Hyperion Development

# Task: Beginner Control Structures - While Loop

www.hyperiondev.com

# Introduction

**Welcome to The  Beginner Control Structures - While Loop Task!**

In this task you will be sequentially exposed to loop statements in order to understand how they can be utilised in reducing lengthy code, preventing coding errors, as well as paving the way towards code reusability. This begins with the while loop, which is the simplest loop of the group.

# Connect with your mentor

CONNECT

**Remember that with our courses - you're not alone!** You can contact your mentor to get support on any aspect of your course.

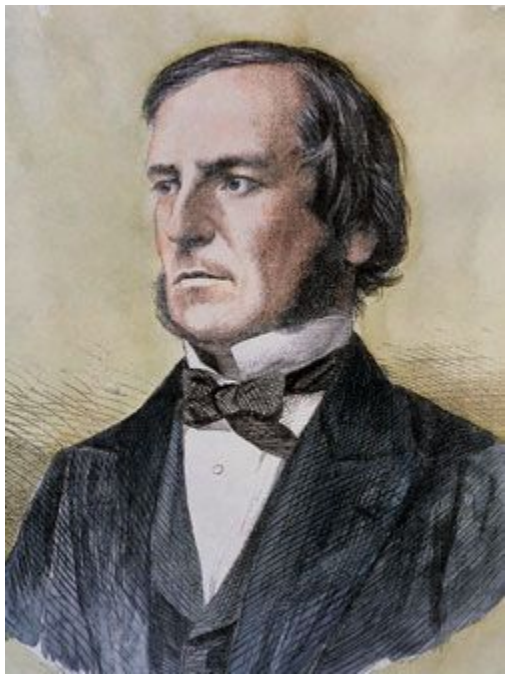
The best way to get help is to login to **www.hyperiondev.com/support** to start a chat with your mentor. You can also schedule a call or get support via email.

Your mentor is happy to offer you support that is tailored to your individual career or education needs. Do not hesitate to ask a question or for additional support!

## A note from the Hyperion Team...

*So by now you should be familiar with the Boolean datatype and the important role Boolean algebra plays in computer programming, but why is it called a Boolean datatype? Well, it was actually named after George Boole (1815 - 1864) an English mathematician who helped establish modern symbolic logic and devised Boolean algebra. Boole is best know as the author of The Laws of Thought (1854) which contains Boolean algebra.*

*Boole was born in Lincoln, Lincolnshire, England. His father, a struggling shoemaker, encouraged him to take an interest in mathematics. Boole went to an elementary school and commercial school for a short time, but he mostly educated himself. By the age of fifteen Boole began teaching and at age nineteen set up his own school in Lincoln. Boole was appointed as a professor of mathematics at Queen's College,Cork in 1849, despite not holding any university degree.*



***George Boole***

-The Hyperion Team

### What is a while Loop?

A while loop is the most general form of loop statements. The while statement repeats its action until the controlling condition becomes false. In other words, the statements indented in the loop repeatedly execute "while" the condition is true (hence the name). The while statement begins with the keyword *while* followed by a boolean expression. The expression is tested before beginning each iteration or repetition. If the test is true then the program passes control to the indented statements in the loop body; if false, the control passes to the first  statement after the body.

Syntax:

```
while boolean expression:
   statement(s)
```

The following code shows a while statement which sums successive even integers 2 + 4 + 6 + 8 + ... until the total is greater than 250. An update statement increments i by 2 so that it becomes the next even integer. This is an event-controlled loop (as opposed to counter-controlled loops like the for loop) because iterations continue until some non-counter-related condition (event) stops the process.

.

```
sum1 = 0
i = 2                    # initial even integer for the sum
while sum1 <= 250:
        sum1 += i
        i += 2           # update statement
        print sum1
```

Compile and run the example.py file to see the execution output of the above program.

### Get into the Loop of Things

Loops are a handy tool that enables programmers to do repetitive tasks with minimal effort. Say we want a program that can count from 1 to 10, we could write the following program:.

```
print 1
print 2
print 3
print 4
```

```
print 5
print 6
print 7
print 8
print 9
print 10
```

The task will be completed just fine, the numbers 1 to 10 will be printed in the output, but there are a few problems with this solution:

- **Flexibility**: what if we wanted to change the start number or end number? We would have to go through and change them, adding extra lines of code where they're needed.
- **Scalability**: 10 repetitions are trivial, but what if we wanted 100 or even 1000 repetitions? The number of lines of code needed would be overwhelming and very tedious for a large number of iterations.
- **Maintenance**: where there is a large amount of code, one is more likely to make a mistake.
- **Feature**: the number of tasks is fixed and doesn't change at each execution.

Using loops we can solve all these problems. Once you get your head around them, they will be invaluable to solving many problems in programming.

Now let's consider the following code:

```
i=0
while i < 10:
    i+=1
    print i
```

If we run the program, the same result is produced, but looking at the code, we immediately see the advantages of loops. Instead of executing 10 different lines of code, line 4 executes 10 times. 10 lines of code have been reduced to just 4. Furthermore, we may change the number 10 to any number we like. Try it yourself, replace the 10 with your own number.

# Instructions

Before you get started we strongly suggest you start using Notepad++ or IDLE to open all text files (.txt) and python files (.py). Do not use the normal Windows notepad as it will be much harder to read.

First read example.py, open it using Notepad++ (Right click the file and select 'Edit with Notepad++') or IDLE.

- example.py should help you understand some simple Python. Every task will have example code to help you get started. Make sure you read all of example.py and try your best to understand.

- You may run example.py to see the output. Feel free to write and run your own example code before doing the Task to become more comfortable with Python.

- You are not required to read the entirety of Additional Reading.pdf, it is purely for extra reference.

# Compulsory Task 1

**Follow these steps:**

- Create a new file called while.py
- Write a program that always asks the user to enter a number.
- When the user enters the negative number -1, the program should stop requesting the user to enter a number,
- The program must then calculate the average of the numbers entered excluding the -1.
- Make use of the while loop repetition structure to implement the program.
- Compile, save and run your file.

# Compulsory Task 2

**Follow these steps:**

- Modify your while.py file to do the following:
  - Require the user to enter their name, with only a certain name being able to trigger the loop.
  - Print out the number of tries it took the user before inputting the correct number.
  - Add a conditional statement that will cause the user to exit the program without giving the average of the numbers entered if they enter a certain input.

- Compile, save and run your file.

**Things to look out for:**

1. Make sure that you have installed and setup all programs correctly. You have setup **Dropbox** correctly if you are reading this, but **Python or Notepad++** may not be installed correctly.
2. If you are not using Windows, please ask your mentor for alternative instructions.

# Give your thoughts..


**Hyperion strives to provide internationally-excellent course content that helps you achieve your learning outcomes.** Think the content of this task, or this course as a whole, can be improved or think we've done a good job?

Click here to share your thoughts anonymously.