

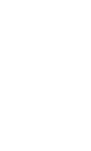
noiseDetail()

Adjusts the character of the noise produced by the [noise\(\)](#) function.

Perlin noise values are created by adding layers of noise together. The noise layers, called octaves, are similar to harmonics in music. Lower octaves contribute more to the output signal. They define the overall intensity of the noise. Higher octaves create finer-grained details.

By default, noise values are created by combining four octaves. Each higher octave contributes half as much (50% less) compared to its predecessor. `noiseDetail()` changes the number of octaves and the falloff amount. For example, calling `noiseDetail(6, 0.25)` ensures that `noise()` will use six octaves. Each higher octave will contribute 25% as much (75% less) compared to its predecessor. Falloff values between 0 and 1 are valid. However, falloff values greater than 0.5 might result in noise values greater than 1.

Examples



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

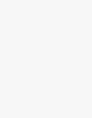
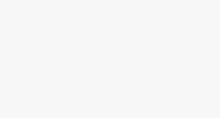
  // Set the noise level and scale.
  let noiseLevel = 255;
  let noiseScale = 0.02;

  // Iterate from top to bottom.
  for (let y = 0; y < height; y += 1) {
    // Iterate from left to right.
    for (let x = 0; x < width / 2; x += 1) {
      // Scale the input coordinates.
      let nx = noiseScale * x;
      let ny = noiseScale * y;

      // Compute the noise value with six octaves
      // and a low falloff factor.
      noiseDetail(6, 0.25);
      let c = noiseLevel * noise(nx, ny);

      // Draw the left side.
      stroke(c);
      point(x, y);

      // Compute the noise value with four octaves
      // and a high falloff factor.
    }
  }
}
```



Syntax

```
noiseDetail(lod, falloff)
```



Parameters

lod Number: number of octaves to be used by the noise.
 falloff Number: falloff factor for each octave.

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

Related References

[noise](#)
 Returns random numbers that can be tuned to feel organic.

[noiseDetail](#)
 Adjusts the character of the noise produced by the `noise()` function.

[noiseSeed](#)
 Sets the seed value for the `noise()` function.

