Parseltongue Piscine - Part04

Building reusable functions and reading data from files

Kai kai@42.us.org

Summary:

Contents

Eat, Sleep, Code, Repeat.

Chapter I

Don't Panic!

Remember the three commandments of 42:

1. Ask the person on the left of you

2. Ask the person on the right of you

3. Read the Manual (i.e. the documentation)!

Chapter II

Building Functions

II.1 FOPP Chapter 12

Go to Runestone: Fundamentals of Python and complete section 12 before the next exercise.

II.2 Exercise 0: Rainbow

	Rainbow
Topics to study : Functions and parameters, ANSI escape codes	
Files to turn in : 00_rainbow.py	
Notes : Credit to StackOverflow for the demonstration code, Function Parameters, Global and Local Variables	

Did you know that it is possible to print colorful text in the terminal?

Here is a sample Python program, which if you save and run it, will show you many options for color combinations that you can get without installing any special libraries.

```python
def print_format_table():
    """
    prints table of formatted text format options
    """
    for style in range(8):
        for foreground_color in range(30,38):
            s1 = ’’
            for background_color in range(40,48):
                format = ’;’.join([str(style), str(foreground_color), str(background_color)])
                s1 += ’\x1b[%sm %s \x1b[0m’ % (format, format)
            print(s1)
        print(’\n’)

print_format_table()
```

Run this program, study this code and think about what is "says", how it does what it does.

Try replacing the part where it says (format, format) with (format, "42") and run it again to see what changes!

Write a program called `00_rainbow.py` which includes a function called write_colorful_text. The function will take in one parameter for the text to be written, one parameter for the "style" (in the range of 0 to 8), one parameter for the "background" (in the range of 40 to 48), and one parameter for the "foreground" (in the range from 30 to 38). The first line looks like this:

```
def print_colorful_text(string, style, foreground, background):
```

Call the write_colorful_text function several times, with different parameters. Your function should print out the word "R A I N B O W" with each letter a different color, in rainbow order. You can decide what color combination is best for rainbow order.

II.2.1 Bonus

Use global variables to label each number with the color or style it represents. You should then be able to call your function like this:

```
print_colorful_text("I am Groot", BOLD, FG_GREEN, BG_DARKBLUE)
```

...or similarly with any other color labels that are useful to you.

This is not the most common color encoding system, by the way. Later on - in Game Design class - we will use RBG format where each color in R, G, and B has a value between 0 and 255.

II.3 Exercise 1: Math Class

	Math Class
Topics to study : Greater than, less than, Math library	
Files to turn in : 01_mathclass.py	
Notes : Math, More on Lists, Importing Libraries, Loops, Statistics	

Take in a set of numbers as command line arguments. Store them as an array and print out the min, max, mean, median, mode and range of the set.

You may not use "import statistics". You may use "import math".

```
?> python3 01_mathclass.py 142 6 13 36 54 4 9 78 78 102
Min: 4
Max: 142
Mean: 52.2
Median: 45
Mode: 78
Range: 138
```

Each mathematical task should take place in its own function. Write a function to find the min, one to find the max, one to find the mean, and so on.

II.3.1 Bonus

As a bonus, but not the main part, you may turn in a version of this program which demonstrates how much easier this is using the Statistics library. :)

Chapter III

Working with Files

III.1 FOPP Chapter 10

Go to Runestone: Fundamentals of Python and complete section 10 before the next exercise.

III.2 Exercise 2: Student Directory

	Student Directory
Topics to study : `File input, dictionary`	
Files to turn in : `02_phonebook.py`	
Notes : `Use the names.txt file provided on the project page. Dictionaries, More Dictionaries, Key Value Pairs`	

Using the attached file names.txt, store the information in a hash or dictionary where first names are associated with last names.

Use your hashtable to identify which first names are shared by more than one student, mentor or admin in h2s. Print out each first name that repeats in the set followed by an array of the last names associated with that first name. Then do the same thing with last names.

```
?> python3 02_phonebook.py
** Shared First Names! **
Elliot (2): [Tregoning, VanHeuman]

** Shared Last Names **
Kardashian (4): [Khloe, Kim, Kourtney, Rob]
```