**Assignment 5**

1. Write a Python script to display the various Date Time formats –

from datetime import datetime, date, timedelta, time --🡪import this module

a) Current date and time

Ans:- print("Current date and time:", datetime.now())  
b) Current year

Ans:- print("Current year:", date.today().year)  
c) Month of year

Ans:- print("Month of year:", date.today().strftime("%B"))  
d) Week number of the year

Ans:- print("Week number of the year:", date.today().strftime("%U"))  
e) Weekday of the week

Ans:- print("Weekday of the week:", date.today().strftime("%A"))  
f) Day of year

Ans:- print("Day of year:", date.today().strftime("%j"))  
g) Day of the month

Ans:- print("Day of the month:", date.today().day)  
h) Day of week

Ans:- print("Day of week:", date.today().weekday())

1. Write a Python program to determine whether a given year is a leap year.

Ans:-

year = 2001

if (year % 4 == 0 and year % 100 != 0) or (year % 400 ==0):

print(f"{year} is a leap year.")

else:

print(f"{year} is not a leap year.")

1. Write a Python program to convert a string to datetime.  
   Sample String : Jul 1 2014 2:43PM  
   Expected Output : 2014-07-01 14:43:00

Ans:-

date\_str = "Jul 1 2014 2:43PM"

date\_time\_obj = datetime.strptime(date\_str, '%b %d %Y %I:%M%p')

print("Converted datetime:", date\_time\_obj)

1. Write a Python program to get the current time in Python.  
   Sample Format :  13:19:49.078205

Ans:- print("Time is :", datetime.now().time())

1. Write a Python program to subtract five days from the current date.  
   Sample Date :  
   Current Date : 2015-06-22  
   5 days before Current Date : 2015-06-17

Ans:-

current\_date = datetime.now().date()

subtract\_date = current\_date - timedelta(days = 5)

print("Current Date:", current\_date)

print("5 days before current date:", subtract\_date)

1. Write a Python program to convert a Unix timestamp string to a readable date.  
   Sample Unix timestamp string : 1284105682  
   Expected Output : 2010-09-10 13:31:22

Ans:-

unix\_timestamp = 1284105682

expected\_date = datetime.fromtimestamp(unix\_timestamp)

print("Expected output:", expected\_date)

1. Write a Python program to print yesterday, today, tomorrow.

Ans:-

today = datetime.now().date()

yesterday = today - timedelta(days = 1)

tomorrow = today + timedelta(days = 1)

print("Yesterday:",yesterday)

print("Today:",today)

print("Tomorrow:",tomorrow)

1. Write a Python program to convert the date to datetime (midnight of the date) in Python.  
   Sample Output : 2015-06-22 00:00:00

Ans:-

sample\_date = date(2015,6,22)

#datetime.min.time() represent (00:00:00) and combine use to create date & time

midnight\_datetime = datetime.combine(sample\_date, datetime.min.time())

print("sample output:", midnight\_datetime)

1. Write a Python program to print the next 5 days starting today.

Ans:-

today = datetime.now().date()

print("Next 5 days starting today:")

for i in range(6):

next\_day = today + timedelta(days = i)

print(next\_day)

1. Write a Python program to add 5 seconds to the current time.  
   Sample Data :  
   13:28:32.953088  
   13:28:37.953088

Ans:-

current\_time = datetime.now()

print("current time:",current\_time.time())

add\_5secs = current\_time + timedelta(seconds = 5)

print("Added after 5 secs:",add\_5secs)

1. Write a Python program to convert Year/Month/Day to Day of Year in Python.

Ans:-

yr = 2024

mnt = 9

day = 22

dt = date(yr,mnt,day)

#Get the day of the year

day\_of\_year = dt.timetuple().tm\_yday

print(f"The day of the year for {dt} is:", day\_of\_year)

1. Write a Python program to get the current time in milliseconds in Python.

Ans:-

current\_time = datetime.now().time()

milliseconds = current\_time.microsecond // 1000

print("Current time:", current\_time)

print("Milliseconds:", milliseconds)

1. Write a Python program to get the week number.  
   Sample Date : 2015, 6, 16  
   Expected Output : 25

Ans:-

sample\_date = date(2015,6,16)

#ISO calendar (year, week number, weekday)

week\_number = sample\_date.isocalendar()[1]

print("Week number of the given date", sample\_date, "is",week\_number)

1. Write a Python program to find the date of the first Monday of a given week.  
   Sample Year and week : 2015, 50  
   Expected Output : Mon Dec 14 00:00:00 2015

Ans:-

import time

print(time.asctime(time.strptime('2015 50 1','%Y %W %w')))

# '%Y %W %w' this year, week number, weekday format is specified

1. Write a Python program to select all the Sundays in a specified year.

Ans:-

def all\_sundays(year):

given\_date = datetime(year,1,1)

given\_date += timedelta(days = (6 - given\_date.weekday())) #First sunday of the year

while given\_date.year == year:

print(given\_date.date())

given\_date += timedelta(days=7)

Example: all\_sundays(2024)

1. Write a Python program to add year(s) to a given date and display the updated date.  
     
   Sample Data : (addYears is the user defined function name)  
   print(addYears(datetime.date(2015,1,1), -1))  
   print(addYears(datetime.date(2015,1,1), 0))  
   print(addYears(datetime.date(2015,1,1), 2))  
   print(addYears(datetime.date(2000,2,29),1))  
     
   Expected Output :  
   2014-01-01  
   2015-01-01  
   2017-01-01  
   2001-03-01

Ans:-

def addYears(given\_date, years):

try:

return given\_date.replace(year=given\_date.year + years).date()

except ValueError: # avoiding leap year issue so using try and except

return given\_date.replace(month=2, day=28, year=given\_date.year + years).date()

print(addYears(datetime(2015,1,1),-1))

print(addYears(datetime(2015,1,1),0))

print(addYears(datetime(2015,1,1),2))

print(addYears(datetime(2000,2,29),1))

1. Write a Python program to drop microseconds from datetime.

Ans:-

print(datetime.now().replace(microsecond=0))

1. Write a Python program to get days between two dates.  
   Sample Dates : 2000,2,28, 2001,2,28  
   Expected Output : 366 days, 0:00:00

Ans:-

date1 = datetime(2000,2,28)

date2 = datetime(2001,2,28)

print(date2 - date1)

1. Write a Python program to get the date of the last Tuesday.

Ans:-

today = datetime.now().date()

day = (dt.weekday()-1) % 7

last\_tuesday = today - timedelta(days = day)

print("Last Tuesday is",last\_tuesday)

1. Write a [Python](https://www.w3resource.com/python-exercises/date-time-exercise/index.php) program to test the third Tuesday of a month.

Ans:-

import calendar

year = 2024

month = 9

#cal object return the list in list for weekdays by calendar

cal = calendar.monthcalendar(year,month)

tuesday = [week[calendar.TUESDAY] for week in cal if week[calendar.TUESDAY] !=0]

#print(cal)

print(tuesday)

1. Write a Python program to get the last day of a specified year and month.

Ans:-

year = 2015

month = 2

print(calendar.monthrange(year,month)[1])

#monthrange return tuple weekday and number of days in a month so give index 1

1. Write a Python program to get the number of days in a given month and year.

Ans:-

year = 2016

month =2

print(calendar.monthrange(year, month))

1. Write a Python program to add a month to a specified date.

Ans:-

from datetime import datetime, date

given\_date = datetime(2016,2,26)

month = given\_date.month + 1

add\_1mnth\_date = given\_date.replace(month=month).date()

print("Give date is:", given\_date.date())

print("Added one month:",add\_1mnth\_date)

1. Write a Python program to count the number of Mondays on the 1st day of the month from 2015 to 2016.

Ans:-

monday\_count = 0

for year in range(2015,2017):

for month in range(1,13):

if datetime(year,month,1).weekday() == 0:

monday\_count += 1

print(monday\_count)

1. Write a Python program to print a string five times, with a delay of three seconds..

Ans:-

import time

str\_to\_print = "Hello world!"

for i in range(5):

print(str\_to\_print)

time.sleep(3)

1. Write a Python program that calculates the date six months from the current date using the datetime module.

Ans:-

from datetime import datetime

from dateutil.relativedelta import relativedelta

current\_date = datetime.now()

six\_months\_later = current\_date + relativedelta(months = 6)

print("current date:",current\_date.date())

print("Added 6 month in currentr date:",six\_months\_later.date());

1. Write a Python program to create 12 fixed dates from a specified date over a given period. The difference between two dates is 20.

Ans:-

from datetime import datetime, timedelta

start\_date = datetime(2024,9,27).date()

def generate\_fixed\_dates(start\_date,num\_dates=12,day\_diff=20):

dates = []

current\_data = start\_date

for i in range(num\_dates):

dates.append(current\_data)

current\_data += timedelta(days=day\_diff)

return dates

fixed\_dates = generate\_fixed\_dates(start\_date)

#print(fixed\_dates)

for i, date in enumerate(fixed\_dates, 1):

print(f"Date{i}:{date}")

1. Write a Python program to get the dates 30 days before and after today.

Ans:-

today = datetime.now().date()

thirty\_days\_before = today - timedelta(days=30)

thirty\_days\_after = today + timedelta(days=30)

print("Today date:", today)

print("30 days before:", thirty\_days\_before)

print("30 days after:", thirty\_days\_after)

1. Write a Python program to get GMT and the local time.

Ans:-

from datetime import datetime, timezone

local\_time = datetime.now()

gmt\_time = datetime.now(timezone.utc)

print("Local Time:",local\_time.strftime('%Y-%m-%d %H:%M:%S'))

print("GMT Time:", gmt\_time.strftime('%Y-%m-%d %H:%M:%S %Z'))

1. Write a Python program to convert a date to a timestamp.

Ans:-

date\_obj = datetime.now()

timestamp = date\_obj.timestamp()

print("Date:", date\_obj)

print("Timestamp:", int(timestamp))

1. Write a Python program to convert a string date to a timestamp.

Ans:-

date\_str = '2024-09-27'

date\_obj = datetime.strptime(date\_str,'%Y-%m-%d')

timestamp = date\_obj.timestamp()

print("String date:",date\_str)

print("Timestamp:",int(timestamp))

1. Write a Python program to calculate the number of days between two dates.

Ans:-

date1 = datetime(2000,2,28)

date2 = datetime(2001,2,28)

print(date2 - date1)

1. Write a Python program to calculate the number of days between two date times.

Ans:-

dt\_str1 = "2024-09-27 14:30:00"

dt\_str2 = "2024-10-15 09:15:00"

dt\_time\_obj1 = datetime.strptime(dt\_str1,'%Y-%m-%d %H:%M:%S')

dt\_time\_obj2 = datetime.strptime(dt\_str2,'%Y-%m-%d %H:%M:%S')

dt\_time\_diff = dt\_time\_obj2 - dt\_time\_obj1

print(dt\_time\_diff)

1. Write a Python program to display the date and time in a human-friendly string.

Ans:-

now = datetime.now()

human\_friendly\_str = now.strftime("Today is %A, %B %d, %Y, and the time is %I:%M %p")

print(human\_friendly\_str)

1. Write a Python program to convert a date to a Unix timestamp.

Ans:-

dt = datetime.now()

timestamp = int(dt.timestamp())

print("Date:", dt.date())

print("Unix Timestamp:",timestamp)

1. Write a Python program to calculate the difference between two dates in seconds.

Ans:-

dt\_str1 = "2024-09-27 14:30:00"

dt\_str2 = "2024-10-15 09:15:00"

dt\_time\_obj1 = datetime.strptime(dt\_str1,'%Y-%m-%d %H:%M:%S')

dt\_time\_obj2 = datetime.strptime(dt\_str2,'%Y-%m-%d %H:%M:%S')

dt\_time\_diff = dt\_time\_obj2 - dt\_time\_obj1

diff\_in\_seconds = dt\_time\_diff.total\_seconds()

print(diff\_in\_seconds)

1. Write a Python program to convert difference of two dates into days, hours, minutes, and seconds.

Ans:-

dt\_str1 = "2024-09-27 14:30:00"

dt\_str2 = "2024-10-15 09:15:00"

dt\_time\_obj1 = datetime.strptime(dt\_str1,'%Y-%m-%d %H:%M:%S')

dt\_time\_obj2 = datetime.strptime(dt\_str2,'%Y-%m-%d %H:%M:%S')

dt\_time\_diff = dt\_time\_obj2 - dt\_time\_obj1

diff\_in\_seconds = dt\_time\_diff.total\_seconds()

#print(diff\_in\_seconds)

print("Days:", diff\_in\_seconds//86400) # 1day = 86400 sec

remaining\_seconds = diff\_in\_seconds % 86400 # pick the remainder for hours

print("Hours:", remaining\_seconds//3600) # 1 hour = 3600 sec

remaining\_seconds %= 3600 # picke the remainder for minutes

print("Minutes:", remaining\_seconds//60) # 1 min = 60 sec

print("Seconds:", remaining\_seconds % 60)

**38.** Write a Python program to get the last modified information of a file.

**39.** Write a Python program to calculate an age in years.

**A**ns:-

today = datetime.now().date()

birthday = datetime(2001,1,1).date()

age = today.year - birthday.year

print(age)

**40.** Write a Python program to get the current date and time information.

**Ans:**-

print("current date and time is",datetime.now())

**41.** Write a Python program to generate a date and time as a string.

Ans:-

print("Current date and time:",datetime.now().strftime("%Y-%m-%d %H:%M:%S"))

**42.** Write a Python program to display formatted text output of a month and start the week on Sunday.

Ans:-

import calendar

year = 2024

month = 9

cal = calendar.TextCalendar(firstweekday=6) # 6 means sunday

formatted\_month = cal.formatmonth(year,month)

print(formatted\_month)

**43.** Write a Python program to print a 3-column [calendar](https://www.w3resource.com/python-exercises/date-time-exercise/index.php) for an entire year.

Ans:-

import calendar

year = 2024

cal = calendar.TextCalendar()

for month in range(1,13):

print(cal.formatmonth(year,month),end = ' ')

if month % 3 ==0: #print new line aftre every 3 month

print("\n -------------------------------------------------------")

**44.** Write a Python program to display a calendar for a locale.

Ans:-

import calendar

import locale

locale.setlocale(locale.LC\_TIME, 'fr\_FR.UTF-8') # for French

year=2024

month=9

cal = calendar.TextCalendar()

formatted\_calendar = cal.formatmonth(year,month)

print(formatted\_calendar)

**45.** Write a Python program to get the current week.

Ans:-

current\_date = datetime.now()

current\_week\_number = current\_date.isocalendar()[1]

print("current date:",current\_date)

print("current week:", current\_week\_number)

**46.** Write a Python program to create a HTML calendar with data for a specific year and month.

Ans:-

import calendar

def create\_html\_calendar(year, month):

html\_calendar = calendar.HTMLCalendar()

month\_calendar = html\_calendar.formatmonth(year, month)

# Open a file to write the HTML

with open(f'calendar\_{year}\_{month}.html', 'w') as file:

file.write('<html>\n')

file.write('<head><title>Calendar</title></head>\n')

file.write('<body>\n')

file.write('<h1>Calendar for {0}/{1}</h1>\n'.format(month, year))

file.write(month\_calendar)

file.write('</body>\n')

file.write('</html>\n')

print(f"Calendar has been created as 'calendar\_{year}\_{month}.html'.")

# Call the function to create the HTML calendar for September 2024

create\_html\_calendar(2024, 9)

**47.** Write a Python program display a list of the dates for the 2nd Saturday of every month for a given year.

Ans:-

from datetime import datetime, timedelta

def get\_second\_saturdays(year):

second\_saturdays = []

# Loop through all months

for month in range(1, 13):

first\_day = datetime(year, month, 1)

# Get the first Saturday of the month

first\_saturday = first\_day + timedelta(days=(5 - first\_day.weekday() + 7) % 7)

# The second Saturday is 7 days after the first Saturday

second\_saturday = first\_saturday + timedelta(days=7)

second\_saturdays.append(second\_saturday)

return second\_saturdays

# Display results

year = 2024

second\_saturdays = get\_second\_saturdays(year)

print("Second Saturdays in", year, ":")

for date in second\_saturdays:

print(date.strftime('%Y-%m-%d'))

**48.** Write a Python program to display a simple, formatted calendar of a given year and month.

Ans:-

import calendar

year = 2024

month = 9

print(calendar.month(year,month))

**49.** Write a Python program to convert a string into datetime

**Please check questions 3**

**50.** Write a Python program to get a list of dates between two dates.

Ans:-

from datetime import datetime, timedelta

def get\_dates\_in\_list(start\_date,end\_date):

date\_list = []

while start\_date < end\_date:

date\_list.append(start\_date.strftime('%Y-%m-%d'))

start\_date += timedelta(days=1)

return date\_list

start\_date = datetime(2024,9,1)

end\_date = datetime(2024,9,27)

dates = get\_dates\_in\_list(start\_date.date(),end\_date.date())

print(dates)

**51.** Write a Python program to generate RFC 3339 timestamp.

Ans:-

from datetime import datetime, timezone

now\_utc = datetime.now(timezone.utc)

print("RFC 3339 Timestamp:",now\_utc.isoformat())

**52.** Write a Python program to get the first and last second.

Ans:-

from datetime import datetime, time

today = datetime.now().date()

first\_sec = datetime.combine(today,time(0,0,0))

print(first\_sec)

last\_sec = datetime.combine(today,time(23,59,59))

print(last\_sec)

**53.** Write a Python program to validate a Gregorian date. The month is between 1 and 12 inclusive, the day is within the allowed number of days for the given month. Leap year's are taken into consideration. The year is between 1 and 32767 inclusive.

Ans:-

from datetime import datetime

def validate\_gregorian\_date(year,month,day):

try:

if not(1<=year<=32767):

return False

datetime(year,month,day)

return True

except ValueError:

return False

dates\_to\_validate = [(2024,2,29),(2023,2,29),(2024,13,1),(2024,4,31),(32767,12,31),(2023,12,31)]

for year,month,day in dates\_to\_validate:

if validate\_gregorian\_date(year,month,day):

print(f"The date {year}-{month}-{day} is valid.")

else:

print(f"The date {year}-{month}-{day} is invalid.")

**54.** Write a [Python](https://www.w3resource.com/python-exercises/date-time-exercise/index.php) program to set the default timezone used by all date/time functions.

Ans:-

from datetime import datetime

import pytz

def set\_default\_timezone(timezone\_str):

timezone = pytz.timezone(timezone\_str)

local\_time = datetime.now(timezone)

return local\_time

if \_\_name\_\_ == "\_\_main\_\_":

timezone\_str = 'America/New\_York'

current\_time = set\_default\_timezone(timezone\_str)

print(f"The current time in {timezone\_str} is: {current\_time.strftime('%Y-%m-%d %H:%M:%S')}")

**55.** The epoch is the point where time starts, and is platform dependent. For Unix, the epoch is January 1, 1970, 00:00:00 (UTC). Write a Python program to find out what the epoch is on a given platform. Convert a given time in seconds since the epoch.  
Sample Output:  
Epoch on a given platform:  
time.struct\_time(tm\_year=1970, tm\_mon=1, tm\_mday=1, tm\_hour=0, tm\_min=0, tm\_sec=0, tm\_wday=3, tm\_yday=1, tm\_isdst=0)  
Time in seconds since the epoch:  
time.struct\_time(tm\_year=1970, tm\_mon=1, tm\_mday=1, tm\_hour=10, tm\_min=0, tm\_sec=0, tm\_wday=3, tm\_yday=1, tm\_isdst=0)

Ans:-

import time

epoch = time.gmtime(0)

print("Epoch on a given platform:")

print(epoch)

seconds\_since\_epoch = 36000 #10 hours since epoch

time\_from\_epoch = time.gmtime(seconds\_since\_epoch)

print("\nTime in seconds since the epoch (36000 seconds):")

print(time\_from\_epoch)

**56.** Write a Python program to get time values with components using local time and gmtime.  
Sample Output:  
localtime:  
tm\_year : 2021  
tm\_mon : 4  
tm\_mday : 13  
tm\_hour : 11  
tm\_min : 20  
tm\_sec : 37  
tm\_wday : 1  
tm\_yday : 103  
tm\_isdst: 0  
gmtime:  
tm\_year : 2021  
tm\_mon : 4  
tm\_mday : 13  
tm\_hour : 11  
tm\_min : 20  
tm\_sec : 37  
tm\_wday : 1  
tm\_yday : 103  
tm\_isdst: 0

Ans:

import time

local\_time = time.localtime()

#print(local\_time)

gmt\_time = time.gmtime()

def print\_time\_components(label,time\_struct):

print(f"{label}:")

print(f"tm\_year : {time\_struct.tm\_year}")

print(f"tm\_mon : {time\_struct.tm\_mon}")

print(f"tm\_mday : {time\_struct.tm\_mday}")

print(f"tm\_hour : {time\_struct.tm\_hour}")

print(f"tm\_min : {time\_struct.tm\_min}")

print(f"tm\_sec : {time\_struct.tm\_sec}")

print(f"tm\_wday : {time\_struct.tm\_wday}")

print(f"tm\_yday : {time\_struct.tm\_yday}")

print(f"tm\_isdst : {time\_struct.tm\_isdst}")

print()

print\_time\_components("localtime",local\_time)

print\_time\_components("gmtime",gmt\_time)

**57.** Write a Python program to get different time values with components timezone, timezone abbreviations, the offset of the local (non-DST) timezone, DST timezone and time of different timezones.  
Sample Output:  
Default Zone:  
TZ : (not set)  
Timezone abbreviations: ('UTC', 'UTC')  
Timezone : 0 (0.0)  
DST timezone 0  
Time : 11:30:05 04/13/21 UTC  
Pacific/Auckland :  
TZ : Pacific/Auckland  
Timezone abbreviations: ('NZST', 'NZDT')  
Timezone : -43200 (-12.0)  
DST timezone 1  
Time : 23:30:05 04/13/21 NZST  
Europe/Berlin :  
TZ : Europe/Berlin  
Timezone abbreviations: ('CET', 'CEST')  
Timezone : -3600 (-1.0)  
DST timezone 1  
Time : 13:30:05 04/13/21 CEST  
America/Detroit :  
TZ : America/Detroit  
Timezone abbreviations: ('EST', 'EDT')  
Timezone : 18000 (5.0)  
DST timezone 1  
Time : 07:30:05 04/13/21 EDT  
Singapore :  
TZ : Singapore  
Timezone abbreviations: ('+08', '+08')  
Timezone : -28800 (-8.0)  
DST timezone 0  
Time : 19:30:05 04/13/21 +08

Ans:-

from datetime import datetime

import pytz

def display\_timezone\_info(timezone\_str):

timezone = pytz.timezone(timezone\_str)

#print(timezone)

current\_time = datetime.now(timezone)

#print(current\_time)

tzname = current\_time.tzname() #timezone name

#print(tzname)

offset = current\_time.utcoffset().total\_seconds() # UTC offset in seconds

#print(offset)

is\_dst = current\_time.dst().total\_seconds() !=0 # check if dst is active

#print(is\_dst)

print(f"{timezone\_str}:")

print(f"TZ : {timezone\_str}")

print(f"Timezone abbreviations: {timezone.\_tzname}")

print(f"Timezone : {int(offset)} ({offset/3600:.1f})")

print(f"DST timezone : {int(is\_dst)}")

print(f"Time :{current\_time.strftime('%H:%M:%S %m/%d/%y %Z')}")

print()

default\_tz = pytz.timezone('UTC')

current\_time\_utc = datetime.now(default\_tz)

print("Default Zone:")

print(f"TZ : (not set)")

print(f"Timezone abbreviations: {('UTC', 'UTC')}")

print(f"Timezone : 0 (0.0)")

print(f"DST timezone : 0")

print(f"Time : {current\_time\_utc.strftime('%H:%M:%S %m/%d/%y UTC')}")

print()

timezones = ['Pacific/Auckland', 'Europe/Berlin', 'America/Detroit', 'Asia/Singapore']

for tz in timezones:

display\_timezone\_info(tz)

**58.** Write a Python program that can suspend execution of a given script for a given number of seconds.  
Sample Output:  
Sorry, Slept for 3 seconds...  
Sorry, Slept for 3 seconds...  
Sorry, Slept for 3 seconds...  
Sorry, Slept for 3 seconds...

Ans:-

import time

def suspend\_exe(sec,repeat):

for i in range(repeat):

print(f"Sorry, Slept for {sec} seconds...")

time.sleep(sec)

suspend\_exe(3,4)

**59.** Write a Python program to convert a given time in seconds since the epoch to a string representing local time.  
Sample Output:  
Tue Apr 13 11:51:51 2021  
Thu Jun 30 18:36:29 1977

Ans:-

import time

def covert\_seconds\_to\_local\_time(seconds):

local\_time = time.localtime(seconds)

time\_str = time.strftime("%a %b %d %H:%M:%S %Y",local\_time)

return time\_str

seconds1 = 1618319511

seconds2 = 236592989

print(covert\_seconds\_to\_local\_time(seconds1))

print(covert\_seconds\_to\_local\_time(seconds2))

**60.** Write a Python program that prints the time, names, representation format, and the preferred date time format in a simple format.  
Sample Output:  
Simple format of time:  
Tue, 13 Apr 2021 12:02:01 + 1010  
Full names and the representation:  
Tuesday, 04/13/21 April 2021 12:02:01 + 0000  
Preferred date time format:  
Tue Apr 13 12:02:01 2021  
Example 11: 04/13/21, 12:02:01, 21, 2021

Ans:-

from datetime import datetime

def print\_time\_formats():

current\_time = datetime.now()

simple\_format = current\_time.strftime("%a, %d %b %Y %H:%M:%S + %f")[:29] #Trim microseconds part

print("Simple format of time:")

print(simple\_format)

full\_representation = current\_time.strftime("%A, %m/%d/%y %B %Y %H:%M:%S + %f")[:35]

print("\nFull names and the representation:")

print(full\_representation)

preferred\_format = current\_time.strftime("%a %b %d %H:%M:%S %Y")

print("\nPreferred date time format:")

print(preferred\_format)

example\_format = current\_time.strftime("%m/%d/%y, %H:%M:%S, %y, %Y")

print("\nExample 11: " + example\_format)

print\_time\_formats()

**61.** Write a Python program that takes a given number of seconds and passes since the epoch as an argument. Print structure time in local time.  
Sample Output:  
Result: time.struct\_time(tm\_year=1983, tm\_mon=2, tm\_mday=19, tm\_hour=21, tm\_min=38, tm\_sec=18, tm\_wday=5, tm\_yday=50, tm\_isdst=0)  
Year: 1983

Ans:-

import time

seconds\_since\_epoch = 413264298

local\_time = time.localtime(seconds\_since\_epoch)

print("Result:",local\_time)

print("Year:", local\_time.tm\_year)

**62.** Write a Python program that takes a tuple containing 9 elements corresponding to structure time as an argument and returns a string representing it.  
Sample Output:  
Result: Sun Jan 22 02:34:06 2020  
Result: Tue Nov 12 02:54:08 1982

Ans:-

import time

def tuple\_to\_time\_string(time\_tuple):

time\_string = time.asctime(time\_tuple)

return time\_string

time\_tuple1 = (2020, 1, 22, 2, 34, 6, 2, 22, -1)

time\_tuple2 = (1982, 11, 12, 2, 54, 8, 1, 316, -1)

print("Result:", tuple\_to\_time\_string(time\_tuple1))

print("Result:", tuple\_to\_time\_string(time\_tuple2))

**63.** Write a [Python](https://www.w3resource.com/python-exercises/date-time-exercise/index.php) program to [parse](https://www.w3resource.com/python-exercises/date-time-exercise/index.php) a string representing time and return the time structure.  
Sample Output:  
String representing time: 22 January, 2020  
time.struct\_time(tm\_year=2020, tm\_mon=1, tm\_mday=22, tm\_hour=0, tm\_min=0, tm\_sec=0, tm\_wday=2, tm\_yday=22, tm\_isdst=-1)  
String representing time: 30 Nov 00  
time.struct\_time(tm\_year=2000, tm\_mon=11, tm\_mday=30, tm\_hour=0, tm\_min=0, tm\_sec=0, tm\_wday=3, tm\_yday=335, tm\_isdst=-1)  
String representing time: 04/11/15 11:55:23  
time.struct\_time(tm\_year=2015, tm\_mon=4, tm\_mday=11, tm\_hour=11, tm\_min=55, tm\_sec=23, tm\_wday=5, tm\_yday=101, tm\_isdst=-1)  
String representing time: 12-11-2019  
time.struct\_time(tm\_year=2019, tm\_mon=12, tm\_mday=11, tm\_hour=0, tm\_min=0, tm\_sec=0, tm\_wday=2, tm\_yday=345, tm\_isdst=-1)  
String representing time: 13::55::26  
time.struct\_time(tm\_year=1900, tm\_mon=1, tm\_mday=1, tm\_hour=13, tm\_min=55, tm\_sec=26, tm\_wday=0, tm\_yday=1, tm\_isdst=-1)

Ans:-

import time

def parse\_time\_string(time\_string,format):

return time.strptime(time\_string,format)

time\_strings = [("22 January, 2020", "%d %B, %Y"),("30 Nov 00", "%d %b %y"),

("04/11/15 11:55:23", "%d/%m/%y %H:%M:%S"),

("12-11-2019", "%d-%m-%Y"),

("13::55::26", "%H::%M::%S")]

for time\_string, fmt in time\_strings:

parsed\_time = parse\_time\_string(time\_string,fmt)

print(f"String representing time: {time\_string}")

print(parsed\_time)

print()