

Python_Lab_3

Task 1 :

Task 1

- Write a function named **'create_student_records'** that takes a list of tuples. Each tuple contains a student's name and their scores in three subjects. The function should return a list of dictionaries, where each dictionary represents a student with keys: name, subject1, subject2, subject3. Use list comprehension to create the list of dictionaries.

```
[26]: # write your code here ^^
def creat_student_records():
    key_list = ["Name","Math","Art","Biology"]
    student_info = [("Abdullah",99,100,98),("Ali",88,70,77),("Malik",100,99,99),("Hassan",100,66,90)]
    student_represnt_list = [dict(zip(key_list,student)) for student in student_info]
    return student_represnt_list

student_reccord = creat_student_records()
print(student_reccord)

[{'Name': 'Abdullah', 'Math': 99, 'Art': 100, 'Biology': 98}, {'Name': 'Ali', 'Math': 88, 'Art': 70, 'Biology': 77}, {'Name': 'Malik', 'Math': 100, 'Art': 99, 'Biology': 99}, {'Name': 'Hassan', 'Math': 100, 'Art': 66, 'Biology': 90}]
```

Task 2 :

Task 2

- Write a function named **'filter_and_sort_students'** that takes the list of dictionaries from Question 1 and returns a sorted list of student names who have an average score of 80 or more across all subjects. Sort the names in alphabetical order.

```
[39]: # write your code here ^^
def filter_and_sort_students(student_reccord):
    filter_student = [student["Name"] for student in student_reccord if (student['Math'] + student['Art'] + student['Biology']) / 3 >= 80]
    filter_student.sort()
    return filter_student

sorted_student = filter_and_sort_students(student_reccord)

print(sorted_student)

['Abdullah', 'Hassan', 'Malik']
```

Task 3 :

Task 3

- Write a function named `get_product_info` that prompts the user to enter information about a product. The function should ask for the product name, product price, and product quantity. If the user enters an invalid input (e.g., a non-numeric value for the price or quantity), the function should handle the error and prompt the user to enter the value again. Return a dictionary containing the product information.

```
[133]: # write your code here ^_^
def get_product_info():
    print ("Enter the product Information")
    while True :
        try :
            pro_name = input ("Enter product name : ")
            if not pro_name :
                raise ValueError("Please Enter the name ")
            try:
                pro_price = float(input("Enter product price : "))
                if pro_price < 0:
                    raise ValueError("Please enter crrect price")
            except ValueError:
                print(" Please enter correct price")
                continue
            try:
                pro_quantity = int(input("Enter product quantity : "))
                if pro_quantity < 0:
                    raise ValueError("Please enter correct quantity")
            except ValueError:
                print(" Please enter correct quantity")
                continue
            product = {" Name " : pro_name, "Price" : pro_price , "Quantity" :pro_quantity }
            return product

        except ValueError as e:
            print("Please try again.")

display_product = get_product_info()
print(displav product)
```

```
Enter the prудuct Information
Enter product name :
Please try again.
Enter product name : ice
Enter product price : k9
Please enter crrect price
Enter product name : ice
Enter product price : 2.5
Enter product quantity : 90
{' Name ' : 'ice', 'Price': 2.5, 'Quantity': 90}
```

Task 4 :

Task 4

- Write a function named **'get_even_numbers'** that takes a list of integers and returns a list of even numbers using list comprehension.

```
[131]: # write your code here ^_^
import random

random_list = [random.randint(0, 50) for _ in range(15)]
print("random list")
print(random_list)

def get_even_numbers(rand_list):
    even_list = [num for num in rand_list if num%2==0]
    return even_list

even_list_print = get_even_numbers(random_list)
print("Even list")
print(even_list_print)

random list
[16, 12, 23, 33, 30, 24, 12, 39, 5, 46, 2, 23, 13, 46, 30]
Even list
[16, 12, 30, 24, 12, 46, 2, 46, 30]
```

Task 5 :

Task 5

- Write a function named **'pair_elements'** that takes two lists of equal length as parameters and returns a list of tuples, where each tuple contains corresponding elements from the two lists. Use the zip function to achieve this.

```
[125]: # write your code here ^_^
list1_rand = [random.randint(0, 50) for _ in range(15)]
list2_rand = [random.randint(0, 50) for _ in range(15)]
print("list 1")
print(list1_rand)
print("list 2")
print(list2_rand)

def pair_elements(list1, list2):
    combined_list = [(num1,num2 )for num1,num2 in (zip(list1, list2))]
    return combined_list

combined_list = pair_elements(list1_rand, list2_rand)
print("combined list")
print(combined_list)

list 1
[43, 26, 42, 37, 12, 25, 22, 4, 14, 49, 16, 49, 40, 14, 12]
list 2
[29, 47, 17, 40, 17, 1, 22, 0, 48, 4, 30, 48, 17, 23, 11]
combined list
[(43, 29), (26, 47), (42, 17), (37, 40), (12, 17), (25, 1), (22, 22), (4, 0), (14, 48), (49, 4), (16, 30), (49, 48), (40, 17), (14, 23), (12, 11)]
```

Activate Windows
Go to Settings to activate W

Task 6 :

Task 6

- Write a function named **'square_dict_lambda'** that takes a list of integers and returns a dictionary where the keys are the integers and the values are their squares. Use a lambda function within a dictionary comprehension to achieve this.

```
[123]: # write your code here ^_^
list1_random = [random.randint(0, 50) for _ in range(15)]
print("Random list ")
print(list1_random)

def square_dict_lambda(rand_list):
    square_dict = {num : ( lambda x : x**2) (num) for num in rand_list}
    return square_dict

square_dict = square_dict_lambda(list1_random)
print("squared dictionary")
print(square_dict)

Random list
[24, 6, 29, 43, 7, 32, 23, 31, 30, 10, 7, 26, 46, 50, 37]
squared dictionary
{24: 576, 6: 36, 29: 841, 43: 1849, 7: 49, 32: 1024, 23: 529, 31: 961, 30: 900, 10: 100, 26: 676, 46: 2116, 50: 2500, 37: 1369}
```