Programming Assignment

Lesson 2

CISC 071

By

<Jeremy Reuwer>

Date: <02/06/2019>

Purpose

* To learn about loops and functions

Rubric

* Correctness: 2 Points. Program should work as specified
* Input/Output: 2 Points. Show the inputs and outputs to the program. If Applicable multiple examples needed
* Coding style/Comments: 1 Points.

Project Assignment

* Use the setup function to set the serial monitor and print “Serial monitor setup done”
* Use the loop function to print “Beginning the loop” then after a 2 second delay print “Middle of the loop” then after another 2 second delay print “End of the loop”

Learning Notes:

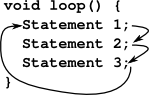
## The Main Loop

As seen in Lesson 1, an Arduino sketch consists of two main functions called **setup()** and **loop()**. The loop() function is the main loop in the Arduino sketch. After statements that only need to be run once have finished being executed in the setup() function, program execution starts in the loop() function.

Once program execution has started in the main loop, the statements in the main loop will be executed continuously until the Arduino is switched off or reset. The main loop is where the actual operational functionality of the Arduino takes place – for example, if the Arduino is programmed to be a flashing light sequencer, then the flashing light functionality will be placed in the main loop.

### Why is it Called a Loop?

Statements in the loop() function will be executed from top to bottom, until the bottom of the loop() function is reached. When the bottom of the loop function is reached, statements are executed from the top of the loop() function again, thus completing the "loop" as shown in the image below.



## More on Functions

Each statement in setup() and loop() consists of a function being called – i.e. being called means that it is executed or run.

### The delay() Function

When the **delay()** function is called in the statement **delay(2000);** then the delay function causes a waiting period of 2 seconds (2000 milliseconds – there are 1000 milliseconds in one second, also written 1000ms). The time of the delay can be changed by passing a different value to delay(), e.g. 3000 will cause a 3 second delay: **delay(3000);**

### The println() Function

The **println()** function sends text out of the serial / USB port of the Arduino and is displayed in the serial monitor window.

The println() function is different from the delay() function in that it has **Serial** and a dot (.) before it: **Serial.println("Text to print.");**

The reason for this notation (Serial.*function\_name()*) is because the function acts on the serial port or Serial object. You will notice in setup() that **Serial.begin()** is called. This is the begin() function acting on the serial port – in this case to set it to the desired speed.

These functions that are preceded by an object name (e.g. Serial) are called "methods" in object oriented programming.

## A Summary of Functions

The following will hopefully clear up what functions are and the terminology used with them. A deeper understanding of functions will only be possible once we start writing our own functions.

### setup() and loop()

**setup()** and **loop()** are two special functions that form part of the structure of an Arduino sketch.

We are actually writing these special functions by giving them a function body (between the opening and closing braces: **{}**) and writing statements in the function body.

The statements in these functions in the above sketch were calling pre-existing functions that perform the tasks that we want, e.g. set up the serial port speed, cause a time delay, write text to the serial monitor window.

The setup() and loop() functions are automatically called at the right time because they are special Arduino functions.

### Calling Functions

By calling or using pre-existing functions, we are using code that someone else has already written.

The delay() function has a function body that contains statements that cause it to perform a delay. We do not see these statements or the function body because they are either part of the Arduino programming language or exist in an external function library.

### Passing a Value to a Function

When a value (e.g. a number or text string) is used by a function, we must pass the value to the function.

#### Passing a Value to the delay() Function

We call the delay() function in the sketch as in the following statement:

**delay(2000);**

The delay value in milliseconds (2000) is said to be passed to the function.

#### Passing a Value to the println() Function

We pass a text string to the println() function as shown in this statement:

**Serial.println("Executing instructions in main loop.");**

We must pass the text string to the function so that the function knows what to send out of the serial / USB port. The text between the opening and closing quotation marks (**""**) is known as a string in programming.

**For further details refer to the Arduino programming reference guide**

<https://playground.arduino.cc/uploads/Main/arduino_notebook_v1-1.pdf>

Program

<

//Author:: JReuwer

//Date:: 2/6/2019

//Version:: 1.0

//Lesson:: 2

void setup() {

Serial.begin(9600);

Serial.println("Serial monitor setup done");

}

void loop() {

Serial.println("Beginning the loop");

delay(2000);

Serial.println("Middle of the loop");

delay(2000);

Serial.println("End of the loop");

}

\>

Inputs/Outputs

<

Serial monitor setup done

Beginning the loop

Middle of the loop

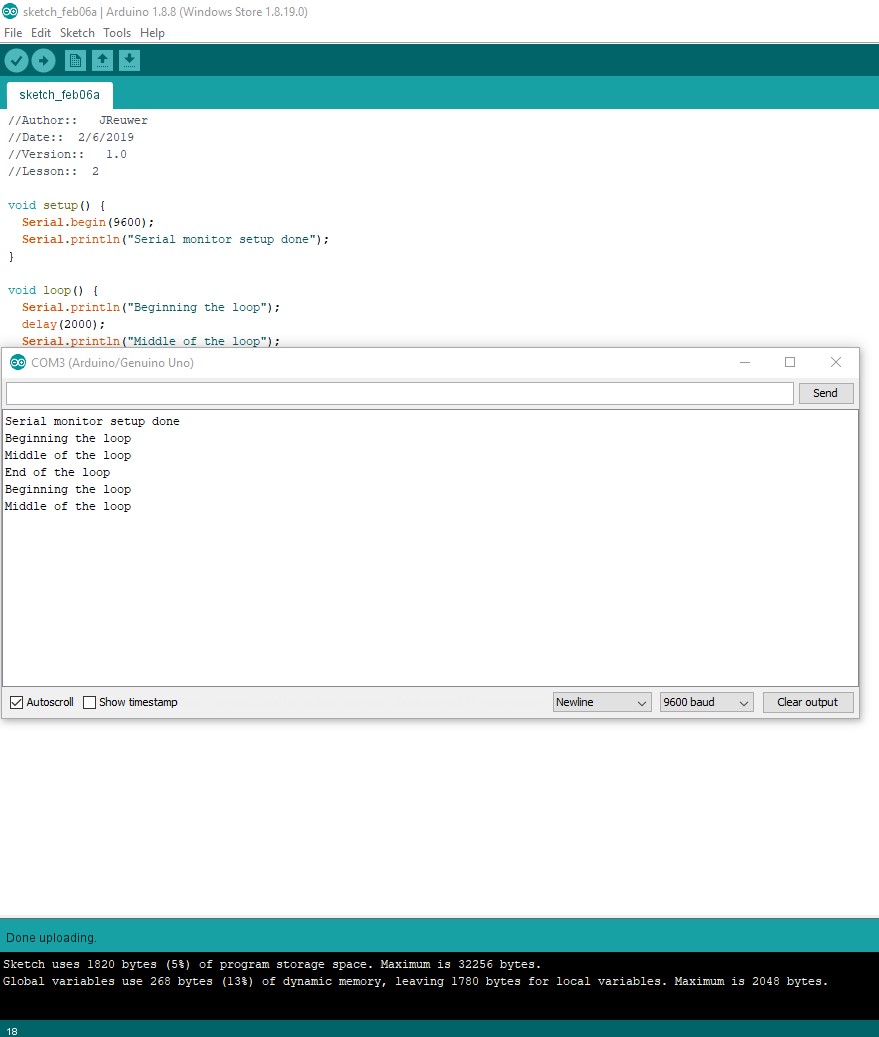
End of the loop

Beginning the loop

Middle of the loop

End of the loop

Beginning the loop…

…

>