



Crash Course: Repetition Structures 1



ST. MARY'S HIGH SCHOOL



In this crash course

- What is a loop?
- Why use a loop?
 - Example 1
- Summary



What is a loop?

- As the name suggests, repetition structures, better known as loops, are used to repeat a programming action or statement as long as one or more conditions are satisfied
- There are three main types of loops: the while loop, the for loop and the do while loop (the first two of which are more commonly used)
- In general, a loop has three main components: an initialization expression, a test expression, and an updating expression



What is a loop?

- The initialization expression serves as a starting or entry point to the loop and indicates at which value you should begin iterating
- The test expression is a condition that must be checked at each iteration, which controls the loop and determines when it should stop
- The updating expression is necessary to repeat the cycle and brings us closer to the test expression or condition
- This will make more sense in a moment



Why use a loop?

- Loops are very useful in that they make your code more modular, cleaner and shorter
- A simple repetition structure can be very efficient and replace many lines of code as shown in the next example
- Loops are also used to navigate through data structures such as arrays and strings so you can access their elements or characters



Why use a loop? – Example 1

- Imagine that you have to draw several vertical lines in Processing across a 500 x 500 unit sketching window at intervals of 50 units
- How would you do this?



Why use a loop? – Example 1

- You could call the line function every time you want to draw the line

The screenshot shows the Processing IDE interface. The code editor on the left contains the following code:

```
1 //Example 1 - Without a loop
2
3 size(500, 500); //Setting the background size
4 background(160); //Setting the background colour
5
6 //Drawing each individual line, one after the other
7 line(0, 0, 0, 500);
8 line(50, 0, 50, 500);
9 line(100, 0, 100, 500);
10 line(150, 0, 150, 500);
11 line(200, 0, 200, 500);
12 line(250, 0, 250, 500);
13 line(300, 0, 300, 500);
14 line(350, 0, 350, 500);
15 line(400, 0, 400, 500);
16 line(450, 0, 450, 500);
17 line(500, 0, 500, 500);
18
19
20
```

The window titled 'Example1' on the right displays the output of the code, which is a 500x500 pixel canvas with a light gray background. It contains 17 vertical gray lines of varying widths, starting from 0 pixels wide at the left edge and increasing to 500 pixels wide at the right edge. The lines are positioned at 50-pixel intervals along the x-axis.



Why use a loop? – Example 1

- Or a you could use a loop (structure explained in more detail later)

The screenshot shows the Processing IDE interface. The code editor on the left contains the following Java code:

```
1 //Example 1 - With a loop
2
3 size(500, 500); //Setting the background size
4 background(160); //Setting the background colour
5
6 //Using a for loop to draw the lines
7 //The inner-workings of this will be explained in a few slides
8 for(int i = 0; i <= 500; i+=50)
9 {
10   line(i, 500, i, 0);
11 }
12
13
14
15
16
17
18
19
20
```

The IDE window title is 'Example1 | Processing 3.4'. The right side of the image shows a preview window titled 'Example1' displaying the result of the code: a gray background with 11 vertical black lines of equal height and width, spaced evenly across the width of the window.



Why use a loop? – Example 1

- Using an arbitrary variable i , the loop begins at $i = 0$, increments 50 units at a time until 500 and is reached, then stops
- In this case, $i = 0$ is the initialization expression, the value we start at
- $i \leq 500$ is the test expression, which is what we check at each iteration (once $i > 500$, the loop can stop)
- $i += 50$ is the update expression, which is what needs to happen to get us to the next iteration (in our case i has to increment by 50 each time)
- In this example, our loop completed 11 iterations and the variable i took on the following values: 0, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500



Summary

- In this crash course, we were introduced to repetition structures
- Repetition structures, also known as loops, are used to repeat a programming action or task
- Every loop consists of an initialization expression, a test expression (condition) and an updating expression
- Loops are useful in that they can replace many lines of code, making your program more clean and compact