Loops,
For-Loops in particular

This is where programs start to get interesting. A *loop* is a code structure that allows a block of code to be executed over and over. You can show that if a programming language has loops, conditional (if) statements and assignments then you can write in it any program that can be written in any language.

There are two kinds of loops, which some call *definite* and *indefinite* and everyone calls *for*-loop and *while*-loops. We'll talk about definite loops first.

The format of a definite loop is

```
for <variable> in <sequence>:
        <body>
```

The body is executed over and over, once for each value in the sequence. Each time the body is executed the variable has one of the values in the sequence.

For example,

```
for name in ["John", "Paul", "George", 'Ringo"]:
    print( name  + " is my favorite Beatle.")
```

This prints

```
John is my favorite Beatle.
Paul is my favorite Beatle.
George is my favorite Beatle.
Ringo is my favorite Beatle.
```

Now, what is a "sequence"??

Here are the most common types of sequences:

- A list of values in square brackets:
  ["Pete", "John", "Keith", "Roger"]
  [1, 2, 3]
  ["One", 1, "Two", 2, "Three", 3]
- A list generated by the **range**() function, which gives sequential lists of integers.
  list(range(1, 4)) is [1, 2, 3]
  list(range(6)) is [0, 1, 2, 3, 4, 5)
  list(range(5, 1, -1)) is [5, 4, 3, 2]

- A string, in which case the variable takes on each character of the string.

- A file value, which is obtained by *opening* a text file:

```
F = open("filePrinter.py", "r")
for line in F:
    print(line)
```

A for-loop that runs through a file has its variable take on each *line* of the file as a string.

Clicker Question: What will this code print?

```
for x in range(0, 5):
        print(x, end=" ")
print()
```

A) 0 1 2 3 4 5

B) 0 1 2 3 4

C) 0
   1
   2
   3
   4
   5

D) It causes an error

Write a program *summer.py* that sums the numbers from 1 to 100.

Then write a program *multiplier.py* that multiplies the numbers from 1 to 100. This is also called 100 factorial.

When loops are nested inside loops you need to think carefully about what they are doing. Remember -- when a for-loop is executed its entire body is executed once for each value in the sequence.

Clicker Question:

What will this print:

```
        for letter in "AB":
                for number in [1, 2]:
                        print( letter, number )
```

A)	B)	C)	D)
A1	A1	B1	A1
A2	B1	A1	A1
B1	A2	B2	B1
B2	A2	A2	B1

Trick Clicker Question

What will the following code do?

```
x = 5
for t in range(1, x):
    print( t )
    x = x+1
```

A) Nothing
B) Give an error message
C) Print the numbers 1 2 3 4 and then stop
D) Print 1 2 3 4 5 6 7 8 9 ... and keep going