Exercise 4 – String Handling

Objective

To consolidate string manipulation in Python. This includes further practise at general Python constructs, such as loops.

Questions

1. Open the script sep.py in a text editor. You'll see a string defined called 'Belgium'. Add code to print:
    1. A line of hyphens the same length as the Belgium string, followed by
    2. the string with the comma separators replaced by colons ':', followed by
    3. the population of Belgium (the second field) plus the population of the capital city (the forth field). Hint: the answer should be 11183818.
    4. A line of hyphens the same length as the Belgium string.

2. In this exercise, much of the code has been written for you! Open the script greek.py in IDLE and run it there (use <F5>) – do not use Windows cmd.exe because the character set used cannot handle the Greek characters.

   The try and except blocks are examples of exception handling in case it is run under cmd.exe – we will cover these later in the course.

   The script has a list of names for the characters in the Greek alphabet, and it displays each one within a loop. The character itself is generated using the chr() built-in function (look it up if you're curious). Within Unicode, Greek lowercase characters start at position 0x03b1 (alpha).

   Currently, the output is a bit messy and insipid, like this:

   Alpha : α
   Beta : β
   Gamma : γ
   Delta : δ
   Epsilon : ε
   Zeta : ζ
   Eta : η

The task in this question is to replace the existing **print** function with another (using **format**) which displays for each character:

The hex value of the character (**pos**)

The character name (**cname**), left justified, maximum 12 characters

A colon separator

The lowercase Greek character (**char**)

The uppercase Greek character

Your output should look something like this:

0x3b1 Alpha : α Α
0x3b2 Beta : β Β
0x3b3 Gamma : γ Γ
0x3b4 Delta : δ Δ
0x3b5 Epsilon : ε Ε
0x3b6 Zeta : ζ Ζ
0x3b7 Eta : η Η

and so on.

If time allows...

3. Examine the file **messier.txt** in the **labs** directory, which contains details of celestial "Messier" objects. It consists of several columns for each object, identified by the 'M' number. The columns are as follows:

MessierNumber CommonName ObjectType Constellation

Note that many have no common name. Read the file using a **for** loop:

```
for line in open('messier.txt', encoding='latin_1'):
 if not line: break
   # The text is in the variable named 'line'
```

Ignore lines that do not start with 'M'. Print the fields from each line delimited with '|' characters. Where there is no common name, use 'no name'. Ignore any lines not beginning with a Messier number. For example:

M1	The Crab Nebula	Supernova remnant	Taurus
M2	no name	Globular cluster	Aquarius
M3	no name	Globular cluster	Canes Venatici

Hint: the header on the file should assist in getting the field positions.

Solutions

Question 1

a) A line of hyphens the same length as the Belgium string, followed by

b) the string with the comma separators replaced by colons ':'., followed by

c) the population of Belgium (the second field) **plus** the population of the capital city (the fourth field). Hint: the answer should be 11183818.

If you did this:

```
print(items[1] + items[3])
```

then you would've got string concatenation, and an apparently very large number! You need to change each value to an int.

d) A line of hyphens the same length as the Belgium string.

```
items = Belgium.split(',')
print('-' * len(Belgium))                    # a)
print(':' . join(items))                  # b)
print(int(items[1]) + int(items[3]))  # c)
print('-' * len(Belgium))                    # d)
```

Question 2

There are at least two ways to format the string, both are shown as "Either/or":

```
greek = ['Alpha', 'Beta', 'Gamma', 'Delta', 'Epsilon', 'Zeta',
         'Eta', 'Theta', 'Iota', 'Kappa', 'Lambda', 'Mu',
         'Nu', 'Xi', 'Omicron', 'Pi', 'Rho', 'Sigma final',
         'Sigma', 'Tau', 'Upsilon', 'Phi', 'Chi', 'Psi', 'Omega'
         ]

# Format required:
#   The hex value of the character.
#   The character name (cname), left justified,
#   maximum 12 characters.
#   A colon separator.
#   The lowercase Greek character.
#   The uppercase Greek character.
```

```
for pos, chr_name in enumerate(greek, start=0x03B1):
    try:
        char = chr(pos)

                Either:
        print("{0:#x} {1:<12s}: {2} {3}".
              format(pos, chr_name, char, char.upper()))

                or:
      print(f"{pos:#x} {chr_name:<12s}: {char} {char.upper()}")

    except UnicodeEncodeError as err:
        print(chr_name, 'unknown', err)
```

If time allows...

Question 3

```
for line in open('messier.txt'):
    if not line: break
    if line.startswith('M'):
        # Slice each field
        mes_num = line[:6].rstrip()
        com_name = line[6:40].rstrip()
        if not com_name: com_name = 'no name'
        obj_type = line[40:64].rstrip()
        const = line[64:].rstrip()
print(f"|{mes_num}|{com_name}|{obj_type}|{const}|")
```