

# updatePixels()

Updates the canvas with the RGBA values in the `pixels` array.

`updatePixels()` only needs to be called after changing values in the `pixels` array. Such changes can be made directly after calling `loadPixels()` or by calling `set()`.

## Examples



```
let img;

// Load the image.
function preload() {
  img = loadImage('/assets/rockies.jpg');
}

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

  // Display the image.
  image(img, 0, 0, 100, 100);

  // Get the pixel density.
  let d = pixelDensity();

  // Calculate the halfway index in the pixels array.
  let halfImage = 4 * (d * width) * (d * height / 2);

  // Load the pixels array.
  loadPixels();

  // Copy the top half of the canvas to the bottom.
  for (let i = 0; i < halfImage; i += 1) {
    pixels[i + halfImage] = pixels[i];
  }

  // Update the canvas.
  updatePixels();

  describe('Two identical images of mountain landscapes, one on top of the other.');
}
```



## Syntax

```
updatePixels([x], [y], [w], [h])
```



## Parameters

x	Number: x-coordinate of the upper-left corner of region to update.
y	Number: y-coordinate of the upper-left corner of region to update.
w	Number: width of region to update.
h	Number: height of region to update.

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

## Related References

**blend**  
Copies a region of pixels from one image to another.

**copy**  
Copies pixels from a source image to a region of the canvas.

**filter**  
Applies an image filter to the canvas.

**get**  
Gets a pixel or a region of pixels from the canvas.

