**Python 05: Functions, Lists and Tuples.**

**Introduction**: This document presents learning activities for Python 05. In Python 05, you will learn basics of functions to organise your large programs. A function is a fragment of code with a name to be reused in different places of the program. Moreover, in this week you will learn how to build a sequence structure using basic data types.

**Note:** Exercises of this learning path can be done using:

1. IDE: Using **BRef-01: Chapter 19, Section: Integrated Development Environment** PyCharm can be installed in your local machine.

**Materials:**

The activities are designed based on these following references:

* **BRef-01**: Book, Bill Lubanovic; "Introducing Python: Modern Computing in Simple Packages"; [Available here](https://www.oreilly.com/library/view/introducing-python-2nd/9781492051374/)
* **ORef-01**: Online Tutorial; Charles Severance; "Python for Everybody"; [Available here](https://books.trinket.io/pfe/index.html)

**Path:**

**Step-01: What is a function?**

**Goals:**

After taking this step, you will be able to:

1. interpret and implement Python programs with Python functions: function definition, calling functions, return of a function, functions with arguments.

**What to Learn?**

1. Using **BRef-01: Chapter 09** answer and experiment the following questions:
   1. What is a function in Python?

A function is basically a block that can be called. Example

Def funcname():

Code..

* 1. What are the main elements of a Python function? Define a simple function that does nothing.

Def do\_nothing():

pass

* 1. How can a function be used (called)?

Do\_nothing()

* 1. How can one return the result of a function?

Def something(a):

Return a

* 1. What are the arguments and/or parameters?

You fill the parameters with arguments.

**Exercises:**

*Note*: In the following exercises you can decide yourself what should be the name of function in your solution. Check [PEP8 Function and Variable Names](https://peps.python.org/pep-0008/#function-and-variable-names).

1. Explain in your own words the difference between arguments and parameters.

The difference is that you make parametes ( can be seen as placeholders ) for arguments. Arguments are then used within the block of the function.

1. Create a function that just prints the word hello. Call the function and run your program. Where the function is *defined*? Where is it *called*?

Def helloworld()

X = “Hello world”

Return X

Helloworld()

You could also do it like

Def helloworld(x):

Return x

Helloworld(“Hello world”)

1. Create a function that takes a text as an argument. The function prints the text it receives. Call the function and run your program.

Def helloworld(x):

Return x

helloworld(“Hello world”)

1. Create a function that takes two numbers as argument. The function adds the numbers together and returns the results. Call the function and run your program.

Graphical user interface, text

Description automatically generated

1. Create two functions, each takes a number as argument. The first one returns the number multiplied by 2 and returns it. The second multiplies it by 10 and returns it. Calling both functions add the two returned numbers together and print it. Run your program and check the results.

Graphical user interface, text, application

Description automatically generated

1. Create two functions. One that prints hello, the other prints bye. Ask the user to input a number, if the number is higher than 10, call the first function. If the number if lower or equal to 10, call the second function. Test your program.

Text

Description automatically generated

1. Create two functions. One that prints hello, the other prints bye. The first functions calls the second one after printing. Call the first function.

Graphical user interface, text, application

Description automatically generated

1. Provide your solutions to the exercises of Python 05: Step-01. **ORef-01: Functions** can be used as extra learning reference.

-

1. Design two exercises of your own. They should improve understanding topics of this step.
2. Install *PyCharm* on your working machine. Implement and run a simple Python program of your choice.
   * It is important to learn how to create a new Python program, how to configure interpreter and how to run the program. Where do you see the results?

<hr>

**Step-02: Tuples.**

**Goals:**

After taking this step, you will be able to:

1. interpret and implement Python programs using tuples.

**What to Learn?**

1. Using **BRef-01: Chapter 07** answer and experiment the following questions:
   1. What is a tuple in Python and how is it defined?

Tuples are stored as multiple items in a single variable

myTuple = (“Item1”, “Item2”, “Item3”)

Tuples are written with round brackets

* 1. How can one combine and compare two (or more) tuples?

You can concatenate the two tuples;

myTuple1 = ("Cucumber", "Paprika")

myTuple2 = ("Banana", "Cherry")

combinedTuple = myTuple1 + myTuple2

print(combinedTuple)

* 1. How can one iterate over the elements of a tuple?

Same as a list.

myTuple1 = ("Cucumber", "Paprika","Banana", "Cherry")

for i in range(0, len(myTuple1)):

print(myTuple1[i])

* 1. How is a tuple modified?

Tuples cannot be modified because it is immutable and cannot be changed. BUT there is a kinda hackfix. You can convert it to a list, change the list and convert it back to a tuple.

Example:   
x = ("apple", "banana", "cherry")

y = list(x)

y[0] = "kiwi"

x = tuple(y)

print(x)

**Exercises:**

1. You can create a tuple with mixed types in it, for example texts and numbers. Can you think of a advantage and a disadvantage of doing this? Implement your example.

myTuple = (“String”, int(1), True, False, 0.2, None)

1. Create a tuple with three numbers in it. Unpack the tuple into three different variables. Print the last one.

Text

Description automatically generated

1. Create a tuple with two numbers in it, create a second tuple with two texts in it. Add them together into a new tuple. Print the new tuple.

Graphical user interface

Description automatically generated

1. Create a tuple with three numbers in it. Use a for loop to iterate over each value. Multiply each value by 2 and print each result.

A screenshot of a computer screen

Description automatically generated with medium confidence

1. Create a function that returns a tuple containing three texts. Call the function. Unpack the tuple into variables. Print the variables.

Graphical user interface, text

Description automatically generated

1. Analyse the two given codes below without executing them. What will be the result of the programs?

a\_tuple = ('Never', 'gonna', 'give', 'you', 'up')

counter = 0

for x in thistuple:

if x[0] == 'g':

counter = counter + 1

else:

counter = counter + 2

print(counter)

def do\_something(x):

rtuple = x,

for i in range(2,11):

rtuple = rtuple + ((x\*i),)

return rtuple

print(do\_something(6))

1. Design two exercises of your own. They should improve understanding topics of this step.

<hr>

**Step-03: Lists.**

**Goals:**

After taking this step, you will be able to:

1. interpret and implement Python programs using lists: defining, offset, slicing, adding new element, modifying an element.

**What to Learn?**

1. Using **BRef-01: Chapter 07** answer and experiment the following questions:
   1. What is a list in Python and how is it defined?

mylist = ["test", "test2"]

* 1. What is the result of *split()* on a string?

Graphical user interface, application, Teams

Description automatically generated

* 1. There are two ways to get items from a list: offset and slice. What are the pros / cons of each? Experiment with some examples.

With slice it becomes two lists but with offset it can take for example only half of the list or the half end of the list.

* 1. How can you add new elements to a list?

.Append

* 1. How can you modify elements of a list?

Select the index of the list and change it. myList[2] = “Viktor”

* 1. How can one iterate over the elements of a list?

For loop. Selecting the index and the length of the list

* 1. *Mutability* is one of the main differences between a tuple and a list. Elaborate this with some example.

Mutability means it is changeable the data. But the tuple cannot be modified. ( Specific data ). You have to convert it to a list and then back to a tuple if you needed.

**Exercises:**

1. Define a list of integers representing scores of a game. Write a program that prints out maximum and minimum of the scores.
2. Extend your program from the previous exercise such that it prints two largest and two smallest elements of the scores.
3. Write a program that asks the user to enter a list of integers. Do the following:
   * Print the total number of items in the list.
   * Print the last item in the list.
   * Print the list in reverse order.
   * Print Yes if the list contains a 5 and No otherwise.
   * Print the number of fives in the list.
   * Remove the first and last items from the list, sort the remaining items, and print the result.
   * Print how many integers in the list are less than 5.
4. Write a program that generates a list of 20 random numbers between 1 and 100.
   * Print the list.
   * Print the average of the elements in the list.
   * Print the largest and smallest values in the list.
   * Print the second largest and second smallest entries in the list
   * Print how many even numbers are in the list.
5. Start with the list [8,9,10]. Do the following:
   * Set the second entry (index 1) to 17
   * Add 4,5, and 6 to the end of the list
   * Remove the first entry from the list
   * Sort the list
   * Double the list
   * Insert 25 at index 3
   * The final list should equal [4,5,6,25,10,17,4,5,6,10,17]
6. Create the following lists using a loop.
   * A list consisting of the integers 0 through 49
   * A list containing the squares of the integers 1 through 50.
   * The list ['a','bb','ccc','dddd', . . . ] that ends with 26 copies of the letter z.
7. Write a program that takes any two lists L and M of the same size and adds their elements together to form a new list N whose elements are sums of the corresponding elements in L and M. For instance, if L=[3,1,4] and M=[1,5,9], then N should equal [4,6,13].
8. Design two exercises of your own. They should improve understanding topics of this step.

<hr>

**Problems:**

**General Notes:**

* Follow [PEP8 Style Guide for Python Code](https://peps.python.org/pep-0008/#code-lay-out) and apply coding style in your solutions.
* After taking Step 01: try to implement a partial solution for each program. More likely it will not be complete. Just think about algorithm and functions you would need. Save your partial solutions in your working folder.
* After taking Step 02: try to continue with your partial solution for each problem. Can you make it complete? Then, it is dones. Do you need Step 03? Save it for later.
* After taking Step 03: All solutions must be complete.
* A primary school teacher needs to automate basic arithmetic (summation, multiplication table, subtraction) exercises for her students. You are asked to implement a program that asks what type of the arithmetic the user needs to practice. Then, the program will generate exercises and the user should give the result. Consider the following features for the program:
  + For each arithmetic operation keep the total number of the exercises 10.
  + The program must be interactive: for example, if the chosen exercise is multiplication table, then the program generates two random numbers (check how python can generate random integers: **Week 01: Step 01, Exercise 2, Code 4**), like 3 and 5; the program prints 3 \* 5 = and the user must give the result; the program will print if the answer was correct or wrong and then the program will generate next question.
  + Your program must be implemented with a function arithmetic\_operation(arithmetic\_type), which can create a sum for each arithmetic operation based on the input given by a user (summation, multiplication, subtraction).
  + Numbers for summation and subtractions will be between 1 and 100.
  + For other aspects of the program feel free to decide your choice.
  + **Extented version**: Extend your program such that it collects all the mistakes from the user and prints them at the end.
  + **Extented version**: The teacher would like to know which questions are difficult for her students. Extend your program such that it measures the time that the students takes to answer each question. For each question collect the information in a tuple like (question, correct or wrong,time). The program collects all the results in a list and prints them at the end.
  + **Extented version**: Sort the list of the results based on their time before printing it.
* In a particular jurisdiction, taxi fares consist of a base fare of 4.00 eur, plus 0.25 eur for every 140 meters traveled. Write a function called calculate\_fare that takes the distance traveled (in kilometers) as its only parameter and returns the total fare as its only result. Write a main program that interacts with the user and calls the function to produce and print the final result.
* If you have 3 straws, possibly of differing lengths, it may or may not be possible to lay them down so that they form a triangle when their ends are touching. For example, if all of the straws have a length of 6 inches, then one can easily construct an equilateral triangle using them. However, if one straw is 6 inches long, while the other two are each only 2 inches long, then a triangle cannot be formed. In general, if any one length is greater than or equal to the sum of the other two then the lengths cannot be used to form a triangle. Otherwise they can form a triangle. Write a function called check\_triangle that determines whether or not three lengths can form a triangle. The function will take 3 parameters and return a Boolean result. In addition, write a program that reads 3 lengths from the user and demonstrates the behaviour of this function, on a correct triangle print possible otherwise impossible.
* In this exercise you will write a function named is\_integer that determines whether or not the characters in a string represent a valid integer. When determining if a string represents an integer you should ignore any leading or trailing white space. Once this white space is ignored, a string represents an integer if its length is at least 1 and it only contains digits, or if its first character is either + or - and the first character is followed by one or more characters, all of which are digits. Write a main program that reads a string from the user and reports whether or not it represents an integer.
  + **Extended version**: Extend your program with a different function remove\_non\_integer that if the given input string contains mixed digits and some alphabetic characters, it removes the alphabetic characters and prints the remaining integer. Example: given -12R0A89s the program will generate -12089. The ersult of +012R0A89s will be 12089.
* Write a function named generate\_random\_password that generates a random password. The password should have a random length of between 7 and 10 characters. Each character should be randomly selected from positions 33 to 126 in the ASCII table. Your function will not take any parameters. It will return the randomly generated password as its only result. Display the randomly generated password in your file’s main program. **Note**: Check how Python can choose random elements of a list or string.
* The Twelve Days of Christmas is a repetitive song that describes an increasingly long list of gifts sent to one’s true love on each of 12 days. A single gift is sent on the first day. A new gift is added to the collection on each additional day, and then the complete collection is sent. The first three verses of the song are shown below. The complete lyrics are available on the internet.

*On the first day of Christmas my true love sent to me: A partridge in a pear tree.*

*On the second day of Christmas my true love sent to me: Two turtle doves, And a partridge in a pear tree.*

*On the third day of Christmas my true love sent to me: Three French hens, Two turtle doves, And a partridge in a pear tree.*

Your task is to write a program that displays the complete lyrics for The Twelve Days of Christmas. Write a function named next\_verse that takes the verse number as its only parameter and displays the specified verse of the song. Then call that function 12 times with integers that increase from 1 to 12. Each item that is sent to the recipient in the song should only appear once in your program, with the possible exception of the partridge. It may appear twice if that helps you handle the difference between “A partridge in a pear tree” in the first verse and “And a partridge in a pear tree” in the subsequent verses.

* Implement a program that takes a quiz. The quiz contains 10 general questions, like *What is the capital of France?*. The user answers the question and the program must check if the answer is correct or not.

**Assignment:**

1. A dataset is given with information of students: student number, first name, last name, date of birth, study program. You are asked to implement a program that given this dataset (as a csv file), the program processes the information. The requested criteria are:
   * Sometimes data values are corrupted. The program must report corupted values. Any invalid or empty value is defined as corrupted.
   * Student number has this format: 7 digits, starting with 0 and second digit (from left) can be either 9 or 8. Example: 0212345 is not valid
   * First name and last names, contains only alphabet.
   * Date of birth has this format: YYYY-MM-DD. Days between 1 and 31, months between 1 and 12 and Years between 1960 and 2004.
   * Study program can have one of these values: INF, TINF, CMD, AI.
   * A template Python file is provided with a function that loads the data set.
   * A Python file and a dataset are provided. The Python file is an incomplete program with todos comments for the solution. The incomplete Python program is available in **/week05/assignment\_data/processstudentscsv.py** and the dataset is in **/week05/assignment\_data/students.csv**.
   * The program should make two separate lists: list of rows with correct values and a list of rows with corrupted values. These two lists will be printed with this format:

### VALID LINES ###

<valid\_line\_1>

<valid\_line\_2>

### CORRUPT LINES ###

<corrupt\_line\_1> => INVALID DATA: [all invalid founds on this line, separated by comma]

<corrupt\_line\_2> => INVALID DATA: [all invalid founds on this line, separated by comma]

**Extra Steps:**

**Problems:**

1. Implement the tic-tac-toe game (console based).
   * For two players and in each round the program asks the players to specify the position.
   * After giving the position by each player, the board is printed in the terminal.
   * The program determines the winner at the end.
   * Employ lists, tuples and functions in your program.
   * **Extended Version**: Extend your program with *two dimensional* lists.