JLUFE **Fall 2021(Sep-Jan)**

Homework Assignment Report

JILIN UNIVERSITY OF FINANCE AND ECONOMICS

College of Managment Science and Information Engineering

BSc in Data Science and Big Data Technology

(2021)

MODULE: Intelligent Technology

Homework Assignment: 05

Functions

04/11/2021

Submitted by:

Milan(米兰) 0318021907632 (2005)
QQ: 3086215265 | Github ID: milaan9

Instructions:

1. I have added tips and required learning resources for each question, which helps you to solve the problems.
2. Finish the assignment on your **OWN**. **Any student find copying/sharing from classmates or internet will get '0' points!!!**
3. After `Accepting this assignment` from ➞ GitHub Clasroom link (https://classroom.github.com/a/I_ScEHXx), Github will create private repository of the assignment in your GitHub Classroom account.
4. In your repository `Clone` ➞ `Download ZIP` in your computer.
5. Change your ➞ **College**, **Major**, **Name**, **Student number**, **Class number**, **QQ number** and **GitHub ID**
6. Once you finish the Assignment convert your .ipynb file into PDF (https://github.com/milaan9/91_Python_Mini_Projects/tree/main/001_Convert_IPython_to_PDF) (both **.ipynb** and **.pdf** file will be required!)
7. To submit your assignment, go to GitHub Classroom repository and `Add file` ➞ `Upload files` ➞ `Commit changes`
   A. Replace the question (**.ipynb**) file with your solution (**.ipynb**) file.
   B. Also, upload (**.pdf**) converted file of your solution (**.ipynb**) file.

Python Assignment 05

Functions → Level 1, 2 & 3

> **Note:** Please create new cell for each question

Part A → Level 1

> **Note:** Please create new cell for each question

1. Area of a circle is calculated as follows: **area =** πr^2 and **perimeter =** $2\pi r$. Write a function that calculates `area_of_circle` and `perimeter_of_circle` by taking user input for value of `r`.
2. Write a function called `add_all_nums` which takes arbitrary number of arguments and sums all the arguments. Check if all the list items are number data types. If not do give a reasonable feedback.
3. Temperature in **°C** can be converted to **°F** using this formula: **°F =** $(°C * 9/5) + 32$. Write a function which converts **°C to °F**, `convert_celsius_2_fahrenheit`.
4. Write a function called `check_season`, it takes a month parameter and returns the season: `Autumn`, `Winter`, `Spring` or `Summer`.
5. Write a function called `calculate_slope` which return the slope of a linear equation
6. Quadratic equation is calculated as follows: $ax^2 + bx + c = 0$. Write a function which calculates solution set of a quadratic equation, `solve_quadratic_eqn`.
7. Declare a function named `print_list`. It takes a list as a parameter and it prints out each element of the list.
8. Declare a function named `reverse_list`. It takes an array as a parameter and it returns the reverse of the array (use loops).

   - ```
     print(reverse_list([1, 2, 3, 4, 5]))
     #[5, 4, 3, 2, 1]
     print(reverse_list1(["A", "B", "C"]))
     #["C", "B", "A"]
     ```

9. Declare a function named `capitalize_list_items`. It takes a list as a parameter and it returns a capitalized list of items
10. Declare a function named `add_item`. It takes a list and an item parameters. It returns a list with the item added at the end.

    - ```
      food_staff = ['Potato', 'Tomato', 'Mango', 'Milk']
      print(add_item(food_staff, 'Fungi'))  #['Potato', 'Tomato', 'Mango', 'Milk', 'Fung
      i']
      numbers = [2, 3, 7, 9]
      print(add_item(numbers, 5))       #[2, 3, 7, 9, 5]
      ```

11. Declare a function named `remove_item`. It takes a list and an item parameters. It returns a list with the item removed from it.

    - ```
      food_staff = ['Potato', 'Tomato', 'Mango', 'Milk']
      print(remove_item(food_staff, 'Mango'))  # ['Potato', 'Tomato', 'Milk']
      numbers = [2, 3, 7, 9]
      print(remove_item(numbers, 3))  # [2, 7, 9]
      ```
      ```

12. Declare a function named `sum_of_numbers` . It takes a number parameter and it adds all the numbers in that range.

    - ```
      print(sum_of_numbers(5))  # 15
      print(sum_all_numbers(10)) # 55
      print(sum_all_numbers(100)) # 5050
      ```

13. Declare a function named `sum_of_odds` . It takes a number parameter and it adds all the odd numbers in that range.
14. Declare a function named `sum_of_even` . It takes a number parameter and it adds all the even numbers in that range.

In [ ]:

```
# Solution:

def area_of_circle(r):
    pi = 3.14
    area = pi*r**2
    print(area)
    return area
def perimeter_of_circle(r):
    pi = 3.14
    perimeter = 2*pi*r
    print(perimeter)
    return perimeter
```

In [ ]:

```
def add_all_nums(num):
    sum=0
    from a in list:
        sum=sum+n
        if type(a)==int:
            print("true")
    print(sum)
```

In [1]:

```

```

```
  File "<ipython-input-1-7e77bbe42415>", line 2
    F=(C*9/5)+32
      ^
SyntaxError: invalid character in identifier
```
```

In [ ]:

```
def check_season(month):
    if month == "February" or month == "March" or month == "April":
        print("Spring")
    if month == "May" or month == "June" or month == "July":
        print("Summer")
    if month == "August" or month == "September" or month == "October":
        print("Autumn")
    if month == "November" or month == "December" or month == "January":
        print("Winter")
    a=check_season("November")
```

In [ ]:

```
def solve_quadratic_eqn(a,b,c):
    m= b**2 - 4*a*c
    if m<0:
        print("此方程无解！")
    elif m==0:
        x=b/(-2*a)
        print("x1=x2=",x)
    else:
        x1=(b+m**0.5)/(-2*a)
        x2 =(b-m**0.5)/(-2*a)
        print("x1=",x1)
        print("x2=",x2)
```

In [ ]:

```
def print_list(list):
    for a in list:
        print(a)
```

In [ ]:

```
def reverse_list(list):
reverselist = []
for a in range(len(list)):
    reverselist.append(list[len(list)-1-a])
return reverselist
print(reverse_list([1, 2, 3, 4, 5]))
#[5, 4, 3, 2, 1]
print(reverse_list(["A", "B", "C"]))
#["C", "B", "A"]
```

In [ ]:

```
def capitalize_list_items(item):
    item = upper(item)
```

Part B → Level 2

Note: Please create new cell for each question

1. Declare a function named `evens_and_odds` . It takes a positive integer as parameter and it counts number of evens and odds in the number.

    - ```
      print(evens_and_odds(100))
      #The number of odds are 50.
      #The number of evens are 51.
      ```

2. Call your function `factorial` , it takes a whole number as a parameter and it return a factorial of the number
3. Call your function `is_empty` , it takes a parameter and it checks if it is empty or not
4. Write different functions which take lists. They should `calculate_mean` , `calculate_median` , `calculate_mode` , `calculate_range` , `calculate_variance` , `calculate_std` (standard deviation).

In [ ]:

```
```

Part C → Level 3

> **Note:** Please create new cell for each question

1. Write a function called `is_prime` , which checks if a number is prime and prints all prime numbers in that range.
2. Write a functions which checks if all items are *unique* in the list.
3. Write a function which checks if all the items of the list are of the *same data type*.
4. Write a function which check if provided variable is a valid *python variable*
5. Go to the data folder and access the **countries-data.py (https://github.com/milaan9/03_Python_Flow_Control/blob/main/countries_details_data.py)** file.

- Create a function called the `most_spoken_languages` in the world. It should return 10 or 20 most spoken languages in the world in descending order
- Create a function called the `most_populated_countries` . It should return 10 or 20 most populated countries in descending order.

In [ ]:

```
# Solution:
```
```