Arrays

Arrays Can Hold and Use Multiple Numbers with the Same Name, and, they let You Write a for-loop to Use Them!

int X[ ] = { 100, 200, 300, 400, 500, 600, 700 }; int Y[ ] = { 100, 100, 100, 200, 200, 200, 300 }; void setup( ) { size( 800, 800 ); background(200, 200, 255); stroke(0, 0, 0); fill(255, 255, 0); } void draw( ) { for (int i = 0; i < X.length; i = i + 1) { ellipse(X[i], Y[i], 100, 50); } }

A Way of Declaring Arrays That Lets You Fill Them Later

int NumPoints = 5000; int W = 800; int H = 800; int[ ] X; int[ ] Y; int[ ] R; int[ ] G; int[ ] B;

Allocate and Filling Arrays

void setup() { X = new int[NumPoints]; Y = new int[NumPoints]; R = new int[NumPoints]; G = new int[NumPoints]; B = new int[NumPoints]; size(800, 800); for (int i = 0; i < NumPoints; i = i + 1) { X[i] = int( random(0, W)); Y[i] = int( random(0, H)); R[i] = int( random(0, 255)); G[i] = int( random(0, 255)); B[i] = int( random(0, 255)); } }

Each point is assigned a random location. Each point is assigned a random color. There is no significance to each point's specific color — it is to look cool!

Allocate the arrays to hold the random points and the random colors. At this point, memory has been given to them, but they don't have any values assigned.

Note: Array indices start at 0 and end at the number of elements minus one. So, an array dimensioned [10] indexes from 0 to 9.

The arrays that will hold the points and the colors. They have only been declared. They don't yet have any memory given to them.

Total number of points

The number of points.
Using the Arrays in a for-loop

```cpp
void draw() {
    background(200, 200, 255);
    stroke(0, 0, 0);
    for (int i = 0; i < NumPoints; i = i + 1) {
        fill(R[i], G[i], B[i]);
        ellipse(X[i], Y[i], 8, 8);
    }
}
```

A Cool Pattern will be made even Cooler

For our next trick, during each frame we are going move each point halfway towards one of three target triangle vertices. Which target to use is chosen at random.

Declaring Arrays

```cpp
int NumPoints = 5000;
int TARGET_SIZE = 40;
int W = 800;
int H = 800;
int [ ] X;
int [ ] Y;
int [ ] R;
int [ ] G;
int [ ] B;
int [ ] XC = { 50, W/2, W-50 };
int [ ] YC = { H-50, 50, H-50 };
```
Setup the arrays to hold the random points and the random colors. At this point, memory has been given to them, but they don't have any values assigned.

Each point is assigned a random location.

Setup the arrays to hold the random points and the random colors. At this point, memory has been given to them, but they don't have any values assigned.

Each point is assigned a random location.

Perform the animation at 2 frames per second so that we can actually see it. Otherwise, it will be too fast.

Each point is assigned a random color. There is no significance to each point’s specific color – it is to look cool!

Draw each point with its color.

Re-compute each point’s position by randomly picking one of the targets (0, 1, or 2) and moving halfway towards it.

Mathematicians call shapes like this "attractors"