**DATE AND TIME**

Compter Language – Concept of epoch 🡪 Jan 1st 1970.

ctime() 🡪 time

datetime 🡪 datetime

**Date Time Examples**

**Code:**

import time,datetime

from calendar import month

epochSeconds = time.time()

print(*"Seconds from 1st Jan 1970 : "*,epochSeconds)

print()

#CONVERT THE ABOVE SECONDS TO CURRENT TIME

currentTime = time.ctime(epochSeconds)

print(*"Current Time from above epoch seconds : "*, currentTime)

print()

dt = datetime.datetime.today()

print(*"Current Date Time (using datetime) : "*, dt)

print()

print(*'Current Date: {}/{}/{}'*.format(dt.day,dt.month,dt.year))

print()

print(*"Current Time: {}:{}:{}"*.format(dt.hour,dt.minute,dt.second))

print()

print(*"Current Day : "*, dt.day)

print()

print(*"Current Month : "*, dt.month)

print()

print(*"Current Year : "*, dt.year)

print()

print(*"Current Minute : "*, dt.minute)

print()

**Output:**

Seconds from 1st Jan 1970 : 1711914396.9683828

Current Time from above epoch seconds : Sun Mar 31 20:46:36 2024

Current Date Time (using datetime) : 2024-03-31 20:46:36.968383

Current Date: 31/3/2024

Current Time: 20:46:36

Current Day : 31

Current Month : 3

Current Year : 2024

Current Minute : 46

Current Second : 36

**Combine Date and Time**

**Code:**

from datetime import \*

da = date(2024,2,14)

tm=time(12,45)

print(*"combined Date Time : "*, datetime.combine(da, tm))

**Output:**

combined Date Time : 2024-02-14 12:45:00

**Sort Dates from a list**

**Code:**

from datetime import \*

datList=[]

d1=date(1976,9,5)

d2=date(1976,8,14)

d3=date(2007,9,10)

d4=date(2013,11,29)

d5=date(2006,6,18)

datList.append(d1)

datList.append(d2)

datList.append(d3)

datList.append(d4)

datList.append(d5)

datList.sort()

print(*"Output of Sorted dates in Ascending Order"*)

print(*"-----------------------------------------"*)

for dt in datList:

print(dt)

print()

**Output:**

Output of Sorted dates in Ascending Order

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

1976-08-14

1976-09-05

2006-06-18

2007-09-10

2013-11-29

**Sleep the method:**

**Code:**

from datetime import date

import time,datetime

datList=[]

d1=date(1976,9,5)

d2=date(1976,8,14)

d3=date(2007,9,10)

d4=date(2013,11,29)

d5=date(2006,6,18)

datList.append(d1)

datList.append(d2)

datList.append(d3)

datList.append(d4)

datList.append(d5)

datList.sort()

dt = datetime.datetime.today()

#PROGAM WILL WAIT FOR 3 SECTONDS

print(*"Executed Before 3 Seconds Sleep: {}:{}:{}"*.format(dt.hour,dt.minute,dt.second))

time.sleep(10)

print(*"---------------TEMPORARILY SLEEPING FOR 10 SECONDS---------------"*)

dt = datetime.datetime.today()

print(*"Executed After 3 Seconds Sleep: {}:{}:{}"*.format(dt.hour,dt.minute,dt.second))

print(*"Output of Sorted dates in Ascending Order"*)

print(*"-----------------------------------------"*)

for dt in datList:

print(dt)

print()

**Output:**

Executed Before 3 Seconds Sleep: 21:59:33

---------------TEMPORARILY SLEEPING FOR 10 SECONDS---------------

Executed After 3 Seconds Sleep: 21:59:43

Output of Sorted dates in Ascending Order

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

1976-08-14

1976-09-05

2006-06-18

2007-09-10

2013-11-29

**Program Execution Time**

**Code:**

from datetime import date

import time,datetime

startTime=time.perf\_counter()

datList=[]

d1=date(1976,9,5)

d2=date(1976,8,14)

d3=date(2007,9,10)

d4=date(2013,11,29)

d5=date(2006,6,18)

datList.append(d1)

datList.append(d2)

datList.append(d3)

datList.append(d4)

datList.append(d5)

datList.sort()

*'''dt = datetime.datetime.today()*

*#PROGAM WILL WAIT FOR 3 SECTONDS*

*print("Executed Before 3 Seconds Sleep: {}:{}:{}".format(dt.hour,dt.minute,dt.second))*

*time.sleep(10)*

*print("---------------TEMPORARILY SLEEPING FOR 10 SECONDS---------------")*

*dt = datetime.datetime.today()*

*print("Executed After 3 Seconds Sleep: {}:{}:{}".format(dt.hour,dt.minute,dt.second))'''*

print(*"Output of Sorted dates in Ascending Order"*)

print(*"-----------------------------------------"*)

for dt in datList:

print(dt)

print()

endTime=time.perf\_counter()

print()

print(*"Start Time of the program : "*,startTime)

print(*"End Time of the program : "*, endTime)

print()

print(*"Execution Time : "*, (endTime-startTime))

**Output:**

Start Time of the program : 1557495.8367142

End Time of the program : 1557495.8368014

Execution Time : 8.719996549189091e-05

<REST OUTPUT FOR DISPLAY ARE REMOVED>

**Example to check the date as valid or not:**

**Code:**

from datetime import \*  
  
def validateCard(expdt):  
 if expdt>datetime.now().date():  
 print("Card Expiry Date - {}/{}/{}".format(expdt.day, expdt.month, expdt.year) + " - is Valid")  
 else:  
 print("Card Expiry Date - {}/{}/{}".format(expdt.day, expdt.month, expdt.year) + " - is InValid")  
  
  
*#vdate = date(input("Please enter Card Expirty Dateate : "))*vdate=date(2027,4,28)  
validateCard(vdate)  
vdate1=date(2023,4,28)  
validateCard(vdate1)

**Output:**

C:\Users\kamal\pythonlab\Scripts\python.exe C:\Users\kamal\PycharmProjects\pythonProjectdemo\creditCardValidate.py

Card Expiry Date - 28/4/2027 - is Valid

Card Expiry Date - 28/4/2023 - is InValid

Process finished with exit code 0

**Simple Project Management Example:**

**Code:**

from datetime import \*  
  
class Project:  
 def \_\_init\_\_(self, name, sdat,edat):  
 self.projName=name  
 self.projStartDate=sdat  
 self.projEndDate=edat  
 self.tasks=[]  
  
 def addTask(self,task):  
 self.tasks.append(task)  
class Task:  
 def \_\_init\_\_(self,name,dur):  
 self.taskName=name  
 self.taskDuration=dur  
 self.resources=[]  
  
 def addResources(self,resource):  
 self.resources.append(resource)  
  
class Resource:  
 def \_\_init\_\_(self, name,skill):  
 self.resName=name  
 self.resSkill=skill  
  
project = Project("Banking - Geneva Team",date(2024,5,1),date(2024,12,31))  
task1 = Task("Create Backend for Geneva Scrub",60)  
task2 = Task("Create Frontend for Geneva Scrub",90)  
task3 = Task("CICD Pipline using DevOps (Geneva Scrub)",30)  
task4 = Task("Deploy in Cloud using AWS (Geneva Scrub)",60)  
  
  
resource1 = Resource("Kamal Kumar","Python Developer")  
resource2 = Resource("Purusoth","Senthil Developer")  
resource3 = Resource("Ragav","DevOps Developer")  
resource4 = Resource("Sathya","Cloud Developer")  
  
task1.addResources(resource1)  
task2.addResources(resource2)  
task3.addResources(resource3)  
task4.addResources(resource4)  
  
project.addTask(task1)  
project.addTask(task2)  
project.addTask(task3)  
project.addTask(task4)  
  
for eachTask in project.tasks:  
 print(eachTask.taskName)  
 for eachRes in eachTask.resources:  
 print(eachRes.resName)  
 print(eachRes.resSkill)  
 print()

**Output :**

C:\Users\kamal\pythonlab\Scripts\python.exe C:\Users\kamal\PycharmProjects\pythonProjectdemo\projectManagement.py

Create Backend for Geneva Scrub

Kamal Kumar

Python Developer

Create Frontend for Geneva Scrub

Purusoth

Senthil Developer

CICD Pipline using DevOps (Geneva Scrub)

Ragav

DevOps Developer

Deploy in Cloud using AWS (Geneva Scrub)

Sathya

Cloud Developer

Process finished with exit code 0

**QUIZ**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A close-up of a form

Description automatically generated

A screenshot of a computer

Description automatically generated

**ASSIGNMENT**

**Code:**

from datetime import \*  
class Event:  
 def \_\_init\_\_(self,stime,etime,ven):  
 self.eventStartTime=stime  
 self.eventEndTime=etime  
 self.eventVenue=ven  
 self.venues=[]  
  
 def addEvent(self,venue):  
 self.venues.append(venue)  
  
class Venue:  
 def \_\_init\_\_(self,vname,vadd):  
 self.venueName=vname  
 self.venueAddress=vadd  
 self.addresses=[]  
  
 def addAddress(self,addr):  
 self.addresses.append(addr)  
  
class Address:  
 def \_\_init\_\_(self,str,cty,cnt,zcode):  
 self.addStreet=str  
 self.addCity=cty  
 self.addCountry=cnt  
 self.addZipCode=zcode  
  
event= Event(time(18,10,50),time(21,30,30),"GAA Club")  
venue1 = Venue("GAA Club", "Adamstown")  
event.addEvent(venue1)  
venue2 = Venue("ACC Community Center", "Adamstown")  
event.addEvent(venue2)  
  
address1 = Address("Adamstown Road", "Lucan", "Ireland","Co Dublin")  
venue1.addAddress(address1)  
address2 = Address("Adamstown Avenue Road", "Adamstown", "Ireland","Co Dublin")  
venue2.addAddress(address2)  
  
print("Start Time : ",event.eventStartTime)  
print("End Time : ",event.eventEndTime)  
  
for eachVen in event.venues:  
 print(eachVen.venueName)  
 print("=======================")  
 for eachaddr in eachVen.addresses:  
 print(eachaddr.addStreet)  
 print(eachaddr.addCity)  
 print(eachaddr.addCountry)  
 print(eachaddr.addZipCode)  
 print()

**Output:**

C:\Users\kamal\pythonlab\Scripts\python.exe C:\Users\kamal\PycharmProjects\pythonProjectdemo\dateAssignment.py

Start Time : 18:10:50

End Time : 21:30:30

GAA Club

=======================

Adamstown Road

Lucan

Ireland

Co Dublin

ACC Community Center

=======================

Adamstown Avenue Road

Adamstown

Ireland

Co Dublin

Process finished with exit code 0