

▼ Coding Tasks - Week 3

Welcome to the Python Programming Exercise Sheet!

In this exercise sheet, we will cover some of the fundamental concepts in Python programming.

Topics covered are: For- and While-Loops and list methods.

DEADLINE: 22nd May 12:15

Your name here: Shanza Amber

Your university mail: samber@uni-osnabrueck.de

Important information:

In order to pass this sheet you need to achieve 10/20 points.

For the best possible grade you require 20/20 points, however, since some harder tasks may take a lot of time you don't have to pressure yourself.

If you complete any three tasks on a sheet you will definitely get a good grade for that sheet.

Hand in your sheet in studip in the respective folder until the deadline.

If you receive no email until a few days after submission you will have passed, the sample solution will also be uploaded around then.

If you receive an email you don't need to worry, you can fail one sheet and also your total points will also be taken into account for the final pass or fail.

▼ Task 1 Looping over strings (4 points)

Given the string below, print each character of the string and count the number of vowels using a loop structure.

Finally, print the number of vowels (AEIOUaeiou) as an output, as stored in the variable `vowel_count`.

Hint: You may want to check again what the `in` operator in python does for statements like `'A' in string`.

```
my_string = "Summer is coming!"
vowel_count = 0

for letter in my_string:
    print(letter)

    if letter.lower() in "aeiou":

        vowel_count += 1

print("The number of vowels is: ", vowel_count)
```

```
S
u
m
m
e
r

i
s

c
o
m
i
n
g
!
The number of vowels is:  5
```

▼ Task 2 For and While conversion (6 points)

For each part you are given an empty list `result`, and a for or while loop.

Convert the for-loops into while-loops and the while-loops into for-loops such that the result will be identical.

Hint: You may use either `result += [element]` or `result.append(element)` for the element you wish to add to the `result` list.

You may find the usage of the condition `number % 2 == 0` useful for an if statement in a while loop for the first task.

▼ Part 1 - Even numbers from 2 to 20

A loop which creates a list of even numbers from 2 to 20.

Convert to a while loop.

```
result = []
for number in range(2, 21, 2):
    result += [number]

print(result)

[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

```
result = []

number = 2
while number <= 20:
    result.append(number)
    number += 2

print(result)

[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

▼ Part 2 - Snake trivia

A loop over a list of strings, then add every string which is longer than 8 characters to the `result` list.

```
python_text = "The Pythonidae, commonly known as pythons, are a family of nonvenomous snakes found in Africa, Asia, and Australia. Among its members, the reticulated python is the largest non-venomous snake in the world."
python_text = python_text.split(" ")
print(python_text)
```

```
['The', 'Pythonidae,', 'commonly', 'known', 'as', 'pythons,', 'are', 'a', 'family', 'of', 'nonvenomous', 'snakes', 'found', 'in', 'Africa', 'Asia', 'and', 'Australia.', 'Among', 'its', 'members,', 'the', 'reticulated', 'python', 'is', 'the', 'largest', 'non-venomous', 'snake', 'in', 'the', 'world.']
```

```
result = []
i = 0

while i < len(python_text):
    if len(python_text[i]) > 8:
        result += [python_text[i]]

    i += 1

print(result)
```

```
['Pythonidae,', 'nonvenomous', 'Australia.', 'currently', 'recognized.', 'naturally', 'non-venomous,', 'constrict', 'suffocate', 'consume', 'the', 'largest', 'non-venomous', 'snake', 'in', 'the', 'world.']
```

```
result = []

for text in python_text:
    if len(text) > 8:
        result.append(text)

print(result)

['Pythonidae,', 'nonvenomous', 'Australia.', 'currently', 'recognized.', 'naturally', 'non-venomous,', 'constrict', 'suffocate', 'consume', 'the', 'largest', 'non-venomous', 'snake', 'in', 'the', 'world.']
```

▼ Task 3 List methods (5 points)

Examine the list methods and find methods suitable for the following four tasks.

Important: Some methods will alter the object they are called on and some methods may or may not return something. For example, `string = string.split(",")` requires the re-assignment to overwrite `string`, while `list.remove()` does not need a re-assignment and will alter the list. For the following methods, test and read so you know which is the case where.

```
print(dir(list))
```

```
['__add__', '__class__', '__class_getitem__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__', '__format__']
```

▼ Part 1

Sort the following list alphabetically from A-Z:

```
names = ["Donald Duck", "Minnie Mouse", "Mickey Mouse", "Goofy", "Tick", "Trick", "Track", "Scrooge McDuck", "Daisy Duck", "Pete", "Pluto"]

names.sort()

print(names)

['Daisy Duck', 'Donald Duck', 'Goofy', 'Mickey Mouse', 'Minnie Mouse', 'Pete', 'Pluto', 'Scrooge McDuck', 'Tick', 'Track', 'Trick']
```

▼ Part 2

Get the index of "March" in the list and print it:

```
months = ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"]

idx = months.index("March")

print (idx, " is the index of ", months[idx])

2 is the index of March
```

▼ Part 3

Retrieve the last object of the list using the `pop()` method and store it in the variable `removed`.

This method removes the last element of a list and alters the list, while also returning it so you can save or use it somewhere.

```
alphabet = ["A", "B", "C", "D", "E", "F", "G", "H"]

removed = alphabet.pop()

print(removed, " was removed from the alphabet list, which is now: ", alphabet)

H was removed from the alphabet list, which is now: ['A', 'B', 'C', 'D', 'E', 'F', 'G']
```

▼ Part 4

Previously we have removed an item from the alphabet list, now we wish to add it back.

Add the "H" letter back into the list by inserting it.

You may need `len(alphabet)` here as argument.

```
alphabet = ["A", "B", "C", "D", "E", "F", "G"]

alphabet.insert(len(alphabet), "H")

print(alphabet)

['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']
```

▼ Part 5

Get the number of occurrences of the integer 2 inside this list and print it:

```
numbers = [1,2,1,3,2,3,2,1,3,2,2,1,3,2,2,2,1,2,2,3,2,1,3,1,2,3,1,2,3,1,2]

n = numbers.count(2)

print("2 appears ", n, " times in the list.")

2 appears 15 times in the list.
```

▼ Task 4 Restaurant Roulette (5 points)

Finally it is the weekend!

You are going out with some friends to enjoy a nice dinner in Osnabrück.

Some of you are really generous that you cannot even decide who is going to pay for the bill.

In order to tackle this issue, you are going to create a program that will collect all of your names into a list and then selects a random name from that list.

Whoever is selected will have to pay for everyone's bill.

Implement a while loop that receives inputs, using, for example, `input("Enter a name or \'stop\' , to stop:")` and stores the input in the list `names`.

Make sure to add a condition to the while loop that will stop the loop, for example checking each iteration if the last input is equal to "stop".

Finally, store the selected name in the variable `selected_name` and print it.

Hint: For this we imported the `random` module, how modules work will be explained in more detail later.

For all intents and purposes you can ignore it and simply look at what it does below the loop you implement.

```
import random

names = []

name = input("Enter a name or 'stop' to stop: ")

while name != "stop":
    names.append(name)
    name = input("Enter a name or 'stop' to stop: ")
selected_name = random.choice(names)

print(selected_name, " will have to pay for everyone's bill.")

Enter a name or 'stop' to stop: bala
Enter a name or 'stop' to stop: stop
bala will have to pay for everyone's bill.
```