

# pixels

An array containing the color of each pixel in the image.

Colors are stored as numbers representing red, green, blue, and alpha (RGBA) values. `img.pixels` is a one-dimensional array for performance reasons.

Each pixel occupies four elements in the pixels array, one for each RGBA value. For example, the pixel at coordinates (0, 0) stores its RGBA values at `img.pixels[0]`, `img.pixels[1]`, `img.pixels[2]`, and `img.pixels[3]`, respectively. The next pixel at coordinates (1, 0) stores its RGBA values at `img.pixels[4]`, `img.pixels[5]`, `img.pixels[6]`, and `img.pixels[7]`. And so on. The `img.pixels` array for a 100×100 `p5.Image` object has  $100 \times 100 \times 4 = 40,000$  elements.

Accessing the RGBA values for a pixel in the image requires a little math as shown in the examples below. The `img.loadPixels()` method must be called before accessing the `img.pixels` array. The `img.updatePixels()` method must be called after any changes are made.

## Examples



```
function setup() {
  createCanvas(100, 100);

  background(200);

  // Create a p5.Image object.
  let img = createImage(66, 66);

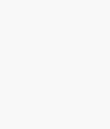
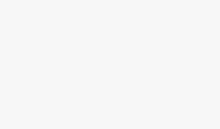
  // Load the image's pixels.
  img.loadPixels();

  for (let i = 0; i < img.pixels.length; i += 4) {
    // Red.
    img.pixels[i] = 0;
    // Green.
    img.pixels[i + 1] = 0;
    // Blue.
    img.pixels[i + 2] = 0;
    // Alpha.
    img.pixels[i + 3] = 255;
  }

  // Update the image.
  img.updatePixels();

  // Display the image.
  image(img, 17, 17);

  describe('A black square drawn in the middle of a gray square.');
}
```



```
function setup() {
  createCanvas(100, 100);

  background(200);

  // Create a p5.Image object.
  let img = createImage(66, 66);

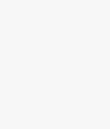
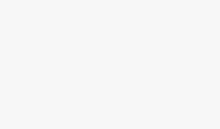
  // Load the image's pixels.
  img.loadPixels();

  // Set the pixels to red.
  for (let i = 0; i < img.pixels.length; i += 4) {
    // Red.
    img.pixels[i] = 255;
    // Green.
    img.pixels[i + 1] = 0;
    // Blue.
    img.pixels[i + 2] = 0;
    // Alpha.
    img.pixels[i + 3] = 255;
  }

  // Update the image.
  img.updatePixels();

  // Display the image.
  image(img, 17, 17);

  describe('A red square drawn in the middle of a gray square.');
}
```



## Related References

### blend

Copies a region of pixels from another image into this one.

### copy

Copies pixels from a source image to this image.

### delay

Changes the delay between frames in an animated GIF.

### filter

Applies an image filter to the image.

