Lab 2: Types

The aim of this lab is to extend our hello world application to make it interactive and to handle different types of data.

This lab is comprised of 5 steps:

1. Make the application interactive
2. Input some numbers
3. Inputting some strings
4. Concatenating numbers and strings
5. Working with / Using the None value

Step 1: Make the application Interactive
Modify your program to take user input as shown in the lecture slides.

An example of the interaction between the program and the user is shown below:

```
Hello World!
Please enter your name: John
Welcome John
```

To do this you can follow the style of application presented in the notes, for example:

```python
# Print out a message and then ask the user for input

print('Hello, World')
user_name = input('Enter your name: ')
print('Welcome', user_name)
```

Step 2: Input some numbers
In this step you should ask the user for two numbers.

Note that the result returned from the `input()` function is a string. You must therefore convert the input string into a number. this can be done using the `int()` function. This will take another type (such as a string or indeed a float) and convert it into an integer. In a similar way the `float()` function will convert an int or a string into a float and the `str()` function convers tints and floats into strings etc.

We can therefore write:

```python
input1 = int(input('Please enter a number: '))
input2 = int(input('Please enter another number: '))
```

Next you should add the two numbers together and then print out the result. For example:

```python
value = input1 + input2
```

You should print these back out to the user to confirm the data they have entered. You could do this just using the `print()` function or you could use a formatted string to help with the layout, for example:

```
print(f'The result of {input1} + {input2} is {value}')
```

Run the program and confirm the output, for example:

```
Please enter a number: 2
Please enter another number: 3
The result of 2 + 3 is 5
```

Finally print out the type of the variable you are holding the result in, for example:

```
The type of the value is <class 'int'>
```

You can do this using the `type()` function.

Step 3: Input two strings
Next ask the user to input two strings, for example:

```
input_string1 = input('Please enter a string: ')
input_string2 = input('Please enter another string: ')
```

You can use the '+' operator to concatenate the two strings together and the print out the result, for example:

```
value = input_string1 + input_string2
print('The new value is', value)
```

What is the result?

Next print out the type of the variable `value` after the above assignment, what is its type now?

Step 4: Concatenate a number and a String
Next we will use the add operator to add a version number to our application.

For example:

```
title = 'Data Processing App Version' + str(1.0)
print(f'The title of this app is {title}')
```

Add the above code and rerun your application. What is the result.

Now remove the `str(1.0)` and replace it with just `1.0` – now rerun; what happens?

Return the code to use str(1.0)

Step 5: Using None
The final part of the lab will get you to create a variable which holds the value None.

Once you have done that, we will print the variable out, test to see fi the value is None, if it is not None and what the type is. For example:

```
user = None
print('user:', user)
print('user is None:', user is None)
print('user is not None:', user is not None)
print('The type of the user', type(user))
```

The output from this is:

```
user: None
user is None: True
user is not None: False
The type of the user <class 'NoneType'>
```