

# draw()

Draws to the framebuffer by calling a function that contains drawing instructions.

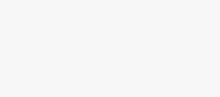
The parameter, `callback`, is a function with the drawing instructions for the framebuffer. For example, calling `myBuffer.draw(myFunction)` will call a function named `myFunction()` to draw to the framebuffer. Doing so has the same effect as the following:

```
myBuffer.begin();
myFunction();
myBuffer.end();
```

## Examples



// Click the canvas to display the framebuffer.



```
let myBuffer;

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

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

  describe('An empty gray canvas. The canvas gets darker and a rotating, multicolor torus appears while the user presses and holds the mouse.');
}

function draw() {
  background(200);

  // Draw to the p5.Framebuffer object.
  myBuffer.draw(bagel);

  // Display the p5.Framebuffer object while
  // the user presses the mouse.
  if (mouseIsPressed === true) {
    image(myBuffer, -50, -50);
  }
}

// Draw a rotating, multicolor torus.
function bagel() {
```

## Syntax

`draw(callback)`



## Parameters

`callback` Function: function that draws to the framebuffer.

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.

