

---

# **Python Exercises Documentation**

***Release 1.0***

**tuxfux.hlp@gmail.com**

March 27, 2014

## CONTENTS

<b>1</b>	<b>UNIT I: Variables Expressions and Statements</b>	<b>1</b>
1.1	Basic Exercises . . . . .	1
<b>2</b>	<b>UNIT II: Control statements</b>	<b>3</b>
2.1	Conditions Exercises . . . . .	3
<b>3</b>	<b>UNIT III: Recursions</b>	<b>5</b>
3.1	Looping Exercises . . . . .	5
<b>4</b>	<b>UNIT IV: Data structures</b>	<b>6</b>
4.1	Data structure Exercises . . . . .	6
<b>5</b>	<b>UNIT V: Functions</b>	<b>10</b>
5.1	Function Exercises . . . . .	10



## UNIT I: VARIABLES EXPRESSIONS AND STATEMENTS

### 1.1 Basic Exercises

#### 1.1.1 Question 1:

Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically. Suppose the following input is supplied to the program: without,hello,bag,world Then, the output should be: bag,hello,without,world

Hints: In case of input data being supplied to the question, it should be assumed to be a console input.

#### 1.1.2 Question 2:

Write a program that accepts sequence of lines as input and prints the lines after making all characters in the sentence capitalized. Suppose the following input is supplied to the program: Hello world Practice makes perfect Then, the output should be: HELLO WORLD PRACTICE MAKES PERFECT

#### 1.1.3 Question 3:

Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically. Suppose the following input is supplied to the program: hello world and practice makes perfect and hello world again Then, the output should be: again and hello makes perfect practice world

Hints: In case of input data being supplied to the question, it should be assumed to be a console input. We use set container to remove duplicated data automatically and then use sorted() to sort the data.

#### 1.1.4 Question 4:

Write a program which accepts a sequence of comma separated 4 digit binary numbers as its input and then check whether they are divisible by 5 or not. The numbers that are divisible by 5 are to be printed in a comma separated sequence. Example: 0100,0011,1010,1001 Then the output should be: 1010 Notes: Assume the data is input by console.

Hints: In case of input data being supplied to the question, it should be assumed to be a console input.

### 1.1.5 Question 5:

Write a program that can convert a integer into a string and print it in console.

Hints:

Use `str()` to convert a number to string.

### 1.1.6 Question 6:

Write a program that can receive two integral numbers in string form and compute their sum and then print it in console.

Hints:

Use `int()` to convert a string to integer.

### 1.1.7 Question 7:

write a program that can accept two strings as input and concatenate them and then print it in console.

Hints:

Use `+` to concatenate the strings

## UNIT II: CONTROL STATEMENTS

### 2.1 Conditions Exercises

#### 2.1.1 Question 1:

WAP to find the min of the two numbers ?

#### 2.1.2 Question 2:

Wap to find the max of the three numbers ?

#### 2.1.3 Question 3:

Wap to find out if a given character is a vowel or consonant ? Note: 'a','e','i','o','u' are the vowels.

#### 2.1.4 Question 4:

Write a script which works out how old a child should be to know a certain word. The classification should be based on the length of the word, as follows:

- 5 years  $\leq$  3 letters
- 6 years  $\leq$  4 letters
- 8 years  $\leq$  6 letters
- 10 years  $\leq$  10 letters
- 12 years = any length

#### 2.1.5 Question 5:

Write a program that accepts a sentence and calculate the number of letters and digits.

Suppose the following input is supplied to the program:

hello world! 123

Then, the output should be:

LETTERS 10 DIGITS 3

Hints:

In case of input data being supplied to the question, it should be assumed to be a console input.

### 2.1.6 Question 6:

Write a program which accepts a string as input to print “Yes” if the string is “yes” or “YES” or “Yes”, otherwise print “No”.

Hints:

Use if statement to judge condition.

### 2.1.7 Question 7:

Write a small password checking script. This will record the username, old password and new password. The rules are that a password is OK if it is >7 characters long, contains some uppercase characters and is different to the old password. The admin user (username ‘admin’) can do whatever they like. Print out whether the new password is OK.

Try doing this as one compound if statement

## UNIT III: RECURSIONS

### 3.1 Looping Exercises

#### 3.1.1 Question 1:

Write a program, which will find all such numbers between 1000 and 3000 (both included) such that each digit of the number is an even number. The numbers obtained should be printed in a comma-separated sequence on a single line.

Hints: In case of input data being supplied to the question, it should be assumed to be a console input.

#### 3.1.2 Question 2:

Write a program that computes the net amount of a bank account based a transaction log from console input. The transaction log format is shown as following:

D 100 W 200

D means deposit while W means withdrawal.

Suppose the following input is supplied to the program: D 300 D 300 W 200 D 100

Then, the output should be: 500

Hints: In case of input data being supplied to the question, it should be assumed to be a console input.



## UNIT IV: DATA STRUCTURES

### 4.1 Data structure Exercises

#### 4.1.1 Question 1:

Perform a list assignment using the Python command line to assign the first 4 prime numbers 2,3,5 and 7 to a list called primes. Hint: you will need to use square brackets [] to assemble the list.

- Append the 5th prime number (11) to this list using the append() method. Print out the primes list
- Create another list called: b of the 6th to the 8th prime numbers ( which are 13, 17 and 19). Add b to the end of primes to create a new list c (adding primes and b together using the + operator). Write what happens when you print c.
- Append the list b to the end of the list primes (using the append() method ). (Explanation: a list item can itself be a list in Python).
- Print the value primes[5][2] . If you carried out the above exercises correctly this should print the value 19. Experiment with other indexes so that you print 17 or 13 in a similar way.

#### 4.1.2 Question 2:

Assign a list to a reference a , containing a regular sequence of 5 - 8 elements, such that if you knew the first 3 elements you would be able to predict the rest. E.G: [3,6,9,12,15,21,24] .

- Using a slice operation assign 2 elements from the middle of your sequence: e.g. 12 and 15 to another list called c .
- Take a backup of your list a in b by assigning: b=a You might need to copy the backup in b back to a if you screw a up.
- Using the del operator twice on indexed elements of list a, remove the 2 elements from the middle of a that you assigned into the list c. E.G if you had (blindly) used the above values your list might now look like: [3,6,9,18,21,24].
- Using a slice assignment operation, restore the list a to its original sequence by inserting list c into the middle of list a.

#### 4.1.3 Question 3:

Create a list made up of all the following 4 lower-case words, 1 word per list item: ["the","dead","parrot","sketch"] . Assign this list to a reference called: parrot . Write and save (as for1.py ) a short program using a for loop, which

prints out each word in turn, with the first letter capitalised, together with the length of each word. Your output should look like this:

```
The 3
Dead 4
Parrot 6
Sketch 6
```

Hint: you can split a string into 2 substrings using 2 slices. You can get a capitalised copy of one sub string using the `.upper()` method which is available if you import `string`. You can rejoin 2 substrings together again using the `+` operator.

- Save a copy of this program as `for2.py`. Edit it so that each time you go through the list and print a word you print one more uppercase letter than the previous word. Your output should look like this:

```
The
DEad
PARrot
SKETCh
```

To do this you will need an integer reference which counts the number of times you have been through the loop.

#### 4.1.4 Question 4:

Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200 (both included). The numbers obtained should be printed in a comma-separated sequence on a single line.

Hints: Consider use `range(begin, end)` method

#### 4.1.5 Question 5:

With a given integral number `n`, write a program to generate a dictionary that contains `(i, i*i)` such that `i` is an integral number between 1 and `n` (both included). and then the program should print the dictionary. Suppose the following input is supplied to the program: 8 Then, the output should be: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}

Hints: In case of input data being supplied to the question, it should be assumed to be a console input. Consider use `dict()`

#### 4.1.6 Question 6:

Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number. Suppose the following input is supplied to the program: 34,67,55,33,12,98 Then, the output should be: ['34', '67', '55', '33', '12', '98'] ('34', '67', '55', '33', '12', '98')

Hints: In case of input data being supplied to the question, it should be assumed to be a console input. `tuple()` method can convert list to tuple

#### 4.1.7 Question 7:

write a program which takes 2 digits, `X,Y` as input and generates a 2-dimensional array. The element value in the `i`-th row and `j`-th column of the array is `i*j` Note: `i=0,1,..., X-1`; `j=0,1,..., Y-1` Example: Suppose the following inputs are given to the program: 3,5 Then, the output of the program should be: [[0,0,0,0,0],[0,1,2,3,4],[0,2,4,6,8]]

Hints: Note: In case of input data being supplied to the question, it should be assumed to be a console input in a comma-separated form.

#### 4.1.8 Question 8:

Write a program that accepts a sentence and calculate the number of upper case letters and lower case letters. Suppose the following input is supplied to the program: Hello world! Then, the output should be: UPPER CASE 1 LOWER CASE 9

Hints: In case of input data being supplied to the question, it should be assumed to be a console input.

#### 4.1.9 Question 9:

Use a list comprehension to square each odd number in a list. The list is input by a sequence of comma-separated numbers. Suppose the following input is supplied to the program: 1,2,3,4,5,6,7,8,9 Then, the output should be: 1,3,5,7,9

Hints: In case of input data being supplied to the question, it should be assumed to be a console input.

#### 4.1.10 Question 10:

Define a function which can print a dictionary where the keys are numbers between 1 and 3 (both included) and the values are square of keys.

Hints:

Use dict[key]=value pattern to put entry into a dictionary. Use \*\* operator to get power of a number.

#### 4.1.11 Question 11:

Define a function which can print a dictionary where the keys are numbers between 1 and 20 (both included) and the values are square of keys.

Hints:

Use dict[key]=value pattern to put entry into a dictionary. Use \*\* operator to get power of a number. Use range() for loops.

#### 4.1.12 Question 12:

Question: Define a function which can generate a dictionary where the keys are numbers between 1 and 20 (both included) and the values are square of keys. The function should just print the values only.

Hints:

Use dict[key]=value pattern to put entry into a dictionary. Use \*\* operator to get power of a number. Use range() for loops. Use keys() to iterate keys in the dictionary. Also we can use item() to get key/value pairs.

#### 4.1.13 Question 13:

With a given tuple (1,2,3,4,5,6,7,8,9,10), write a program to print the first half values in one line and the last half values in one line.

Hints:

Use [n1:n2] notation to get a slice from a tuple.

#### 4.1.14 Question 14:

Write a program to generate and print another tuple whose values are even numbers in the given tuple (1,2,3,4,5,6,7,8,9,10).

Hints:

Use “for” to iterate the tuple Use tuple() to generate a tuple from a list.

#### 4.1.15 Question 15:

Write a program which can filter even numbers in a list by using filter function. The list is: [1,2,3,4,5,6,7,8,9,10].

Hints:

Use filter() to filter some elements in a list. Use lambda to define anonymous functions.

#### 4.1.16 Question 16:

Write a program which can map() to make a list whose elements are square of elements in [1,2,3,4,5,6,7,8,9,10].

Hints:

Use map() to generate a list. Use lambda to define anonymous functions.

#### 4.1.17 Question 17:

Write a program which can map() and filter() to make a list whose elements are square of even number in [1,2,3,4,5,6,7,8,9,10].

Hints:

Use map() to generate a list. Use filter() to filter elements of a list. Use lambda to define anonymous functions.

## UNIT V: FUNCTIONS

### 5.1 Function Exercises

#### 5.1.1 Question 1:

Define a function `max()` that takes two numbers as arguments and returns the largest of them. Use the if-then-else construct available in Python. (It is true that Python has the `max()` function built in, but writing it yourself is nevertheless a good exercise.)

#### 5.1.2 Question 2:

Define a function `max_of_three()` that takes three numbers as arguments and returns the largest of them.

#### 5.1.3 Question 3:

Write a function `max_in_list()` that takes a list of numbers and returns the largest one.

#### 5.1.4 Question 4:

Define a function that computes the length of a given list or string. (It is true that Python has the `len()` function built in, but writing it yourself is nevertheless a good exercise.)

#### 5.1.5 Question 5:

Write a function that takes a character (i.e. a string of length 1) and returns `True` if it is a vowel, `False` otherwise.

#### 5.1.6 Question 6:

Define a function `sum()` and a function `multiply()` that sums and multiplies (respectively) all the numbers in a list of numbers. For example, `sum([1, 2, 3, 4])` should return 10, and `multiply([1, 2, 3, 4])` should return 24.

#### 5.1.7 Question 7:

Define a function `reverse()` that computes the reversal of a string. For example, `reverse("I am testing")` should return the string "gnitset ma I".

### 5.1.8 Question 8:

Define a function `is_palindrome()` that recognizes palindromes (i.e. words that look the same written backwards). For example, `is_palindrome("radar")` should return `True`.

### 5.1.9 Question 9:

Write a function `is_member()` that takes a value (i.e. a number, string, etc) `x` and a list of values `a`, and returns `True` if `x` is a member of `a`, `False` otherwise. (Note that this is exactly what the `in` operator does, but for the sake of the exercise you should pretend Python did not have this operator.)

### 5.1.10 Question 10:

Define a function `overlapping()` that takes two lists and returns `True` if they have at least one member in common, `False` otherwise. You may use your `is_member()` function, or the `in` operator, but for the sake of the exercise, you should (also) write it using two nested `for`-loops.

### 5.1.11 Question 11:

Define a function `generate_n_chars()` that takes an integer `n` and a character `c` and returns a string, `n` characters long, consisting only of `c`:s. For example, `generate_n_chars(5,"x")` should return the string `"xxxxx"`. (Python is unusual in that you can actually write an expression `5 * "x"` that will evaluate to `"xxxxx"`. For the sake of the exercise you should ignore that the problem can be solved in this manner.)

### 5.1.12 Question 12:

Define a procedure `histogram()` that takes a list of integers and prints a histogram to the screen. For example, `histogram([4, 9, 7])` should print the following:

```
****
*****
*****
```

### 5.1.13 Question 13:

Write a function `find_longest_word()` that takes a list of words and returns the length of the longest one.

### 5.1.14 Question 14:

Write a function `filter_long_words()` that takes a list of words and an integer `n` and returns the list of words that are longer than `n`.

### 5.1.15 Question 15:

A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: The quick brown fox jumps over the lazy dog. Your task here is to write a function to check a sentence to see if it is a pangram or not.

### 5.1.16 Question 16:

Write a function `char_freq()` that takes a string and builds a frequency listing of the characters contained in it. Represent the frequency listing as a Python dictionary. Try it with something like `char_freq("abbabcbdbabdbbabababcbcbab")`.

Write a program which can compute the factorial of a given numbers. The results should be printed in a comma-separated sequence on a single line. Suppose the following input is supplied to the program: 8 Then, the output should be: 40320

Hints: In case of input data being supplied to the question, it should be assumed to be a console input.

### 5.1.17 Question 17:

Python has many built-in functions, and if you do not know how to use it, you can read document online or find some books. But Python has a built-in document function for every built-in functions. Please write a program to print some Python built-in functions documents, such as `abs()`, `int()`, `raw_input()` And add document for your own function

**Hints:** The built-in document method is `__doc__`