



{x:0, y:40} rgb(255, 255, 255)

◀

Objects

X

M

Properties

Menu Graph

CanvasRenderingConte...

HTMLImageElement

Graph

canvas: HTMLCanvasEl...

context: CanvasRende...

image: HTMLImageElement

W: 192

H: 192

Menu

Object

Methods ☒ sort

Graph

getPixelAt

grayscale

invert

loadImage

mirrorEffectHorizontal

mirrorEffectReversed

mirrorEffectVertical

Menu

allKeysIn

allValuesOf

[object Menu] Graph — 5 properties and 11 methods

Ref: [Pixel manipulation](#)



{x:0, y:40} rgb(255, 255, 255)

◀

Objects

X

M

Properties

Menu Graph

CanvasRenderingConte...

HTMLImageElement

Graph

canvas: HTMLCanvasEl...

context: CanvasRende...

image: HTMLImageElement

W: 192

H: 192

Menu

Object

Methods ☒ sort

Graph

getPixelAt

grayscale

invert

loadImage

mirrorEffectHorizontal

mirrorEffectReversed

mirrorEffectVertical


Menu

allKeysIn

allValuesOf

_.mirrorEffectHorizontal()

Ref: [Pixel manipulation](#)



{x,y} {R,G,B}

Ref: [Pixel manipulation](#)

Menu Graph

CanvasRenderingConte...

HTMLImageElement

Graph

canvas: HTMLCanvasEl...

context: CanvasRende...

image: HTMLImageElement

W: 192

H: 192

Menu

Object

Methods ☒ sort

Graph

getPixelAt

grayscale

invert

loadImage

mirrorEffectHorizontal

mirrorEffectReversed

mirrorEffectVertical

Menu

allKeysIn

allValuesOf

[object Menu] Graph — 5 properties and 11 methods

Sample code

```

function mirrorEffectVertical1() {
  let id = this.context.getImageData(0, 0, this.W, this.H);
  let w=this.W;
  let h=this.H;
  console.log(w,h)
  let data = id.data;
  let id2 = this.context.getImageData(0, 0, this.W, this.H);
  let tempData = id2.data;

  for (let i = 0; i < w; i += 1) {
    for (let j = 0; j < h; j += 1) {

      data[w*i*4+4*j] = tempData[w*(w-i)*4+4*(j)] // red
      data[w*i*4+4*j+1] = tempData[w*(w-i)*4+(j)+1] // green
      data[w*i*4+4*j+2] = tempData[w*(w-i)*4+(j)+2] // blue
      data[w*i*4+4*j+3] = 255

    }
  }
  this.context.putImageData(id, 0, 0);
}

```