

remove()

Deletes the framebuffer from GPU memory.

Calling `myBuffer.remove()` frees the GPU memory used by the framebuffer. The framebuffer also uses a bit of memory on the CPU which can be freed like so:

```
// Delete the framebuffer from GPU memory.
myBuffer.remove();
```

```
// Delete the framebuffer from CPU memory.
myBuffer = undefined;
```

Note: All variables that reference the framebuffer must be assigned the value `undefined` to delete the framebuffer from CPU memory. If any variable still refers to the framebuffer, then it won't be garbage collected.

Examples



```
// Double-click to remove the p5.Framebuffer object.

let myBuffer;

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

  // Create an options object.
  let options = { width: 60, height: 60 };

  // Create a p5.Framebuffer object and
  // configure it using options.
  myBuffer = createFramebuffer(options);

  describe('A white circle at the center of a dark gray square
disappears when the user double-clicks.');
}

function draw() {
  background(200);

  // Display the p5.Framebuffer object if
  // it's available.
  if (myBuffer) {
    // Draw to the p5.Framebuffer object.
    myBuffer.begin();
    background(100);
    circle(0, 0, 20);
  }
}
```

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.

