

15-104 Introduction to Computing for Creative Practice

Fall 2022

09 More Loops

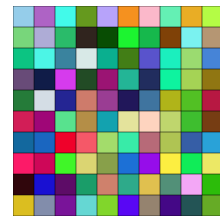
Instructor: Tom Cortina, tcortina@cs.cmu.edu, GHC 4117, 412-268-3514

1



Nested Loops

```
var side = 40;
function setup() {
  createCanvas(400,400);
  background(220);
}
function draw() {
  for (var row = 0; row < 10; row += 1) {
    for (var col = 0; col < 10; col += 1) {
      fill(random(0,256), random(0,256), random(0,256));
      rect(col*side, row*side, side, side);
    }
  }
  noLoop(); // not related to the for loop!
}
```



Why are **row** and **col**
flipped in the **rect**
statement?

2

Tracing Example

Shorter loops:

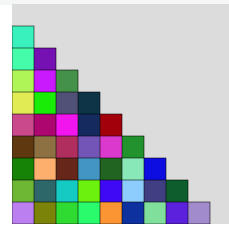
```
for (var row = 0; row < 5; row += 1) {
  for (var col = 0; col < 4; col += 1) {
    ...
  }
}
```

row	0	0	0	0	0	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	5
col	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	

3

Nested Loops (one change)

```
var side = 40;
function setup() {
  createCanvas(400,400);
  background(220);
}
function draw() {
  for (var row = 0; row < 10; row += 1) {
    for (var col = 0; col < row; col += 1) {
      fill(random(0,256), random(0,256), random(0,256));
      rect(col*side, row*side, side, side);
    }
  }
  noLoop(); // not related to the for loop!
}
```



4

Tracing Example

Shorter loops:

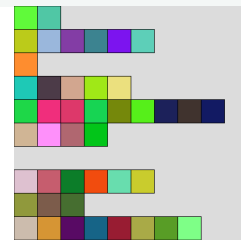
```
for (var row = 0; row < 5; row += 1) {
  for (var col = 0; col < row; col += 1) {
    ...
  }
}
```

row	0	1	1	2	2	2	3	3	3	3	4	4	4	4	4	5
col	0	0	1	0	1	2	0	1	2	3	0	1	2	3	4	

5

Nested Loops

```
var side = 40;
function setup() {
  createCanvas(400,400);
  background(220);
}
function draw() {
  for (var row = 0; row < 10; row += 1) {
    var limit = floor(random(0,11));
    for (var col = 0; col < limit; col += 1) {
      fill(random(0,256), random(0,256), random(0,256));
      rect(col*side, row*side, side, side);
    }
  }
  noLoop(); // not related to the for loop!
}
```



6

What is the result?

```
function setup() {
  createCanvas(480, 240);
  strokeWeight(2);
}

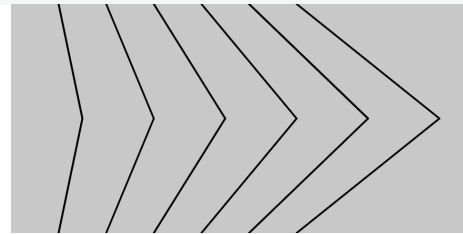
function draw() {
  background(200);
  for (var i = 50; i <= 300 ; i += 50) {
    line(i, 0, 1.5*i, 120);
    line(1.5*i, 120, i, 240);
  }
}
```

7

What is the result?

```
function setup() {
  createCanvas(480, 240);
  strokeWeight(2);
}

function draw() {
  background(200);
  for (var i = 50; i <= 300 ; i += 50) {
    line(i, 0, 1.5*i, 120);
    line(1.5*i, 120, i, 240);
  }
}
```



8



Draw this:

The canvas size is 400 X 300.

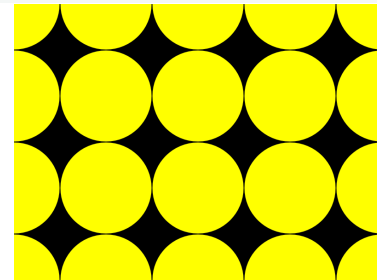
Each circle has a diameter of 100.

The background is black.

The circles are yellow.

Write a list of the centers of all of the circles.

What pattern(s) do you see? How many loops will you need?



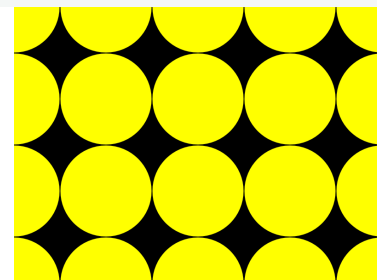
9



Draw this:

(0, 0)	(0, 200)
(100, 0)	(100, 200)
(200, 0)	(200, 200)
(300, 0)	(300, 200)
(400, 0)	(400, 200)


(0, 100)	(0, 300)
(100, 100)	(100, 300)
(200, 100)	(200, 300)
(300, 100)	(300, 300)
(400, 100)	(400, 300)



y changes less frequently, so
y should be the outer loop

x changes more frequently, so
x should be the inner loop

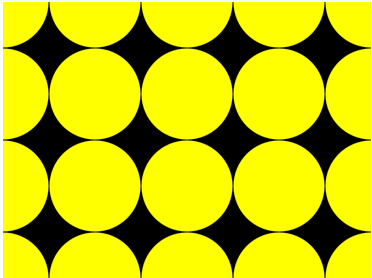
10




Draw this:

```
function setup() {
  createCanvas(400, 300);
  background(0);
}

function draw() {
  fill(255, 255, 0); // yellow
  for (var y = 0; y <= 300; y += 100) {
    for (var x = 0; x <= 400; x += 100) {
      circle(x, y, 100);
    }
  }
}
```



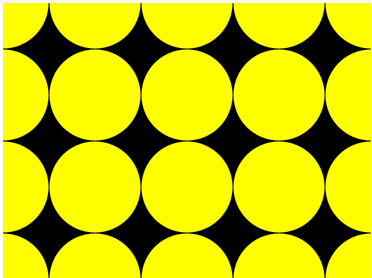
11



Draw this:

```
function setup() {
  createCanvas(400, 300);
  background(0);
}

function draw() {
  fill(255, 255, 0); // yellow
  for (var row = 0; row <= 3; row += 1)
    for (var col = 0; col <= 4; col += 1) {
      circle(col*100, row*100, 100);
    }
}
```



12

Draw this:

```
function setup() {
  createCanvas(400, 300);
  background(0);
}

function draw() {
  fill(255, 255, 0); // yellow
  for (var row = 0; row < 4; row += 1)
    for (var col = 0; col < 5; col += 1) {
      circle(col*100, row*100, 100);
    }
}
```

13

Chess Board


```
var grayvalue = 255;

function setup() {
  createCanvas(320, 320);
}

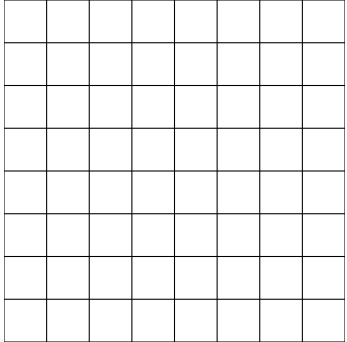
function draw() {
  // to be completed
}
```

14

Chess Board




```
function draw() {
  rectMode(CENTER);
  for (var y = 20; y <= 300; y += 40) {
    for (var x = 20; x <= 300; x += 40) {
      fill(grayvalue);
      rect(x, y, 40, 40);
    }
  }
}
```



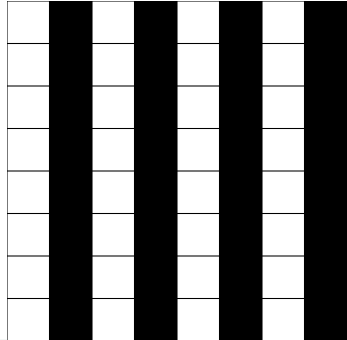
15

Chess Board



```
function draw() {
  rectMode(CENTER);
  for (var y = 20; y <= 300; y += 40) {
    for (var x = 20; x <= 300; x += 40) {
      fill(grayvalue);
      rect(x, y, 40, 40);
      grayvalue = 255 - grayvalue;
    }
  }
}
```

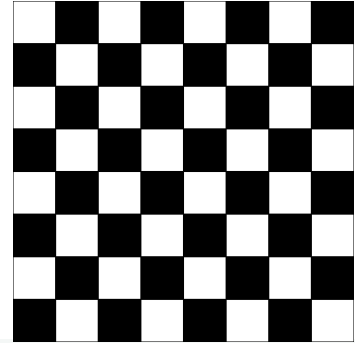
This instruction flips grayvalue from 0 to 255 or from 255 to 0. Do you see why?



16

Chess Board

```
function draw() {
  rectMode(CENTER);
  for (var y = 20; y <= 300; y += 40) {
    for (var x = 20; x <= 300; x += 40) {
      fill(grayvalue);
      rect(x, y, 40, 40);
      grayvalue = 255 - grayvalue;
    }
    grayvalue = 255 - grayvalue;
  }
}
```

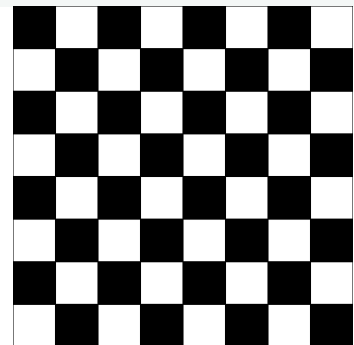


17

Another Way

```
function setup() {
  createCanvas(320, 320);
}

function draw() {
  rectMode(CORNER);
  for (var row = 0; row < 8; row += 1) {
    for (var col = 0; col < 8; col += 1) {
      fill( ((row+col) % 2) * 255 );
      rect(col*40, row*40, 40, 40);
    }
  }
}
```



Modulo operator $x \% y$ (for $x > 0, y > 0$): Divide x by y and keep the remainder.
 When $y = 2$, there are only two remainders, 0 (for even x) and 1 (for odd x).
 So the two fills are $0 * 255 = 0$ (black) when $row+col$ is even
 and $1 * 255 = 255$ (white) when $row+col$ is odd.

18

