Tutorial 7 – Iterations

Functions

In Python, the line “if __name__ == "__main__":” is used to control what code runs when you execute a file.

It simply means “run this part only if this file is the main program.” When you run the file directly, Python sets its special name (__name__) to "__main__", so the code under that line runs.

But if you import the same file into another script, that section won’t run automatically — only the functions you call will. This lets you write files that can be both run on their own and safely imported into other programs.

Exercises

1) **Fill in the blanks:**

```
Exercise 1:

# Make this program print numbers from 1 to 3

num = 1                     # initialize the counter at 1
while num <= 3 :            # loop while num is less than or equal to 3
    print( num )            # display the current number
    num = num + 1           # increase num by 1 each time (increment)
print("Finished!")          # message printed after loop ends

Expected Result:
1
2
3
Finished!
```

2) **Fill in the blanks:**

```
Exercise 2:
# Fix this countdown so it prints 3, 2, 1, Go!

n = 3                       # start counting down from 3
while n > 0:                # keep looping while n is greater than 0
    print(n)                # show the current countdown value
    n = n - 1               # decrease n by 1 each time (decrement)
print("Go!")                # print message when countdown completes

Expected Result:
3
2
1
Go!
```

3) Fill in the blanks:

```
Exercise 3:
# Complete this so it stops when the user types "quit"

while True:                                 # start an infinite loop
    line = input("Enter text: ")          # ask the user for some input
    if line == "quit":                      # if the user types "quit"...
        break                               # ...exit the loop immediately
    print("You entered:", line)             # otherwise, echo back their input
print("Done!")                              # message printed after loop ends

Expected Result:
Enter text: hello
You entered: hello
Enter text: test
You entered: test
Enter text: quit
Done!
```

4) Fill in the blanks:

```
Exercise 4:
# Print "Hello!" 3 times

for i in range(0,1,2):          # repeat 3 times (i takes values 0, 1, 2)
    print("Hello!")            # body of the loop — runs once per repetition
print("Done!")                  # message printed after the loop finishes

Expected result:
Hello!
Hello!
Hello!
Done!
```

5) Fill in the blanks:

```
Exercise 5:
# Print numbers 1 to 5

for num in range(1, 6 ):   # start at 1 and stop before 6 → prints 1–5
    print( num )           # print the current number each time
print("All done!")         # message after loop completes

Expected result:
1
2
3
4
5
All done!
```

6) Fill in the blanks

```
Exercise 6:
# Add numbers from 1 to 3

total = 0                  # start total at 0 before the loop begins
for n  in range(1, 4 ):     # range(1, _) gives 1, 2, 3
    total = total  + n     # add the current number to running total
print("Total:", total )    # display the total after loop ends

Expected result:
Total: 6
```

7) Practice Question

Write a short program that keeps asking the user to enter numbers, one at a time.

When the user types "done", your program should stop asking and then print out:

- how many numbers were entered
- the total (sum) of all the numbers
- and the average (total ÷ count)

Hints:

- Think about what needs to repeat — that's what your while loop will do.
- Create two variables before the loop starts:
    - one to keep track of how many numbers you've seen (count)

    - one to add them all up (total)

- Inside the loop, use input() to ask the user for a number.
- If the user types "done", use a break statement to stop the loop.
- Otherwise, convert the input into a number (use float()), then update your count and total each time.
- After the loop finishes, print your results in a friendly message.