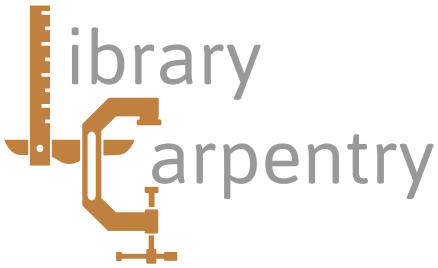

Software skills for librarians

Module 3: Programming in Python
Exercises

1. Open your text editor and enter the 'Hello World' program from the slides. Now save it to the file hellow.py:

   a. Run this program from within your editor, or the command line.
   b. Now start the Python shell and enter the program interactively.
   c. Make a deliberate mistake, for example by omiting one of the quotation marks, and try again.
   d. What are the advantages and disadvantages of the two methods for running the program.

2. Working in pairs list any programming languages you may have heard of; do you know anything about these languages?

3. Modify the hellow.py program by declaring a variable:

   a. Read a name into this variable using the `raw_input()` function.
   b. Modify the print statement so that it outputs that name.
   c. Add extra lines to change the content of the variable and print it again.

4. Declare a list containing a few strings (like the names of famous authors):

   a. Sort the list into order.
   b. Reverse the sorted list and print it out.
   c. Create a new list by taking a slice from the first one, and then print this out.
   d. Could a tuple have been used instead? Why?

5. Open the program marcdict.py in your text editor. This program reads the textual representation of a MARC record into a dictionary. You can add your own code to the end of the program:

   a. Create a list from the keys of the dictionary record.
   b. Extract the 245 field; what data type is it?
   c. Now extract the contents of subfield $c from the 245 field.
   d. How could this program be modified so that the two indicator digits from each field could be retained.

6. Write a program which prompts users to enter their favourite programming language. Print an appropriate message depending on whether it is Python.

7. Modify the previous program to print another specific message if the language is Java.

8. Write a program which uses a loop to read a few numbers from the input.

   a. Store these numbers in a list first, and then calculate the average afterwards.
   b. Modify the program so that you do not need to store the numbers.
   c. Which would you expect to run faster?

9. The program isbn.py contains code to validate an ISBN number. Modify this by making it into a function which accepts a string and returns a boolean value. Test this with a few known values.

10. Further modify this program to create an ISBN class.

    a. Provide an `__init__()` method which stores an ISBN number in a class variable.
    b. Provide another method to validate the number.
    c. Test this by creating a few ISBN objects and call the validate() method on each.

11. a. Modify the program in exercise 8 to read the numbers from a file instead. Test your program using the file numbers.txt. [hint: you can convert a string to a number using `num=float(str)`. Windows users should use the file numbers_win.txt instead which has different line endings]
    b. There are at least three ways of writing this; can you outline each one?

12. Modify the program in exercise 5 to save the MARC record to a new text file.

13. Using all of the code from the previous exercise create a Python module which defines a MARC record class. You should store the record in a dictionary, but give some thought to how you handle repeated fields (like multiple 650 fields). You should provide:

    a. An `__init__()` method which creates an empty record.
    b. Methods to read and write records to text files.
    c. A method to access a field using the subscript notation [hint: implement `__getitem__(self, index)`]
    d. A method to add a new field.