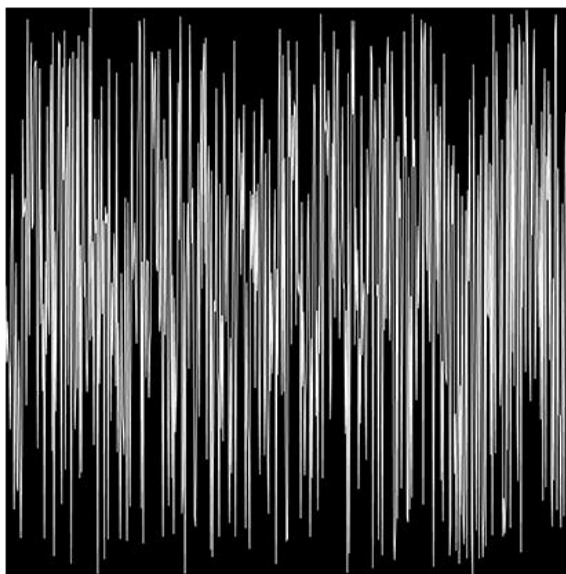


## Noise

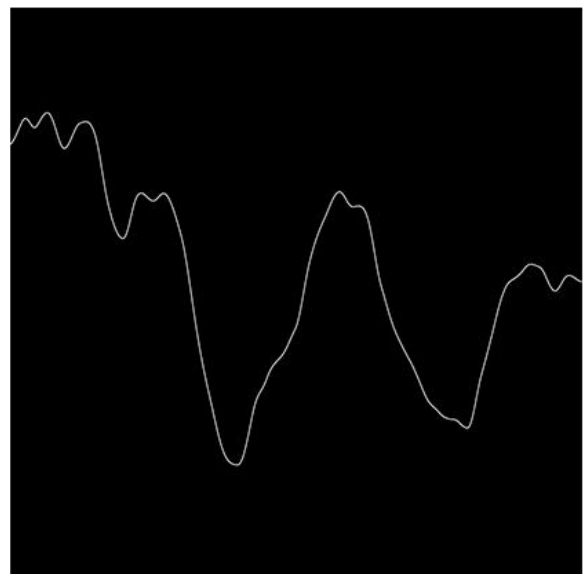
12 min

As you may have noticed in the previous exercise, sometimes the `random()` function can be too random. There may be times when you want to create random movements that are more natural. Thankfully, p5.js has a `noise()` function that does exactly this!

The `noise()` function returns a random value based on the [Perlin noise](#). When you call the `noise()` function, it generates a naturally ordered sequence of random numbers. The diagram below shows Perlin noise over time, which is represented as the x-axis. Notice how smooth the curve is for the `noise()` graph. In contrast, notice how jumpy the curve is for the `random()` graph. This is because the `random()` function selects purely random numbers with no relation to the previously selected value. Whereas, the `noise()` function generates values that produce a more harmonic sequence of numbers.



Random



Noise

You can generate a noise value with the following syntax:

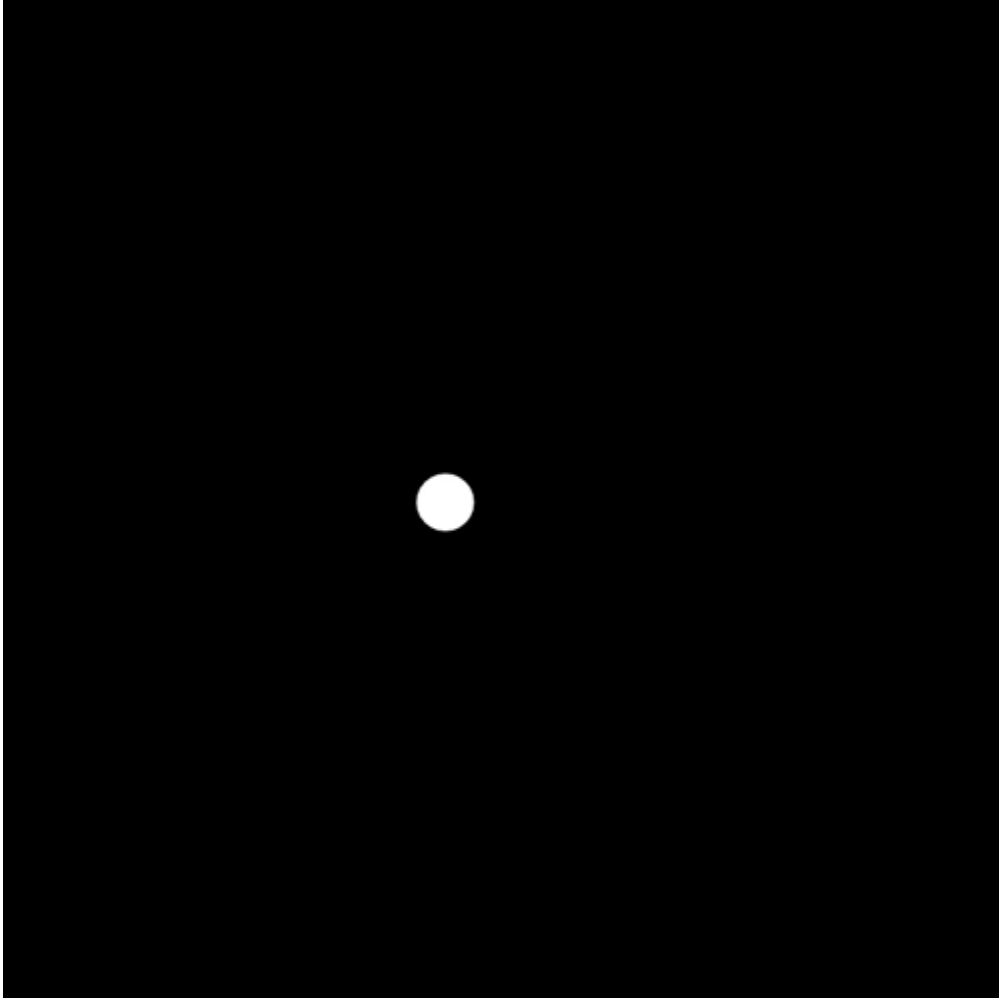
```
let num = noise(x, y);
```

Let's take a closer look:

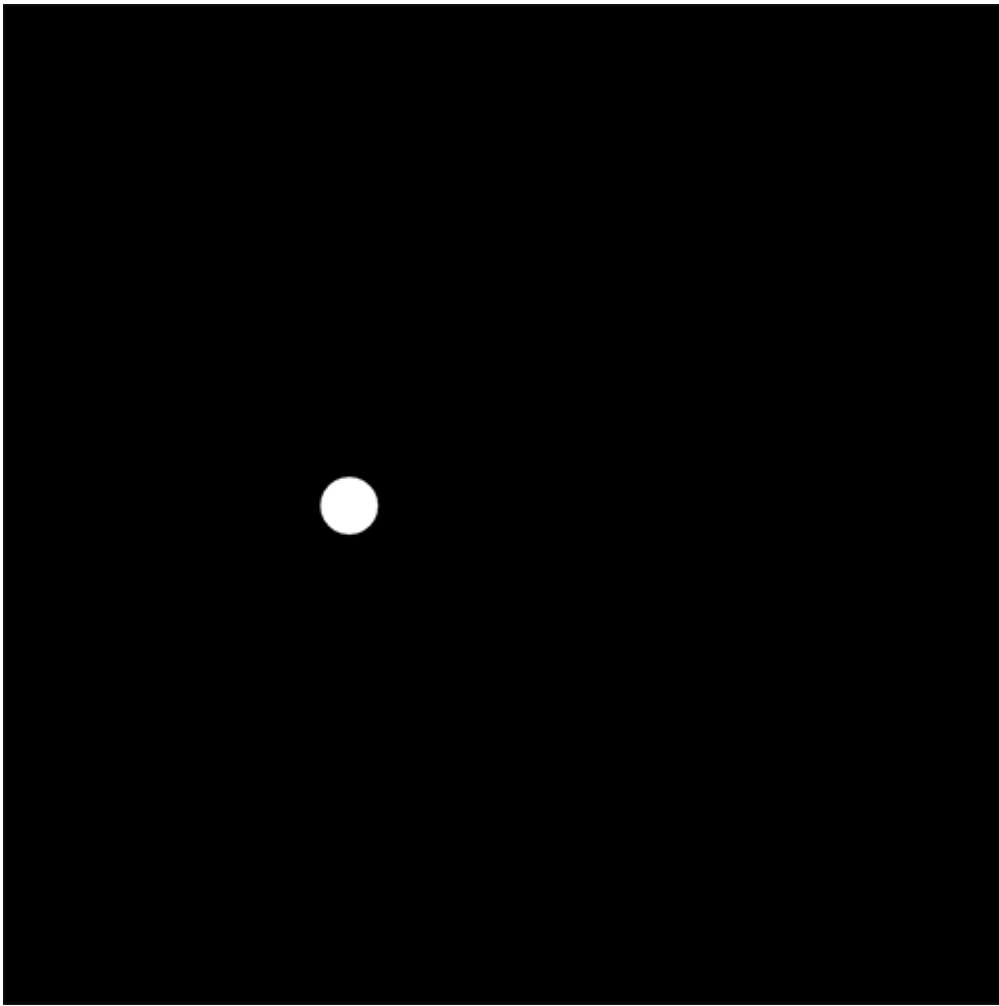
- The `x` value specifies the x-coordinate in noise space.
- The `y` value specifies the y-coordinate in noise space.

- The number returned when you call the `noise()` function is between 0 and 1.
- The `noise()` function can't be called without at least one [argument](#).

The difference between the `random()` and `noise()` function is especially noticeable when used in animations. You can see in the example below how sporadic the motion is when the `random()` function is used for the ellipse's x position.



The animation below uses the `noise()` function for the x position of the ellipse. Compare the movement with the above ellipse.



In order to create this wandering ellipse animation, we need to first set a variable for the value we wish to increment along the x-axis. We do this by writing:

```
let xOffset = 0;
```

Next, we set the `x` variable to be equal to:

```
map(noise(xOffset), 0, 1, 0, width);
```

This function maps the value generated from the `noise()` function, which will be between 0 and 1, to a number between 0 and `width`. You can read more about how to use the `map()` function in the [Tips for Creating Animations with p5.js](#) article.

We then pass the `x` variable into the x position of the ellipse:

```
ellipse(x, 200, 24, 24);
```

The last step is to increment our `xOffset` value by writing:

```
xOffset += 0.01;
```

When we put all the steps together, you get the below code that creates these nice, semi-random movements.

```
let xOffset = 0;  
let x = map(noise(xOffset), 0, 1, 0, width);  
  
fill(255);  
ellipse(x, 200, 24, 24);  
xOffset += 0.01;
```

## Instructions

1.

In this exercise, you will use the noise function to make a wandering ball animation. First, inside the `draw()` function, set the `x` variable to equal:

```
width * noise(t + 15)
```

Be sure to write this somewhere after the `fill()` function.

Hint

Remember, that the number returned when you call a `noise()` function is between 0 and 1. This is why we need to multiply the value generated by `noise()` with the `width` value.

2.

Still inside the `draw()` function, set the `y` variable to equal:

```
height * noise(t + 5)
```

We will be using this variable for the ellipse's y position.

Hint

The number returned when you call a `noise()` function is between 0 and 1. This is why we need to multiply the value generated by `noise()` with the `height` value.

3.

Now, draw an ellipse with the `x` variable for the x position and the `y` variable for the y position. Set the width and height of the ellipse to be 150.

Hint

Remember that you can draw an ellipse with the following syntax:

```
ellipse(x, y, ellipseWidth, ellipseHeight);
```

4.

As the last step, increment the `t` value by 0.005. Your ellipse should now be gracefully gliding around the canvas!

Hint

Remember that you can increment a value by writing:

```
x += amount;
```

This takes the current value of the variable and then adds `amount` to it.

sketch.js

```
// Global variables
let t;
let x; // X position
let y; // Y position

function setup() {
  createCanvas(windowWidth, windowHeight);
  background(0);
  t = 0;
}

function draw() {
  background(0, 16);

  noStroke();
  fill(255);

  // TODO: Set the x position to incorporate the noise function
  x = width * noise(t + 15);

  // TODO: Set the y position to incorporate the noise function
  y = height * noise(t + 5);

  // TODO: Draw an ellipse with the variables for x and y passed in
  ellipse(x, y, 150, 150);

  // TODO: Increment the t variable to control the noise amount
  t += 0.005;
}
```