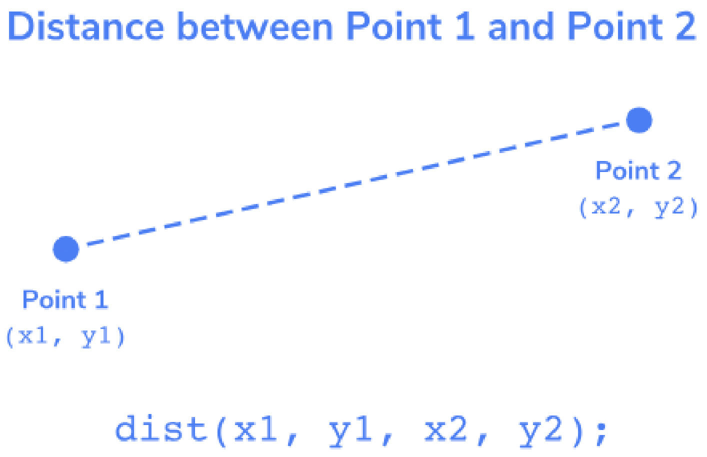
Understanding the dist() Function

13 min

The `dist()` function is a powerful tool for calculating the distance between two points, stationary or dynamic. This function is oftentimes used in conjunction with mouse events. Understanding how and when to use `dist()` can simplify calculations and make fun and interesting sketches.



The `dist()` function takes four arguments: the x and y coordinates of the first point (`x1` and `y1` in the diagram above) and the x and y coordinates of the second point (`x2` and `y2` above).

For example, you can use the `dist()` function to calculate the dynamic distance between the center of the canvas and the mouse position as follows:

```
let distance = dist(mouseX, mouseY, width / 2, height / 2);
```

The `distance` variable in the above code will return the distance between the center of the canvas and the mouse position wherever the mouse is.

Instructions

Checkpoint 1 Passed

1.

Let’s create a spotlight effect using the mouse position. The spotlight will reveal and distort the grid of ellipses being drawn across the canvas!

At the bottom of the `setup()` function in **sketch.js**, assign the `canvas_dist` variable to the `dist()` function that calculates the distance between the following two points:

- The origin of the canvas at (0, 0)
- The bottom-right corner of the canvas at (`width`, `height`)

This will calculate the diagonal distance across the canvas.
Hint
You can calculate the diagonal distance across the canvas from the top-left corner at the origin to the bottom-right corner at (`width`, `height`) using the `dist()` function. Remember that the function takes 4 arguments: x and y positions of two endpoints.

Checkpoint 2 Passed

2.

Now, to determine the distance between the mouse position and each ellipse’s position, we will use the `dist()` function again.

Inside the nested `for` loops in the `draw()` function block, use the `dist()` function to calculate the distance between `mouseX`, `mouseY`, and the `i` and `j` iterator variables. Create and store the output of the `dist()` function's calculation in a variable called `size`.

Hint

Calculate the distance between the mouse position at (`mouseX`, `mouseY`) and the position of each ellipse in the grid at (`i`, `j`) using the `dist()` function.

Store the result of the `dist()` function in a newly created variable called `size`. Remember to do this inside the nested `for` loops.

3.

Below the `size` variable initialization, re-assign `size` to:

```
(size / canvas_dist) * 70
```

This sets the `size` variable to be proportionate to the size of the canvas. Here, we multiply by 70 to scale the spotlight appropriately.

Hint

Reassign the `size` variable to the calculation of `size` in proportion to the size of the canvas so we can begin making the spotlight effect.

Make sure to do this below our `dist()` calculation from the previous step.

4.

At the end of the `draw()` function inside the nested `for` loops, a grid of ellipses is drawn at x position `i` and y position `j`.

Currently, the ellipses have a width and height of 10 pixels—replace them with the `size` variable.

Run the code. Move the mouse across the canvas and see the spotlight effect happen!

Optionally, try uncommenting the second `fill()` function and run the code. Play around with the iteration statement in the `for` loops and the value that the `size` variable is being factored by! All of these will affect the outcome of the sketch differently.

Hint

By replacing the ellipse's width and height with the `size` variable, the mouse's position on the canvas proportionally changes the size of every ellipse around a certain radius of the mouse.

sketch.js

```
// Stores diagnal distance across the canvas
let canvas_dist;

function setup() {
  createCanvas(windowWidth, windowHeight);
  noStroke();

  // TODO: Use the dist() function to calculate the diagonal distance across the canvas
  canvas_dist = dist(0, 0, width, height);
}
```

```
function draw() {
  background(255, 40);

  // Nested for loops tp draw a grid of ellipses
  for (let i = 0; i <= width; i += 15) {
    for (let j = 0; j <= height; j += 15) {
      // TODO: Calculate the distance between mouse position and each ellipse's position
      let size = dist(mouseX, mouseY, i, j);
      // TODO: Reassign size to be proportional to the size of the canvas
      size = (size / canvas_dist) * 70;
      fill(0);
      // Try uncommenting the second fill function below:
      //fill(j, i,size, 10 );
      // TODO: set width and height of ellipse to size variable
      ellipse(i, j, size, size);
    }
  }
}
```