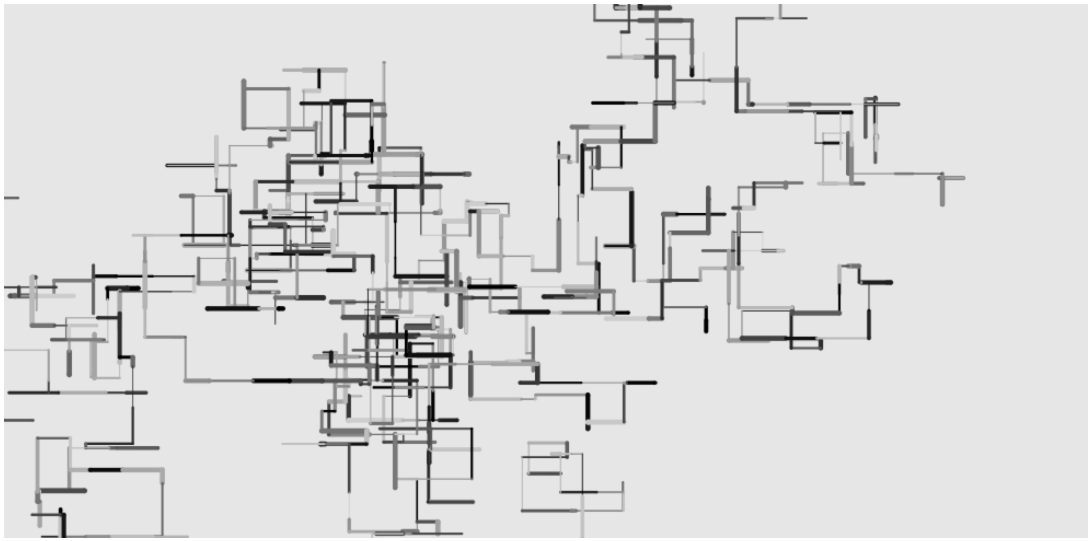
Using Point



Point

Description Draws a point, a coordinate in space at the dimension of one pixel. The first parameter is the horizontal value for the point, the second value is the vertical value for the point, and the optional third value is the depth value. Drawing this shape in 3D with the z parameter requires the P3D parameter in combination with `size()` as shown in the above example.

Syntax

```
point(x, y)
point(x, y, z)
```

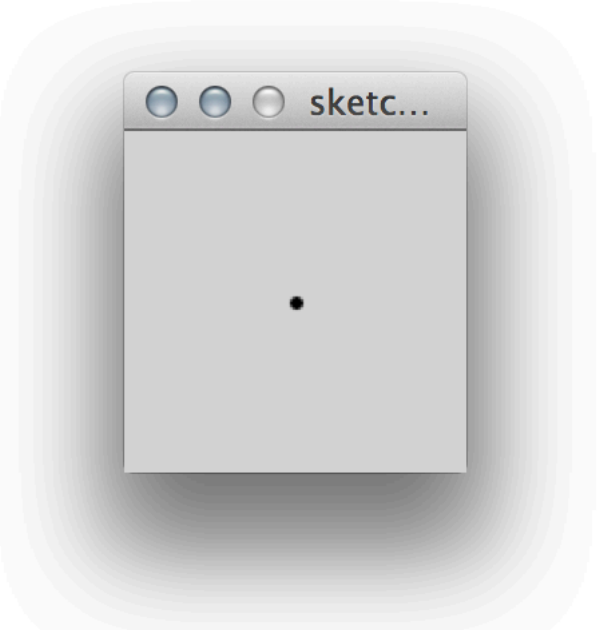
Parameters

x	float: x-coordinate of the point
y	float: y-coordinate of the point
z	float: z-coordinate of the point

Returns `void`

Draw a Point

```
// draw a point
strokeWeight(5);
point(width/2, height/2);
```



Delete that…

Draw a Line with Point

```
void moveRight(float startX, float startY, float moveCount) {
  for(float i=0; i<moveCount; i++) {
    point(startX+i, startY);
    xpos = startX + i;
    ypos = startY;
  }
}
```

Setup

```
float xpos;
float ypos;
float strokeW;
float pointCount;

void setup() {
  background(random(100,255));
  size(500,500);
  xpos = random(width);
  ypos = random(height);
}
```

Try this…

```
float xpos;
float ypos;
float strokeW;
float pointCount;

void setup() {
  size(500,500);
  background(255);
  xpos = width/2;
  ypos = height/2;
}

void draw() {
  strokeW = random(1,10);
  pointCount = random(1,20);
  stroke(0);
  strokeWeight(strokeW);
  moveRight(xpos,ypos,pointCount);
}

void moveRight(float startX, float startY, float moveCount) {
  for(float i=0; i<moveCount; i++) {
    point(startX+i, startY);
    xpos = startX + i;
  }
}
```

Challenge #1

Create a function for up, down, right and left…

Challenge #2

Keep the line drawing on the screen infinitely…

Hint: employ edge detection!

Calling the Functions in draw()

```
if (random(100)>70) {
  strokeWeight(strokeW);
  moveLeft(xpos,ypos,pointCount);
} else if (random(100)>65) {
  strokeWeight(strokeW);
  moveUp(xpos,ypos,pointCount);
} else if (random(100)>55) {
  strokeWeight(strokeW);
  moveDown(xpos,ypos,pointCount);
} else {
  strokeWeight(strokeW);
  moveRight(xpos, ypos, pointCount);
}
```

Check Edge…

```
if(xpos > width || xpos < 0 || ypos > height || ypos < 0) {
  xpos = random(width);
  ypos = random(height);
} else {
  if (random(100)>70) {
    strokeWeight(strokeW);
    moveLeft(xpos,ypos,pointCount);
  } else if (random(100)>65) {
    strokeWeight(strokeW);
    moveUp(xpos,ypos,pointCount);
  } else if (random(100)>55) {
    strokeWeight(strokeW);
    moveDown(xpos,ypos,pointCount);
  } else {
    strokeWeight(strokeW);
    moveRight(xpos, ypos, pointCount);
  }
}
```

Finished code minus functions…

```
float xpos;
float ypos;
float strokeW;
float pointCount;

void setup() {
  //size(displayWidth,displayHeight);
  size(400,400);
  background(random(100,255));
  xpos = random(width);
  ypos = random(height);
}

void draw() {
  strokeW = random(1,3);
  pointCount = random(2,10);
  stroke(random(10),random(100),random(200));
  if(xpos > width || xpos < 0 || ypos > height || ypos < 0) {
    xpos = random(width);
    ypos = random(height);
  } else {
    if (random(100)>70) {
      strokeWeight(strokeW);
      moveLeft(xpos,ypos,pointCount);
    } else if (random(100)>65) {
      strokeWeight(strokeW);
      moveUp(xpos,ypos,pointCount);
    } else if (random(100)>55) {
      strokeWeight(strokeW);
      moveDown(xpos,ypos,pointCount);
    } else {
      strokeWeight(strokeW);
      moveRight(xpos, ypos, pointCount);
    }
  }
}
```

- You create the functions based on…

```
void moveRight(float startX, float startY, float moveCount) {
  for(float i=0; i<moveCount; i++) {
    point(startX+i, startY);
    xpos = startX + i;
    ypos = startY;
  }
}
```

Challenge #3

Create a “Lines” class and instantiate 10 or more on the screen…

Hint...

```
class Lines {
  // class variables
  float xpos;
  float ypos;
  float strokeW;
  float lineLength;

  // constructor
  Lines(float tempX, float tempY, float tempStroke, float tempLength) {
    xpos = tempX;
    ypos = tempY;
    strokeW = tempStroke;
    lineLength = tempLength;
  }
  // display
  void display() {
    strokeW = random(1,2);
    lineLength = random(1,50);
    //stroke(random(100,255));
    stroke(random(200),random(100),random(10));
    if(xpos > width || xpos < 0 || ypos > height || ypos < 0) {
      xpos = random(width);
      ypos = random(height);
    } else {
      if (random(100)>90) {
        strokeWeight(strokeW);
        moveLeft(xpos,ypos,lineLength);
      } else if (random(100)>80) {
        strokeWeight(strokeW);
        moveUp(xpos,ypos,lineLength);
```

Hint...

```
//Lines myLines1;
Lines[] myLines = new Lines [10];

void setup() {
  background(random(2,50));
  size(1000,500);
  //myLines1 = new Lines(random(width),random(height),random(1,5),random(1,20));
  for (int i=0; i<myLines.length; i++) {
    myLines[i] = new Lines(random(width),random(height),random(1,5),random(1,20));
  }
}

void draw() {
  //myLines1.display();
  for (int i=0; i<myLines.length; i++) {
    Lines iLines = myLines[i];
    iLines.display();
  }
}
```

Challenge #4

Create four diagonal functions for additional directions…

Challenge #5

Evaluate and tweak randomness for direction, color and line width…

Challenge #6

Have the program reset every 500 frames…