# Python Basics:

# Data types

# Text Type: str

# Numeric Types: int, float

# Sequence Types: list, tuple, range

# Mapping Type: dict

# Boolean Type: bool

# Example Variables and data types

age = 25

name = "John"

is\_student = False

x = {"name" : "John", "age" : 36}

y = ["apple", "banana", "cherry"]

f = frozenset({"apple", "banana", "cherry"})

r = range(6)

print(type(f))

# Casting:

a = int("3")

b = int(1.0)

c = str(3.0) # z will be '3.0'

d = str(3) # z will be '3'

e = float(3) # z will be 3.0

print(c)

# Basic operations

result = 5 + 3

greeting = "Hello, " + "World"

# Control Flow:

age = 10

# Conditional statements

if age >= 18:

print("You are an adult.")

elif age >= 13:

print("You are a teenager.")

else:

print("You are a child.")

# Loops

for i in range(5):

print(i)

0

1

2

3

4

# Data Structures:

# Lists

my\_list = [1, 2, 3, 4]

print(my\_list[0]) # Accessing elements

my\_list.append(5) # Appending elements

print(my\_list) # Accessing elements

1

[1, 2, 3, 4, 5]

# Dictionaries

my\_dict = {"name": "John", "age": 25}

print(my\_dict["name"]) # Accessing values

my\_dict["city"] = "New York" # Adding key-value pairs

# Sets and Tuples

my\_set = {1, 2, 3} # Sets have unique elements

my\_tuple = (1, 2, 3) # Tuples are immutable

# Functions:

# Function definition

def greet(name):

return "Hello, " + name

# Function call

message = greet("Alice")

print(message)

# String Manipulation:

# String methods

text = " Hello, World! "

trimmed\_text = text.strip()

words = text.split(',')

formatted\_text = "Name: {}, Age: {}".format("John", 25)

# Concatenation and formatting

full\_name = "John" + " " + "Smith"

formatted\_greeting = f"Hello, {name}"

# Printing:

a = "Hello"

print(a)

b = "Hello, World!"

print(b[2:5])

b = "Hello, World!"

print(b[:5])

b = "Hello, World!"

print(b[2:])

# Modify Strings

a = "Hello, World!"

print(a.upper())

a = "Hello, World!"

print(a.lower())

# Resources (e.g., Suggested Websites):

# LeetCode

# HackerRank

# GeeksforGeeks

# Python's official documentation

Before working on the solution:

* These things should be straightforward and you should all know these topics
* So No Use of Internet to find any help. 🙂
* Ask, if you have any questions.

1. Write a python function to accept a string and count the number of vowels and consonants.

Example:

vowel\_counter("hello world")

The number of vowels is 3

The number of consonant is 7

Solution:

def vowel\_counter(inp):

#initializing counters for vowels and consonants

v=0

c=0

#running a for loop to check all the characters of the input string

for i in inp:

#checking if the character is a vowel, then increasing v by 1

if i in ["a","e","i","o","u"]:

v=v+1

#checking if the character is a space, then pass and do nothing

elif i == " ":

pass

#checking if the character is a consonant, then increasing c by 1

else :

c=c+1

#print the values of v and c

print( f"The number of vowels is {v}")

print( f"The number of consonant is {c}")

2. Write a Python function char\_frequency(s) that takes a string s as input and returns a dictionary where the keys are characters in the string, and the values are the frequencies of those characters in the string.The function should not be case-sensitive, meaning 'A' and 'a' should be considered the same character.

# Example usage

input\_string = "Hello, World!"

result = char\_frequency(input\_string)

print(result)

{'h': 1, 'e': 1, 'l': 3, 'o': 2, 'w': 1, 'r': 1, 'd': 1}

Solution:

def char\_frequency(s):

# Convert the input string to lowercase to make it case-insensitive

s = s.lower()

# Initialize an empty dictionary to store character frequencies

frequency\_dict = {}

# Iterate through each character in the string

for char in s:

# Check if the character is alphanumeric

if char.isalnum():

# If the character is already in the dictionary, increment its frequency

if char in frequency\_dict:

frequency\_dict[char] += 1

# If the character is not in the dictionary, add it with a frequency of 1

else:

frequency\_dict[char] = 1

return frequency\_dict

3. Given the data provided:

data = '''{"class":[

{

"name": "John Smith",

"age": 30,

"city": "New York",

"is\_student": false,

"grades": [95, 87, 92, 88, 76]

},

{

"name": "Emily Smith",

"age": 20,

"city": "New Jersey",

"is\_student": true,

"grades": [55, 67, 82, 88, 96]

},

{

"name": "Jacob Dum",

"age": 10,

"city": "New Jersey",

"is\_student": true,

"grades": [25, 57, 42, 88, 76],

"hobbies": [

{

"instruments": {

"guitar": "acoustic",

"drums": "electric"

}

},

"sports",

"games"

]

}

]

}'''

Do the following:

* Print out a list of actual student names
* Print out the list of grades from each individual in this list
* Print out the average grade for each student
* For Jacob Dum, print all the instruments he plays

Tips:

* JSON
* Create functions for each task.

Solution:

import json

json\_data = json.loads(data)

#1a

def student\_grades():

for students in range(len(json\_data['class'])):

grades = json\_data['class'][students]['grades']

print(grades)

#1b

def student\_names():

for students in range(len(json\_data['class'])):

names = json\_data['class'][students]['name']

print(names)

#1c

def avg\_grades():

for students in range(len(json\_data['class'])):

names = json\_data['class'][students]['name']

grades = json\_data['class'][students]['grades']

avg\_grades = sum(grades) / len(grades)

print(f'{names}:{avg\_grades}')

#1d

def student\_hobbies():

for students in json\_data['class']:

if students['name'] == "Jacob Dum":

for hobbies in students['hobbies'][0]['instruments']:

print(hobbies)

student\_grades()

student\_names()

avg\_grades()

student\_hobbies()

----------------------------------------------------------------------------------------------------------------------------

**Day 2**

Exercise: You are given a list of numbers. Your task is to write a Python program to calculate the sum of all the even numbers in the list.

1. Solve this problem using a traditional for loop iterating through the list
2. Solve this problem using the range and len functions.
3. Modify one of your functions from the previous parts to return a single integer value.
   1. Create an additional function called power\_of\_return() and create a variable that adds 50 to the result of your original function.
   2. What is the difference between return and print?

#Example

numbers = [2, 7, 14, 6, 9, 10, 28, 36, 45, 8]

#Output should be 104 for the above list

Solution:

Part 1.

def count\_even(numbers:list):

# Initialize a variable to hold the sum of even numbers

even\_sum = 0

# Use a for loop and the range function to iterate through the list

for value in numbers:

# Check if the current number is even and add it to the sum if it is

if value % 2 == 0:

even\_sum += value

# Print the sum of even numbers

print(f"Sum of even numbers:{even\_sum}")

# Define the list of numbers

numbers = [2, 7, 14, 6, 9, 10, 28, 36, 45, 8]

#Test your function

count\_even(numbers)

Part 2

def count\_even(numbers:list):

# Initialize a variable to hold the sum of even numbers

even\_sum = 0

# Use a for loop and the range function to iterate through the list

for value in range(len(numbers)):

# Check if the current number is even and add it to the sum if it is

if numbers[value] % 2 == 0:

even\_sum += numbers[value]

# Print the sum of even numbers

print(f"Sum of even numbers:{even\_sum}")

# Define the list of numbers

numbers = [2, 7, 14, 6, 9, 10, 28, 36, 45, 8]

#Test your function

count\_even(numbers)

Part 3

def count\_even(numbers:list):

# Initialize a variable to hold the sum of even numbers

even\_sum = 0

# Use a for loop and the range function to iterate through the list

for value in range(len(numbers)):

# Check if the current number is even and add it to the sum if it is

if numbers[value] % 2 == 0:

even\_sum += numbers[value]

# Print the sum of even numbers

return even\_sum

def power\_of\_return(numbers):

num = count\_even(numbers)+50

print(num)

# Define the list of numbers

numbers = [2, 7, 14, 6, 9, 10, 28, 36, 45, 8]

#Test your function

power\_of\_return(numbers)

Exercise: Write a Python program for converting temperatures between fahrenheit and celsius.. Create multiple functions:

1. Create a function called celsius\_to\_fahrenheit that converts celsius to fahrenheit
2. Create a function called fahrenheit\_to\_celsius that converts fahrenheit to celsius
3. Create a function called main that uses the user input and provides the correct temperature conversion function.
   1. For example:
   2. print("1. Celsius to Fahrenheit")
   3. print("2. Fahrenheit to Celsius")
   4. choice = input("Enter your choice (1/2): ")

Solution:

# Define a function to convert Celsius to Fahrenheit

def celsius\_to\_fahrenheit(celsius):

fahrenheit = (celsius \* 9/5) + 32

return fahrenheit

# Define a function to convert Fahrenheit to Celsius

def fahrenheit\_to\_celsius(fahrenheit):

celsius = (fahrenheit - 32) \* 5/9

return celsius

# Define the main function

def main():

print("Temperature Converter")

print("1. Celsius to Fahrenheit")

print("2. Fahrenheit to Celsius")

choice = input("Enter your choice (1/2): ")

if choice == '1':

celsius = float(input("Enter temperature in Celsius: "))

result = celsius\_to\_fahrenheit(celsius)

print(f"{celsius}°C is equal to {result}°F")

elif choice == '2':

fahrenheit = float(input("Enter temperature in Fahrenheit: "))

result = fahrenheit\_to\_celsius(fahrenheit)

print(f"{fahrenheit}°F is equal to {result}°C")

else:

print("Invalid choice. Please enter 1 or 2.")

main()