

ধরুন, আপনি প্রায়ই কিছু সংখ্যার গড় বের করতে আগ্রহী। সব সময় গড় বের করার সঙ্কেত না লিখে একবারেই লিখে তা বারবার ব্যবহার করাটা যুক্তিযুক্ত। মডিউলের ধারণা অনেকটা এখান থেকেই এসেছে। মডিউল হচ্ছে একটি ফাইল, যেখানে প্রয়োজনমতো ফাংশন রাখা যায়। নোটপ্যাড চালু করে টাইপ করুন নিচে কোডগুলো :

```
def avg(num):
    r=0
    numlist = num.split()
    for n in numlist:
        r+=int(n)
    res = r/len(numlist)
    return res
```

পাইথনে মডিউলের ব্যবহার ও অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং

মৃগাল কান্তি রায় দীপ

পাইথন প্রোগ্রামিং পর্ব-৪

ফাইলটি “average.py” নামে সেভ করুন আপনার ডেস্কটপে।

এবার আরেকটি ফাইলে টাইপ করুন নিচের কোডগুলো :

```
import average
def showtext(buttontext):
    if buttontext=="OK":
        entry.insert(t.END, ". The average is: "+str(average.avg(entry.get())))
    else:
        entry.delete(0,t.END)
import tkinter as t
root=t.Tk()
t.Button(root,text="Get Average",
        command=lambda x="OK":
showtext(x)).grid(row=1,column=0)
t.Button(root,text="Clear All",
        command=lambda x="CA":
showtext(x)).grid(row=1,column=1)
label=t.Label(root, text="Numbers to average
(Give a space between them)")
label.grid(row=0,column=0)
entry=t.Entry(root, width=40)
entry.grid(row=0,column=1)
root.mainloop()
```

এটি “frontend.py” নামে সেভ করুন ডেস্কটপে। লক্ষ করুন, ফাইলের নামে উদ্ধৃতি চিহ্ন দেয়া হয়েছে। উদ্ধৃতি চিহ্ন ছাড়া সেভ করলে তা সাধারণ টেক্সট ফাইল হিসেবে সেভ হবে। এভাবে তৈরি হলো আপনার গড় বের করার মডিউল average, যা frontend.py ফাইলটির মাধ্যমে এখানে ব্যবহার হয়েছে এবং পরবর্তী সময়ে প্রয়োজন মতো ব্যবহার করতে পারবেন।

পাইথনের কোড বা সঙ্কেত আপনি উইন্ডোজের নোটপ্যাডেই সেভ করতে পারবেন। আবার <http://pype.sourceforge.net> থেকে সফটওয়্যারটি নামিয়ে নিয়ে আরও সহজেই কোড সেভ করতে ও চালাতে পারেন।

আধুনিক প্রোগ্রামিং ভাষার প্রায় সবকটিতেই অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং করার সুবিধা রয়েছে। পাইথনও এর ব্যতিক্রম নয়। অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিংয়ের (ওওপি) সুবিধা হচ্ছে যেকোনো সমস্যাই ছোট ছোট ভাগে ভাগ করা যায়। উদাহরণস্বরূপ, আপনি এটি ব্যাংকের গ্রাহক তথ্য সংরক্ষণের জন্য সফটওয়্যার তৈরি

করবেন। এটি কিভাবে ওওপির ধারণায় প্রকাশ করবেন তা নিয়ে কিছু ধারণা দেয়া যাক। প্রথমেই আপনাকে একটি class তৈরি করতে হবে। ক্লাস হচ্ছে ভ্যারিয়েবল, মেথড ইত্যাদির সমন্বয়ে তৈরি একটি কাঠামো মাত্র, যা এ ক্ষেত্রে গ্রাহকের নাম, হিসাব নম্বর ইত্যাদি সংরক্ষণ করবে। ক্লাস আগে থেকে নির্ধারিত থাকে না, প্রয়োজনমতো তৈরি করতে হয়। অর্থাৎ ক্লাস হচ্ছে প্রয়োজনীয় তথ্যের কাঠামো মাত্র, কোনো অর্থবহ তথ্য এখানে থাকে না। এখন গ্রাহকের তথ্য যোগ করতে চাইলে ওই ক্লাসের একটি নতুন instance বানাতে হবে, যা object নামে পরিচিত। এখন গ্রাহক ক্লাসের একটি উদাহরণ দেখুন :

```
class BankClient:
    name = ""
    acc_no = 0
    address = ""
    acc_bal = 0
    def createProfile(self,p_name,p_address):
        self.name=p_name
        self.address=p_address
    def createAccount(self,a_acc_no,a_acc_bal):
        self.acc_no=a_acc_no
        self.acc_bal=a_acc_bal
    def showDetails(self):
        print("Mr. "+self.name+" has Account Balance: "+str(self.acc_bal))
```

এখানে BankClient হচ্ছে একটি ক্লাস, যার কিছু ভ্যারিয়েবল ও মেথড রয়েছে। লক্ষ করুন, ভ্যারিয়েবল ক্লাসের প্রথমেই ডিক্লেয়ার করা হয়েছে। মেথড হচ্ছে ক্লাসের ভেতরের ফাংশনগুলো। তারপর নিচের সঙ্কেতগুলো লিখুন :

```
client1 = BankClient()
client1.createProfile("Rony","Dhaka")
client1.createAccount(1122,1000)
client2 = BankClient()
client2.createProfile("Johny","Khulna")
client2.createAccount(2211,2000)
client1.showDetails()
client2.showDetails()
```

এর ফলে তৈরি হয়ে যাবে ক্লায়েন্টের তথ্য, যা showDetails() মেথডের মাধ্যমে দেখা যাবে। এভাবে চেষ্টা করুন আরও কিছু ক্লায়েন্টের তথ্য যোগ করার।

```
class BankClient:
    name = ""
    acc_no = 0
    address = ""
    acc_bal = 0
    def createProfile(self,p_name,p_address):
        self.name=p_name
        self.address=p_address
    def createAccount(self,a_acc_no,a_acc_bal):
        self.acc_no=a_acc_no
        self.acc_bal=a_acc_bal
    def showDetails(self):
        print("Mr. "+self.name+" has Account Balance: "+str(self.acc_bal))
```

এখন যদি আপনি গ্রাহকের ঋণের তথ্য যোগ করতে চান, নতুন করে আবার গ্রাহকের ক্লাস লেখার প্রয়োজন নেই। নতুন LoanInfo নামে একটি ক্লাস যোগ করুন। LoanInfo ক্লাসটি হবে BankClient-এর Child ক্লাস। BankClient

ক্লাসটি হবে Parent ক্লাস। নতুন তৈরি করা ক্লাসটি তার নিজস্ব মেথড এবং ভ্যারিয়েবল ছাড়াও Parent ক্লাসের মেথড এবং ভ্যারিয়েবলগুলোও ব্যবহার করতে পারবে। এক ক্লাসের বৈশিষ্ট্য আরেক ক্লাসে এভাবে প্রসারিত করাকে বলা হয় Inheritance এবং এটি ব্যাপকভাবে ব্যবহার হওয়া একটি ধারণা। LoanInfo(BankClient) ক্লাসটির সঙ্কেতগুলো দেখুন :

```
class LoanInfo(BankClient):
    loan_Type=""
    loan_Amount=0
    def ApprovedLoan(self,l_type,l_amount):
        self.loan_Amount=l_amount
        self.loan_Type=l_type
    def LoanInfo(self):
        print("Loan Type: "+self.loan_Type+" and Amount: "+str(self.loan_Amount))
```

এবার নিচের সঙ্কেতগুলো লিখলেই দেখতে পাবেন গ্রাহকের তথ্যের সাথে সাথে ঋণের তথ্যও দেখাচ্ছে।

```
client3 = LoanInfo()
client3.createProfile("Salman", "Khulna")
client3.createAccount(2445, 10000)
client3.ApprovedLoan("House", 200000)
client4 = LoanInfo()
client4.createProfile("Amir", "Rajshahi")
client4.createAccount(2477, 10900)
client4.ApprovedLoan("Consumer", 208000)
client3.showDetails()
client3.LoanInfo()
client4.showDetails()
client4.LoanInfo()
```

শেখার জন্য পাইথনে লেখা একটা সহজ স্ক্রিপ্ট :

এবার একটু পাইথন Script নিয়ে দেখুন। একটা সহজ Script। টেক্সট এডিটর যেমন নোটপ্যাড বা পাইথন এডিটর ওপেন করুন। অথবা এমএসডস কমান্ড প্রম্পটে ওপেন করে লিখুন python

```
^ start > cmd > python
```

এখন কমপিউটারের System Platform, System Version, OS name, Current time বের করার জন্য একটা Script লিখুন।

০১. কোড এডিটরে লিখুন :

```
print('A Simple Python Script to find out system platform, system version, OS name & Current time')
import sys
sys.platform
sys.version
import os
os.name
import time
time.asctime(time.localtime(time.time()))
```

০২. এবার এটি সেভ করুন script.py নামে।

০৩. এরপর ফাইলটি এমএসডস কমান্ড প্রম্পটে রান করুন।

এবার কোডগুলো কী কাজ করে, তা দেখা যাক :

import sys এই লাইনটি দিয়ে sys library moduleটি লোড করা হয়েছে।

একইভাবে বাকি importগুলোও একই ধরনের কাজ করেছে।

পাইথনে কোনো function বা variable-এর নাম প্রিন্ট করতে চাইলে print লিখে ১ম ব্রাকেটের মধ্যে শুধু function বা variableটির নাম লিখলেই হয়।

যেমন : print(sys.platform) **ফল**

ফিডব্যাক : mkrdip@yahoo.com