

Tasca3_Sprint3_Control_Structures

January 15, 2021

1 Tasca 3, Sprint 3 Data Structures

```
In [1]: ## This is a function defined to print some of the outputs in blue color.
        # This is to distinguish certain messages if there are many lines of output.
        BLUE = '34m'

        def display_colored_text(color, text):
            colored_text = f"\033[{color}]{text}\033[00m"
            return colored_text
```

1.1 Exercise 1

Create a program that rates you a numeric variable based on the Suspès/Aprovat/Notable/Excel·lent scale.

Solution:

We have to write a code that tells the exam grades based on the value inserted by user.

Standard Spanish grading scales are:

0 - 4.99 Suspès
5 - 6.99 Aprovat
7 - 8.99 Notable
9 - 10 Excel·lent

For this, I have created a function (**exam_grades**) that uses *if-else* statements to assign grades based on the conditions above. Then, I use a **user-input to enter a value**, and calls the function *exam_grades* to display the grade rating of the value.

If the entered value is less than 0 or more than 10, an error message is displayed.

```
In [2]: # @Pre: True
        # Post: Message displaying grade correspondint to entered value.
        def exam_grades(grade):
            try:
                assert(grade >= 0 and grade <=10), "Error: Number not acceptable! Please enter a valid value."
                if(grade >=0 and grade <= 10 ):
                    if(grade < 5):
                        print(display_colored_text(BLUE, 'Your grade is Suspès'))
                    elif(grade >= 5 and grade < 7):
```

```

        print(display_colored_text(BLUE, 'Your grade is Aproved'))
    elif(grade >= 7 and grade < 9):
        print(display_colored_text(BLUE, 'Your grade is Notable'))
    else:
        print(display_colored_text(BLUE, 'Your grade is Excel.lent'))
except AssertionError as msg:
    print(msg)

```

```

#exam_grades(-1)
#exam_grades(3.6)

```

```

grade = float(input("Check grade for value = "))
exam_grades(grade)

```

```

Check grade for value = 8.3
Your grade is Notable

```

1.2 Exercise 2

Using the following Programiz tutorial: *Python Input, Output and Import* create a program that asks you two numbers. It should show you a message saying if the first one is bigger, the second one is bigger or they are the same.

Solution:

To achieve this, I have created a function (**num_compare**) which uses *if-else* statements to determine if one of the numbers is bigger than the other or if the two numbers are the same. Then, I use **user-input of two numbers separated by space**, which are then assessed by calling the function *num_compare*.

```

In [3]: # @Pre: True
        # Post: Message dsplaying the comparision of two numbers.
def num_compare(num):
    if (num[0] > num[1]):
        print('The first number is bigger')
    elif (num[1] > num[0]):
        print("The second number is bigger")
    else:
        print("Two numbers enetered are equal")

#num1 = [3004.6, 987]
#num_compare(num1)

num = input("Enter two numbers here: ").split()
num_compare(num)

```

```

Enter two numbers here: 7 883
The second number is bigger

```

1.3 Exercise 3

Create a program that asks you for your name, and asks you for a number. If the number is 0, it should display an error message. Otherwise, it should show the repeated name as many times as the number indicates. For example, "Joan Joan Joan".

Solution:

To achieve this, I have created three functions:

1. **your_name**, asking the user to *enter a name*. This function will keep on asking if non-letters are detected.
2. **your_number**, asking the user to *enter a number*. This function will keep on asking if non-numerical values or negative numbers are detected.
3. **repeat_name**, which uses assertion error if the number is 0 or prints the name as many times as the number

```
In [4]: # @pre: True
        # Post: A name, will keep on asking to enter unless it has *only letters*
        def your_name():
            while True:
                name=input("\nEnter Name: ")
                if (name.isalpha()):
                    break
                else : #invalid
                    print ("Invalid Name! Please enter letters only.\n")
            return name

        # @pre: True
        # Post: A number, will keep on asking to enter again unles it is a non-negative numeri
        def your_number():
            while True:
                number=input("Enter a number: ")
                if (number.isnumeric() and int(number) >= 0):
                    break
                else : #invalid
                    print ("Invalid Number! Only numerics and no negatives.\n")
            return number

        name = your_name()
        number = your_number()

        # @pre: True
        # Post: Repeats the name as many times as the value of number, returns an error if the
        def repeat_name(name, number):
            try:
                assert(int(number) != 0), "Error: 2020 not found!"
                print((name + ' ')* int(number))
```

```

except AssertionError as msg:
    print(msg)

repeat_name(name, number)

```

```

Enter Name: Neelu
Enter a number: 6
Neelu Neelu Neelu Neelu Neelu Neelu

```

1.4 Exercise 4

Create a program that given any list, tells you if it is symmetrical or not. If so, tell me how many items it has.

Solution:

A list is symmetric if the first row is the same as the first column, the second row is the same as the second column and so on.

To achieve this,

1. I define a function called **symmetric**.
2. The input for this function is a list.
3. Output of this function is a boolean **True if the list is symmetric and False if it is not**.
4. The counter inside this function also displays how many elements are there in the list.
5. I have also made a **user-input for list of lists**, in which you have to enter the number of sublists n and then individually enter the elements (n times n) of the lists one by one. The list will be created with the entered values or strings by putting them in rows. For example, if the number of list is chosen as 3, then you have to enter 9 values, first three of which forms the first sublist (i.e., first row), the following three elements forms the second list (i.e., second column) and the last three becomes the third list (i.e., third row).
6. This list is then given to the function *symmetric*, which returns either True or False depending on if the list is symmetric or not and also returns a value indicating how many elements are in the list if it is symmetric. Additionally, if the list is symmetric then the function prints the answer "The list is symmetric and has n elements."

```

In [5]: # @Pre: True
        # Post: True (symmetric) or False (not symmetric)
def symmetric(checklist):
    # First, check if the list is square or not
    length = len(checklist)
    for value in range(0,length):
        if len(checklist) != len(checklist[value]):
            return False
    checkcolumn = 0 # counter set to 0
    list1 = [] # sublist for row
    list2 = [] # sublist for column
    i = 0 # counter for elements

```

```

while checkcolumn < len(checklist):
    list1+=checklist[checkcolumn]
    for row in checklist:
        i = i + 1
        list2.append(row[checkcolumn])
    if list1 != list2:
        return False
    else:
        checkcolumn+=1
print(display_colored_text(BLUE, 'The list is symmetric and has ' + str(i) + ' elements.'))
#print('The list is symmetric and has ' + str(i) + ' elements.')
return True, i

# The checklist by user-input (works for both numbers and strings)
dim = int(input("Enter the number of sublist "))
print("\n")
checklist = [(input("Enter item: ")) for _ in range(dim)]
print("List is: ")
print(checklist)

symmetric(checklist)

```

Enter the number of sublist 3

```

Enter item: 1
Enter item: 2
Enter item: 3
Enter item: 2
Enter item: 3
Enter item: 4
Enter item: 3
Enter item: 4
Enter item: 1
List is:
[['1', '2', '3'], ['2', '3', '4'], ['3', '4', '1']]
The list is symmetric and has 9 elements.

```

Out[5]: (True, 9)

```

In [6]: checklist = [[1, 5, 2, 5],
                    [5, 2, 4, 2],
                    [2, 7, 1, 8]]
        symmetric(checklist)

```

Out[6]: False

1.5 Exercise 5

Create a program that given a list, tells you how many numbers match your position. For example [3,4,2,0,2,3,6] 2 and 6 coincide.

Solution:

Here, I need to check if the number value matches the number position. Important to remember is that in Python, the index of list starts with 0 by default meaning that a list with n elements has index from 0 to n .

To achieve this check, I take following steps:

1. A function **enterlist** is defined to initiate user-input prompt for entering a list. First, the size/length of the list has to be entered and then the elements separated by space. If the length of the list and the number of elements entered does not match, the user-input will prompt again asking to enter correct number of elements.
2. A second function called **match** is defined to check if the value of numbers and their positions match. To check this, a simple for loop is run with if-else conditions, and the corresponding message is displayed: False for no match and True for match. At the end of the loop, the counter calculates how many True values are for the match and displays a message accordingly.

```
In [7]: def enterlist():
        while True:
            n = int(input("Enter the size of the list "))
            list1 = list(int(num) for num in input("Enter the list numbers separated by space "))
            length = len(list1)
            if length == n:
                break
            else:
                print("Invalid! Please enter " + str(n) + " elements")
        return list1

def match(list1):
    i = 0 # counter
    length = len(list1)
    for ii in range(0,length):
        if list1[ii] != ii:
            print ('Position ' + str(ii) + ': ' + str(list1[ii]) + ' --> False')
        else:
            i = i + 1
            print ('Position ' + str(ii) + ': ' + str(list1[ii]) + ' --> True')
    #print("\n")
    #print ('Total ' + str(i) + ' numbers match from the list.')
    print(display_colored_text(BLUE, 'Total ' + str(i) + ' numbers match from the list'))

list1 = enterlist()
print("List entered: ", list1)
print("\n")

match(list1)
```

Enter the size of the list 8
Enter the list numbers separated by space 1 44 2 66 4 5 93 8
List entered: [1, 44, 2, 66, 4, 5, 93, 8]

Position 0: 1 --> False
Position 1: 44 --> False
Position 2: 2 --> True
Position 3: 66 --> False
Position 4: 4 --> True
Position 5: 5 --> True
Position 6: 93 --> False
Position 7: 8 --> False
Total 3 numbers match from the list.