PYTHON FUNDAMENTALS

Loops

LEARNING OBJECTIVES

- To understand the uses of a for loop
- To understand the uses of a while loop
- To tell the difference between for and while loops
- To write programs using both for and while loops

Moving on. Loop de loop

Imagine doing the same thing over and over again.

For example, if I asked you to make me a cup of tea…

And then asked you again to make everyone in the room a cup of tea…

Or if I asked you to update stocks in a warehouse…

Iteration in coding

For loops

If I said to you, make a list of your 3 favourite drinks and print each one…

1) I’d expect you to have a good time doing it

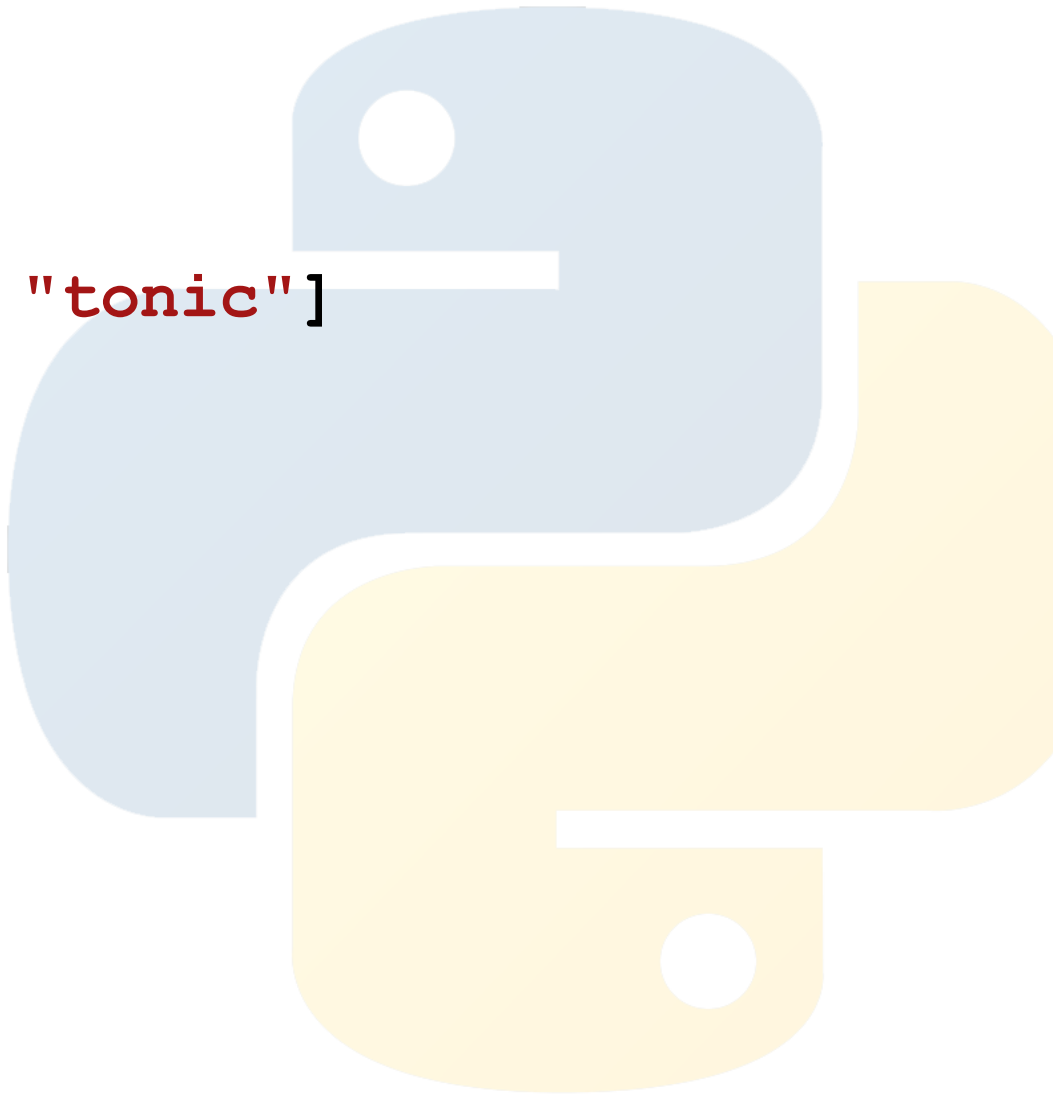
2) I’d expect you to do something like this

```
favourite_drinks = ["coke", "fanta", "tonic"]

print(favourite_drinks[0])
print(favourite_drinks[1])
print(favourite_drinks[2])
```

But imagine if I said 1000 drinks

Let’s make this code work for us.

```python
favourite_drinks = ["coke", "fanta", "tonic"]

for drink in favourite_drinks:
    print(drink)
```



‘drink’ is the index variable name

```
favourite_drinks = ["coke", "fanta", "tonic"]

for drink in favourite_drinks:
    print(drink)
```

The list name just created above

The ‘action’ taken is to print each ‘drink as looping through the list, one at a time

```
favourite_drinks = ["coke", "fanta", "tonic"]

for i in favourite_drinks:
    print(i)
```

*i stands for index, which is widely used in for loops.

```
for thing in iterable:
    #do stuff
```

Range

Remember that code?

```
for i in range(10):
    print(i)
```

Remember that code?

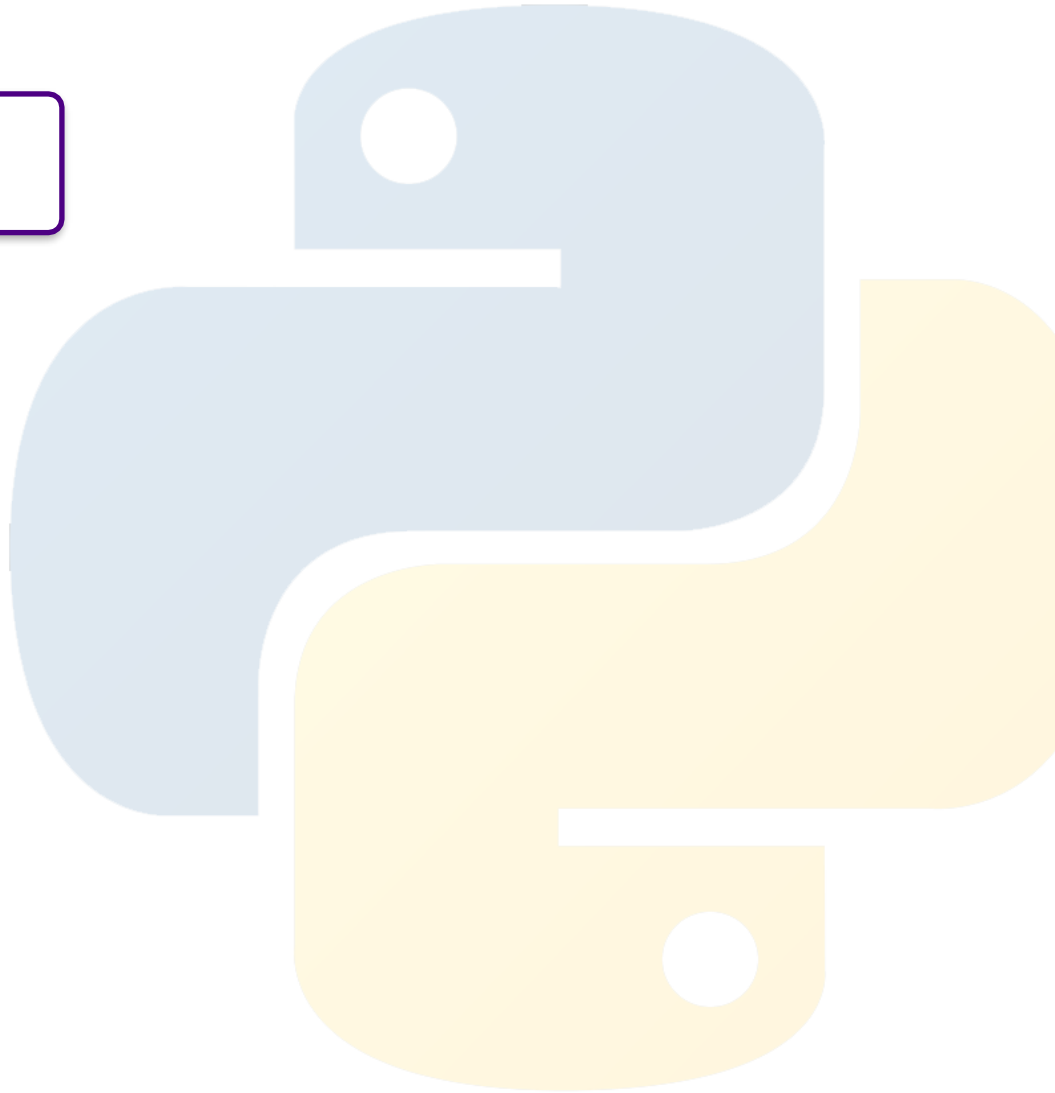
```
for __ in range():
    #action
```

```
for i in range(10):
    print(i)
```

Remember that code?

‘i’ stands for index, but can be any name

```
for i in range(10):
    print(i)
```



Remember that code?

Range 10, means 0 - 9

```
for i in range(10):
    print(i)

#Expected: 0-9
```

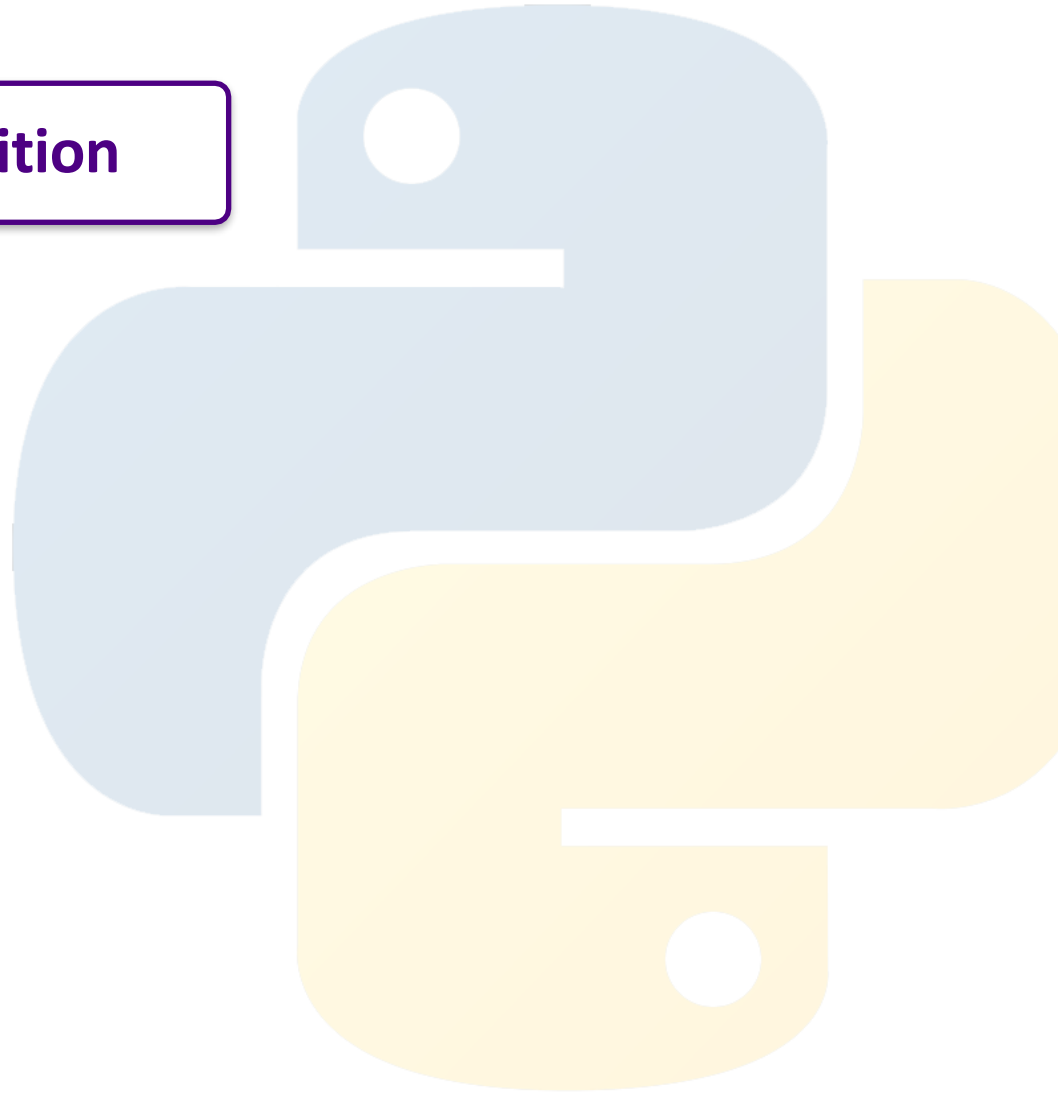
Remember that code?

```
for i in range(0, 10):
    print(i)
```

Remember that code?

First value is start position

```
for i in range(0, 10):
    print(i)
```



Remember that code?

Second value is stop position (not including)

```
for i in range(0, 10):
    print(i)

#Expected: 0-9
```

Remember that code?

```
for i in range(5, 11):
    print(i)
```

Test this out with different numbers to see how range with two values works

Remember that code?

```
for i in range(5, 11):
    print(i)

#Expected: 5-10
```

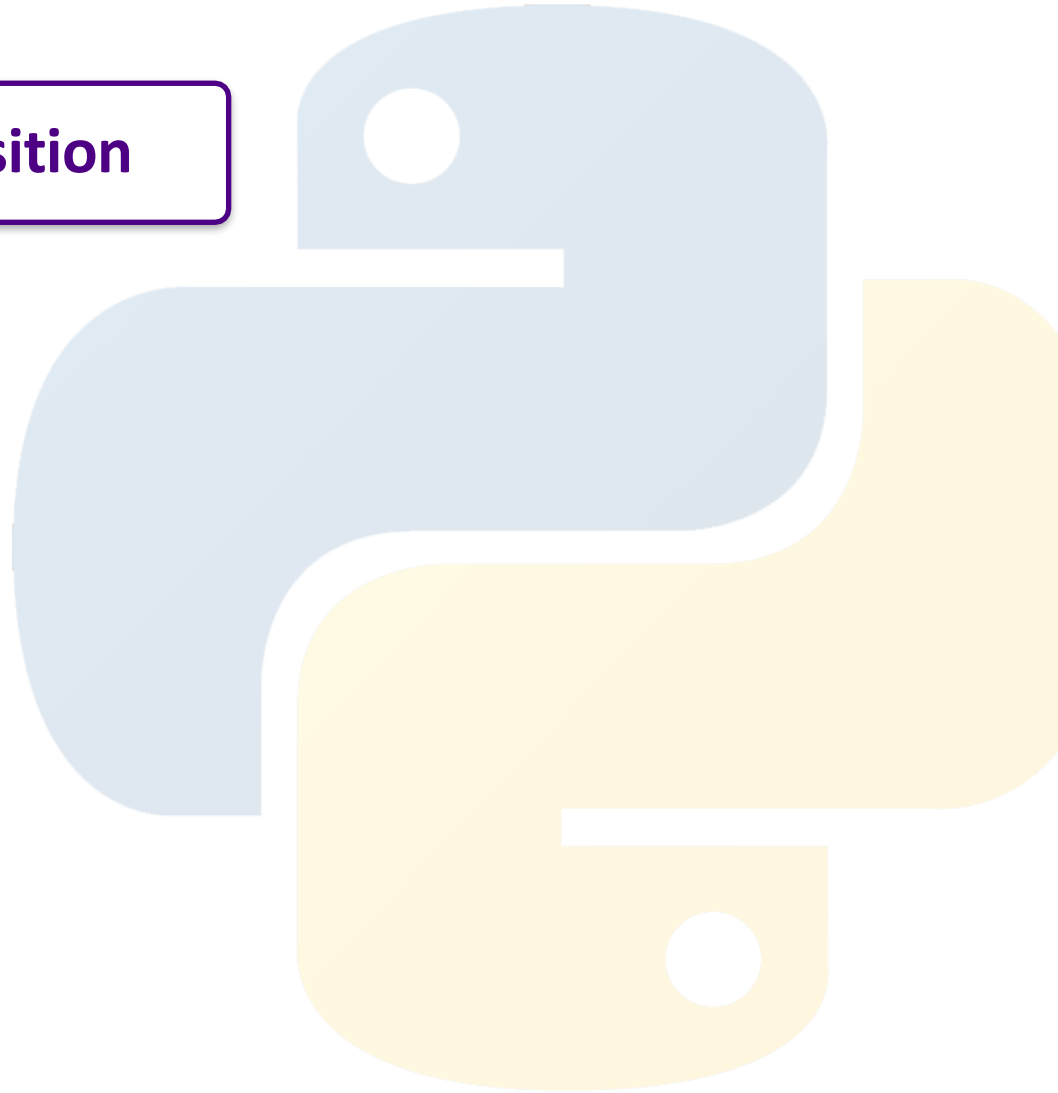
Remember that code?

```
for i in range(0, 10, 1):
    print(i)
```

Remember that code?

First value is start position

```
for i in range(0, 10, 1):
    print(i)
```



Remember that code?

Second value is stop position (not including)

```
for i in range(0, 10, 1):
    print(i)
```

Remember that code?

```
for i in range(0, 10, 1):
    print(i)
```

Third value is 'step' - how the for loops iterates. The default will be 1, but you can change this

Remember that code?

```
for i in range(0, 10, 1):
    print(i)

#Expected: 0-9
```

Remember that code?

```
for i in range(2, 12, 2):
    print(i)

#Expected: 2, 4, 6, 8, 10
```

Activity(1):

Create a list of 4 favourite films

Use a for loop to show each film in the list

Create a function called film_check() that checks if the 3rd film in the list is Ghostbusters.

If it is, it should print “yey it’s ghostbusters”. If it isn’t, it should print “booo, we want ghostbusters”

Activity(2):

If you can create a for loop to print 0-9 on the terminal, how can you create one to count backwards from 9-0?

Consider the different ways we've used range and give it a go. Research if necessary!

while loops

for loops run a finite, or limited number of times.

A while loop is a little different

A while loop will run infinitely and do its job until a condition is met

```
num = 0

while num < 10:
    num += 1
    print(num)
```

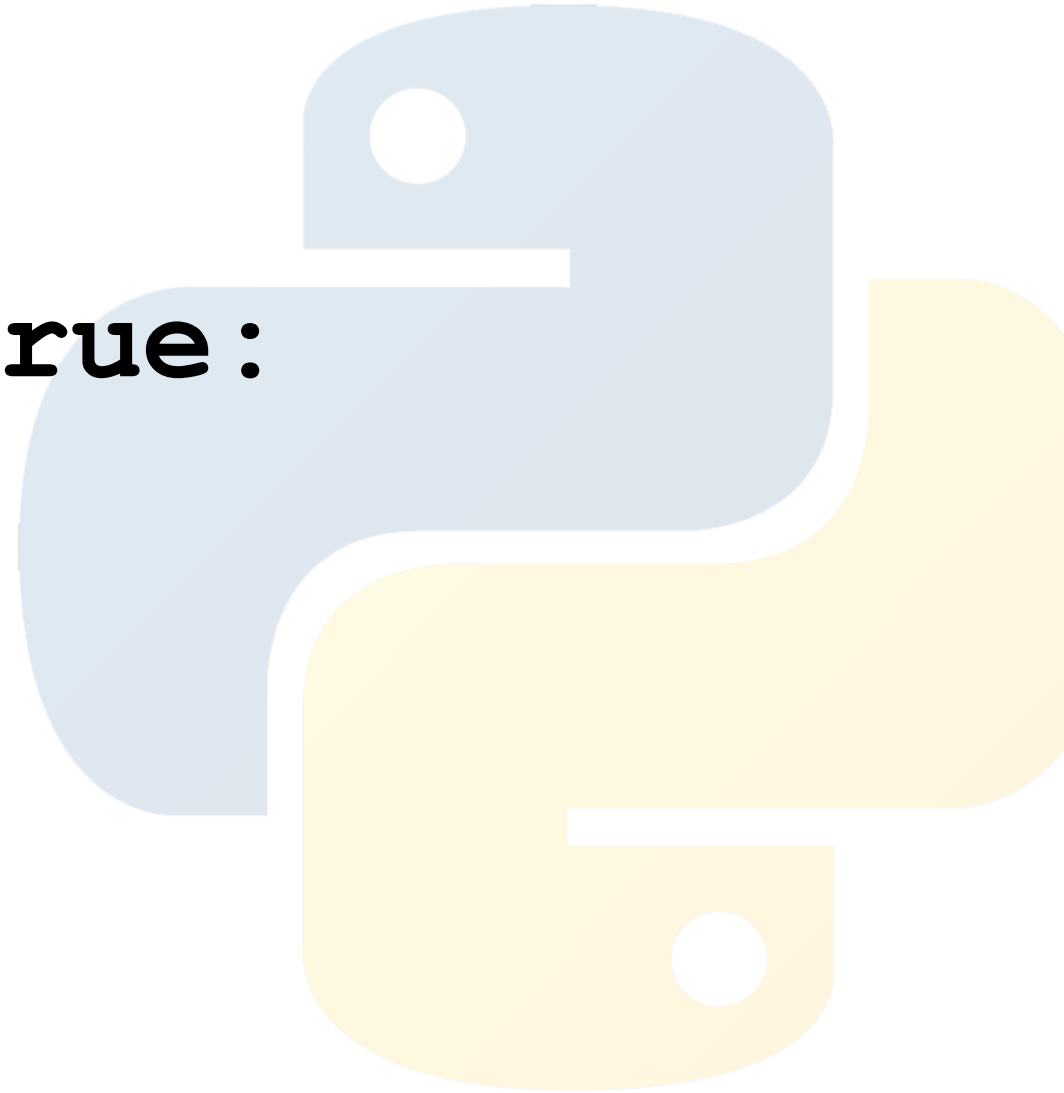
```
num = 0

while num < 10:
    num += 1
    print(num)
```

The loop only runs while this statement is true

Add 1 to num and print it

*Once num becomes >= 10, the statement is no longer true so the while loop stops

```
while statement_is_true:
    #do this
```



Let’s use the example of a random number generator

I want to generate random numbers until a certain integer is found

I can’t use a for loop, because I don’t know how many times the loop will have to iterate

So I use a while loop

```
import random

rand_num = random.randint(0,50)

my_num = 50

while rand_num != my_num:
    print(rand_num)
    rand_num = random.randint(0,50)

print("You've found {}!".format(my_num))
```

LEARNING OBJECTIVES

- To understand the uses of a for loop
- To understand the uses of a while loop
- To tell the difference between for and while loops
- To write programs using both for and while loops

Activity(1):

Create a for loop that prints “hello world” 13 times. Now, convert this into a while loop that does the same job.

Activity(2):

Create a variable, use a loop to generate a random number between 1 and 30 six times. For each random number generated, check if this number of divisible by 7 or not.

Activity(3):

Create a while loop to randomly cycle through a list of card suits until a given card suit is reached

```
cards = ["Diamond", "Spade", "Club", "Heart"]
```

Create a variable called current_card and use a list method to randomly give it a value from the list every time the loop runs. Then compare this to the suit you want to find to stop the while loop.

Extra activity:

Create a loop that asks a user to input a number, then displays the multiplication table for that number up to 12 e.g. if I input 1, I should see this

1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10
1 x 11 = 11
1 x 12 = 12

Incorporate another loop so the program starts again and ask the user for a new number every time it finishes.

Extra activity (DIFFICULT):

Create a program that checks whether all numbers between 1 and 20 are prime numbers or not.

Extra reading:

Research on do...while loops, find out about the difference between for loop, while loop and do...while loop. Give an example of each. What are the pros and cons?