Double-click (or enter) to edit

Coding Tasks - Week 2

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: String methods, indexing, and slicing.

DEADLINE: 21st May until 23:59
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 the submission deadline 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 Multiline printing and number of characters (5 points)

Part 1

We have learned that there are different ways of printing in multiple lines, using linebreaks for example.
Now it is time to write a program that splits the given string into a list of lines.
Hint: You may need to look up the `split()` function, and make sure the variable `programmer_joke` is overwritten with the method result.

```
programmer_joke = "That\ncode\nlooks\nlike\nit\nis\nsnaking\naround"
programmer_joke = programmer_joke.split('\n')
print(type(programmer_joke))
```

```
<class 'list'>
```

```
print(programmer_joke)
```

```
['That', 'code', 'looks', 'like', 'it', 'is', 'snaking', 'around']
```

Part 2

By now you are familiar with receiving user input using the `input()` function.
Write a program that receives two user inputs and stores them in `string_a` and `string_b`.
If the first string provided is longer than the second string, print out `True`, if not, print `False`.

```
string_a = input("Enter the first input: ")
string_b = input("Enter the second input: ")

if len(string_a) > len(string_b):
    print(True)
else:
    print(False)
```

```
Enter the first input: 55
Enter the second input: 7
True
```

Task 2 If-statements, methods, and strings (4 points)

Write three conditional statements (filled in for the `True`), to handle the following four cases.
For this you may need to use string methods, you can look at this link.
You can find a single method in the list which will solve either conditional, an example is `char.isalnum()` which will return True if all letters in `char` are alphanumerical.
Please handle the following cases for the if-statements in order: 1) The char is not an alphanumerical letter. 2) The char is a number. 3) The char is upper-case. 4) The char is lower-case.

```
char = 'CAT'


if not char.isalnum():
    print("The character <", char, "> is not alphanumerical.")

if char.isdigit():
    print("The character <", char, "> is a number.")

elif char.isupper():
    print("The character <", char, "> is an upper case alphabetical.")

elif char.islower():
    print("The character <", char, "> is a lower case alphabetical.")

else:
    print("The character <", char, "> neither of the former options.")
```

```
The character < CAT > is an upper case alphabetical.
```

Task 3 Slicing and Indexing (5 points)

In this task you will receive a list and are asked to change something in it or to access only specific elements of the list.
For each cell you will have a `result` variable with a print statement right after for checking, make sure you change the value of `result` accordingly.

Retrieve the `"Cherry"` element of the list.

```
shopping_list = ["Apple", "Banana", "Cherry", "Tomato", "Potato", "Onion"]

result = shopping_list[2]
print(result)
```

```
Cherry
```

link textRetrieve the first two elements of the list.

```
shopping_list = ["Apple", "Banana", "Cherry", "Tomato", "Potato", "Onion"]

result = shopping_list[:2]

print(result)
```

```
['Apple', 'Banana']
```

Retrieve the first two and the last two elements of the list.
HINT: You may want to use list-addition `list + list`

```
from re import S
shopping_list = ["Apple", "Banana", "Cherry", "Tomato", "Potato", "Onion"]

result = shopping_list[:2]+ shopping_list[-2:]
print(result)
```

```
['Apple', 'Banana', 'Potato', 'Onion']
```

Retrieve every second item of the list, starting from the Apple.

```
shopping_list = ["Apple", "Banana", "Cherry", "Tomato", "Potato", "Onion"]

result = shopping_list[::2]
print(result)
```

```
['Apple', 'Cherry', 'Potato']
```

Retrieve the list in reverse.

```
shopping_list = ["Apple", "Banana", "Cherry", "Tomato", "Potato", "Onion"]

result = shopping_list[::-1]
print(result)
```

```
['Onion', 'Potato', 'Tomato', 'Cherry', 'Banana', 'Apple']
```

Retrieve a list with at least one dairy product, fruit, and sweet, by taking elements from each of the presented lists (don't just make a new list, that will not give you points).
HINT: You can do this by using the `[list[idx]]+[list[idx]]+[list[idx]]` operation.

```
shopping_list_fruit = ["Apple", "Banana", "Cherry", "Tomato", "Potato", "Onion"]
shopping_list_dairy = ["Joghurt", "Milk"]
shopping_list_sweets = ["Chocolate", "Popsicle", "Gummy bears"]

result = [shopping_list_fruit[1]] + [shopping_list_dairy[0]] + [shopping_list_sweets[2]]
print(result)
```

```
['Banana', 'Joghurt', 'Gummy bears']
```

Task 4 Manipulating strings (6 points)

In this task you are required to manipulate a string `file_name` through sequence of actions as follows.
Try different file extensions for testing, make sure it works for different file names as well as intended.

```
file_name = "my_picture.jpg"
```

Check whether the file contains special characters like colon ":" or coma "," or dollar sign "$", using an if statement.
Then, and only if the symbol is in the file name, replace it with an underscore character (`"_"`), using string methods.

Finally, check the file type from the extension, and assign the variable `file_type` the following values:

- `".txt"` and `".doc"` and `".pdf"` should output `"Document file"`
- `".jpg"` and `".png"` should output `"Image file"`
- `".mp4"` and `".mov"` should output `"Video file"`
- for other types, or if no extension exists, it should print `"Unsupported file format"`

HINT: For this you may want to take a look at the last three characters of the string in an if statement.

```
file_name = "my_picture.jpg"

if ":" in file_name or "," in file_name or "$" in file_name:
    file_name = file_name.replace(":", "_").replace(",", "_").replace("$", "_")

if file_name.endswith(".txt") or file_name.endswith(".doc") or file_name.endswith(".pdf"):
    file_type = "Document file"

elif file_name.endswith(".jpg") or file_name.endswith(".png"):
    file_type = "Image file"

elif file_name.endswith(".mp4") or file_name.endswith(".mov"):
    file_type = "Video file"

else:
    file_type = "Unsupported file format"

print(file_type)
```

```
Image file
```