Python tasks – Gabor – 31 Aug 2018
Python basics exercises

Unless indicated, each script in this section is to receive command line input. You should write a separate script for each task.

1. Have two numbers input, and find the sum of these two numbers.

2. Given a space separated list of integers, output the sum of them

3. Given a space separated list of integers, output the list of those divisible by three.

4. Calculate the area of a triangle given three side lengths. eg. 13 14 15 ➔ 84

5. Given an input, output a string composed of every other character. eg. Aardvark ➔ Arvr

6. Count the number of each vowel in an input string.

7. Print the first n Fibonacci numbers. eg. 6 ➔ 1, 1, 2, 3, 5, 8

8. Check whether an input number is prime or not.

9. Given an input of two space separated numbers, print all the prime numbers between them.

10. Given an input of one or two space separated numbers, implement 8 or 9 as appropriate.

11. Given two integers as input, print the value of $f(k) = k^2 - 3k + 2$ for each integer between the two inputs. eg. `2 5` ➔ `0, 2, 6, 12`

12. Given an input string, determine whether the string contains only digits

13. Given an input string, determine whether the string contains only 0 and 1 characters, and if so assume it's a binary string, convert it to a number, and print the decimal value.

14. Given an input string, determine whether it contains only 0-9 and a-f or A-F characters, and if so assume it's a hex string, convert it to a number, and print the decimal value.

Slicing and other string exercises

Example input strings:

1. `234`
2. `The quick brown fox ran over dogs.`
3. `"Don't quote me," she said.`

For the following exercises, assume you have a string in variable `s` and `s` is sufficiently long for each of the following questions to make sense. Write a one line expression for …

15. … the character at position 2.

16. … the fifth character.

17. … the number of characters in the string.

18. … the first character.

19. … the last character.

20. … the penultimate character.

21. … the five character long substring starting at position 3.

22. … a substring consisting of the last 5 characters of the string.

23. … the substring starting at the third character and continuing to the end of the string.

24. … the three character long substring starting five characters before the end of the string.

25. … a substring containing every other character from the string.

26. … a string consisting of every 3rd character from the input, starting from its 2nd character.

27. … the input string reversed.

28. … a string consisting of every 3rd character from the input string, starting from its penultimate character.

29. … the position of the first space in the input string.

30. … the string shifted to the right by one (ie. original string with the last character removed).

31. … the string shifted to the left by one (ie. original string with the first char removed).

32. … the string all in lower case.

For the remaining exercises on the next two pages, a word means a substring of `s` that is delimited on both sides by either a space, the start of `s` or the end of `s`, regardless of whether it is a genuine English word or not.

Write a one line expression for …

33. … the first word in the input string

34. … a list of all the space delimited substrings of the input string. For example,

```
234    ➔     ["234"]
12 35  ➔     ["12", "35"]
The quick brown fox ran over dogs.  ➔
      ["The", "quick", "brown", "fox", "ran", "over", "dogs."]
"Don't quote me," she said."  ➔
      ['"'+"Don't", "quote", 'me,"', 'she', 'said.']
```

35. … the number of words there are in your input expression.

36. … a list of all the characters (including duplicates) in the string.
eg. `"foo"` ➔ `["f", "o", "o"]`

37. … rearranging all the characters of the input string so that the characters are all in increasing order. Example: `"quick"` ➔ `"cikqu"`

38. … whether or not the input string is a palindrome (assume the input string has only lower case letters with no grammatical symbols.)

Still more string exercises

Given a string input (either from the command line or `input('Enter input')`) in a variable `s`, write a routine to …

39. … remove all punctuation from a string.
    eg. `"Don't quote me," she said.` ➔ `Dontquotemeshesaid`

40. … determine all the words that start and end in a vowel

41. … determine a character with the most number of occurrences.

42. … determine all characters that occur the most.

43. … determine all the words having at least three vowels

44. … capitalize the starting letter of every word of the input

45. … find an adjacent pair of characters with the greatest difference between their ascii values.

46. … find the first non-repeated character within a string

47. … remove all duplicate characters from a string. Eg. `detection` ➔ `detcion`

48. … print out the string with each word in the string reversed.

49. … treat the first word of the input as a search string to be found in the rest of the string. The script should return the number of occurrences that were found. Overlapping occurrences count:
    eg. `ana Anaphylactic banana` ➔ `2` (found at positions 14 and 16)

50. … treat the first word of the input as a search string to be found in the rest of the string, treats the second word as a replacement for the first, and treats the rest of the input as the string to be searched.
    eg. `b Ba baby boy` ➔ `BaaBay Baoy`

51. Write a script that accepts two strings as input and determines whether the two strings are anagrams of each other.

52. Have a loop and for each execution of the loop show the prior name that was entered, and ask for a new name. If there is no name given for the input, then exit the script. What you show as the name when the loop is first entered is up to you.