**1. Add the current date to the text file today.txt as a string.**

<https://github.com/snehankitc/Python-basic-assignment-Program/blob/main/Assignment_21_Program_1>

**2. Read the text file today.txt into the string today\_string**

<https://github.com/snehankitc/Python-basic-assignment-Program/blob/main/Assignment_21_Program_2>

**3. Parse the date from today\_string.**

Here is a sample code in Python to parse the date from the string today\_string, which was obtained by reading the contents of the file "today.txt":

import datetime

# Open the file in read mode and read its contents into a string variable

with open("today.txt", "r") as file:

today\_string = file.read()

# Parse the date from the string

date = datetime.datetime.strptime(today\_string, "%Y-%m-%d").date()

This code uses the strptime method from the datetime module to parse the date from the string today\_string using the format string "%Y-%m-%d". The method returns a datetime object, from which the date can be obtained using the date method.

**4. List the files in your current directory**

Here is a sample code in Python to list the files in the current directory:

import os

# Get a list of all files in the current directory

files = [f for f in os.listdir() if os.path.isfile(f)]

# Print the list of files

print(files)

This code uses the os module to get a list of all the files in the current directory using the listdir method. The isfile method from the os.path module is used to filter the list and only include regular files (not directories). The list of files is then printed to the console.

**5. Create a list of all of the files in your parent directory (minimum five files should be available).**

<https://github.com/snehankitc/Python-basic-assignment-Program/blob/main/Assignment_21_Program_5>

**6. Use multiprocessing to create three separate processes. Make each one wait a random number of**

**seconds between one and five, print the current time, and then exit.**

<https://github.com/snehankitc/Python-basic-assignment-Program/blob/main/Assignment_21_Program_6>

**7. Create a date object of your day of birth.**

Here is a sample code in Python to create a date object of the user's day of birth using user input:

import datetime

# Get the user's birth year, month, and day

birth\_year = int(input("Enter your birth year: "))

birth\_month = int(input("Enter your birth month (1-12): "))

birth\_day = int(input("Enter your birth day (1-31): "))

# Create a date object of the user's day of birth

birthday = datetime.date(birth\_year, birth\_month, birth\_day)

# Print the birthday

print(f"Your birthday is: {birthday}")

This code uses the input function to get the user's birth year, month, and day. The datetime.date class is then used to create a date object with the birth year, month, and day. The date object is then printed to the console using the print function.

**8. What day of the week was your day of birth?**

import datetime

# Get the user's birth year, month, and day

birth\_year = int(input("Enter your birth year: "))

birth\_month = int(input("Enter your birth month (1-12): "))

birth\_day = int(input("Enter your birth day (1-31): "))

# Create a date object of the user's day of birth

birthday = datetime.date(birth\_year, birth\_month, birth\_day)

# Get the day of the week for the birthday

day\_of\_week = birthday.strftime("%A")

# Print the day of the week

print(f"You were born on a {day\_of\_week}.")

**9. When will you be (or when were you) 10,000 days old?**

import datetime

# Get the user's birth year, month, and day

birth\_year = int(input("Enter your birth year: "))

birth\_month = int(input("Enter your birth month (1-12): "))

birth\_day = int(input("Enter your birth day (1-31): "))

# Create a date object of the user's day of birth

birthday = datetime.date(birth\_year, birth\_month, birth\_day)

# Calculate the date when the user will be (or was) 10,000 days old

ten\_thousand\_days\_later = birthday + datetime.timedelta(days=10000)

# Print the date when the user will be (or was) 10,000 days old

print(f"You will be (or were) 10,000 days old on: {ten\_thousand\_days\_later}")