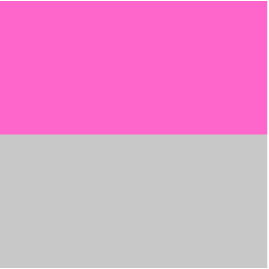


# updatePixels()

Updates the framebuffer with the RGBA values in the **pixels** array.

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

## Examples



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

  background(200);

  // Create a p5.Framebuffer object.
  let myBuffer = createFramebuffer();

  // Load the pixels array.
  myBuffer.loadPixels();

  // Get the number of pixels in the
  // top half of the framebuffer.
  let numPixels = myBuffer.pixels.length / 2;

  // Set the framebuffer's top half to pink.
  for (let i = 0; i < numPixels; i += 4) {
    myBuffer.pixels[i] = 255;
    myBuffer.pixels[i + 1] = 102;
    myBuffer.pixels[i + 2] = 204;
    myBuffer.pixels[i + 3] = 255;
  }

  // Update the pixels array.
  myBuffer.updatePixels();

  // Draw the p5.Framebuffer object to the canvas.
```

This page is generated from the comments in [src/webgl/p5.Framebuffer.js](#) . Please feel free to edit it and submit a pull request!

## Related References

### autoSized

Toggles the framebuffer's autosizing mode or returns the current mode.

### begin

Begins drawing shapes to the framebuffer.

### color

An object that stores the framebuffer's color data.

### createCamera

Creates a new p5.Camera object to use with the framebuffer.

### p5.js

### Resources

- Reference
- Tutorials
- Examples
- Contribute
- Community
- About
- Start Coding
- Donate

### Information

- Download
- Contact
- Copyright
- Privacy Policy
- Terms of Use

### Socials

- GitHub ↗
- Instagram ↗
- X ↗
- YouTube ↗
- Discord ↗
- Forum ↗

