

9. TUPLES

Tuples are another data structure that we can use to group items of data together. We have already used tuples to return multiple values from a function, see *Chapter 4 – Functions*.

In the same way as a string cannot be changed, added or edited when it has been created, tuples are also known as immutable data structures.

Tuples can be written in two different ways; in this example the three data items have been grouped together simply by separating the variables using commas. This could also have been shown:

```
def find_names():
    """asks for two name string and returns"""
    y_name = get_name("your name")
    f_name = get_name("your first friend's name")
    s_name = get_name("your second friend's name")
    return y_name, f_name, s_name
```

The return statement is a tuple as three data items have been grouped together separated by commas. This means that the three items can be grouped together and then 'unpacked' when the function is called.

```
def get_name(p):
    """asks for name string and returns"""
    n = input("Please enter {0}: ".format(p))
    return n

def find_names():
    """asks for two name string and returns"""
    y_name = get_name("your name")
    f_name = get_name("your first friend's name")
    s_name = get_name("your second friend's name")
    return y_name, f_name, s_name

def print_message(y, f, s):
    """prints message using two name variables"""
    print("Hello {0}, your best friends are {1} and {2}".format(y, f, s))

def main():
    """ runs all programs"""
    y_name, f_name, s_name = find_names()
    print_message(y_name, f_name, s_name)
```

main()  The tuple is unpacked when the function is called here

Tuples are also useful in grouping data as key/value pairs in dictionaries to create records.

Look at this example:

```
# create a student dictionary
studentDict = {
    5521: ('Jonny', 'Smith', 'W10'),
    5423: ('Kyle', 'Jones', 'H10'),
    5761: ('Sara', 'Greene', 'M10')}

#iterate through the data

for k,v in studentDict.items():
    if k == 5423:
        print("Student name:{0} {1}. Form {2}".format(v[0],v[1],v[2]))
```

```
>>>
Student name:Kyle Jones. Form H10
>>>
```

Using a tuple, we can match the key, e.g. student ID number, with the record about the student rather than just one data item.