

QUARTER_PI

A `Number` constant that's approximately 0.7854.

`QUARTER_PI` is one-fourth the value of the mathematical constant π . It's useful for many tasks that involve rotation and oscillation. For example, calling `rotate(QUARTER_PI)` rotates the coordinate system `QUARTER_PI` radians, which is an eighth of a turn (45°).

Note: `TWO_PI` radians equals 360°, `PI` radians equals 180°, `HALF_PI` radians equals 90°, and `QUARTER_PI` radians equals 45°.

Examples

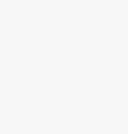
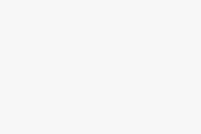


```
function setup() {
  createCanvas(100, 100);

  background(200);

  // Draw an arc from 0 to QUARTER_PI.
  arc(50, 50, 80, 80, 0, QUARTER_PI);

  describe('A one-eighth slice of a circle drawn in white on a
gray background.');
}
```



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

  background(200);

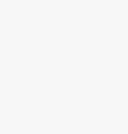
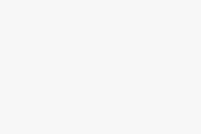
  // Translate the origin to the center.
  translate(50, 50);

  // Draw a line.
  line(0, 0, 40, 0);

  // Rotate an eighth turn.
  rotate(QUARTER_PI);

  // Draw the same line, rotated.
  line(0, 0, 40, 0);

  describe('Two black lines that form a "V" opening towards
the bottom-right corner of a gray square.');
}
```



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

  describe(
    'A red circle and a blue circle oscillate from left to
right on a gray background. The red circle appears to chase
the blue circle.'
);

function draw() {
  background(200);

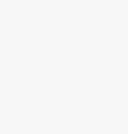
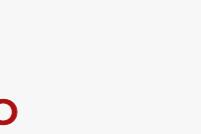
  // Translate the origin to the center.
  translate(50, 50);

  // Calculate the x-coordinates.
  let x1 = 40 * sin(frameCount * 0.05);
  let x2 = 40 * sin(frameCount * 0.05 + QUARTER_PI);

  // Style the oscillators.
  noStroke();

  // Draw the red oscillator.
  fill(255, 0, 0);
  circle(x1, 0, 20);

  // Draw the blue oscillator.
  fill(0, 0, 255);
  circle(x2, 0, 20);
}
```



This page is generated from the comments in [src/core/constants.js](#). Please feel free to edit it and submit a pull request!

Related References

[ADD](#)
[ALT](#)
[ARROW](#)
[AUTO](#)

AUTO allows us to automatically set the width or height of an element (but not both), based on the current height and width of the element.

