Reading and Research - Iteration Statements

These tasks are designed to introduce you to the programming topic we will be studying in class next lesson. You **must** complete these activities prior to the lesson.

Repetition

We often want computers to repeat some process or block of code several times. Programming languages provide structures that enable you to repeat blocks of instructions over and over again. This type of repetition is known as **Iteration**.

Task 1

Read through the following pages on the Python School website and investigate the difference between the two types of loop:

1. For Loops in Python
2. Examples of For Loops (watch the video)
3. While Loops in Python
4. Examples of While Loops (watch the video)

In the space below write the general rules for when each type of loop should be used:

Loop Type	Description
`for`	*If you know how many times the loop wants to be repeated before entering it then this is used.*
`while`	*When the number of repetitions of the loop is unknown and so the loop is repeated until certain conditions are met.*

Task 2

For the sample code below:

- Fill in the expected results columns with the output that you would expect to appear on the screen when the code is executed in the **Python Shell**.
- Key in the code to the Shell and check if your expected results are correct.

Code	Expected Results	Does Actual Result Match Expected?

“ `

for count in range(5):
print(“Hello World”)

`| *Expected- Hello world printed 5 times* |`
`| *Actual- Hello world 5 times* |`
counter = 0
while counter < 4:
print(“Hello World”)
counter += 1
“ `| *Expected - Hello world repeated until it occurs 4 times*|
|*Actual - Hello world repeated 4 times*|

Task 3

Consider the following tasks and indicate whether you would use a `for` loop or a `while` loop to perform the tasks:

Task	`for` or `while` loop?
Write a program to display 5 asterisks on screen (i.e. *****).	for
Write a program which asks the user how many asterisks they want to be shown on the screen and then it displays them.	while
Write a program which asks the user to key in a password. If the user enters anything other than ‘secret’ reprompt the user for the password until they enter the password correctly	while
Write a program which asks the user to key in a message followed by a number and then display that message the number of times they have requested.	for

Task	`for` or `while` loop?
Write a program which asks the user for a number and then displays the times table for that number. E.g. if they input 5 it will show the 5 times table (1*5 = 5, 2*5 = 10 etc.)	for

Dry Running Algorithms

This consists of a careful, step-by-step simulation on paper of how an algorithm would be executed by a computer. It is usually carried out on the algorithm expressed as pseudo-code or as a flowchart and it allows the programmer to check that an algorithm is correct before he/she spends time implementing it in a programming language. The results of the dry-run are recorded in a trace table - see **page 25-26** in Bond for a worked example of a dry-run.

Note that a dry run can also be known as a hand trace, or desk check.

Task 4

The algorithm below is written in **pseudo-code**:

```
result ← 0
REPEAT
    INPUT number
    result ← result + number
UNTIL number = 0
```

Hand trace the above algorithm using the numbers 2, 6, 34, 12, 0 in order as input. Use the table on the right to record your results.

number	result
0	0
2	*2*
6	*6*
12	*12*
34	*34*

Task 5

Convert the algorithm in **Task Four** into a Python program adding an extra line of code which will output the value that is held in `result`.

Now test your program with the same data used in the dry run in Task Four. Verify that the value in the variable `result` matches the value on the last line of your trace table.

Task 6

Write a test the following two programs:

1. Write a program that will ask the user for a message and the number of times they want that message displayed. Then output the message that number of times.
2. Write a program which asks the user to key in a password. If the user enters anything other than 'secret' reprompt the user for the password until they enter the password correctly. When the password is entered correctly display the message 'You now have access to the system'.

Task 7

Paste the code from each of the exercises in **Task 6** into the space below.

```
6a)#Harry Robinson
#10-10-2014
#Asking for message and umbers of repitions
```

```
count = int(input("Give the number of times you want the message
repeated"))
message = input("Give your message: ")
for count in range(count):
    print(message)

 6b)#Harry Robinson
#10-10-2014
#Password secret program

password = ""
while password != "secret":
    password = input("Give the password: ")
    if password == "secret":
        print("Thanks you entered the right password")
    else:
        print("Incorrect password try again")
```

Summary

In this R&R you have investigated iteration. You have seen how iteration is expressed in pseudo-code and you have seen the syntax of the 2 types of loops (`for` and `while` in Python) and have had the opportunity to create programs that use these statements.

Please make sure you have completed this R&R fully before your next programming lesson as it will form the basis of the initial classroom discussion and starter tasks.