Exercises Week 2-3 (Strings)

EX.6-1 UPPER/lower CASE

Open VSC then -> File -> Open folder -> select “Python course” folder.

Create a new file “ex6-1.py”

TASK: Your task is to take any string, for example:

my_str = "Life's too short to be serious. Smile and laugh more!"

and perform the following transformations:

(1) Convert the entire string to uppercase.
(2) Convert the entire string to lowercase.
(3) Convert the entire string to title case, where the first letter of each word is capitalized.

Test your function with different strings. Does it work as expected?

EXAMPLE:

```
Life's too short to be serious. Smile and laugh more!
LIFE'S TOO SHORT TO BE SERIOUS. SMILE AND LAUGH MORE!
life's too short to be serious. smile and laugh more!
Life'S Too Short To Be Serious. Smile And Laugh More!
```

HINT: Use Python functions **upper**(), **lower**(), **title**()

EX.6-2 Random password generator

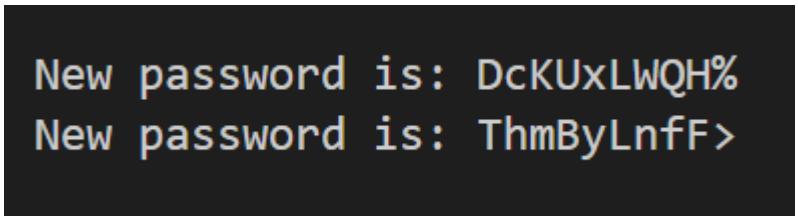
Create a new file “ex6-2.py”

TASK (1): Write a Python function that generates a random password with the following criteria:

(1) The generated password should be a combination of random characters (10 characters in total).
(2) First nine characters are lowercase and uppercase letters, or digits.
(3) Last character is one of punctuation characters (@, #, $, %, &, *, etc).

Your function should return the generated password.

EXAMPLE:



HINT:

(*) Use a list of alphanumeric characters for first nine digits. Use **choice**() function from **random** module to select a random character from it:

char_alphanumeric = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"

(**) Use a list of punctuation characters for the last character. Use **choice**() function from **random** module to select a random character from it:

char_punctuation = "!$%&'()*+,-/:;<=>?@[\]^_{|}~"

TASK (2). Modify your code to get a password of any length. Pass a length of the password to your function as a parameter.

TASK (3) - optional: Modify your code using functions **string.ascii_letters**(), **string.digits**(), **string.punctuation**() from module **string** instead of using a list of characters from TASK 1.

First, import module **string** then print (for your reference):

print(string.ascii_lowercase)

print(string.ascii_uppercase)

print(string.ascii_letters)

print(string.digits)

print(string.punctuation)

print(string.printable)

EX.6-3 Greeting function

Create a new file “ex6-3.py”

TASK (1): Create a Python function that generates a greeting based on the time provided as a string in the format "HH:MM", for example, "07:15" or "18:59".

Requirements:

(1) The function should take a time string as a parameter.
(2) The function should return "Good morning" if the time is between "00:00" and "11:59", "Good afternoon" if the time is between "12:00" and "17:59", "Good evening" if the time is between "18:00" and "23:59", and "Hello" in all other cases.
(3) Test your function with different values of the parameter, including "00:00", "11:59", "15:00", "18:00", and incorrect inputs such as "23:88", "0000000", and "15:99". Observe if the function works as expected and provides the correct greetings for each time.

EXAMPLE:

```
00:00 Good morning
11:59 Good morning
15:00 Good afternoon
18:00 Good evening
23:88 Hello
0000000 Hello
15:99 Good afternoon
```

HINT: Use **if-elif-else** statement and **>= <=** to compare strings.

TASK (2) – advanced: Let’s explore datetime datatype. Modify your code to make the time comparison more accurate. Use **strptime()** function from **time** module to convert the string to time variable, for example:

time_variable = strptime(“15:45”, '%H:%M')

Test your function for correct and incorrect input.
Learn more here: https://www.freecodecamp.org/news/python-string-to-datetime-how-to-convert-an-str-to-a-date-time-with-strptime/

https://www.w3schools.com/python/python_datetime.asp

Print the current date and time using function **datetime.now()** from module **datetime**