

TWO_PI

A `Number` constant that's approximately 6.2382.

`TWO_PI` is twice the value of the mathematical constant π . It's useful for many tasks that involve rotation and oscillation. For example, calling `rotate(TWO_PI)` rotates the coordinate system `TWO_PI` radians, which is one full turn (360°). `TWO_PI` and `TAU` are equal.

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

Examples

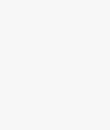
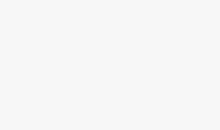


```
function setup() {
  createCanvas(100, 100);

  background(200);

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

  describe('A white circle drawn on a gray background.');
}
```



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

  background(200);

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

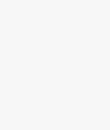
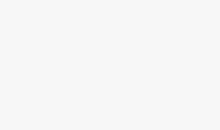
  // Draw a line.
  line(0, 0, 40, 0);

  // Rotate a full turn.
  rotate(TWO_PI);

  // Style the second line.
  strokeWeight(5);

  // Draw the same line, shorter and rotated.
  line(0, 0, 20, 0);

  describe(
    'Two horizontal black lines on a gray background. A thick line extends from the center toward the right. A thin line extends from the end of the thick line.'
  );
}
```



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

  describe(
    'A red circle with a blue center oscillates from left to right on a gray background.'
  );
}

function draw() {
  background(200);

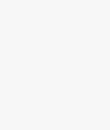
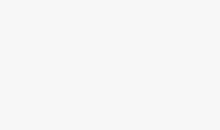
  // Translate the origin to the center.
  translate(50, 50);

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

  // Style the oscillators.
  noStroke();

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

  // Draw the blue oscillator, smaller.
  fill(0, 0, 255);
  circle(x2, 0, 10);
}
```



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.

