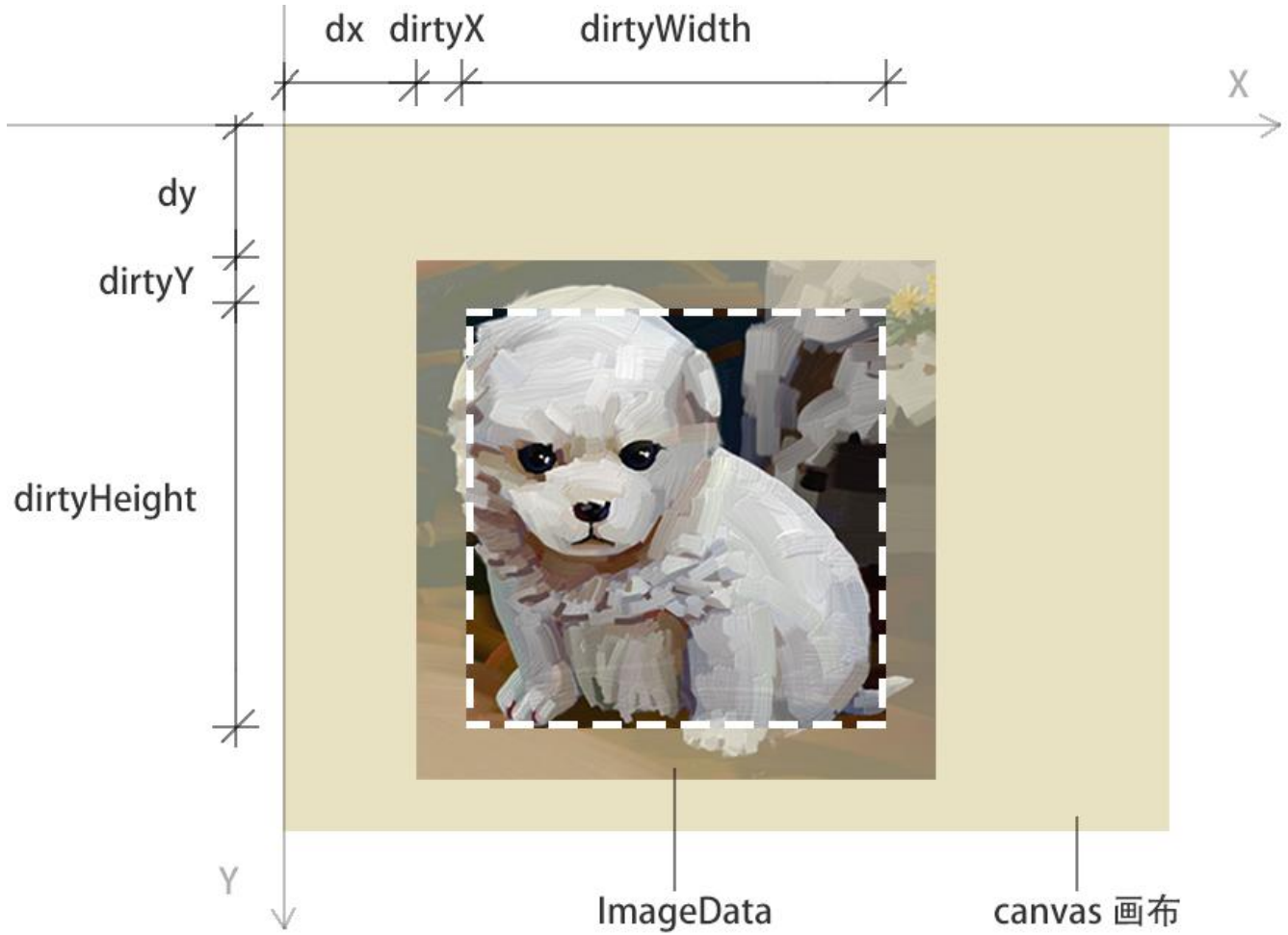
- new ImageData()
  - new ImageData(width, heidth)
  - new ImageData(Uint8ClampedArray, width, heidth)
- ctx.createImageData()
  - ctx.createImageData(width, heidth)
  - ctx.createImageData(ImageData)

2. 获取canvas 的ImageData() 对象（可以以此原理获取真实图片的数据）

- ctx.getImageData(x, y, width, height)

拿到 ImageData 后可以干什么？

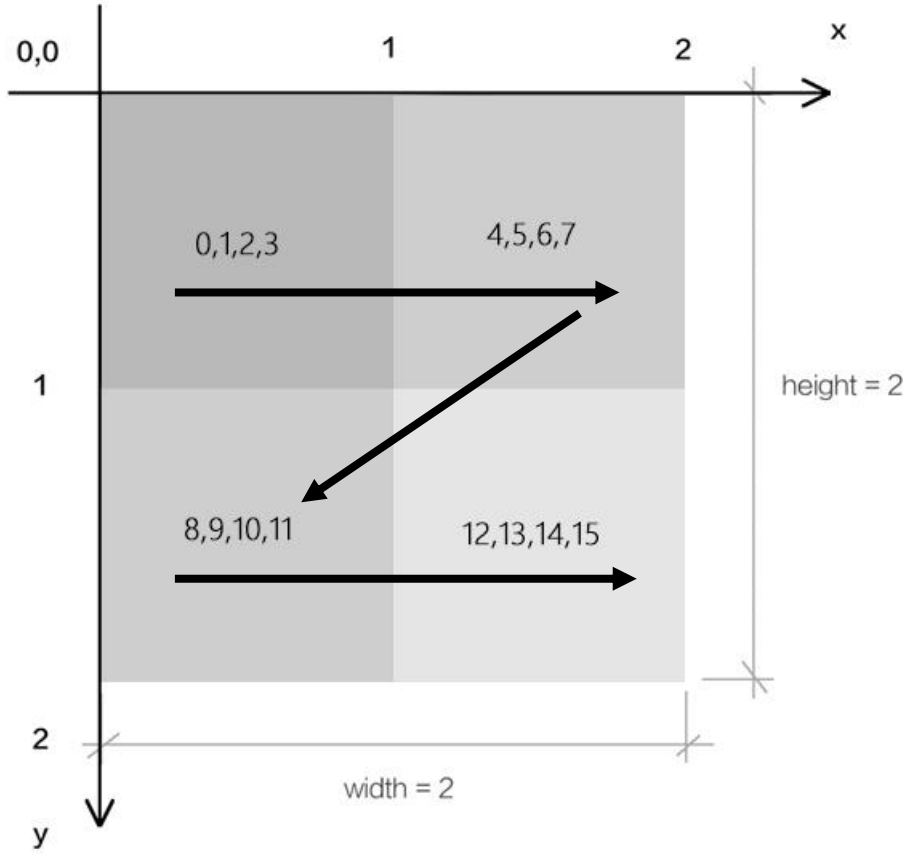
- 在canvas 中显示ImageData：putImageData(ImageData, dx, dy, dirtyX, dirtyY, dirtyWidth, dirtyHeight)



理解 ImageData 中的像素集合和图像栅格的对应关系

- ImageData 对象的属性：
  - data：Uint8ClampedArray [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
  - width：2
  - heidth：2

- data 中的数值与栅格的对应关系：
  - 先从第一行开始，遍历每一列，遍历完成后
  - 进入下一行，遍历每一列，遍历完成后
  - 进入下一行，遍历每一列，遍历完成后
  - 进入下一行，遍历每一列，遍历完成后
  - ……
  - 直到遍历完最后一行的最后一列。



遍历像素的方法

- 像素遍历：每隔4 个数据遍历一次，简单快捷
- 行列遍历：基于行列遍历，可获取像素点的位置信息

逐像素遍历

```
for(let
i=0;i<arr.length;i+=4){
        let r=data[i+0];
        let g=data[i+1];
        let b=data[i+2];
        let a=data[i+3];

console.log(r,g,b,a)
}
```

行列遍历

```
for(let y=0;y<h;y++){
      for(let x=0;x<w;x++){
           let ind=(y*w+x)*4;
           let r=data[ind];
           let g=data[ind+1];
           let b=data[ind+2];
           let a=data[ind+3];
           console.log(r,g,b,a)
       }
   }
```

图像置灰

- 要点：灰度算法 const lm =0.299*r+0.587*g+0.114*b ;

马赛克效果

- 要点：获取一区域的像素颜色，然后将此颜色赋给此区域的所有像素。

ImageData 的功能非常强大

我可以通过不同的算法，对ImageData 中的像素进行不同的处理。比如调整图片的色调，检测图像边缘，实现艺术效果，人脸识别……

原图　　边界检测　　Low Polly 风格