String Handling

Objectives

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

Reference Material

Chapter 04 String Handling. You will also need the Python online documentation.

Questions

1. Open the script **`sep.py`** in a text editor. You will see a string defined called 'Belgium'. Add code to print:
   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 forth field). Hint: the answer should be 11183818.
   d) 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 **`unichr()`** built-in function (look it up if you are curious). Within Unicode, Greek lowercase characters start at position 0x03b1 (alpha).

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

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

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

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

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

The lowercase Greek character (**char**)

The uppercase Greek character

Use string formatting to layout each display line.

Your output should look something like this:

```
3b2 Alpha        α Α
3b3 Beta         β Β
3b4 Gamma        γ Γ
3b5 Delta        δ Δ
3b6 Epsilon      ε Ε
3b7 Zeta         ζ Ζ
```

and so on..

If time allows…

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

   M number Common name Type Constellation

   Note that many have no common name.

   Read the file in a for loop in the following manner:

```
for line in open('messier.txt'):
  if not line: break
     # The text is available through variable '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 forth field). Hint: the answer should be 11183818.

If you did this:

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

then you would have 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

Format required:
The hex value of the character
The character name (cname), left justified, maximum 10 characters
The lowercase Greek character
The uppercase Greek character

```
print "%x %-12s %s %s" % \
      (pos,cname,char,char.upper())
```

The full program is:

```
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

pos = 0x03B1
for cname in greek:
    try:
        char = unichr(pos)
        #print cname,":",char
```

```
                print "%x %-12s %s %s" % \
                      (pos,cname,char,char.upper())
                pos += 1
            except UnicodeEncodeError as err:
                print cname, 'unknown'
```

If time allows…

Question 3

```
for line in open('messier.txt'):
    if not line: break
    if line.startswith('M'):
        # Slice each field
        Mnum = line[:6].rstrip()
        Name = line[6:40].rstrip()
        if not Name: Name = 'no name'
        Type = line[40:64].rstrip()
        Cons = line[64:].rstrip()

        print "|"+Mnum+"|"+Name+"|"+Type+"|"+Cons+"|"
```