rotate()

Rotates the coordinate system.

By default, the positive x-axis points to the right and the positive y-axis points downward. The `rotate()` function changes this orientation by rotating the coordinate system about the origin. Everything drawn after `rotate()` is called will appear to be rotated.

The first parameter, `angle`, is the amount to rotate. For example, calling `rotate(1)` rotates the coordinate system clockwise 1 radian which is nearly 57°. `rotate()` interprets angle values using the current angleMode().

The second parameter, `axis`, is optional. It's used to orient 3D rotations in WebGL mode. If a p5.Vector is passed, as in `rotate(QUARTER_PI, myVector)`, then the coordinate system will rotate `QUARTER_PI` radians about `myVector`. If an array of vector components is passed, as in `rotate(QUARTER_PI, [1, 0, 0])`, then the coordinate system will rotate `QUARTER_PI` radians about a vector with the components `[1, 0, 0]`.

By default, transformations accumulate. For example, calling `rotate(1)` twice has the same effect as calling `rotate(2)` once. The push() and pop() functions can be used to isolate transformations within distinct drawing groups.

Note: Transformations are reset at the beginning of the draw loop. Calling `rotate(1)` inside the draw() function won't cause shapes to spin.

Examples

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

  describe(
    "A white rectangle on a gray background. The rectangle's long axis runs from top-left to bottom-right."
  );
}

function draw() {
  background(200);

  // Rotate the coordinate system 1/8 turn.
  rotate(QUARTER_PI);

  // Draw a rectangle at coordinates (50, 0).
  rect(50, 0, 40, 20);
}
```

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

  describe(
    "A white rectangle on a gray background. The rectangle's long axis runs from top-left to bottom-right."
  );
}

function draw() {
  background(200);

  // Rotate the coordinate system 1/16 turn.
  rotate(QUARTER_PI / 2);

  // Rotate the coordinate system another 1/16 turn.
  rotate(QUARTER_PI / 2);

  // Draw a rectangle at coordinates (50, 0).
  rect(50, 0, 40, 20);
}
```

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

  // Use degrees.
  angleMode(DEGREES);

  describe(
    "A white rectangle on a gray background. The rectangle's long axis runs from top-left to bottom-right."
  );
}

function draw() {
  background(200);

  // Rotate the coordinate system 1/8 turn.
  rotate(45);

  // Draw a rectangle at coordinates (50, 0).
  rect(50, 0, 40, 20);
}
```

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

  describe(
    'A white rectangle on a gray background. The rectangle rotates slowly about the top-left corner. It disappears and reappears periodically.'
  );
}

function draw() {
  background(200);

  // Rotate the coordinate system a little more each frame.
  let angle = frameCount * 0.01;
  rotate(angle);

  // Draw a rectangle at coordinates (50, 0).
  rect(50, 0, 40, 20);
}
```

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

  describe("A cube on a gray background. The cube's front face points to the top-right.");
}

function draw() {
  background(200);

  // Rotate the coordinate system 1/8 turn about
  // the axis [1, 1, 0].
  let axis = createVector(1, 1, 0);
  rotate(QUARTER_PI, axis);

  // Draw a box.
  box();
}
```

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

  describe("A cube on a gray background. The cube's front face points to the top-right.");
}

function draw() {
  background(200);

  // Rotate the coordinate system 1/8 turn about
  // the axis [1, 1, 0].
  let axis = [1, 1, 0];
  rotate(QUARTER_PI, axis);

  // Draw a box.
  box();
}
```

Syntax

```
rotate(angle, [axis])
```

Parameters

angle	Number: angle of rotation in the current angleMode().
axis	p5.Vector|Number[]: axis to rotate about in 3D.

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

Related References

applyMatrix
Applies a transformation matrix to the coordinate system.

resetMatrix
Clears all transformations applied to the coordinate system.

rotate
Rotates the coordinate system.

rotateX
Rotates the coordinate system about the x-axis in WebGL mode.