

# createImage()

Creates a new `p5.Image` object.

`createImage()` uses the `width` and `height` parameters to set the new `p5.Image` object's dimensions in pixels. The new `p5.Image` can be modified by updating its `pixels` array or by calling its `get()` and `set()` methods. The `loadPixels()` method must be called before reading or modifying pixel values. The `updatePixels()` method must be called for updates to take effect.

Note: The new `p5.Image` object is transparent by default.

## Examples



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

  background(200);

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

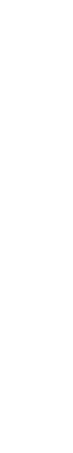
  // Load the image's pixels into memory.
  img.loadPixels();

  // Set all the image's pixels to black.
  for (let x = 0; x < img.width; x += 1) {
    for (let y = 0; y < img.height; y += 1) {
      img.set(x, y, 0);
    }
  }

  // Update the image's pixel values.
  img.updatePixels();

  // Draw 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 into memory.
  img.loadPixels();

  // Create a color gradient.
  for (let x = 0; x < img.width; x += 1) {
    for (let y = 0; y < img.height; y += 1) {
      // Calculate the transparency.
      let a = map(x, 0, img.width, 0, 255);

      // Create a p5.Color object.
      let c = color(0, a);

      // Set the pixel's color.
      img.set(x, y, c);
    }
  }

  // Update the image's pixels.
  img.updatePixels();

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

  describe('A square with a horizontal color gradient that transitions from gray to black.');
}
```



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

  background(200);

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

  // Load the pixels into memory.
  img.loadPixels();
  // Get the current pixel density.
  let d = pixelDensity();

  // Calculate the pixel that is halfway through the image's pixel array.
  let halfImage = 4 * (d * img.width) * (d * img.height / 2);

  // Set half of the image's pixels to black.
  for (let i = 0; i < halfImage; 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's pixels.
}
```

## Syntax

```
createImage(width, height)
```

### Parameters

`width` Integer: width in pixels.  
`height` Integer: height in pixels.

### Returns

`p5.Image`: new `p5.Image` object.

This page is generated from the comments in `src/image/image.js`. Please feel free to edit it and submit a pull request!

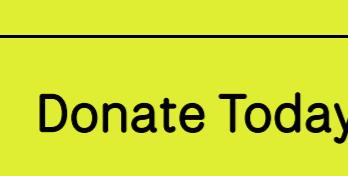
## Related References

`blend` Images into this one from another

`copy` Copies pixels from a source image to

`delay` Changes the delay between frames in

`filter` Applies an image filter to the image.



Donate Today! Support p5.js and the Processing Foundation.

[Start Coding](#) [Donate](#)

Socials

[GitHub](#)

[Instagram](#)

[YouTube](#)

[Forum](#)