

Iteration in Programming

loops

Produced Dr. Siobhán Drohan
by: Mr. Colm Dunphy
 Mr. Diarmuid O'Connor

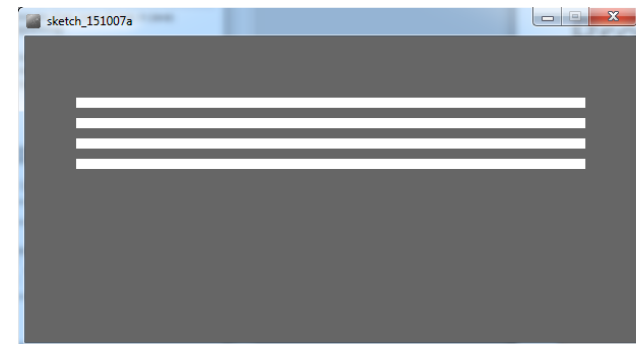


Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
<http://www.wit.ie/>

Problem:

- Draw a rectangle 4 times that has a gap of 10 pixels between each one.



Form of loop

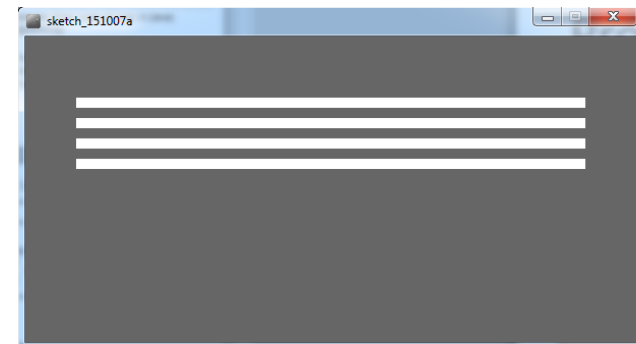
- Draw a rectangle 4 times that has a gap of 10 pixels between each one.
 - Without loop:

```
rect(50, 60, 500, 10);
```

```
rect(50, 80, 500, 10);
```

```
rect(50, 100, 500, 10);
```

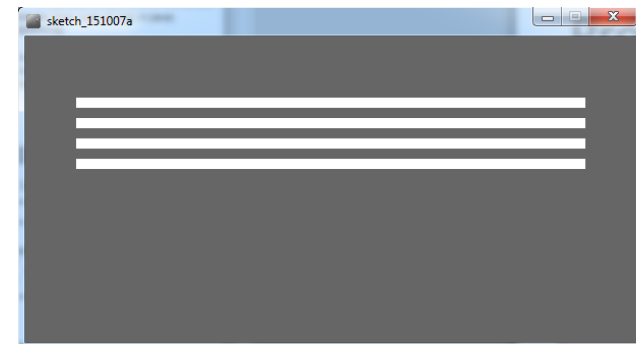
```
rect(50, 120, 500, 10);
```



Form of loop

- Draw a rectangle 4 times that has a gap of 10 pixels between each one.
 - With a loop:

```
rect(50, yCoordinate, 500, 10);
```
 - But do this 4 times
(adding 20 onto the yCoordinate variable each time).



Loops in Programming

- There are three types of loop in (Java) programming:
 - **While** loops
 - **For** loops
 - **Do While** loops

WHILE LOOPS

Construction of **while** loop

Declare and initialise **loop control variable (LCV)**

while (boolean condition based on **LCV** is true)

{

 “do the job to be repeated”

 “update the **LCV**”

}

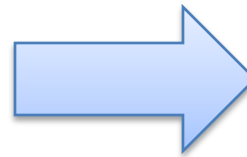
This structure should always be used

Simplen **while** Loop

This basic while loop, produces this output.

```
int i = 1;

while (i <=5)
{
    println("Hello World");
    i++;
}
```

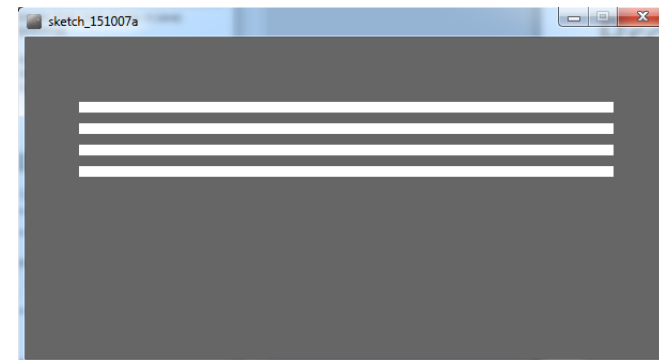


```
Hello World
Hello World
Hello World
Hello World
Hello World
```


Processing Example 2.13

```
Example_2_13 | Processing 3.3.6
File Edit Sketch Debug Tools Help

Example_2_13
1 int yCoordinate = 60;
2
3 size(600, 300);
4 background(102);
5 fill(255);
6 noStroke();
7
8 int i = 0;
9 while(i < 4)
10 {
11     rect(50, yCoordinate, 500, 10);
12     yCoordinate += 20;
13     i++;
14 }
15
```

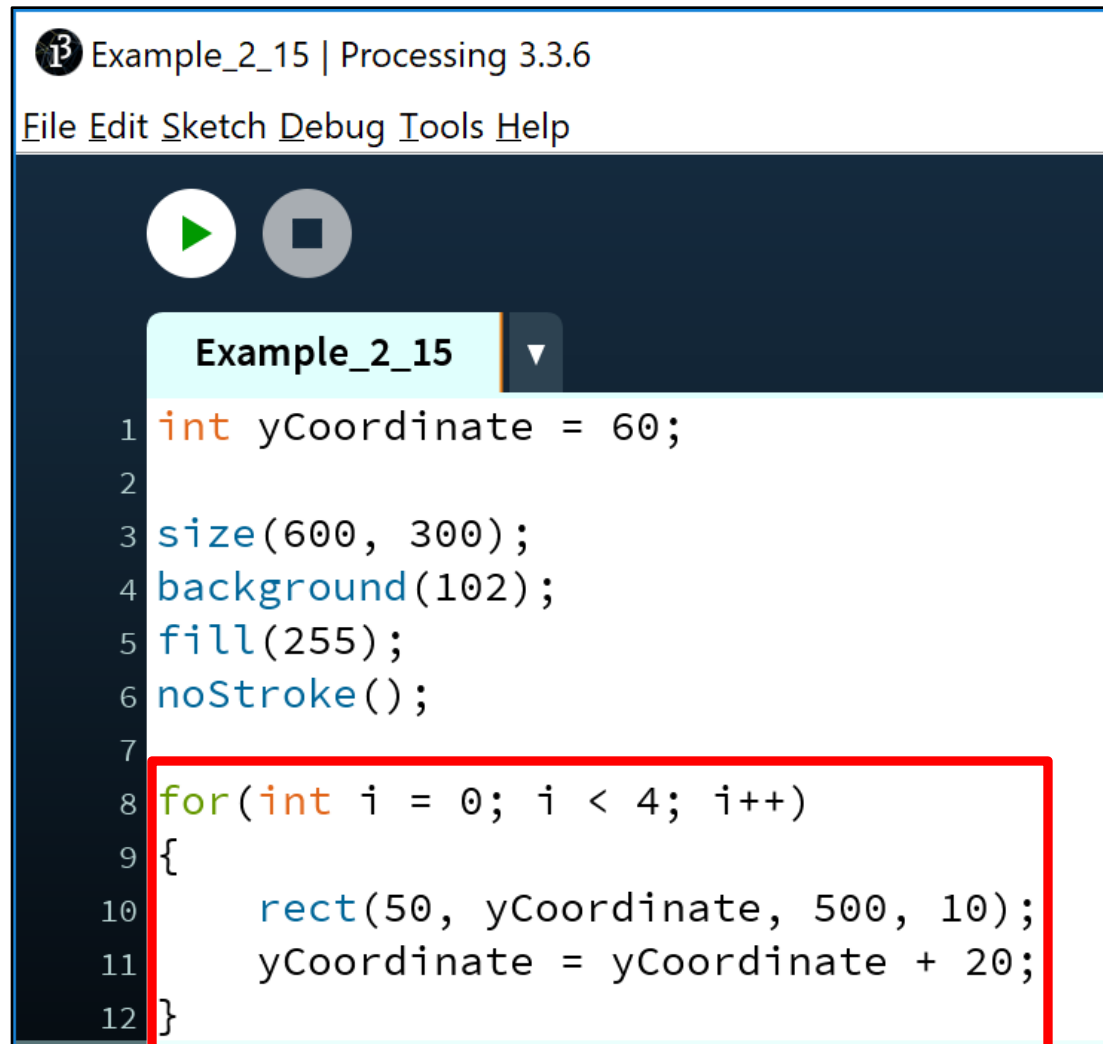


Some Study Exercises

1. Change the code so that “Hello World” is printed out 10 times.
2. Change the code so that the numbers from 1 to 10 (inclusive) are printed out, one line at a time.
3. Change the code so that the numbers from 10 to 1 are printed out.

FOR LOOPS

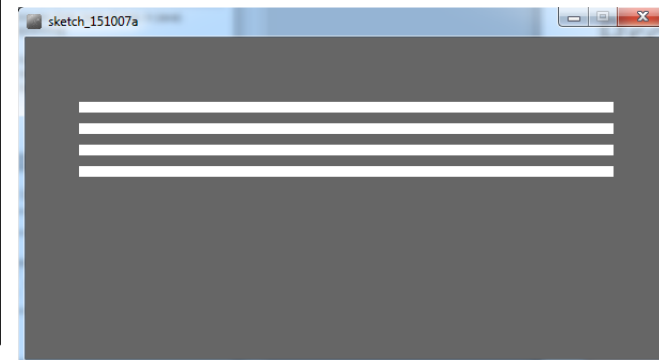
Processing Example 2.15



```
Example_2_15 | Processing 3.3.6
File Edit Sketch Debug Tools Help

Example_2_15
1 int yCoordinate = 60;
2
3 size(600, 300);
4 background(102);
5 fill(255);
6 noStroke();
7
8 for(int i = 0; i < 4; i++)
9 {
10     rect(50, yCoordinate, 500, 10);
11     yCoordinate = yCoordinate + 20;
12 }
```

This code does the same as the previous slide, except that we use a different loop: **for**



For loop syntax

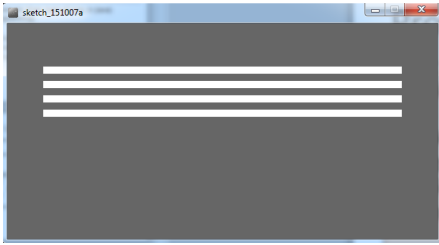
```
for(int i = 0; i < 4; i++)
```

```
for(initialization; boolean condition; post-body action)  
{  
    statements to be repeated  
}
```

For loop syntax

```
for(int i = 0; i < 4; i++)
```

initialization	<code>int i = 0;</code>	Initialise a loop control variable (LCV) e.g. i. It can include a variable declaration.
boolean condition	<code>i < 4;</code>	Is a valid boolean condition that typically tests the loop control variable (LCV).
post-body action	<code>i++</code>	A change to the loop control variable (LCV). Contains an assignment statement.



for versus while

Variable **i** is the Loop Control Variable (**LCV**). It must be:

- initialised, tested and changed.

int i = 0 is the **initialisation**.

i < 4 is the boolean condition i.e. the **test**

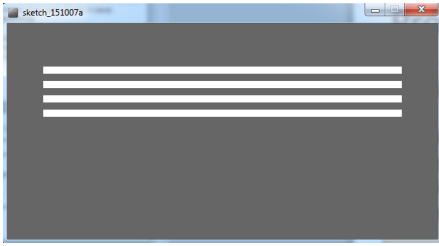
i++ is the post-body action i.e. the **change**.

Example 2.15
(**for** loop)

```
for(int i = 0; i < 4; i++)  
{  
  rect(50, yCoordinate, 500, 10);  
  yCoordinate += 20;  
}
```

Example 2.13
(**while** loop)

```
int i = 0;  
while(i < 4)  
{  
  rect(50, yCoordinate, 500, 10);  
  yCoordinate += 20;  
  i++;  
}
```



for versus while

Example 2.15
(**for** loop)

```
for(int i = 0; i < 4; i++)  
{  
  rect(50, yCoordinate, 500, 10);  
  yCoordinate += 20;  
}
```

Example 2.13
(**while** loop)

```
int i = 0;  
while(i < 4)  
{  
  rect(50, yCoordinate, 500, 10);  
  yCoordinate += 20;  
  i++;  
}
```

Removing the code to do the

- initialisation, test and change

What we are left with is the code that you want looped / repeated

i.e. draw the rectangle and update the yCoordinate variable.

Note:

it is exactly the same for each loop