createGraphics()

Creates a p5.Graphics object.

`createGraphics()` creates an offscreen drawing canvas (graphics buffer) and returns it as a p5.Graphics object. Drawing to a separate graphics buffer can be helpful for performance and for organizing code.

The first two parameters, `width` and `height`, are optional. They set the dimensions of the p5.Graphics object. For example, calling `createGraphics(900, 500)` creates a graphics buffer that's 900×500 pixels.

The third parameter is also optional. If either of the constants `P2D` or `WEBGL` is passed, as in `createGraphics(900, 500, WEBGL)`, then it will set the p5.Graphics object's rendering mode. If an existing HTMLCanvasElement is passed, as in `createGraphics(900, 500, myCanvas)`, then it will be used by the graphics buffer.

The fourth parameter is also optional. If an existing HTMLCanvasElement is passed, as in `createGraphics(900, 500, WEBGL, myCanvas)`, then it will be used by the graphics buffer.

Note: In WebGL mode, the p5.Graphics object will use a WebGL2 context if it's supported by the browser. Check the webglVersion system variable to check what version is being used, or call `setAttributes({ version: 1 })` to create a WebGL1 context.

Examples

```
//  Double-click to draw the contents of the graphics buffer.

let pg;

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

  background(180);

  // Create the p5.Graphics object.
  pg = createGraphics(50, 50);

  // Draw to the graphics buffer.
  pg.background(100);
  pg.circle(pg.width / 2, pg.height / 2, 20);

  describe('A gray square. A smaller, darker square with a white circle at its center appears when the user double-clicks.');
}

// Display the graphics buffer when the user double-clicks.
function doubleClicked() {
  if (mouseX > 0 && mouseX < 100 && mouseY > 0 && mouseY < 100) {
    image(pg, 25, 25);
  }
}
```

```
//  Double-click to draw the contents of the graphics buffer.

let pg;

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

  background(180);

  // Create the p5.Graphics object in WebGL mode.
  pg = createGraphics(50, 50, WEBGL);

  // Draw to the graphics buffer.
  pg.background(100);
  pg.lights();
  pg.noStroke();
  pg.rotateX(QUARTER_PI);
  pg.rotateY(QUARTER_PI);
  pg.torus(15, 5);

  describe('A gray square. A smaller, darker square with a white torus at its center appears when the user double-clicks.');
}

// Display the graphics buffer when the user double-clicks.
function doubleClicked() {
  if (mouseX > 0 && mouseX < 100 && mouseY > 0 && mouseY < 100) {
```

Syntax

```
createGraphics(width, height, [renderer], [canvas])
```

```
createGraphics(width, height, [canvas])
```

Parameters

width	Number: width of the graphics buffer.
height	Number: height of the graphics buffer.
renderer	Constant: either P2D or WEBGL. Defaults to P2D.
canvas	HTMLCanvasElement: existing canvas element that should be used for the graphics buffer..

Returns

p5.Graphics: new graphics buffer.

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

Related References

createFramebuffer
Creates a new p5.Framebuffer object with the same WebGL context as the graphics buffer.

remove
Removes the graphics buffer from the web page.

reset
Resets the graphics buffer's transformations and lighting.

blendMode
Sets the way colors blend when added to the canvas.