

text()

Draws text to the canvas.

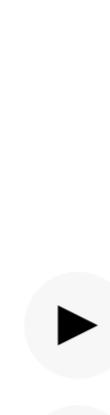
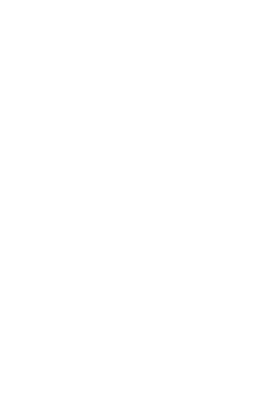
The first parameter, `str`, is the text to be drawn. The second and third parameters, `x` and `y`, set the coordinates of the text's bottom-left corner. See [textAlign\(\)](#) for other ways to align text.

The fourth and fifth parameters, `maxWidth` and `maxHeight`, are optional. They set the dimensions of the invisible rectangle containing the text. By default, they set its maximum width and height. See [rectMode\(\)](#) for other ways to define the rectangular text box. Text will wrap to fit within the text box. Text outside of the box won't be drawn.

Text can be styled a few ways. Call the `fill()` function to set the text's fill color. Call `stroke()` and `strokeWeight()` to set the text's outline. Call `textSize()` and `textFont()` to set the text's size and font, respectively.

Note: `WEBGL` mode only supports fonts loaded with `loadFont()`. Calling `stroke()` has no effect in `WEBGL` mode.

Examples



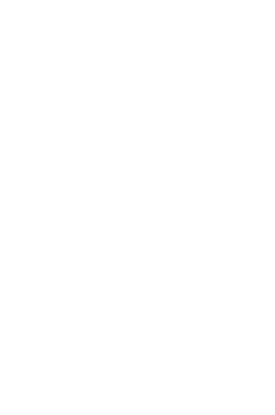
```
function setup() {
  background(200);
  text('hi', 50, 50);

  describe('The text "hi" written in black in the middle of a gray square.');
}
```



```
function setup() {
  background('skyblue');
  textSize(100);
  text('🌈', 0, 100);

  describe('A rainbow in a blue sky.');
}
```



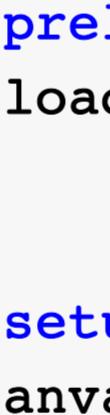
```
function setup() {
  textSize(32);
  fill(255);
  stroke(0);
  strokeWeight(4);
  text('hi', 50, 50);

  describe('The text "hi" written in white with a black outline.');
}
```



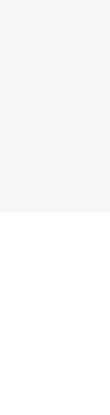
```
function setup() {
  background('black');
  textSize(22);
  fill('yellow');
  text('rainbows', 6, 20);
  fill('cornflowerblue');
  text('rainbows', 6, 45);
  fill('tomato');
  text('rainbows', 6, 70);
  fill('limegreen');
  text('rainbows', 6, 95);

  describe('The text "rainbows" written on several lines, each in a different color.');
}
```



```
function setup() {
  background(200);
  let s = 'The quick brown fox jumps over the lazy dog.';
  text(s, 10, 10, 70, 80);

  describe('The sample text "The quick brown fox..." written in black across several lines.');
}
```



```
let font;

function preload() {
  font = loadFont('/assets/inconsolata.otf');
}

function setup() {
  createCanvas(100, 100, WEBGL);
  textSize(32);
  textAlign(CENTER, CENTER);
}

function draw() {
  background(0);
  rotateY(frameCount / 30);
  text('p5*js', 0, 0);

  describe('The text "p5*js" written in white and spinning in 3D.');
}
```

Syntax

```
text(str, x, y, [maxWidth], [maxHeight])
```

Parameters

<code>str</code>	<code>String Object Array Number Boolean</code> : text to be displayed.
<code>x</code>	Number: x-coordinate of the text box.
<code>y</code>	Number: y-coordinate of the text box. See rectMode() for other options.
<code>maxWidth</code>	Number: maximum width of the text box. See rectMode() for other options.

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

Related References

[font](#) The font's underlying `opentype.js` font

[textBounds](#) Returns the bounding box for a string

[textToPoints](#) Returns an array of points outlining a string of text written using the font

[image](#) Draws an image to the canvas.



[Resources](#)

[Reference](#)

[Tutorials](#)

[Examples](#)

[Contributes](#)

[Community](#)

[About](#)

[Start Coding](#)

[Donate](#)

[Information](#)

[Download](#)

[Contact](#)

[Copyright](#)

[Privacy Policy](#)

[Terms of Use](#)

[Socials](#)

[GitHub](#) ↗

[Instagram](#) ↗

[YouTube](#) ↗

[Discord](#) ↗

Donate Today! Support p5.js and the Processing Foundation.

