Learn Python by solving problems

1. Introduction

Exercise 1 – Hello.py

# Write a program to enter your name (first name, surname), the program should respond hello to you.

Exercise 2 – Area of a surface.py

# Enter the width and length of a surface. The program should compute the area. Enter the values as floating point numbers and print the units of measurement used.

Exercise 3 – Age in five years.py

# Create a program that will tell your age in 5 years.

Exercise 4 – Converter Calculator.py

# Write a program that converts days in seconds, Celsius in Fahrenheit, miles in centimeters.

Exercise 5 – How many can you fit.py

# A shelf has x > 100 cm. Determine how many y < 15 cm products can you fit on it.

Exercise 6 – Savings account.py

# You just open a savings account that earns 2.1 interest per year. Display the amount you will have in your account over 1, 2, 3 years. Initial deposit is 45.

# Use only 2 decimal places for each amount.

Exercise 7 – Grand total.py

# Enter the price of a meal at the restaurant. Determine the tax paid for that meal at a tax rate of your choice and tip at the rate of 10% of meal amount without the tax.

# Display meal price, tax, tip and grand total on different lines in a user friendly format.

Exercise 8 – Sum of digits.py

# Determine the sum of digits for a 3 digits number. The program should display the following format for example 254: 2+5+4=11

Exercise 9 – Self checkout.py

# Determine how much change you need to provide if a shopper pays with cash at a checkout machine. Input the total to pay and the amount paid by the shopper.

# The machine should return as few coins as possible. Consider it is loaded with 1¢, 5¢, 10¢, 25¢, 50¢, $1.

# ex: for a total of $17.12 shopper paid $20 and received back 2 x $1 , 1 x 50c , 1 x 25c , 1 x 10c , 0 x 5c , 3 x 1c

Exercise 10 – Where is it from.py

# Enter a piece of text. Ask for a start and end number and then display just that section between.

Exercise 11 – How many letters.py

# Take 2 inputs, first and last name. Merge them with a space between and calculate the number of letters in your full name.

Exercise 12 – Distance on Earth.py

# Create a program where you enter the latitude and longitude of 2 points on Earth in degrees. The result should be the distance between them in kilometers.

# For lat1,long1 and lat2,long2 distance is: distance= 6371.01(average radius of Earth in km)\*arccos(sin(lat1)\*sin(lat2)+cos(lat1)\*cos(lat2)\*cos(long1-long2))

# You need to convert input from degrees in radians as Python`s trigonometric functions operate in radians.

1. Decision making

Exercise 13 – Biggest number.py

# Determine the biggest number between 2 numbers and display it an appropriate message.

Exercise 14 – Odd or even.py

# Decide if a number is odd or even.

Exercise 15 – Is it between.py

# Enter a number. If the number is between -10 and 10 then tell if it is positive, negative or 0. If number is outside range show an error message.

Exercise 16 – Leap Year.py

# Determine if a year is a leap year.

Exercise 17 – Make your choice.py

# Display square and triangle on screen. Ask user to choose a shape, if square is chosen, ask for length of 1 side and print the area.

# If triangle is chosen ask for height and base and display area. Your program should work regardless of user using or not uppercase letters for answer.

# Display an error message if no shape is selected.

Exercise 18 – Triangles.py

# Write a program that reads the lengths of a triangle sides and determine if it is equilateral (all sides the same), isosceles (2 sides same length) or scalene (all sides different).

Exercise 19 – Is it a hoiday.py

# Write a program that reads a day and a month. If the date matches: Christmas, New Year or Halloween, print the holiday.

Exercise 20 – Astrological sign.py

# Read a birthday and determine the astrological sign.

Exercise 21- Will you get a raise.py

# At a company employees get a raise every year based on performance. The values used for performance are only: 0 for Bad performance, 0.25 for Good performance

# and 0.50 for Exceptional Performance. Raise is equal to £2000 multiplied by performance. Write a program that can calculate this.

Exercise 22 – Overcharged.py

# A phone plan includes 100 minutes and messages and 6 GB of data for £10 a month. Every minute over limit is charged at 0.20p, every text at 0.08p and every 1 GB or less at £5.

# There is also a charge of 0.36p per month to support emergency call centers. The entire bill is subject to 6 percent tax.

# Take input number of minutes, texts and GB used and calculate the total of the bill with each charge printed on it`s own line.

Exercise 23 – Next day.py

# Take as input a day, month, year and display the date in the next format DD:MM:YYYY. Create a program that displays the next day in the same format.

# Make sure it works correct for leap years. Assume the user will always enter a valid date.

Exercise 24 – Black or white.py

# Letters on a chess board represent columns and numbers represent rows. Create a program that reads a position and prints if it is black or white.

1. Loops

Exercise 25 – Name name name.py

# Enter your name and display it 5 times. Upgrade the program to display it for a number of times chosen by you.

Exercise 26 – Na m e n a m e n a m e.py

# Enter your name and a number. Display your name one letter at the time on same line for that number of times.

Exercise 27 – Total.py

# Create a program where you add 5 numbers to the total. After each number print the total.

# Extend the program by asking if the numbers should be added.

Exercise 28 – Conversion table.py

# Convert from Celsius to Fahrenheit 0 to 100 degrees, only multiples of 5.

Exercise 29 – One million chronometer.py

# How long it takes to display all numbers from 0 to 1 million in steps of 10?

Exercise 30 – Binary to decimal.py

# Read a binary number as a string and convert it to a decimal number.

Exercise 31 – Palindrome.py

# A palindrom is a word that reads the same backwards as forwards. ex: eye, madam, level, radar. Check if a string is a palindrom.

Exercise 32 – Random number.py

# Print a random number under 100.

Exercise 33 – What to eat.py

# Chose a random meal from a menu of 5.

Exercise 34 – Coin flip 1.0.py

# Flip a coin and try to guess the result.

Exercise 35 – Test.py

# Create a small test using 2 different random numbers every question. Use only addition. 10 questions with one point for each, print score at the end.

Exercise 36 – Find that number.py

# Generate a number under 100 and keep adding numbers under 10 until you reach it or pass it.

# If you pass it you lose and game resets, reach it to win. Display total after each number.

Exercise 37 – Game developer.py

# Generate a number from 1 to 1000. Try to guess it in 10 attemps. Print too low or too high depending on your guess.

Exercise 38 – Polygon perimeter.py

# On a x,y axis compute the perimeter of a polygon. Enter the coordonates for each corner until x is skipped.

Exercise 39 – 10 elephants.py

# Using the song :

# 1 elephant swings happily

# On a spider web this morning

# The web is strong, it doesn't break

# So he calls another elephant

# Write a program that will ask you for the number of elephants. If number is right, print next lyrics up to 5.

# If number is wrong print a error and ask user to try again.

Exercise 40 – Coin flip 2.0.py

# Spin a coin. How many times you have to spin it to get the same outcome four consecutive times.

# Run the simulation 5 times, print the result on each row in the follwing format: H T H T T T T (7 flips)

1. Functions

Exercises 41-61 – Revision.py

# Revise previous exercises, try to create functions for as many as you can.

Exercise 62 – Is it prime.py

# Create a function that can determine if number is prime.

Exercise 63 – Next prime.py

# Take a number as input and create a function to determine the next prime number after it.

Exercise 64 – 3 numbers.py

# Take 3 random numbers as parameters and determine the average value.

Exercise 65 – Online shipping.py

# An online retailer offers shipping at the cost of 3.10 for the first item and 0.95 for each item after.

# Create a function that takes as input the number of items ordered and prints the shipping cost.

Exercise 66 – Days this month.py

# Create a function that will determine how many days are in a month.

# It should have 2 parameters, month and year. Make sure it works for leap years.

Exercise 67 – Counting letters.py

# Write a function that will calculate the number of uppercase and lowercase letters in a string.

Exercise 68 – Perfect.py

# Create a function that will determine if a number is perfect.

# A perfect number is a positive integer that is equal to the sum of its positive divisors excluding the number itselt.

# Check how many perfect numbers are in 1 million.

Exercise 69 – Password generator.py

# Create a function that will generate a password. Use ASCII table, in special items from 33 to 126 to create it.

# It should have between 8 and 15 characters. Create a main function that will display 5 passwords.

Exercise 70 – Strong password.py

# Create a function that will check if you have a strong password, at least 10 characters, 1 uppercase letter,

# 1 number and 1 symbol.

# Test this function on the password generated in the previous exercise and keep track how many attemps it takes to generate a strong password.

Exercise 71 – Calculator.py

# Create a simple calculator function that will be able to calculate + and – operations. It should work like every calculator,

# after first 2 numbers are calculated, you can keep making operations to the result.

# Break this problem in 3 separate functions: addition, substraction, main.