

# Python

## Lab Book



## Document Revision History

---

Date	Revision No.	Author	Summary of Changes
09-May-2015	1.0	Vishal	Content Creation

## Table of Contents

---

Document Revision History .....	2
Table of Contents .....	3
Getting Started .....	4
Overview .....	4
Setup Checklist for Python .....	4
Day 1	5
Day 2	6

## Getting Started

---

### Overview

This lab book comprises 'To Do' assignments for Python. Work out the 'To Do' assignments that are given.

### Setup Checklist for Python

Here is what is expected on your machine for the lab to work.

#### Minimum System Requirements

- Intel Pentium 90 or higher (P166 recommended)
- Microsoft Windows 95, 98, or NT 4.0, 2k, XP.
- Memory: 32MB of RAM (64MB or more recommended)

#### Please ensure that the following is done:

- Python 3.6.4

## Day 1

---

1. Input your name into a variable called `$name` and then print "Hello, <your name here>".
2. Write a program that adds two numbers and then prints out whether the sum of those two numbers is positive or negative.
3. Write a program that stores a number and keeps trying to get user input until the user enters the number correctly. As soon as the correct number is entered, it prints: Correct!
4. Input your first name and last name as two separate variables, labeled as `$firstname` and `$lastname` respectively. Concatenate them together using the dot operator `'.'` into a new variable called `$wholename`. Then print out the `$wholename`.
5. Write a program to accept an input string from the user and toggle the character cases.

For example, `$str=" Hello How Are You?"`

O/p : hELLO hOW aRE yOU

6. Write a program which will perform sum and multiplication ,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.
7. Write a program 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.)
8. Write a program that has two lists and print 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.
9. Write a program for histogram that takes a list of integers and prints a histogram to the screen. For example, `histogram([4, 9, 7])` should print the following:

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

## Day 2

---

1. 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.)
2. The function `max()` from exercise 1) and the function `max_of_three()` from exercise 2) will only work for two and three numbers, respectively. But suppose we have a much larger number of numbers, or suppose we cannot tell in advance how many they are? Write a function `max_in_list()` that takes a list of numbers and returns the largest one.
3. Write a program that maps a list of words into a list of integers representing the lengths of the corresponding words.
4. Write a function `find_longest_word()` that takes a list of words and returns the length of the longest one. Modify the same to do with lambda expression.
5. 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`. Modify the same to do with lambda expression.
6. Write a version of a palindrome recognizer that also accepts phrase palindromes such as `"Go hang a salami I'm a lasagna hog."`, `"Was it a rat I saw?"`, `"Step on no pets"`, `"Sit on a potato pan, Otis"`, `"Lisa Bonet ate no basil"`, `"Satan, oscillate my metallic sonatas"`, `"I roamed under it as a tired nude Maori"`, `"Rise to vote sir"`, or the exclamation `"Dammit, I'm mad!"`. Note that punctuation, capitalization, and spacing are usually ignored.
7. 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.
8. Represent a small bilingual lexicon as a Python dictionary in the following fashion `{"merry": "god", "christmas": "jul", "and": "och", "happy": "gott", "new": "nytt", "year": "år"}` and use it to translate your Christmas cards from English into Swedish. That is, write a function `translate()` that takes a list of English words and returns a list of Swedish words.
9. 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("abbabcbdbabdbdbabababcbcbab")`.

10. Create a module called mathematics.py and provide subroutines (should be defined generally and should work for any number of arguments) such as:

Add

Sub

Sort the values

Max

Sort

Use the module in a program and apply the functions on two array variables (say a and b) to:

Add two arrays (and store it in c)

Subtract two arrays (and store it in d)

Find the minimum and maximum value of the resultant array (c or d)

Sort the resultant array (c or d)

11. Try above programme with package.
12. Create a Date class, which represents the Date with its attributes. Write a UseDate class, which makes use of the Date class to instantiate, and call methods on the object.
13. WAP to read data from one file and writes in second file.
14. WAP which will display different function of math and numpy library.