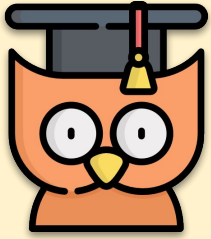


Lesson X - For Loops



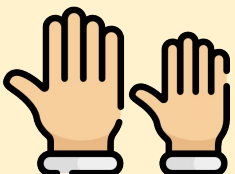
Learning Outcomes:

- We will learn about **for** loops and their uses.
- We will learn how **for** loops attributes can be changed
- We will learn how to use a **for** loop in a practical sense by implementing a lives system in a game.



Revision topics before starting make sure you know:

- ✓ The x-axis & y-axis in Processing
- ✓ Variables
- ✓ Drawing targets
- ✓ Keyboard input
- ✓ Image loading

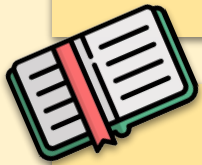


REMEMBER: Put up your hand. We love to help!



Introduction to Loops

When we are writing code, sometimes we would like to **repeat certain blocks of code multiple times** until some condition is met (the user presses a key, it has reached a certain limit, etc.). Loops are used to save you time so you don't have to write the same piece of code over and over again. One of the most commonly used type of loop in coding are **for loops**.



Anatomy of a **for** loop

The image below illustrates the syntax of a **for** loop:

```
for (int i = 0; i < 10; i++) {  
    println("Random Number " + i + ": " + random(0, 1000));  
}
```

The diagram shows the following components of the for loop syntax:

- Declare counter**: Points to `int i = 0`
- Set counter limit**: Points to `i < 10`
- Increment counter**: Points to `i++`
- Code block**: Points to the block of code inside the curly braces: `println("Random Number " + i + ": " + random(0, 1000));`

We start by declaring a counter `i`; we check if it's less than 10; if it is, we do the code in the block and then increment the counter (`i++`); then we check again, and continue this loop until "`i < 10`" is no longer true.

The code block in this case will print out 10 random values between 0 and 1000, numbering each random value 0 to 9. The console output can be seen below:

```
Random Number 0: 636.5524  
Random Number 1: 873.3519  
Random Number 2: 424.71402  
Random Number 3: 826.12085  
Random Number 4: 259.87137  
Random Number 5: 465.68362  
Random Number 6: 896.1115  
Random Number 7: 703.57385  
Random Number 8: 768.21576  
Random Number 9: 954.2379
```



Let's get coding!

- 1 Create a new Processing sketch and call it **For_Loops_Lesson1**.
- 2 Draw **twenty** circles of **random sizes**, in **random positions** and with **random fills**.



Expert Tip

Remember, **void draw()** is a loop that runs continuously. So, if you have a **for** loop in your **void draw()** it will repeat (which you don't want in this case). Call the function **noLoop()** after your **for** loop has finished so **void draw()** will stop running.

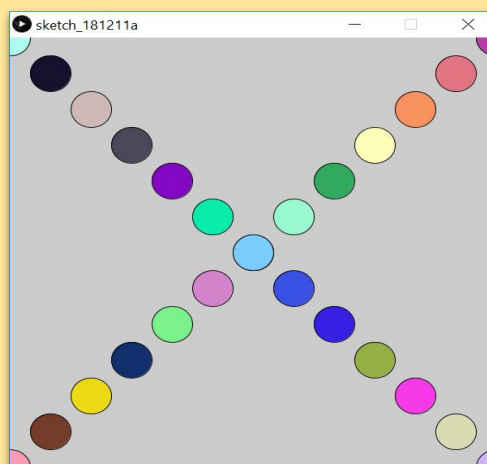
- 3 Change your code to draw circles of equal size, going **diagonally right & down** on the canvas.

Hint: Instead of incrementing your **for** loop by 1 (**i++**), you can increment it by another value, e.g. **i += 50**.



- 4 Add more ellipses going **diagonally left & down**.

Hint: Instead of increasing your **for** loop, you can decrease it by some value, e.g. **i -= 50**.



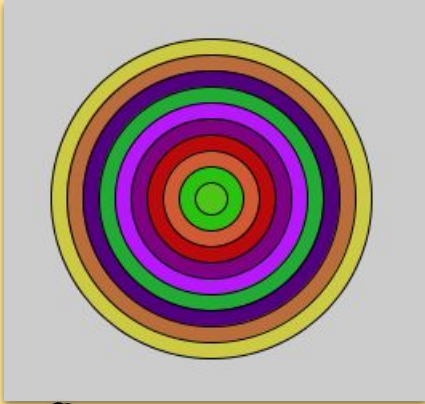
Be Careful!

When decreasing the counter, start the counter from the maximum value and set the limit to the minimum value.



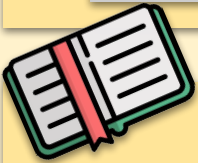
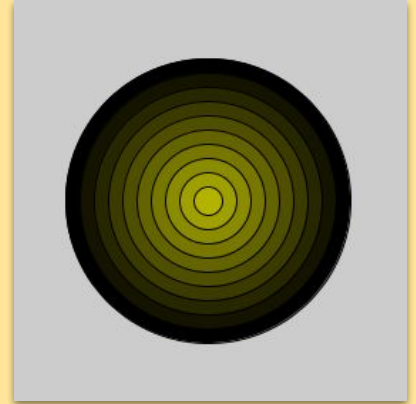
Challenge Task

Create a target using a **for** loop.



Challenge Task

Using the same **for** loop as the previous challenge, make the target's fill fade from black into yellow.



Changing the Length of **for** Loops

In the previous section we set a counter limit that would exit the **for** loop when the condition is no longer true, e.g. i is less than 10 ($i < 10$). Sometimes we may want to change the counter limit while the program is running.

```

int numCircles;

void setup() {
  size(800, 800);

  numCircles = 1;
}

void draw() {
  background(0);

  for (int i = 0; i < numCircles; i++) {
    ellipse(i*50, height/2, 50, 50);
  }

  if (keyPressed) {
    if (keyCode == LEFT && numCircles > 1) {
      numCircles--;
    } else if (keyCode == RIGHT) {
      numCircles++;
    }
  }
}

```

In the example, the **for** loop will draw circles going across the canvas. The number of circles drawn will depend on the variable **numCircles**. We are increasing/decreasing **numCircles** with user input so the number of times the **for** loop executes will vary.



Let's get coding!

- 1 Create a new Processing sketch and call it **For_Loops_Lesson2**.
- 2 Draw a target using a **for** loop, using a variable to set how many times the **for** loop will execute.
- 3 Add user input so the user can increase/decrease the number of times the **for** loop executes (and the number of circles being drawn in the **for** loop).



Be Careful!

Ensure that the variable you use to set the counter limit does not go below 1. Otherwise, the condition `i < 0` will cause the program to crash.



Challenge Task

Create a simple game that involves the following:

- A player sprite that moves around the screen using user input.
- The player starts off with **five** “lives”.
- The “lives” are represented by love hearts in the top-left of the screen. You can find an image of a love heart to use online. The love hearts should be displayed using a **for** loop, so the **image()** function should only be written once inside a **for** loop.
- Something happens that causes the player to lose “lives”, e.g. the player hits of the edge of the screen, the player collides with an enemy, etc.

