Python lab

ASSSIGNMENT : Q1 to Q15

Prn : 1272240312

Name : Saurabh Nawale

Q1.

```python
S = str(input("Enter the string"))

S = S.replace("A", "")
S = S.replace("E", "")
S = S.replace("I", "")
S = S.replace("O", "")
S = S.replace("U", "")
S = S.replace("a", "")
S = S.replace("e", "")
S = S.replace("i", "")
S = S.replace("o", "")
S = S.replace("u", "")

print(S)

# Output
# Enter the string Abhijeet
# bhjt
```

Q6

```python
class Library:
    def __init__(self, acc_number, title, author, publisher):
        self.acc_number = acc_number
        self.title = title
        self.author = author
        self.publisher = publisher

    def read(self):
        print(f"Accession Number: {self.acc_number}")
        print(f"Title: {self.title}")
        print(f"Author: {self.author}")
        print(f"Publisher: {self.publisher}")
    def compute(self, days_late):
        fine = lambda days: days * 5
        total_fine = fine(days_late)
        print(f"Number of Days Late: {days_late}")
        print(f"Fine Charged: Rupees {total_fine}")

    def display(self):
        print("Book Details:")
        print(f"Accession Number: {self.acc_number}")
        print(f"Title: {self.title}")
        print(f"Author: {self.author}")
        print(f"Publisher: {self.publisher}")



if __name__ == "__main__":

    book = Library("A123", "Python Programming", "Abhijeet", "TechBooks Inc.")
    print("Reading Book Details:")
    book.read()

    print("\nCalculating Fine for 7 Days Late:")
    book.compute(7)
    print("\nDisplaying Book Details Again:")
    book.display()
# output
# Reading Book Details:
# Accession Number: A123
# Title: Python Programming
# Author: Abhijeet
# Publisher: TechBooks Inc.

# Calculating Fine for 7 Days Late:
# Number of Days Late: 7
# Fine Charged: Rupees 35

# Displaying Book Details Again:
# Book Details:
# Accession Number: A123
# Title: Python Programming
# Author: Abhijeet
# Publisher: TechBooks Inc.
# PS D:\MCA1-2\python>
```

Q7-A

```
A = str(input("Enter the IP Address: "))
segments = A.split('.')

normalized_segments = [str(int(segment)) for segment in segments]
A = '.'.join(normalized_segments)
print(f"IP without leading zeros: {A}")

# output:
# PS D:\MCA1-2\python> python Q7-A.py
# Enter the IP Address: 011.110.001.110
# IP without leading zeros: 11.110.1.110
# PS D:\MCA1-2\python>
```

Q7-B

```
A = str(input("Enter the String "))

words = A.split()

five_char_words = [word for word in words if len(word) == 5]

print("5-character-long words:", five_char_words)

# PS D:\MCA1-2\python> python Q7-B.py
# Enter the String i am omkar
# 5-character-long words: ['omkar']
```

Q8

```
import pandas as pd

file_path = input("Enter the path to the CSV file: ")
df = pd.read_csv(file_path)

df_filled = df.fillna(df.mean(numeric_only=True)) #numeric_only=True

print("Last 5 rows of the DataFrame:")
print(df_filled.tail())


# output
# Enter the path to the CSV file: c:\Users\HP\Downloads\customers-100.csv
# 96     97   CeD220bdAaCfaDf   ...        2021-07-10         https://novak-allison.com/
# 96     97  CeD220bdAaCfaDf  ...        2021-07-10         https://novak-
allison.com/
# 97     98  28CDbC0dFe4b1Db  ...        2021-09-
18              https://www.ross.com/
# 98     99  c23d1D9EE8DEB0A  ...        2021-08-
11               http://watkins.info/
# 99    100  2354a0E336A91A1  ...        2020-03-11  http://www.hatfield-
saunders.net/
```

Q9

```
import pandas as pd

file_path = 'C:\Users\HP\Downloads\Sample-Spreadsheet-10-rows.csv'

df = pd.read_csv(file_path)
columns = df.columns.tolist()
print("\nList of columns:", columns)

last3 = df.iloc[-3:, :3]
print("\nLast three rows and first three columns:")
print(last3)

# output
#  python Q9.py
# List of columns: ['1', 'Abhijeet', 'Unnamed: 2', '3', '-213.25', '38.94',
'35', 'Nunavut', 'Storage & Organization', '0.8']

# Last three rows and first three columns:
#     1 Abhijeet  Unnamed: 2
# 6   8  dheeraj         NaN
# 7   9    nitin         NaN
# 8  10   vedant         NaN
```

Q10

```
import pandas as pd
import numpy as np

data = {
    "City": ["Delhi", "Bengaluru", "Chennai", "Mumbai", "Kolkata"],
    "MaxTemp": [40, 31, 35, 29, 39],
    "MinTemp": [32, 25, 27, 21, 23],
    "RainFall": [24.1, 36.2, 40.8, 35.2, 41.8]
}
```

```
df = pd.DataFrame(data)


sum_columns = df.select_dtypes(include=np.number).sum()
mean_rainfall = df["RainFall"].mean()
median_maxtemp = df["MaxTemp"].median()
column_names = df.columns.tolist()

print("Sum of numeric columns:")
print(sum_columns)
print("\nMean of RainFall column:", mean_rainfall)
print("\nMedian of MaxTemp column:", median_maxtemp)
print("\nColumn Names:", column_names)

# Output
# PS D:\MCA1-2\python> python Q10.py
# Sum of numeric columns:
# MaxTemp     174.0
# MinTemp     128.0
# RainFall    178.1
# dtype: float64

# Mean of RainFall column: 35.620000000000005

# Median of MaxTemp column: 35.0

# Column Names: ['City', 'MaxTemp', 'MinTemp', 'RainFall']
```

Q11

```
import numpy as np
array = np.array([[1, 2, 3],
                  [4, 5, 6],
                  [7, 8, 9],
                  [10, 11, 12]])

columnM = np.mean(array, axis=0)
print("Array:")
print(array)
print("\nColumn-wise Mean:")
print(columnM)

# output
# PS D:\MCA1-2\python> python Q11.py
# Array:
# [[ 1  2  3]
#  [ 4  5  6]
#  [ 7  8  9]
#  [10 11 12]]

# Column-wise Mean:
# [5.5 6.5 7.5]
```

Q12

```
import numpy as np
import pandas as pd

array = np.array([1, 2, 2, 3, 3, 3, 4, 4, 4, 4])
series = pd.Series(array)
frequency_count = series.value_counts()
print("Frequency count of unique items:")
print(frequency_count)

# output
# PS D:\MCA1-2\python> python Q12.py
# Frequency count of unique items:
# 4    4
# 3    3
# 2    2
# 1    1
# Name: count, dtype: int64
```

Q13

```
import pandas as pd

path = input("Enter the Csv file path ")
df = pd.read_csv(path)

filled = df.fillna(df.mean(numeric_only=True))
print("frist 10 Coloums ")
print(filled.head(10))
print("All Coloums")
print(filled.all)



# output
# Enter the Csv file path c:\Users\HP\Downloads\customers-100.csv
#    Index      Customer Id  ... Subscription
Date                          Website
```

```
# 0      1  DD37Cf93aecA6Dc  ...        2020-08-
24   http://www.stephenson.com/
# 1      2  1Ef7b82A4CAAD10  ...        2021-04-
23        http://www.hobbs.com/
# 2      3  6F94879bDAfE5a6  ...        2020-03-
25     http://www.lawrence.com/
# 3      4  5Cef8BFA16c5e3c  ...        2020-06-02   http://www.good-
lyons.com/
# 4      5  053d585Ab6b3159  ...        2021-04-17  https://goodwin-
ingram.com/
# 5      6  2d08FB17EE273F4  ...        2020-02-
25       http://www.berger.net/
# 6      7  EA4d384DfDbBf77  ...        2021-08-
24          https://www.le.com/
# 7      8  0e04AFde9f225dE  ...        2021-04-12  https://hammond-
ramsey.com/
# 8      9  C2dE4dEEc489ae0  ...        2020-01-
13     https://www.bullock.net/
# 9     10  8C2811a503C7c5a  ...        2021-11-
08           https://arias.com/
# [10 rows x 12 columns]
# All Coloums
# <bound method DataFrame.all of     Index      Customer Id  ... Subscription
Date                             Website
# 0       1  DD37Cf93aecA6Dc  ...        2020-08-
24          http://www.stephenson.com/
# 1       2  1Ef7b82A4CAAD10  ...        2021-04-
23               http://www.hobbs.com/
# 2       3  6F94879bDAfE5a6  ...        2020-03-
25            http://www.lawrence.com/
# 3       4  5Cef8BFA16c5e3c  ...        2020-06-02         http://www.good-
lyons.com/
# 4       5  053d585Ab6b3159  ...        2021-04-17        https://goodwin-
ingram.com/
#
..     ...              ...  ...               ...
   ...
# 95     96  cb8E23e48d22Eae  ...        2022-01-30           http://hayes-
perez.com/
# 96     97  CeD220bdAaCfaDf  ...        2021-07-10         https://novak-
allison.com/
# 97     98  28CDbC0dFe4b1Db  ...        2021-09-
18               https://www.ross.com/
# 98     99  c23d1D9EE8DEB0A  ...        2021-08-
11                http://watkins.info/
# 99    100  2354a0E336A91A1  ...        2020-03-11  http://www.hatfield-
saunders.net/
```

```
# [100 rows x 12 columns]>
```

Q14

```
import numpy as np
import pandas as pd

array = np.array([[1,2,3,4,5],
                 [1,2,3,4,5],
                 [1,2,3,4,5],
                 [1,2,3,4,5],
                 [1,2,3,4,5]])
print(array)
print("2*2 array")
bf = array[-2:, -2:]
print(bf)

# output
# PS D:\MCA1-2\python> python Q14.py
# [[1 2 3 4 5]
#  [1 2 3 4 5]
#  [1 2 3 4 5]
#  [1 2 3 4 5]
#  [1 2 3 4 5]]
# 2*2 array
# [[4 5]
#  [4 5]]
```

Q15

```
year = int(input("Enter the year"))
if year < 0:
    print("invalid input check it ")
else:
    if year % 4 == 0:
          print("Year is leap year ")
    else:
         print("its not leap year")

# Output
# PS D:\MCA1-2\python> python Q15.py
# Enter the year -2
# invalid input check it
# PS D:\MCA1-2\python> python Q15.py
# Enter the year 2024
# Year is leap year
# PS D:\MCA1-2\python>
```

Q16

```
def sumn(n):
     if n == 1:
        return 1
     return n + sumn(n - 1)

n =  int(input("Enter the value of n  "))
sumn(n)
print( "The sum is ", sumn(n))
print(f"The sum of the first {n} natural numbers is: {sumn(n)}")

# output
# PS D:\MCA1-2\python> python Q16.py
# Enter the value of n  10
# The sum is  55
```

Q17

```
dict1 = {'a': 1, 'b': 2, 'c': 3, 'd': 4}
dict2 = {'c': 3, 'd': 4, 'e': 5, 'f': 6}

B = dict1.keys() - dict2.keys()
print("Keys in dict1 but not in dict2:" , B )

# output
# python Q17.py
# Keys in dict1 but not in dict2: {'a', 'b'}
```

Q18

```
# Define a tuple
my_tuple = (10, 20, 30, 40, 50, 60, 70, 80, 90, 100)

A = my_tuple[3] #fourth_from_front

B = my_tuple[-4] #fourth_from_last


print("4th element from the front:", A)
print("4th element from the last:", B)

# output
# python Q18.py
# 4th element from the front: 40
# 4th element from the last: 70
# PS D:\MCA1-2\python>
```

Q19

```
def fibonacci(n):
    a, b = 0, 1
    while a <= n:
        print(a, end=" ")
        a, b = b, a + b
    print()

num = int(input("Enter a number: "))
fibonacci(num)

# PS D:\MCA1-2\python> python Q19.py
# Enter a number: 10
# 0 1 1 2 3 5 8
```