

Python assignment-2(practical 1)

1. Display the difference in dates:

#input:

#static

```
from datetime import date

d1 = date(2025,9,1)
d2 = date(2025,9,13)

difference = d2-d1

print("date 1:", d1)
print("date 2:", d2)
print("difference:",difference.days,"days")
```

#output

```
""" date 1: 2025-09-01
    date 2: 2025-09-13
    difference: 12 days """
```

#user input

```
from datetime import datetime

d1 = input("Enter first date (dd-mm-yyyy):")
d2 = input("Enter second date (dd-mm-yyyy):")

date1 = datetime.strptime(d1, "%d-%m-%Y")
date2 = datetime.strptime(d2, "%d-%m-%Y")

diff = abs((date2-date1).days)
print("Difference in days:", diff)
```

#output

```
""" Enter first date (dd-mm-yyyy):16-03-2005
    Enter second date (dd-mm-yyyy):15-09-2025
    Difference in days: 7488 """
```

2. Display time since epoch in hours and minutes:

#input:

```
import time
```

```
#current time in seconds from epoch
```

```
seconds_since_epoch = time.time()
```

```
#convert to hours and minutes
```

```
hours = int(seconds_since_epoch//3600)
```

```
minutes = int((seconds_since_epoch%3600)//60)
```

```
print("second since epoch :",int(seconds_since_epoch))
```

```
print("time since epoch:",hours,"hours and" , minutes,"minutes")
```

#output

```
““““ second since epoch : 1757952436
```

```
time since epoch: 488320 hours and 7 minutes””””
```

3. Display your age in years, months and days

#input:

```
from datetime import date
```

```
byear = int(input("Enter birth year:"))
```

```
bmonth = int(input("Enter birth month:"))
```

```
bday = int(input("Enter bith day:"))
```

```
birth_date = date(byear,bmonth,bday)
```

```
today = date.today()
```

```
years= today.year - birth_date.year
```

```
months= today.month - birth_date.month
```

```
days= today.day - birth_date.day
```

```
if days < 0:
```

```
    months -= 1
```

```
    days += 30
```

```
if months < 0:
```

```
    years -=1
```

```
    months += 12
```

```
print(f"your age: {years} years, {months} months, {days} days")
```

#output

```
"""" Enter birth year:2005
```

```
    Enter birth month:03
```

```
    Enter bith day:16
```

```
    your age: 20 years, 5 months, 29 days """"
```

4. Display trigonometric table of sin, cos and tan:

#input:

```
import math

print("Angle \t Sin \t\t Cos \t\t Tan")

print("-"*40)

for angle in range(0,91,15):

    rad = math.radians(angle)

    sin_val = round(math.sin(rad), 4)

    cos_val = round(math.cos(rad), 4)

    try:

        tan_val = round(math.tan(rad), 4)

    except:

        tan_val = "∞"

    print(f"{angle:>3}°\t{sin_val:<8}\t{cos_val:<8}\t{tan_val}")
```

#output

““““

Angle	Sin	Cos	Tan
0°	0.0	1.0	0.0
15°	0.2588	0.9659	0.2679
30°	0.5	0.866	0.5774
45°	0.7071	0.7071	1.0
60°	0.866	0.5	1.7321
75°	0.9659	0.2588	3.7321
90°	1.0	0.0	1.633123935319537e+16

””””

5. Generate 10 random numbers:

#input:

```
import random
```

```
print("10 Random Numbers:")
```

```
for i in range(10):
```

```
    print(random.randint(1,100000))
```

#output

““““

10 Random Numbers:

51785

80090

35272

29285

35968

51934

90674

89337

7242

42577

””””

6. Authentication: Ask username, password and compare

#input:

```
u_name="vivek"
```

```
pass_word= "123456"
```

```
username = input("Enter username: ")
```

```
password = input("Enter password: ")
```

```
if username == u_name and passwod == pass_word:
```

```
    print("Login successful")
```

```
else:
```

```
    print("Username an password is not valid")
```

#output

```
""""
```

```
Enter username: vivek
```

```
Enter password: 123456
```

```
Login successful
```

```
""""
```

7. Authentication: Ask username, password and compare with encryption

#input:

```
import base64
```

```
c_username = "vivek"
```

```
c_password = "123"
```

```
encrypted_password = base64.b64encode(c_password.encode()).decode()
```

```
username = input("Enter username: ")
```

```
password = input("Enter password:")
```

```
entered_encrypted = base64.b64encode(password.encode()).decode()
```

```
if username == c_username and entered_encrypted == encrypted_password:
```

```
    print("login successful")
```

```
else:
```

```
    print("invalid username and password")
```

#output

```
"""
```

```
Enter username: vivek
```

```
Enter password:123
```

```
login successful
```

```
"""
```

8. Authentication: Ask username, password and compare with hashing

#input:

```
import hashlib
```

```
c_username = "vivek"
```

```
c_password_hash = hashlib.sha256("1234".encode()).hexdigest()
```

```
username = input("Enter Username: ")
```

```
password = input("Enter Password: ")
```

```
password_hash = hashlib.sha256(password.encode()).hexdigest()
```

```
if username == c_username and password_hash == c_password_hash:
```

```
    print("Login success")
```

```
else:
```

```
    print("Invalid username and password")
```

#output

```
"""
```

```
Enter Username: vivek
```

```
Enter Password: 1234
```

```
Login success
```

```
"""
```


9. Convert string "Hello\$World" into Base64

#input:

```
import base64
```

```
text = "Hello$World"
```

```
print("Original String :", text)
```

```
encoded = base64.b64encode(text.encode())
```

```
print("Base64 Encoded:", encoded.decode())
```

#output

```
"""
```

```
Original String : Hello$World
```

```
Base64 Encoded: SGVsbG8kV29ybGQ=
```

```
"""
```

10. Code for String Manipulation

#1A:Create a string made of the first, middle and last character

#input:

str1 = "vivek"

print("Original String is", str1)

res=str1[0]

l=len(str1)

mi = int(l/2)

res = res+str1[mi]

res= res+str1[l-1]

print("New String:",res)

#output

"""

Original String is vivek

New String: vvk

"""

1B: Create a string made of the middle three characters

#input:

```
str1 = "JhonDipPeta"
```

```
str2 = "JaSonAy"
```

```
def get_middle_three_chars(str1):
```

```
    print("Original String is ",str1)
```

```
    mi=int(len(str1)/2)
```

```
    res=str1[mi-1:mi+2]
```

```
    print("Middle three chars are:", res)
```

```
get_middle_three_chars("JhonDipPeta")
```

```
get_middle_three_chars("JaSonAy")
```

#output

```
"""
```

```
Original String is  JhonDipPeta
```

```
Middle three chars are: Dip
```

```
Original String is  JaSonAy
```

```
Middle three chars are: Son
```

```
"""
```

2: Append new string in the middle of a given string

#input:

```
s1 = "Atul"
```

```
s2 = "Kelly"
```

```
def append_middle(s1,s2):
```

```
    print("Original String are",s1,s2)
```

```
    mi = int(len(s1)/2)
```

```
    x=s1[:mi:]
```

```
    x=x+s2
```

```
    x=x+s1[mi:]
```

```
    print("After appending new string in middle:", x)
```

```
append_middle("Ault","Kelly")
```

#output

```
"""
```

```
Original String are Ault Kelly
```

```
After appending new string in middle: AuKellylt
```

```
"""
```

#3: Create a new string made of the first, middle, and last characters of each input string

#input:

```
s1 = "America"
```

```
s2 = "Japan"
```

```
def mix_string(s1,s2):
```

```
    first_char = s1[0] + s2[0]
```

```
    middle_char = s1[int(len(s1)/2):int(len(s1)/2)+1] + s2[int(len(s2)/2):int(len(s2)/2)+1]
```

```
    last_char = s1[len(s1) - 1] + s2[len(s2) - 1]
```

```
    res = first_char + middle_char + last_char
```

```
    print("Mix String is ", res)
```

```
mix_string(s1,s2)
```

#output

```
"""
```

```
Mix String is AJrpan
```

```
"""
```

4: Arrange string characters such that lowercase letters should come first

#input:

```
str1 = "PYnAtivE"

print("Original String:", str1)

lower=[]

upper=[]

for char in str1:

    if char.islower():

        lower.append(char)

    else:

        upper.append(char)

sorted_str = ''.join(lower+upper)

print('Result:', sorted_str)
```

#output

““““

Original String: PYnAtivE

Result: ntivPYAE

””””

#5: Count all letters, digits, and special symbols from a given string

#input:

```
str1 = "P@#yn26at^&i5ve"
```

```
def find_digits_chars_symbols(str1):
```

```
    char_count = 0
```

```
    digit_count = 0
```

```
    symbol_count = 0
```

```
    for char in str1:
```

```
        if char.isalpha():
```

```
            char_count +=1
```

```
        elif char.isdigit():
```

```
            digit_count +=1
```

```
        else:
```

```
            symbol_count +=1
```

```
    print("Chars =", char_count, "Digits =", digit_count, "Symbol =", symbol_count)
```

```
print("total counts of char, digits and symbols \n")
```

```
find_digits_chars_symbols(str1)
```

#output

```
"""
```

```
total counts of char, digits and symbols
```

```
Chars = 8 Digits = 3 Symbol = 4
```

```
"""
```

#6: Create a mixed String using the following rules

#Given two strings, s1 and s2. Write a program to create a new string s3 made of the first char of s1, then the last char of s2, Next, the second char of s1 and second last char of s2, and so on. Any leftover chars go at the end of the result.

#input:

s1 = "Abc"

s2 = "Xyz"

s1_length = len(s1)

s2_length = len(s2)

length = s1_length if s1_length > s2_length else s2_length

result = ""

s2 = s2[::-1]

for i in range(length):

 if i < s1_length:

 result = result + s1[i]

 if i < s2_length:

 result = result + s2[i]

print(result)

#output

““““

AzbycX

””””

7: String characters balance Test**#input:**

```
def string_balance_test(s1,s2):  
    flag = True  
    for char in s1:  
        if char in s2:  
            continue  
        else:  
            flag = False  
    return flag  
  
s1 = "Yn"  
s2 = "PYnative"  
flag = string_balance_test(s1,s2)  
print("s1 and s2 are balanced:", flag)
```

```
s1 = "Ynf"  
s2 = "PYnative"  
flag = string_balance_test(s1,s2)  
print("s1 and s2 are balanced: ",flag)
```

#output

““““

s1 and s2 are balanced: True

s1 and s2 are balanced: False

””””

#8: Find all occurrences of a substring in a given string by ignoring the case

#input:

```
str1 = "Welcome to USA. usa awesome, isn't it?"
```

```
sub_string = "USA"
```

```
temp_str = str1.lower()
```

```
count = temp_str.count(sub_string.lower())
```

```
print("The USA count is:", count)
```

#output

```
"""
```

```
The USA count is: 2
```

```
"""
```

#9: Calculate the sum and average of the digits present in a string

#input:

```
str1 = "PYnative29@#8496"
```

```
total = 0
```

```
cnt = 0
```

```
for char in str1:
```

```
    if char.isdigit():
```

```
        total+=int(char)
```

```
        cnt +=1
```

```
avg = total/cnt
```

```
print("Sum is:", total,"Average is:",avg)
```

#output

```
““““
```

```
Sum is: 38 Average is: 6.333333333333333
```

```
””””
```

#10: Write a program to count occurrences of all characters within a string

#input:

```
str1 = "Apple"
```

```
char_dict = dict()
```

```
for char in str1:
```

```
    count = str1.count(char)
```

```
    char_dict[char] = count
```

```
print('Result:', char_dict)
```

#output

```
"""
```

```
Result: {'A': 1, 'p': 2, 'l': 1, 'e': 1}
```

```
"""
```

Mca 1

Name: vivek jadav

Roll no:09

#11: Reverse a given string

#input:

```
str1 = "PYnative"
```

```
print("Original string is :",str1)
```

```
str1 = str1[::-1]
```

```
print("Reversed string is:",str1)
```

#output

““““

Original string is : PYnative

Reversed string is: evitanYP

””””

#12: Find the last position of a given substring

#input:

```
str1 = "Emma is a data scientist who knows Python. Emma works at google."
```

```
print("Original String is:",str1)
```

```
index = str1.rfind("Emma")
```

```
print("Last occurrence of Emma starts at index:", index)
```

#output

```
"""
```

```
Original String is: Emma is a data scientist who knows Python. Emma works at google.
```

```
Last occurrence of Emma starts at index: 43
```

```
"""
```

#13: Split a string on hyphens

#input:

```
str1 = "Emma-is-a-data-scientist"
```

```
print("Original string is:", str1)
```

```
sub_strings = str1.split("-")
```

```
print("Displaying each substring")
```

```
for sub in sub_strings:
```

```
    print(sub)
```

#output

““““

Original string is: Emma-is-a-data-scientist

Displaying each substring

Emma

is

a

data

scientist

””””

#14: Remove empty strings from a list of strings

#input:

```
str_list=["Emma","Jon","", "Kelly",None,"Eric",""]
```

```
res_list=[]
```

```
for s in str_list:
```

```
    if s:
```

```
        res_list.append(s)
```

```
print(res_list)
```

#output

```
"""
```

```
['Emma', 'Jon', 'Kelly', 'Eric']
```

```
"""
```


#15: Remove special symbols / punctuation from a string

#input:

```
import string
```

```
str1 = "/*Jon is @developer & musician"
```

```
print("Original sting is ",str1)
```

```
new_str = str1.translate(str.maketrans("",string.punctuation))
```

```
print("New string is ", new_str)
```

#output

```
““““
```

```
Original sting is /*Jon is @developer & musician
```

```
New string is Jon is developer musician
```

```
””””
```

#16: Removal all characters from a string except integers

#input:

```
Str1 = 'I am 25 years and 10 months old'
```

```
print("Original string is", str1)
```

```
res = "".join([item for item in str1 if item.isdigit()])
```

```
print(res)
```

#output

```
"""
```

```
Original string is I am 25 years and 10 months old
```

```
2510
```

```
"""
```

#17: Find words with both alphabets and numbers**#input:****str1 = "Emma25 is Data scientist50 and AI Expert"****print("The original string is:" + str1)****res=[]****temp = str1.split()****for item in temp:****if any(char.isalpha() for char in item) and any(char.isdigit() for char in item):****res.append(item)****print("Displaying words with alphabets and numbers")****for i in res:****print(i)****#output****““““**

The original string is:Emma25 is Data scientist50 and AI Expert

Displaying words with alphabets and numbers

Emma25

scientist50

””””

#18: Replace each special symbol with # in the following string

#input:

```
import string
```

```
str1 = '/*Jon is @developer & musician!!'
```

```
print("The original string is :", str1)
```

```
replace_char = '#'
```

```
for char in string.punctuation:
```

```
    str1 = str1.replace(char, replace_char)
```

```
print("the string after replacement: ", str1)
```

#output

““““

The original string is : /*Jon is @developer & musician!!

the string after replacement: ##Jon is #developer # musician##

””””