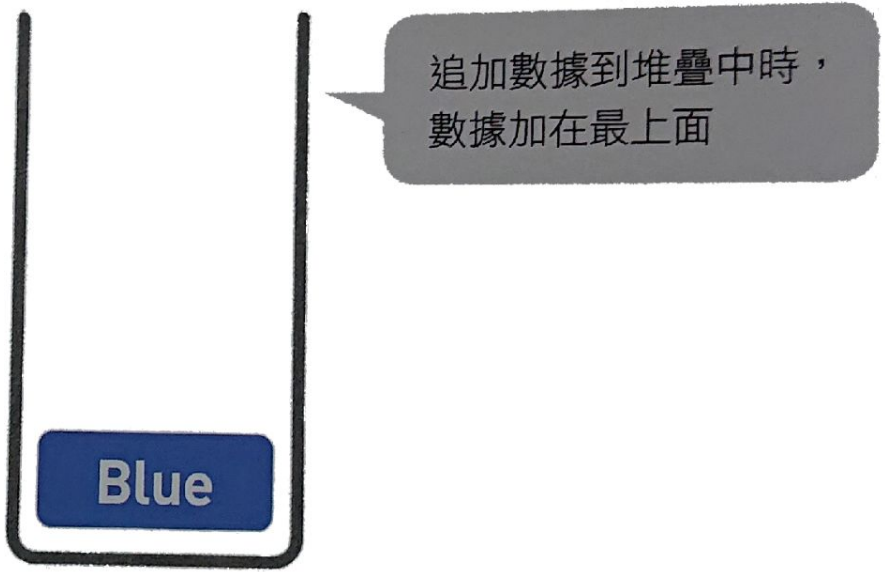
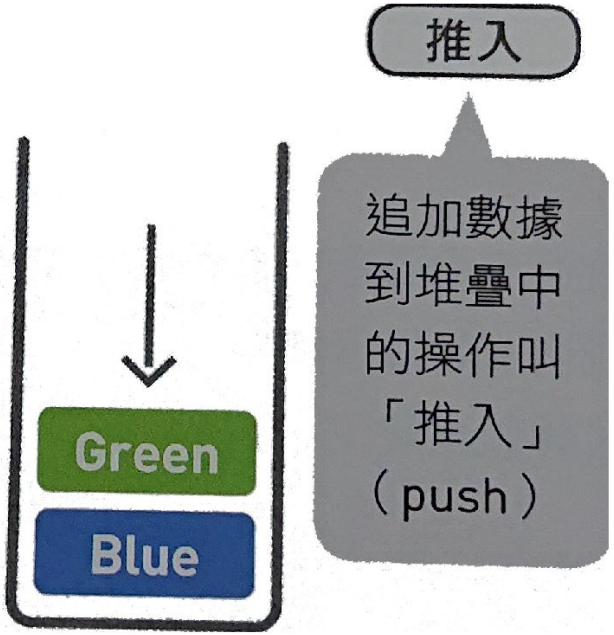
遞迴01

範例程序01

```
void setup(){
  size(800,600,P3D);
  rectMode(CENTER);
}
void draw() {
  background(255);
  camera(0, 0, 600, 0, 0, 0, 0, 1, 0);
  strokeWeight(2);
  stroke(200, 0, 0);
  line(0, -height, 0, 0, height, 0);
  stroke(0, 100, 0);
  line(-width, 0, 0, width, 0, 0);
  noStroke();
  fill(150);
  pushMatrix();
  translate(300,-200);
  pushMatrix();
  rotate(radians(frameCount));
  rect(0, 0, 200, 60);
  popMatrix();
  rect(0, 0, 200, 60);
  popMatrix();
  rect(0, 0, 100, 60);
}
```

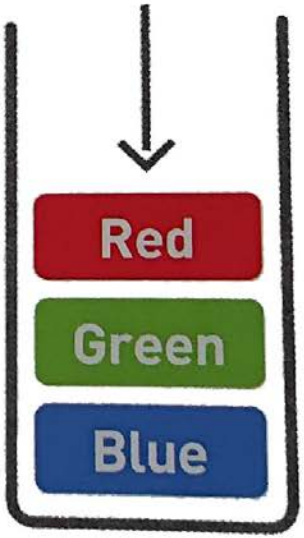
堆疊(stack)

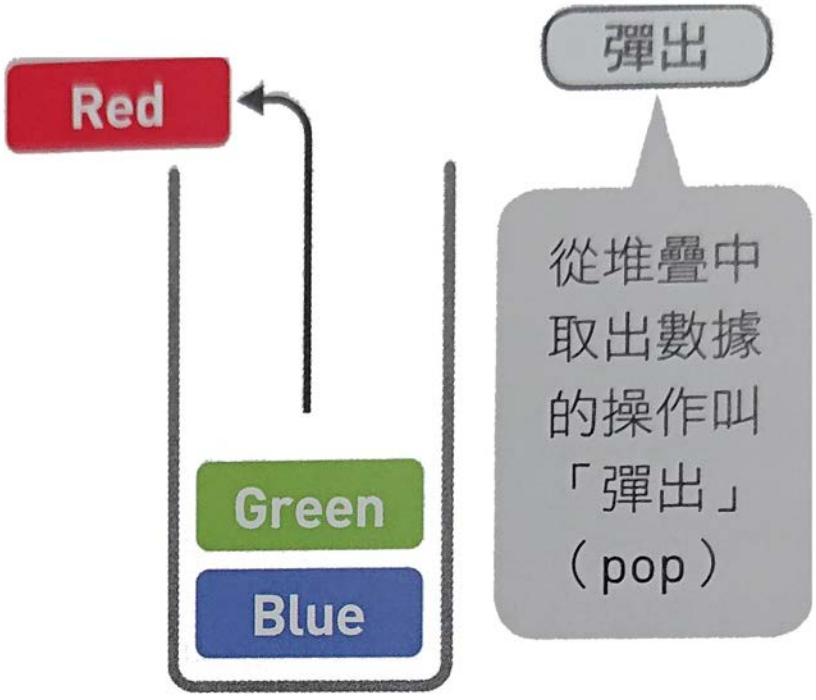

追加數據到堆疊中時，
數據加在最上面
Blue

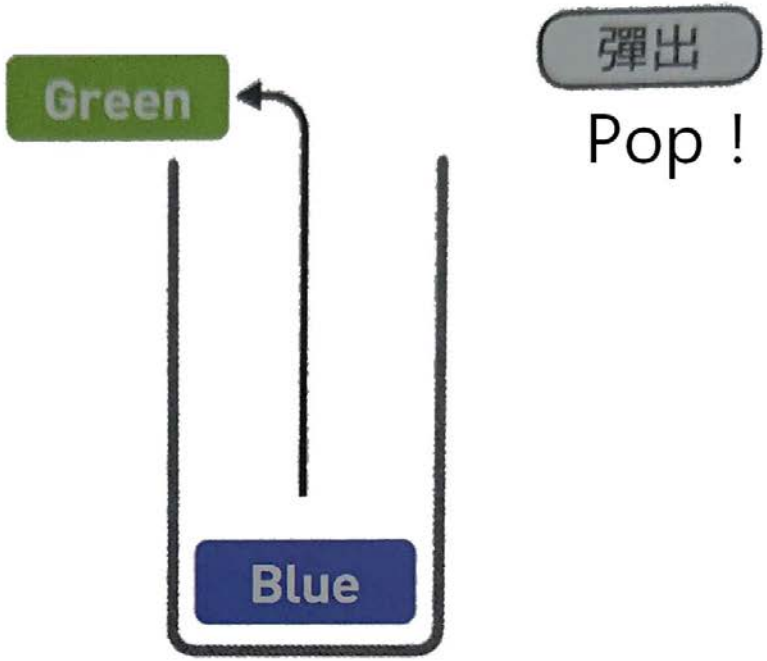

推入
追加數據到堆疊中的操作叫「推入」（push）
Green
Blue

推入

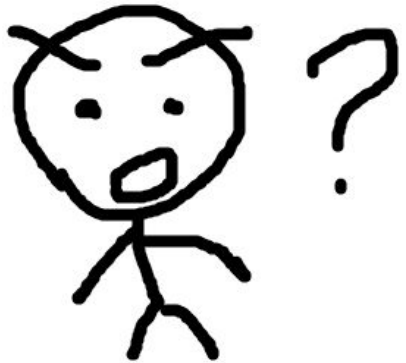
Push

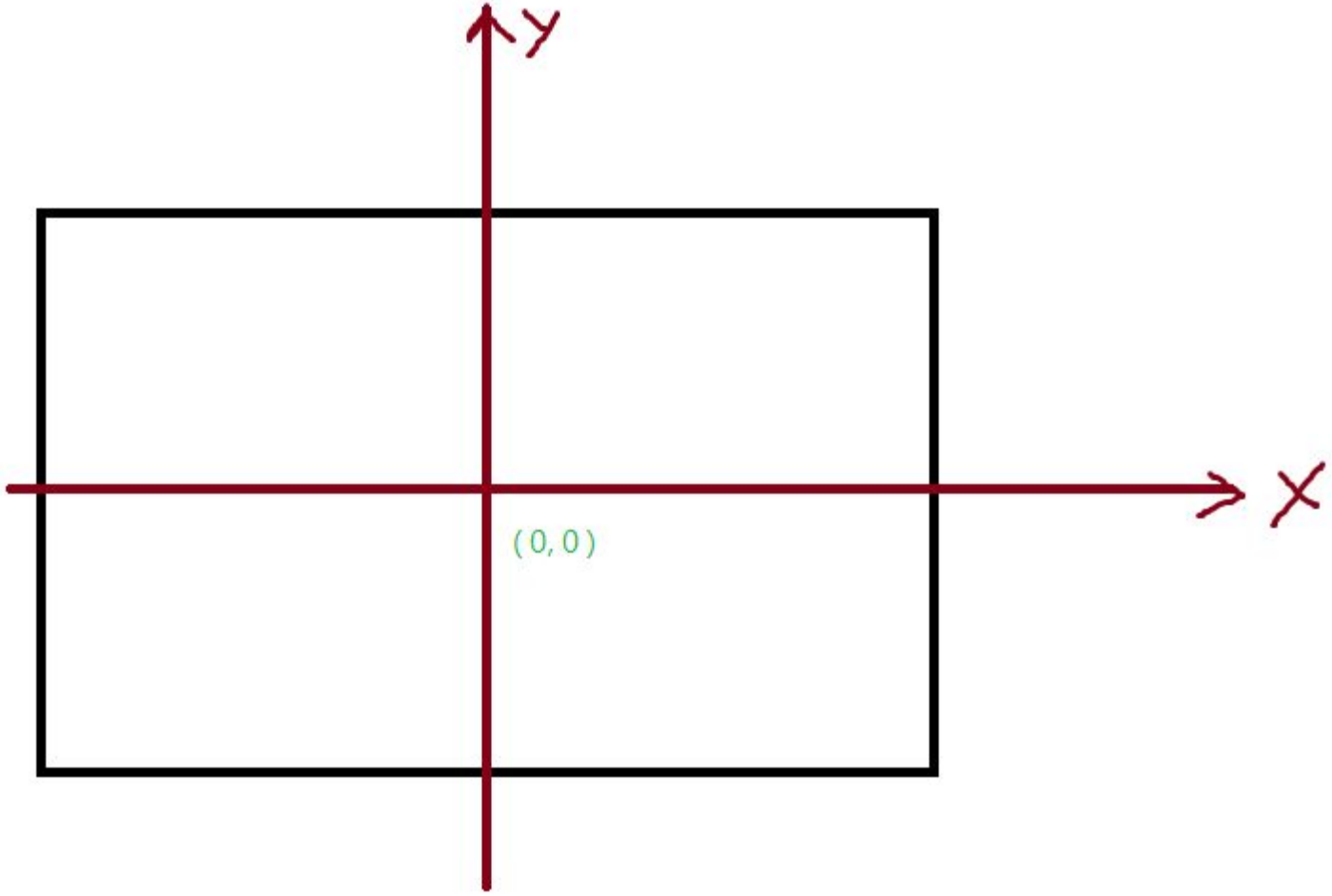


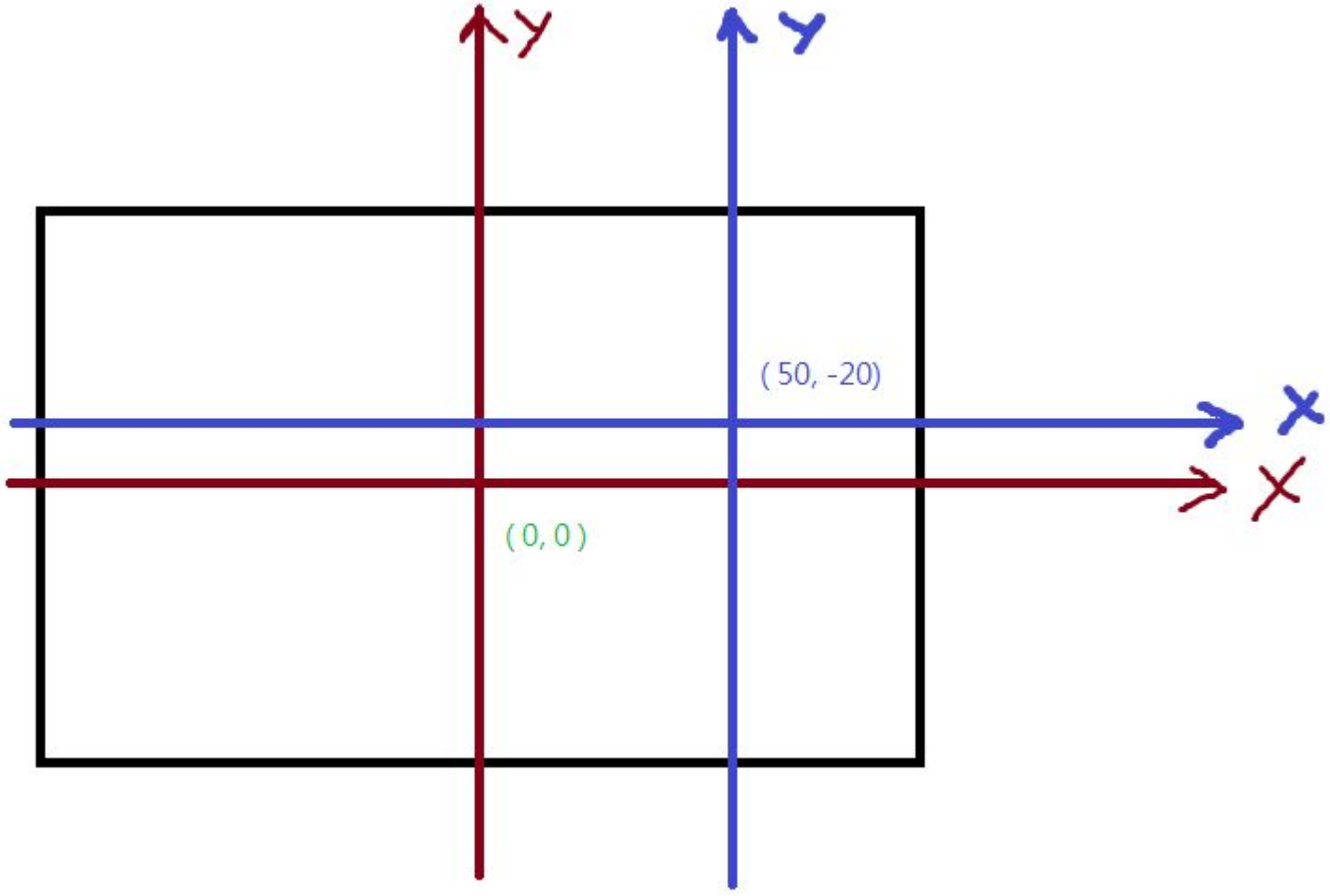

Red
Green
Blue
彈出
從堆疊中
取出數據
的操作叫
「彈出」
（pop）

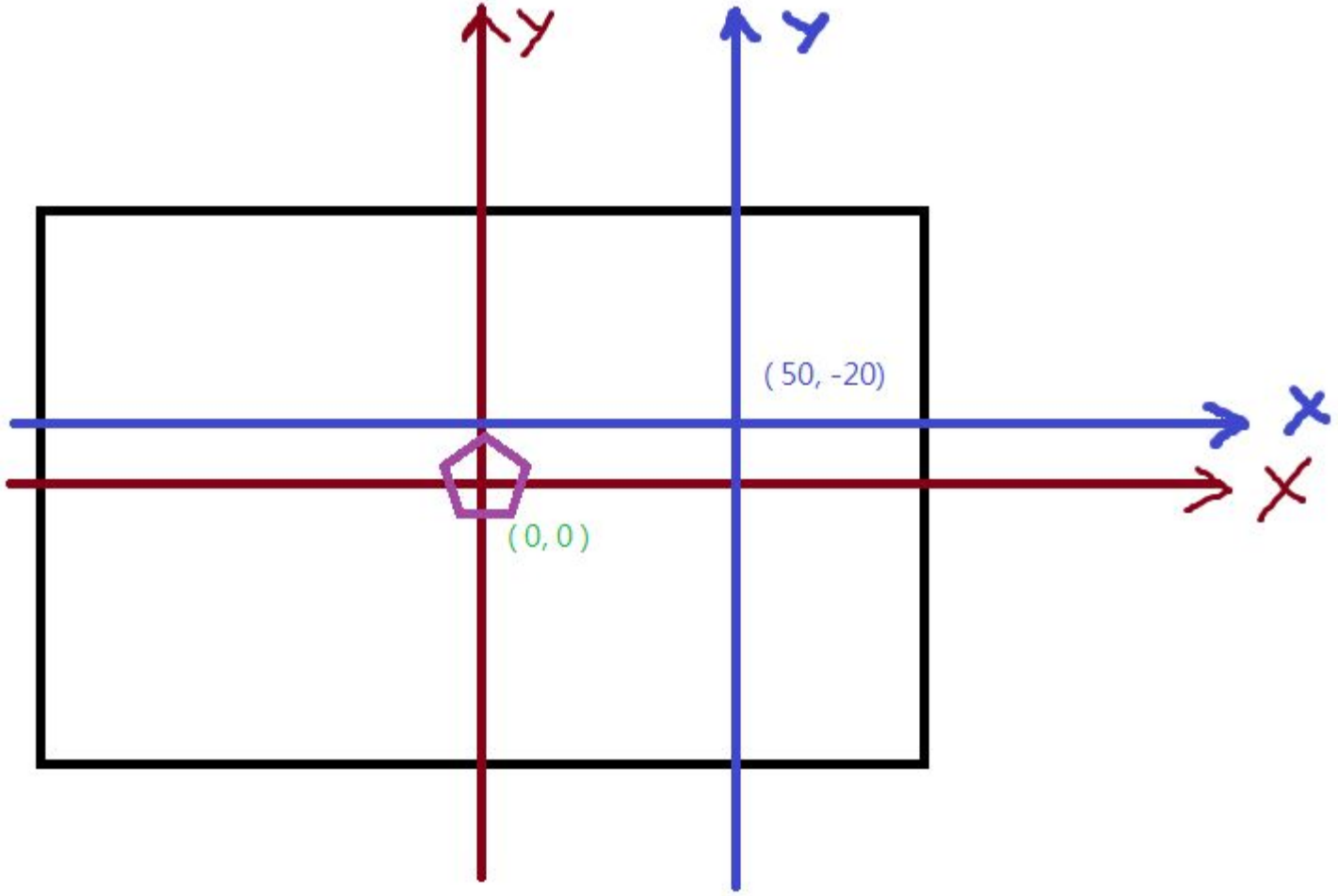

Green
Blue
弾出
Pop !

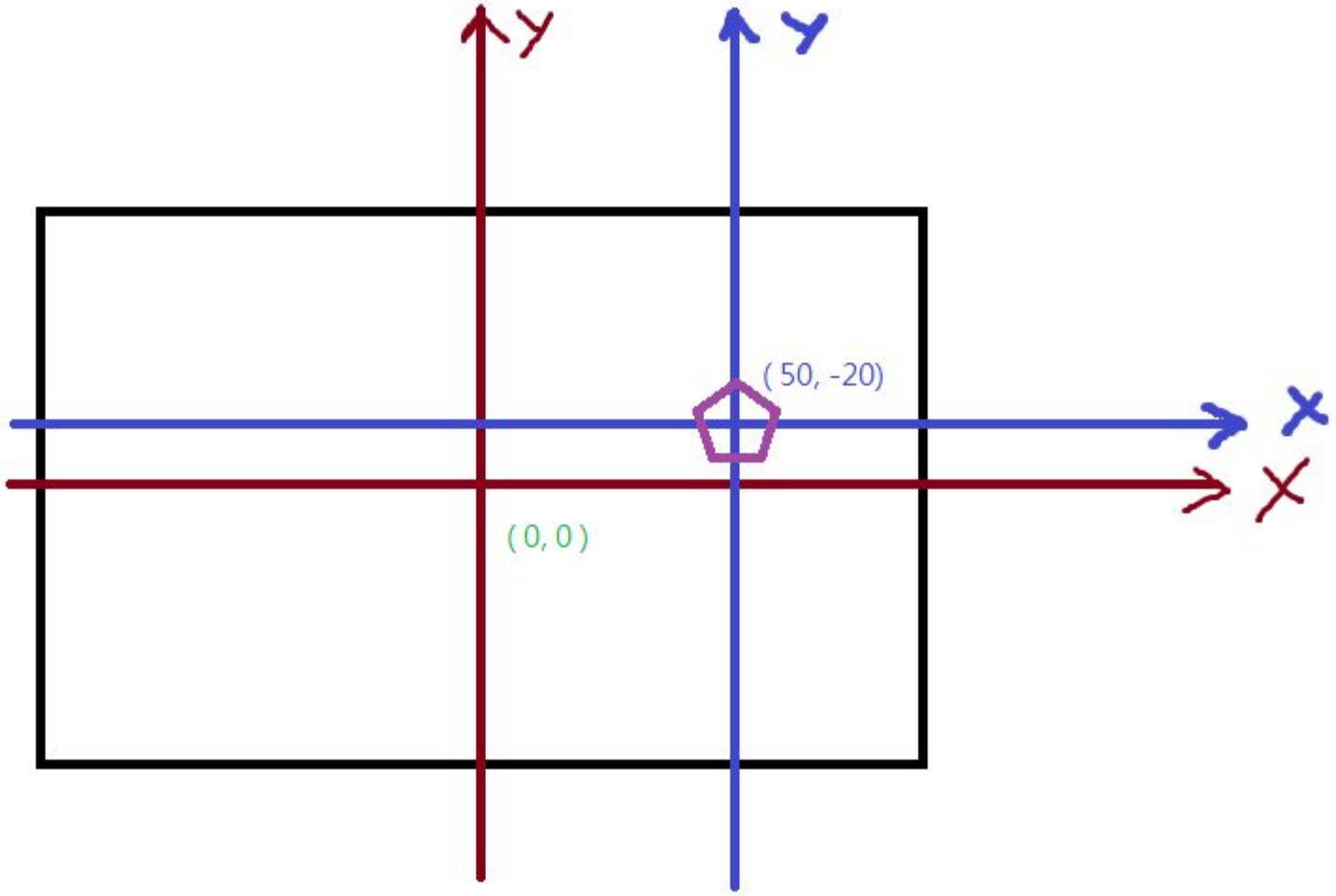
Push/Pop ???

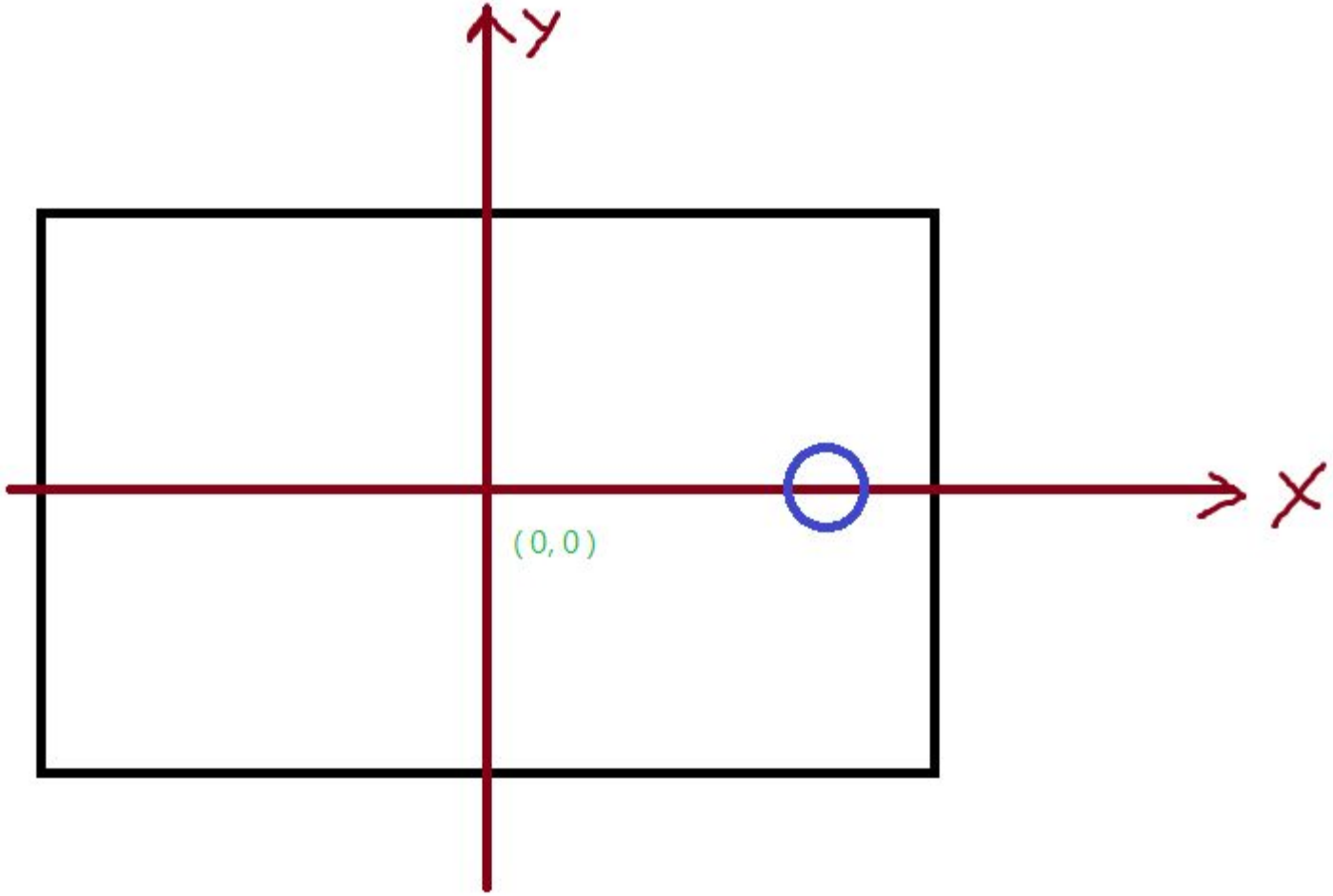


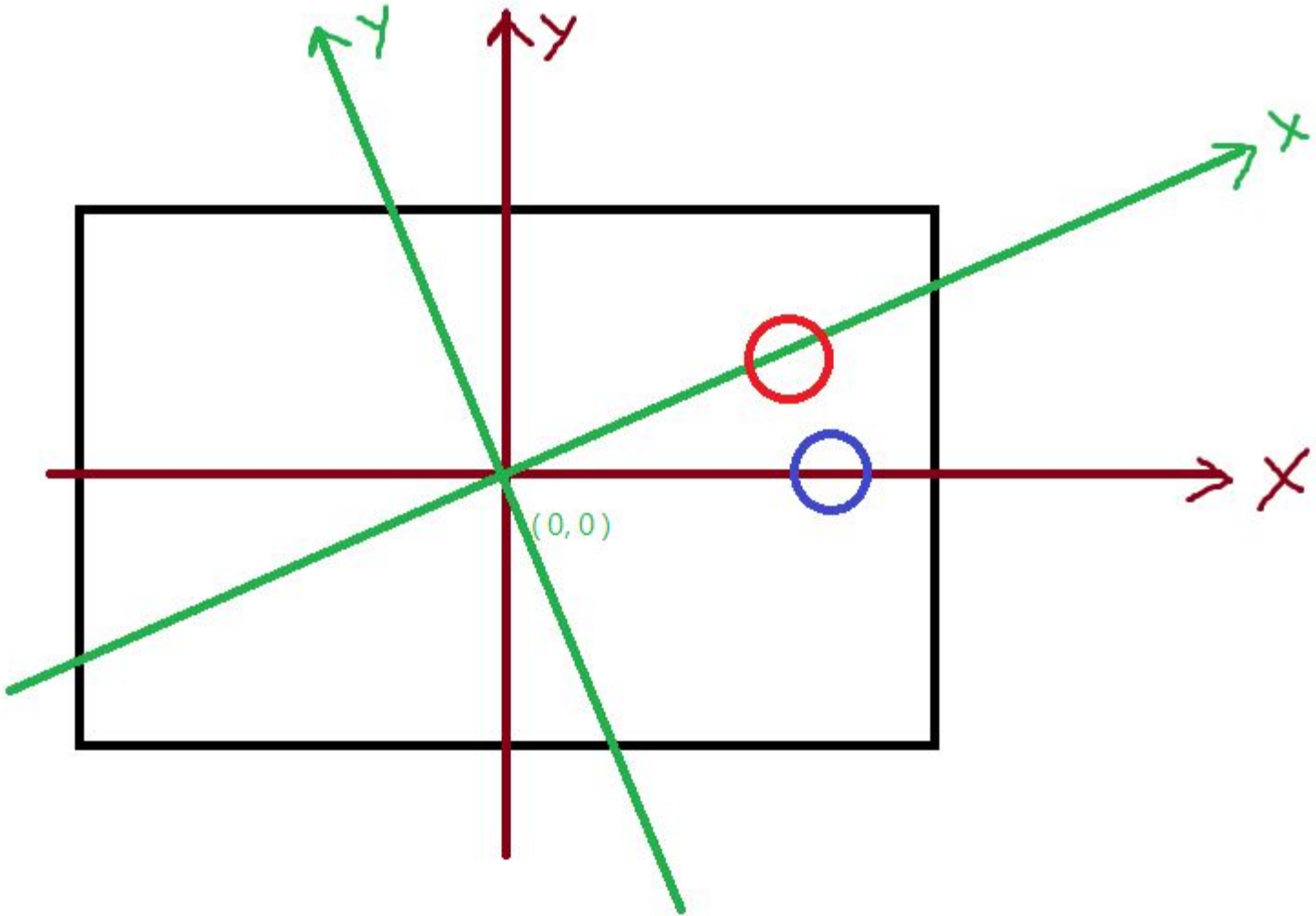

y
x
(0, 0)

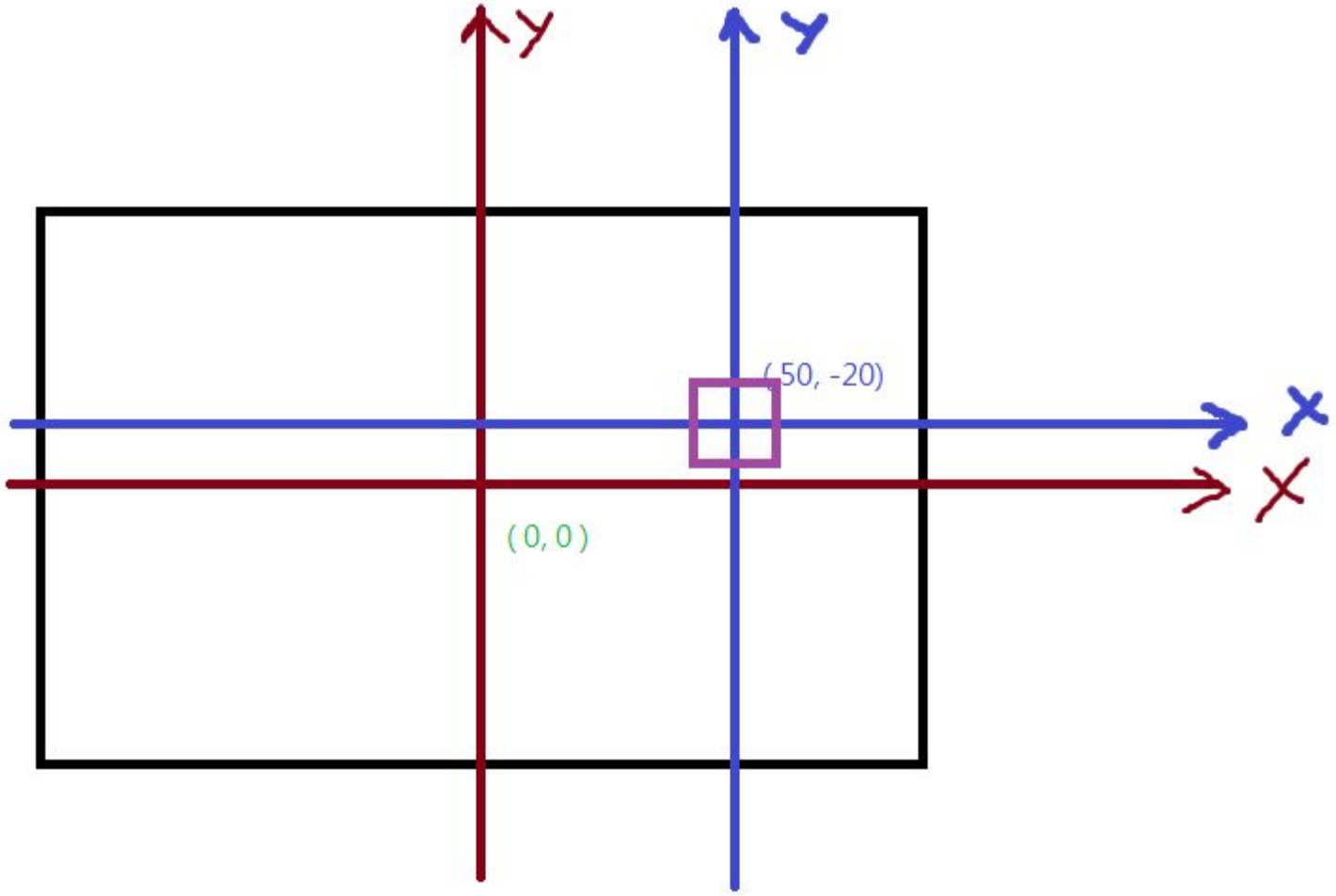

y
y
(50, -20)
x
X
(0, 0)

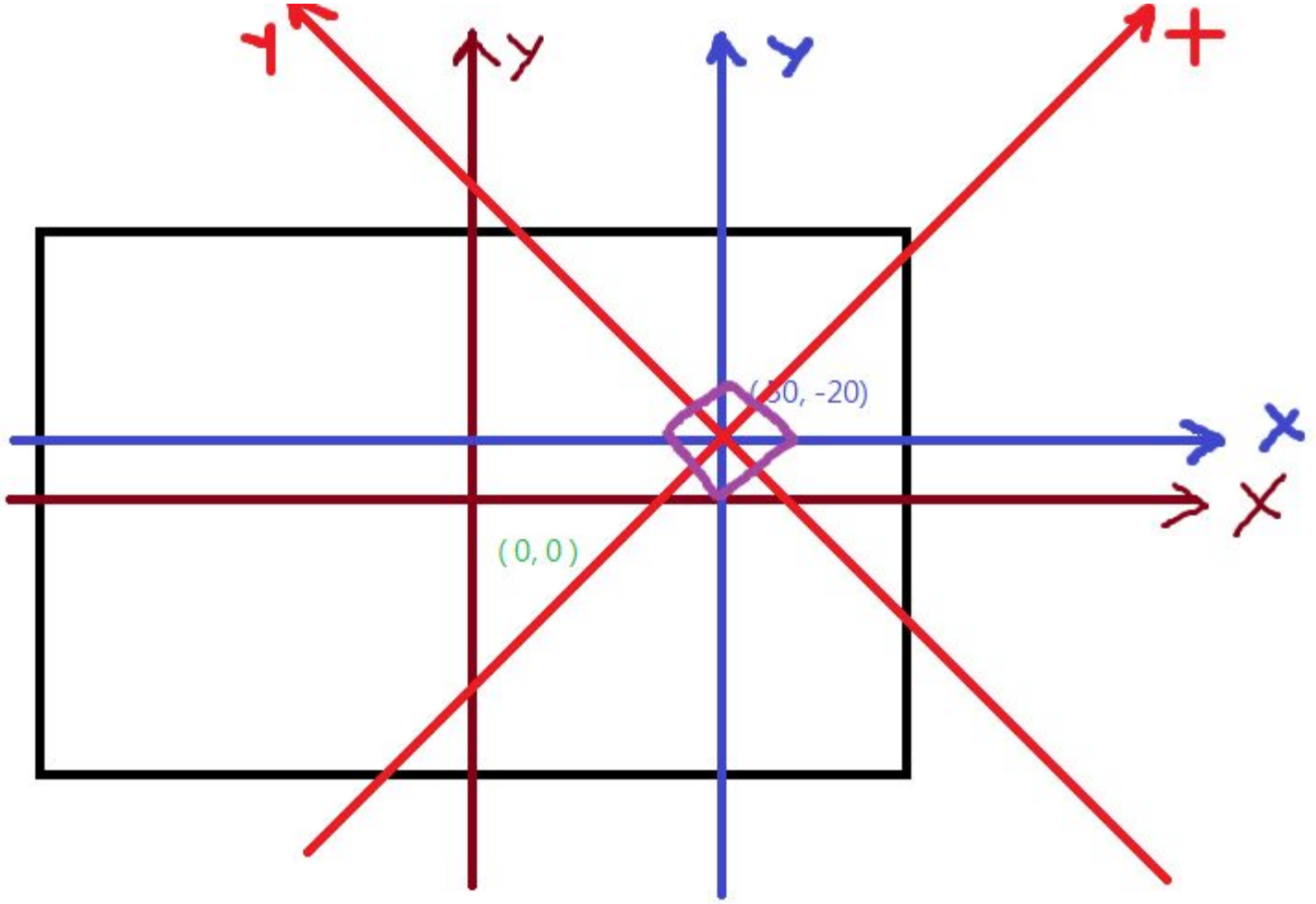

y
y
(50, -20)
x
X
(0, 0)

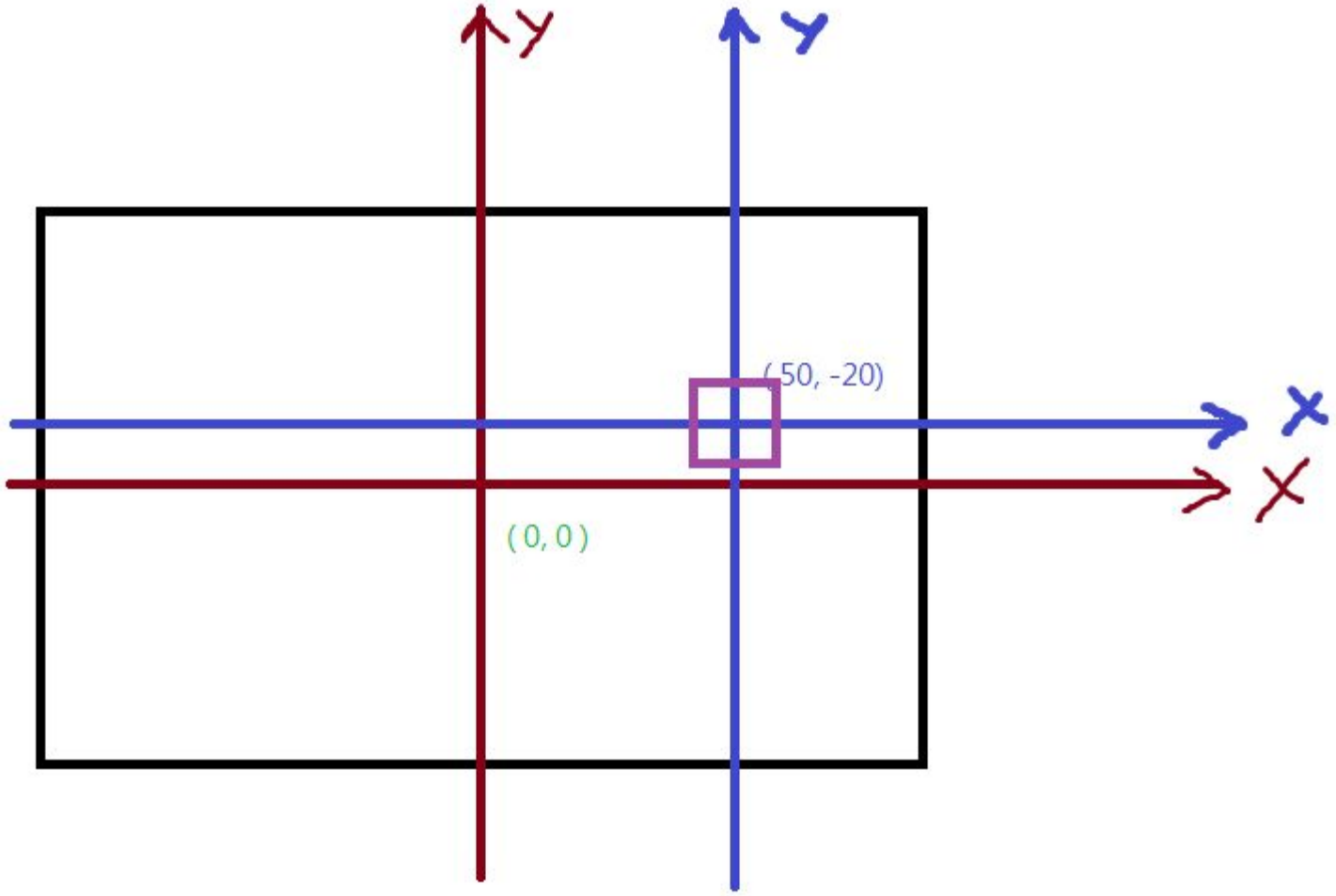

y
y
(50, -20)
x
X
(0, 0)


y
x
(0, 0)

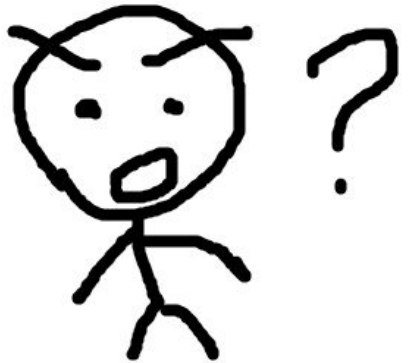

y
y
x
X
(0, 0)


y
y
(50, -20)
x
X
(0, 0)


y
y
y
+
(30, -20)
x
X
(0, 0)

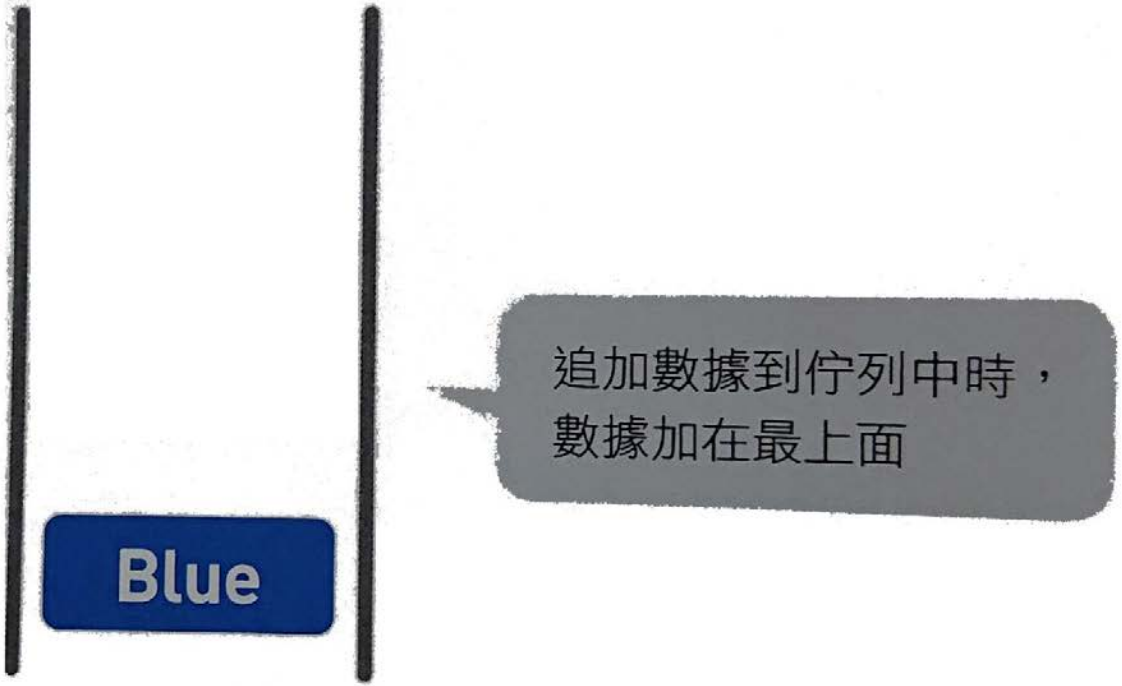

y
y
(50, -20)
x
X
(0, 0)

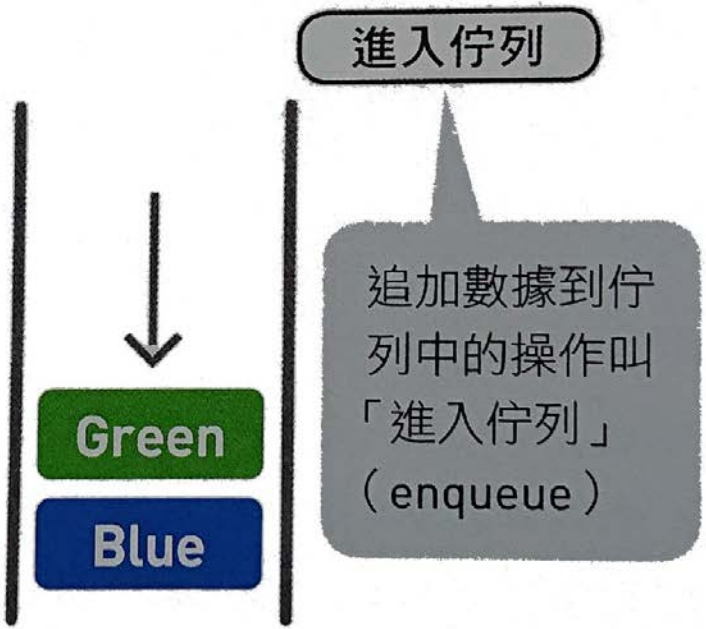
先旋轉再平移?

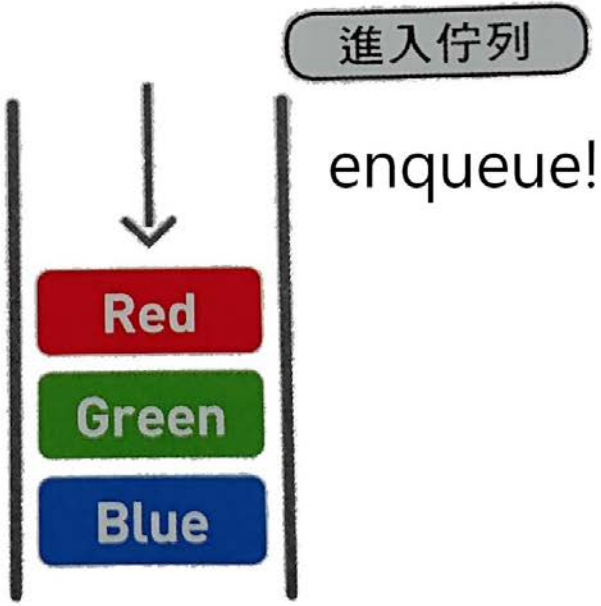


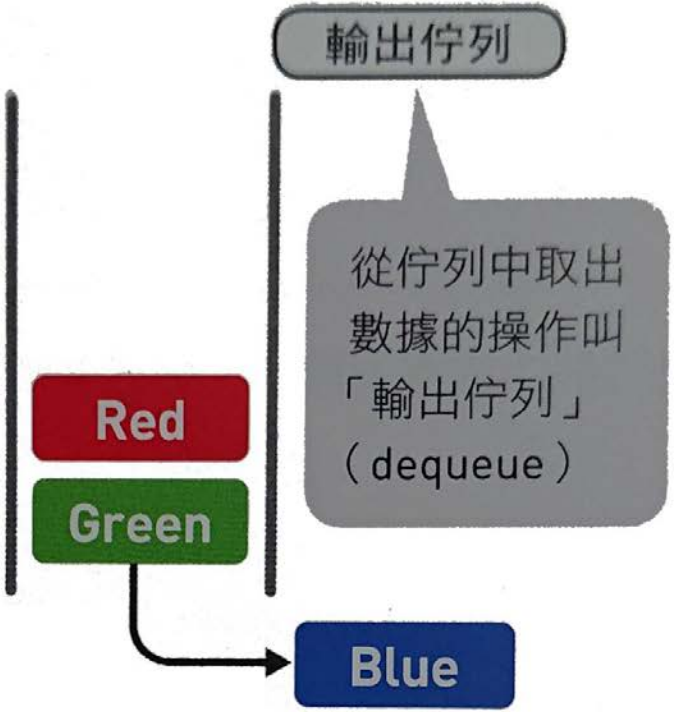
佇列

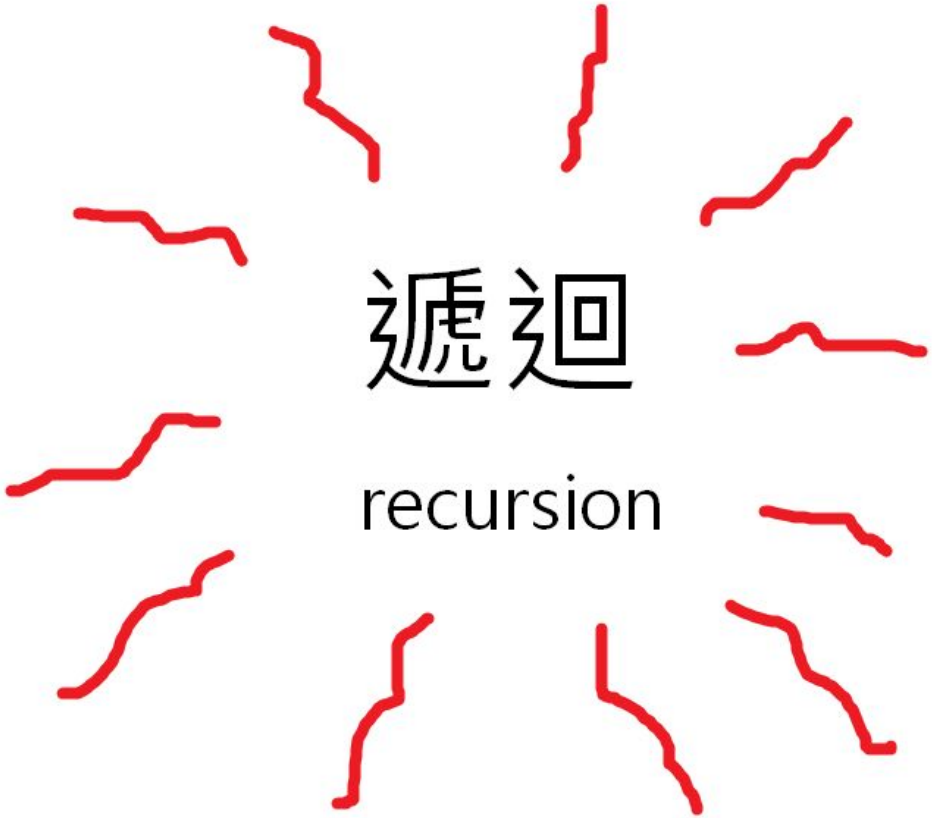
Queue


Blue
追加數據到佇列中時，
數據加在最上面


進入佇列
追加數據到佇列中的操作叫「進入佇列」（enqueue）
Green
Blue


進入佇列
enqueue!
Red
Green
Blue


輸出佇列
從佇列中取出
數據的操作叫
「輸出佇列」
（dequeue）
Red
Green
Blue


遞迴
recursion

DEMO
CODE

```
void setup() {
  size(800, 600, P3D);
  rectMode(CENTER);
}

void draw() {
  background(255);
  drawSquare(width/2, height/2, 100);
}

void drawSquare(float x, float y, float R) {
  rect(x, y, R, R);
}
```

自己呼叫自己

DEMO 2

CODE

```
void setup() {
  size(800, 600, P3D);
  rectMode(CENTER);
}
void draw() {
  background(255);
  drawSquare(width/2, height/2, 100);
}
void drawSquare(float x, float y, float R) {
  rect(x, y, R, R);
  drawSquare(x, y-R, R/2);
}
```

stack overflow

堆疊溢位

☺ 呼叫太多次堆疊導致程式的記憶體爆掉

☺ 程式設計的問答網站

遞迴？ 堆疊？

Recursion/ Stack

遞迴? 堆疊?

Recursion/ Stack

如何才不會stack overflow?

遞迴? 堆疊?

Recursion/ Stack

如何才不會stack overflow?

⇩

設定遞迴的停止條件

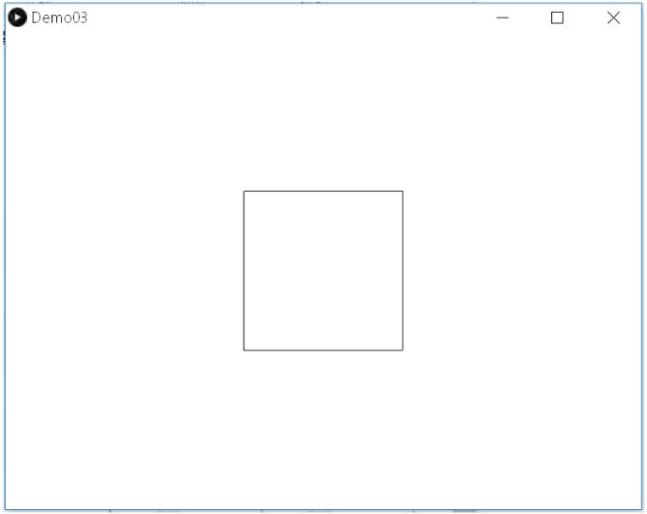
DEMO 3
CODE

```
void setup() {
  size(800, 600, P3D);
  rectMode(CENTER);
}
void draw() {
  background(255);
  drawSquare(width/2, height/2, 200, 5);
}
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

```
drawSquare(width/2, height/2, 200, 4);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x = 400, y = 300, R = 200, i =4



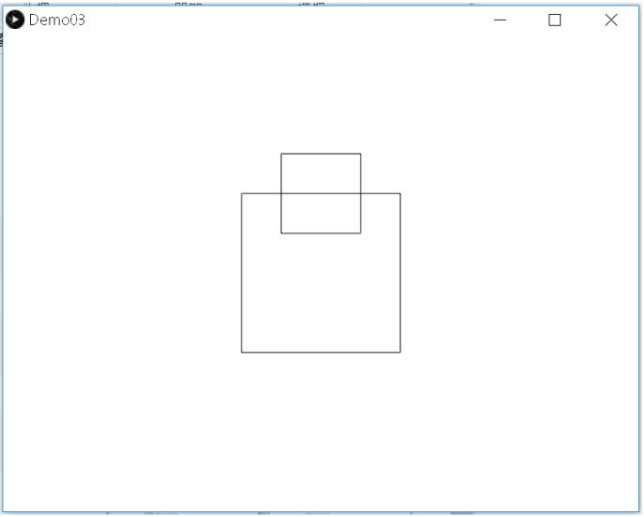
```
drawSquare(width/2, height/2, 200, 4);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=300, R=200, i=4

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=200, R=100, i=3



```
drawSquare(width/2, height/2, 200, 4);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

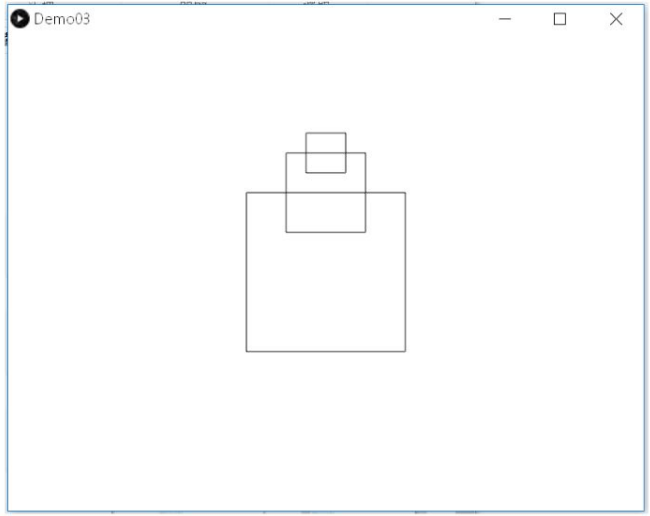
x=400, y=300, R=200, i=4

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=200, R=100, i=3

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=150, R=50, i=2



```
drawSquare(width/2, height/2, 200, 4);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=300, R=200, i=4

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

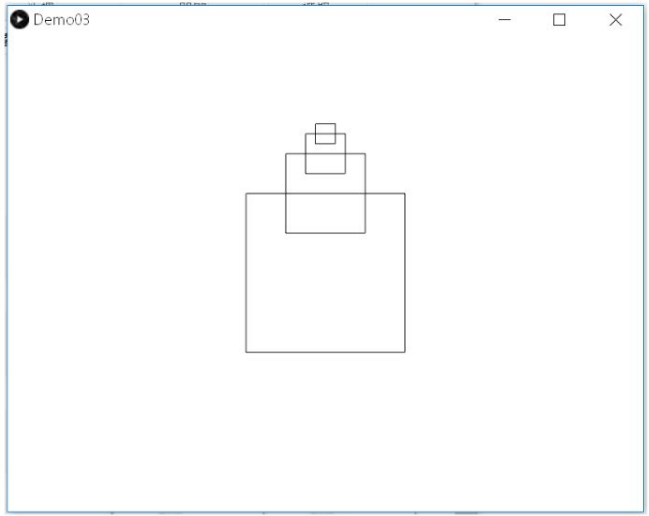
x=400, y=200, R=100, i=3

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=150, R=50, i=2

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=125, R=25, i = 1



```
drawSquare(width/2, height/2, 200, 4);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=300, R=200, i=4

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=200, R=100, i=3

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

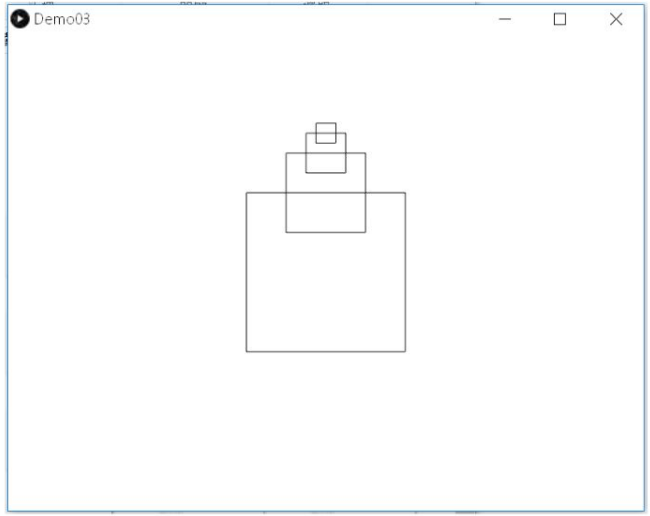
x=400, y=150, R=50, i=2

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=125, R=25, i = 1

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y = 112.5, R=12.5, i = 0



```
drawSquare(width/2, height/2, 200, 4);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=300, R=200, i=4

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=200, R=100, i=3

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

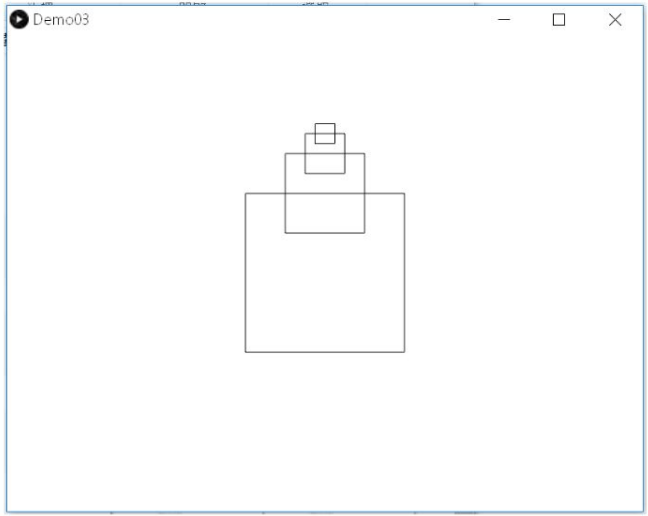
x=400, y=150, R=50, i=2

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=125, R=25, i = 1

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y = 112.5, R=12.5, i = 0



```
drawSquare(width/2, height/2, 200, 4);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=300, R=200, i=4

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

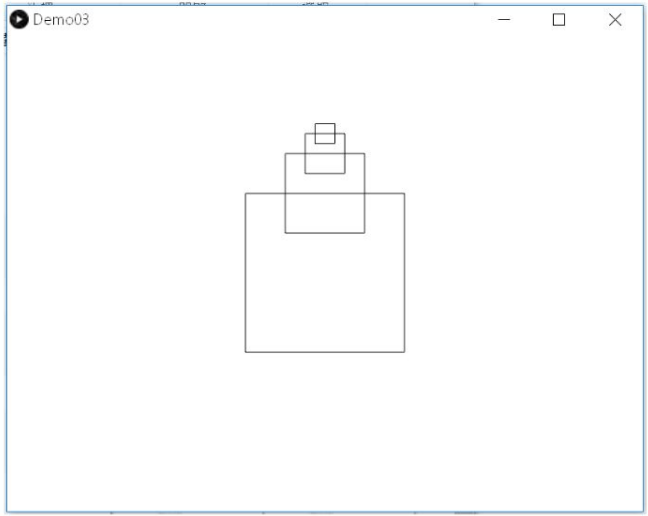
x=400, y=200, R=100, i=3

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=150, R=50, i=2

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=125, R=25, i = 1



```
drawSquare(width/2, height/2, 200, 4);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

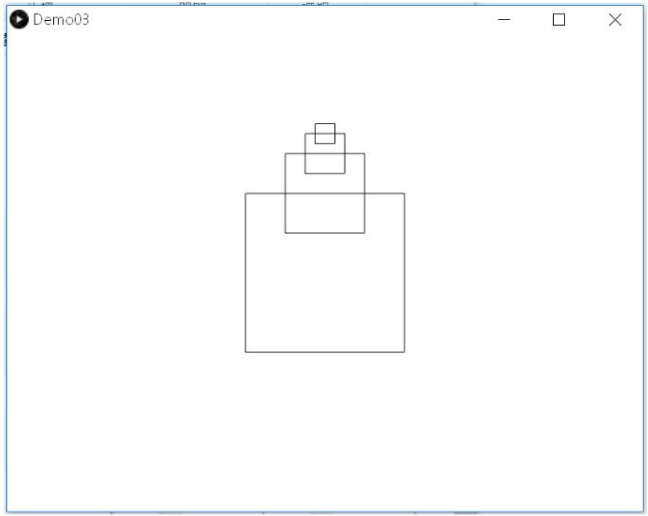
x=400, y=300, R=200, i=4

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=200, R=100, i=3

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=150, R=50, i=2



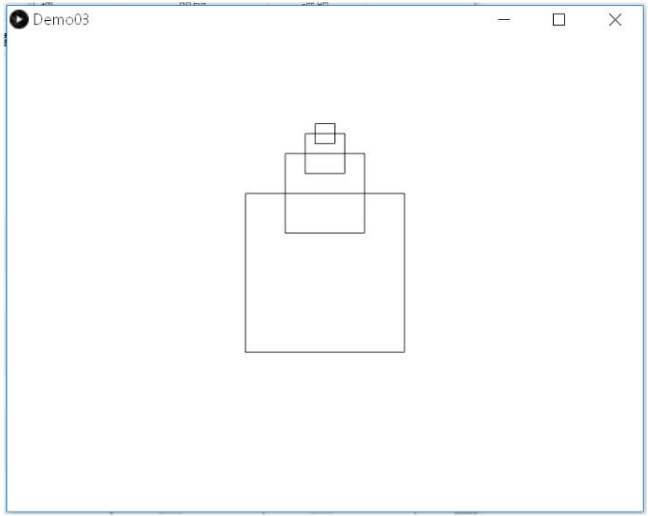
```
drawSquare(width/2, height/2, 200, 4);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

x=400, y=300, R=200, i=4

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

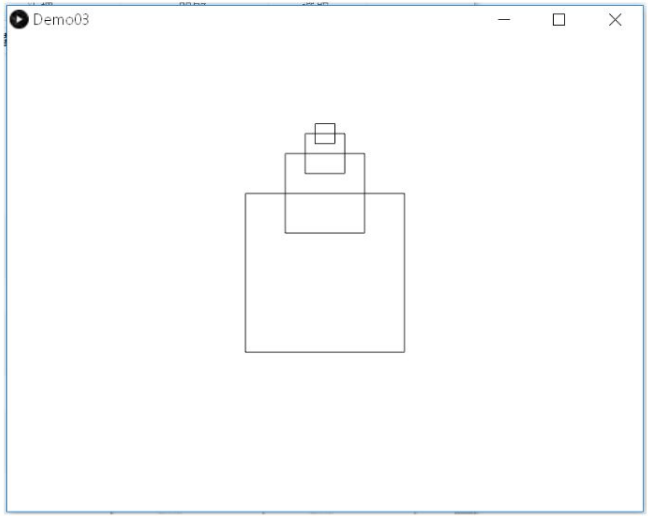
x=400, y=200, R=100, i=3



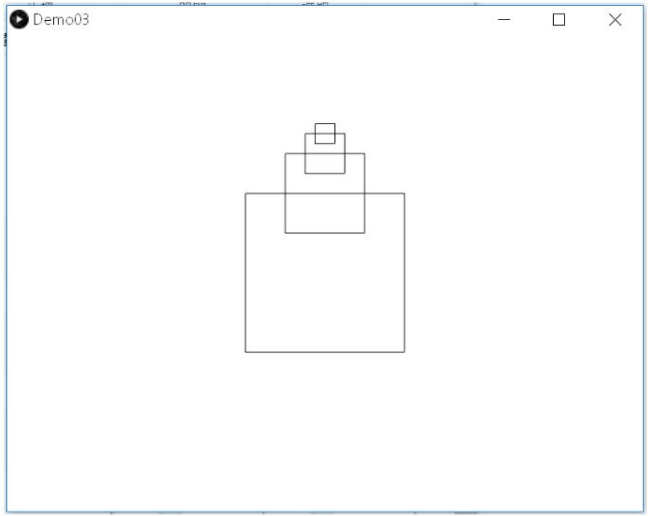
```
drawSquare(width/2, height/2, 200, 4);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    drawSquare(x, y-R/2, R/2, i-1);
  }
}
```

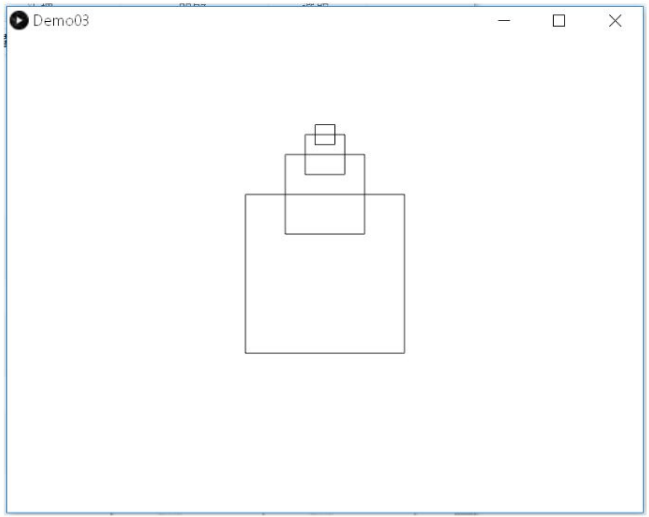
x=400, y=300, R=200, i=4

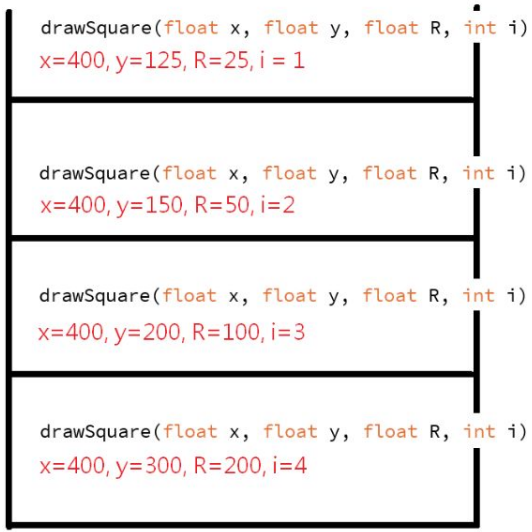


```
drawSquare(width/2, height/2, 200, 4);
```



```
drawSquare(width/2, height/2, 200, 4);
```





Stack

$n!$

$$n! = n \times (n-1) \times (n-2) \times \ldots\ldots \times 2 \times 1$$

DEMO 4
CODE

```
void setup() {
  size(800, 600, P3D);
  println(factorial(5));
}

void draw() {
  background(255);
}

int factorial(int n) {
  if (n <= 0) {
    return 1;
  } else {
    return n*factorial(n-1);
  }
}
```

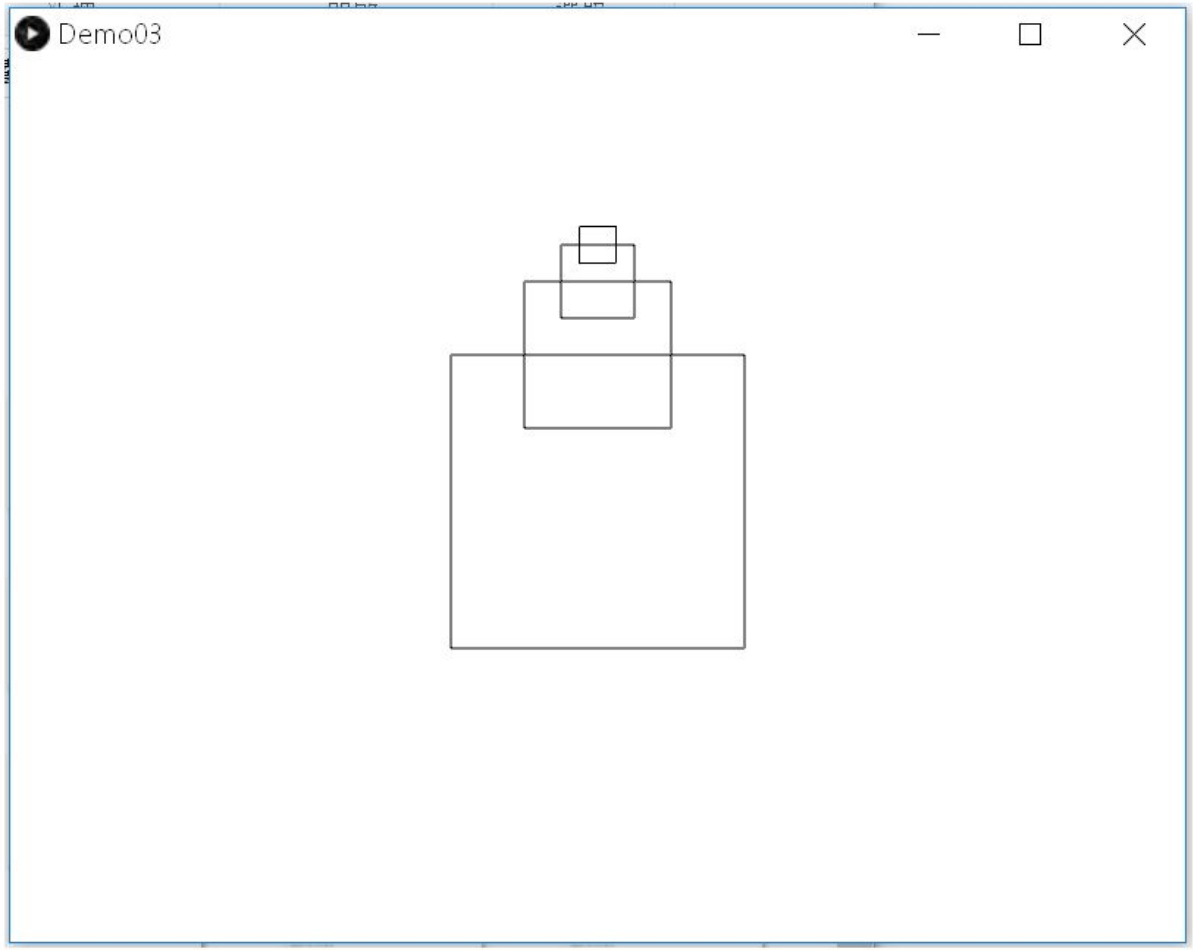
DEMO 5

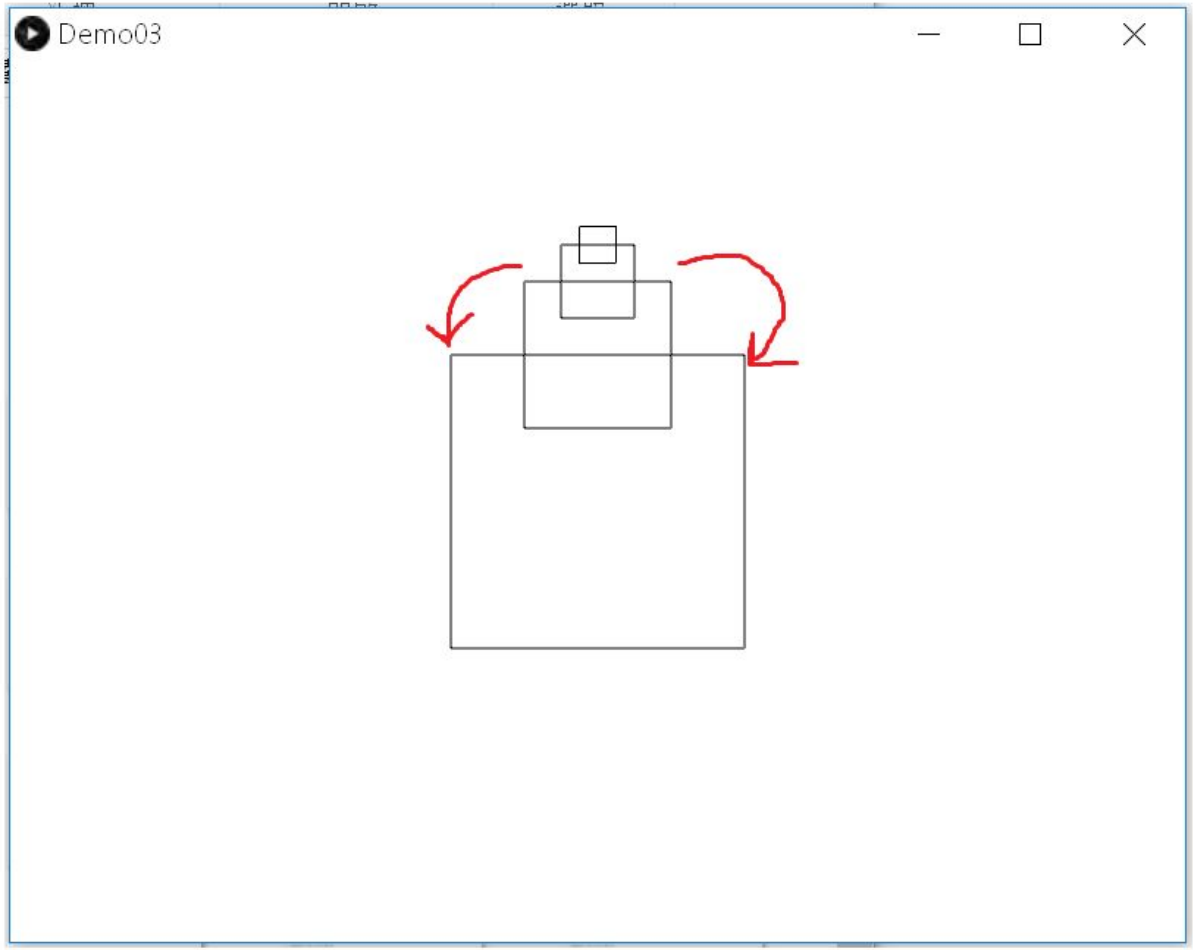
CODE (非遞迴)

```
void setup() {
  size(800, 600, P3D);
  println(factorial(5));
}

void draw() {
  background(255);
}

int factorial(int n) {
  int sum = 1;
  for (int i = n; i >= 1; i--) {
    sum = sum*i;
  }
  return sum;
}
```


Demo03


Demo03

DEMO 6
CODE

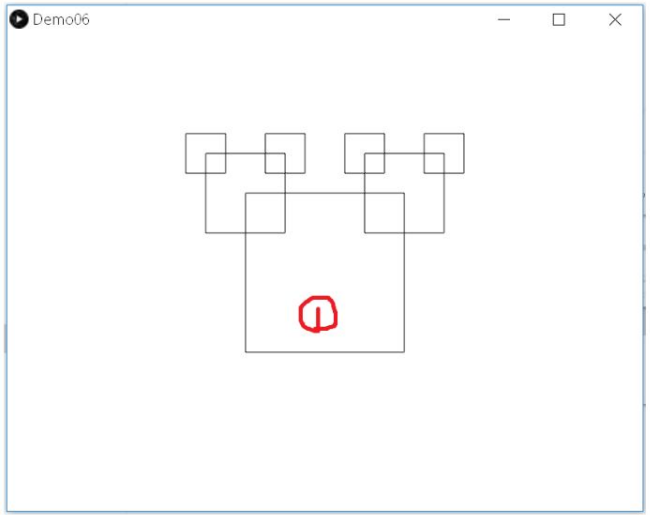
```
void setup() {
  size(800, 600, P3D);
  rectMode(CENTER);
}

void draw() {
  background(255);
  drawSquare(width/2, height/2, 200, 3);
}
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);
    drawSquare(x+R/2, y-R/2, R/2, i-1);
  }
}
```

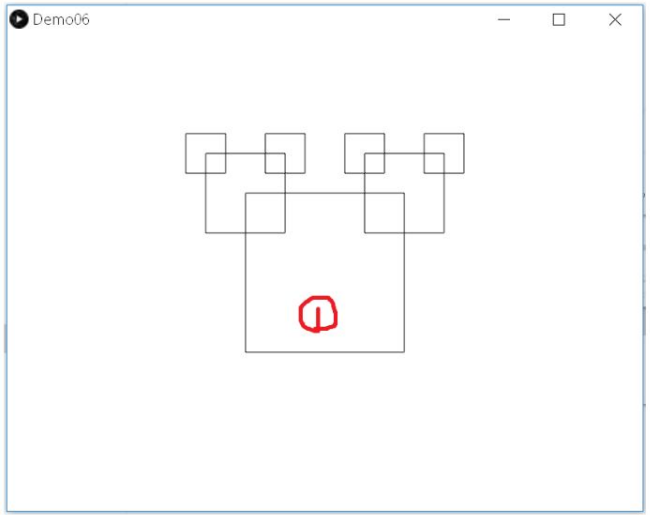
```
drawSquare(width/2, height/2, 200, 3);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);
    drawSquare(x+R/2, y-R/2, R/2, i-1);
  }
}
```



```
drawSquare(width/2, height/2, 200, 3);
```

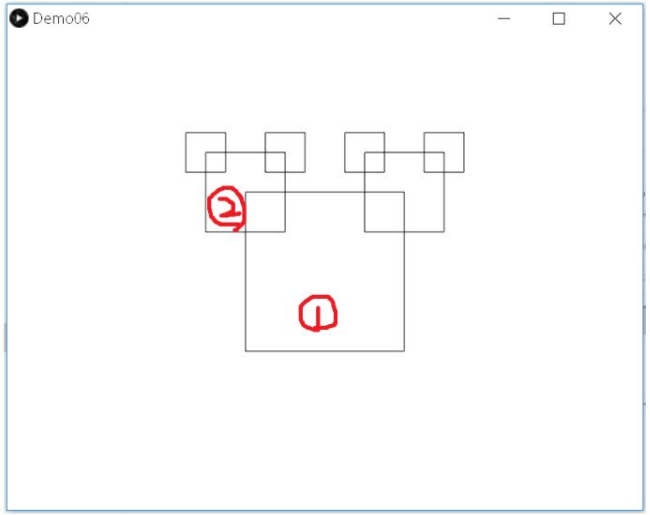
```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);
    drawSquare(x+R/2, y-R/2, R/2, i-1);
  }
}
```



```
drawSquare(width/2, height/2, 200, 3);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);
    drawSquare(x+R/2, y-R/2, R/2, i-1);
  }
}
```

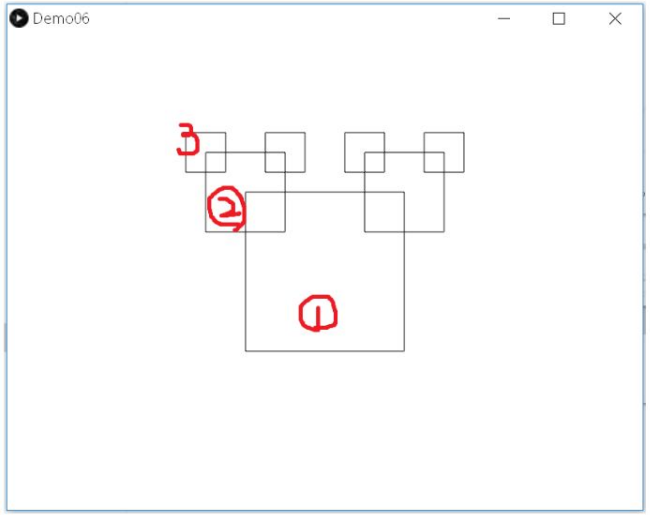
```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);
    drawSquare(x+R/2, y-R/2, R/2, i-1);
  }
}
```



```
drawSquare(width/2, height/2, 200, 3);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);
    drawSquare(x+R/2, y-R/2, R/2, i-1);
  }
}
```

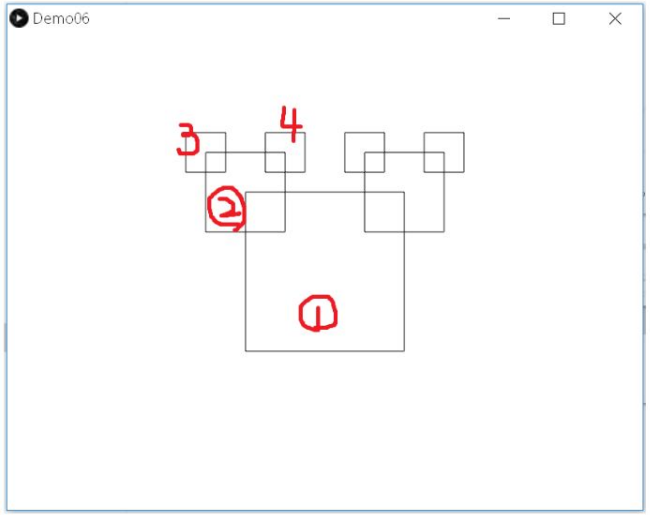
```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);
    drawSquare(x+R/2, y-R/2, R/2, i-1);
  }
}
```



```
drawSquare(width/2, height/2, 200, 3);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);
    drawSquare(x+R/2, y-R/2, R/2, i-1);
  }
}
```

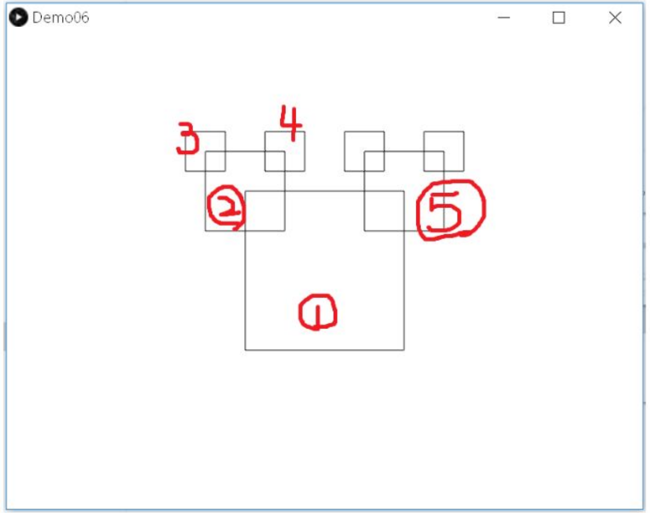
```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);
    drawSquare(x+R/2, y-R/2, R/2, i-1);
  }
}
```



```
drawSquare(width/2, height/2, 200, 3);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);
    drawSquare(x+R/2, y-R/2, R/2, i-1);
  }
}
```

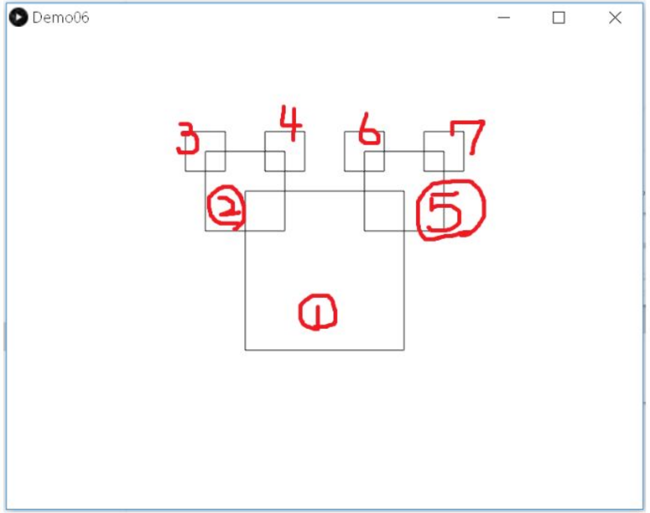
```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);
    drawSquare(x+R/2, y-R/2, R/2, i-1);
  }
}
```




```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);
    drawSquare(x+R/2, y-R/2, R/2, i-1);
  }
}
```

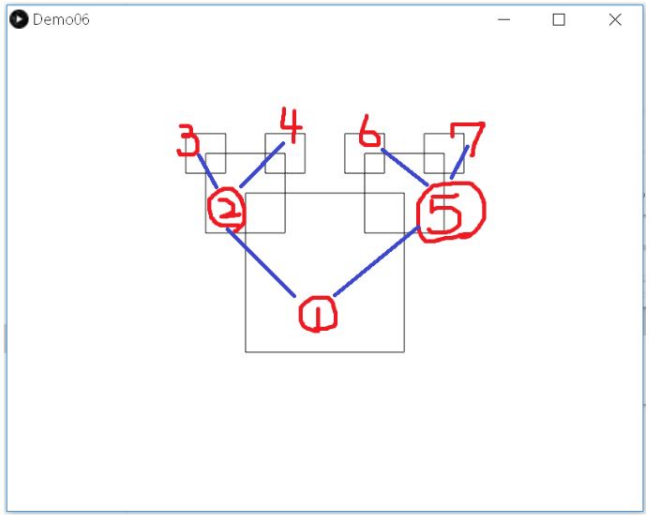
```
drawSquare(width/2, height/2, 200, 3);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);
    drawSquare(x+R/2, y-R/2, R/2, i-1);
  }
}
```




```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);
    drawSquare(x+R/2, y-R/2, R/2, i-1);
  }
}
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);
    drawSquare(x+R/2, y-R/2, R/2, i-1);
  }
}
```




```
drawSquare(width/2, height/2, 200, 3);
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);                          ①
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);
    drawSquare(x+R/2, y-R/2, R/2, i-1);
  }
}
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);                          ②
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);        3
    drawSquare(x+R/2, y-R/2, R/2, i-1);        4
  }
}
```

```
void drawSquare(float x, float y, float R, int i) {
  if (i > 0) {
    rect(x, y, R, R);                          ⑤
    //drawSquare(x, y-R/2, R/2, i-1);
    drawSquare(x-R/2, y-R/2, R/2, i-1);        6
    drawSquare(x+R/2, y-R/2, R/2, i-1);        7
  }
}
```

Example->Topics -> Fractals and L-Systems -> Tree