clip()

Defines a shape that will mask any shapes drawn afterward.

The first parameter, `callback`, is a function that defines the mask. Any shapes drawn in `callback` will add to the mask shape. The mask will apply to anything drawn after `clip()` is called.

The second parameter, `options`, is optional. If an object with an `invert` property is passed, as in `beginClip({ invert: true })`, it will be used to set the masking mode. `{ invert: true }` inverts the mask, creating holes in shapes that are masked. `invert` is `false` by default.

Masks can be contained between the push() and pop() functions. Doing so allows unmasked shapes to be drawn after masked shapes.

Masks can also be defined with beginClip() and endClip().

Examples

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

  background(200);

  // Create a mask.
  clip(mask);

  // Draw a backing shape.
  square(5, 5, 45);

  describe('A white triangle and circle on a gray
background.');
}

// Declare a function that defines the mask.
function mask() {
  triangle(15, 37, 30, 13, 43, 37);
  circle(45, 45, 7);
}
```

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

  background(200);

  // Create an inverted mask.
  clip(mask, { invert: true });

  // Draw a backing shape.
  square(5, 5, 45);

  describe('A white square at the top-left corner of a gray
square. The white square has a triangle and a circle cut out
of it.');
}

// Declare a function that defines the mask.
function mask() {
  triangle(15, 37, 30, 13, 43, 37);
  circle(45, 45, 7);
}
```

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

  background(200);

  noStroke();

  // Draw a masked shape.
  push();
  // Create a mask.
  clip(mask);

  // Draw a backing shape.
  square(5, 5, 45);
  pop();

  // Translate the origin to the center.
  translate(50, 50);

  // Draw an inverted masked shape.
  push();
  // Create an inverted mask.
  clip(mask, { invert: true });

  // Draw a backing shape.
  square(5, 5, 45);
  pop();

  describe('In the top left, a white triangle and circle. In
the bottom right, a white square with a triangle and circle
cut out of it.');
}

// Declare a function that defines the mask.
```

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

  describe('A silhouette of a rotating torus colored
fuchsia.');
}

function draw() {
  background(200);

  // Create a mask.
  clip(mask);

  // Draw a backing shape.
  noStroke();
  fill('fuchsia');
  plane(100);
}

// Declare a function that defines the mask.
function mask() {
  push();
  rotateX(frameCount * 0.01);
  rotateY(frameCount * 0.01);
  scale(0.5);
  torus(30, 15);
  pop();
}
```

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

  describe('A silhouette of a rotating torus colored with a
gradient from cyan to purple.');
}

function draw() {
  background(200);

  // Create a mask.
  clip(mask);

  // Draw a backing shape.
  noStroke();
  beginShape(QUAD_STRIP);
  fill(0, 255, 255);
  vertex(-width / 2, -height / 2);
  vertex(width / 2, -height / 2);
  fill(100, 0, 100);
  vertex(-width / 2, height / 2);
  vertex(width / 2, height / 2);
  endShape();
}

// Declare a function that defines the mask.
function mask() {
  push();
  rotateX(frameCount * 0.01);
  rotateY(frameCount * 0.01);
```

Syntax

```
clip(callback, [options])
```

Parameters

callback	Function: a function that draws the mask shape.
options	Object: an object containing clip settings.

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

Related References

background
Sets the color used for the background of the canvas.

beginClip
Starts defining a shape that will mask any shapes drawn afterward.

clear
Clears the pixels on the canvas.

clip
Defines a shape that will mask any shapes drawn afterward.