Knowledge Organiser: Python 3 Fundamentals

Functions

We use functions to break up our code into small chunks. These chunks are easier to read, understand and maintain. If there are bugs, it's easier to find bugs in a small chunk than the entire program. We can also re-use these chunks.

```
def greet_user(name):
    print(f”Hi {name}”)


greet_user(“John”)
```

Parameters are placeholders for the data we can pass to functions. **Arguments** are the actual values we pass.

We have two types of arguments:

- Positional arguments: their position (order) matters
- Keyword arguments: position doesn't matter - we prefix them with the parameter name.

```
# Two positional arguments
greet_user(“John”, “Smith”)

# Keyword arguments
calculate_total(order=50, shipping=5, tax=0.1)
```

Our functions can return values. If we don't use the return statement, by default **None** is returned. None is an object that represents the absence of a value.

```
def square(number):
    return number * number

result = square(2)
print(result)  # prints 4
```

While loops

The while loop is used to repeat a section of code an unknown number of times until a specific condition is met. In this example, Print i as long as i is less than 5.

```
i = 1
while i < 5:
    print(i)
    i += 1
```

Output: 1 2 3 4

For loops

A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string). The range() function defaults to 0 as a starting value, however it is possible to specify the starting value by adding a parameter: range(1, 5), which means values from 1 to 4 (but not including 6) will be printed.

```
for i in range(1, 5):
    print(i)
```

Output: 1 2 3 4