

Object Oriented and Functional Programming with Python

by Dina Alhaj Ibrahim
14 Oct 2022

A photograph of a person's hand pointing their index finger towards a computer monitor. The monitor displays a dark-themed code editor with Python script content. The hand is positioned such that the index finger is directly pointing at the screen.

```
mirror_mod = modifier_obj
# mirror object to mirror
mirror_mod.mirror_object
operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

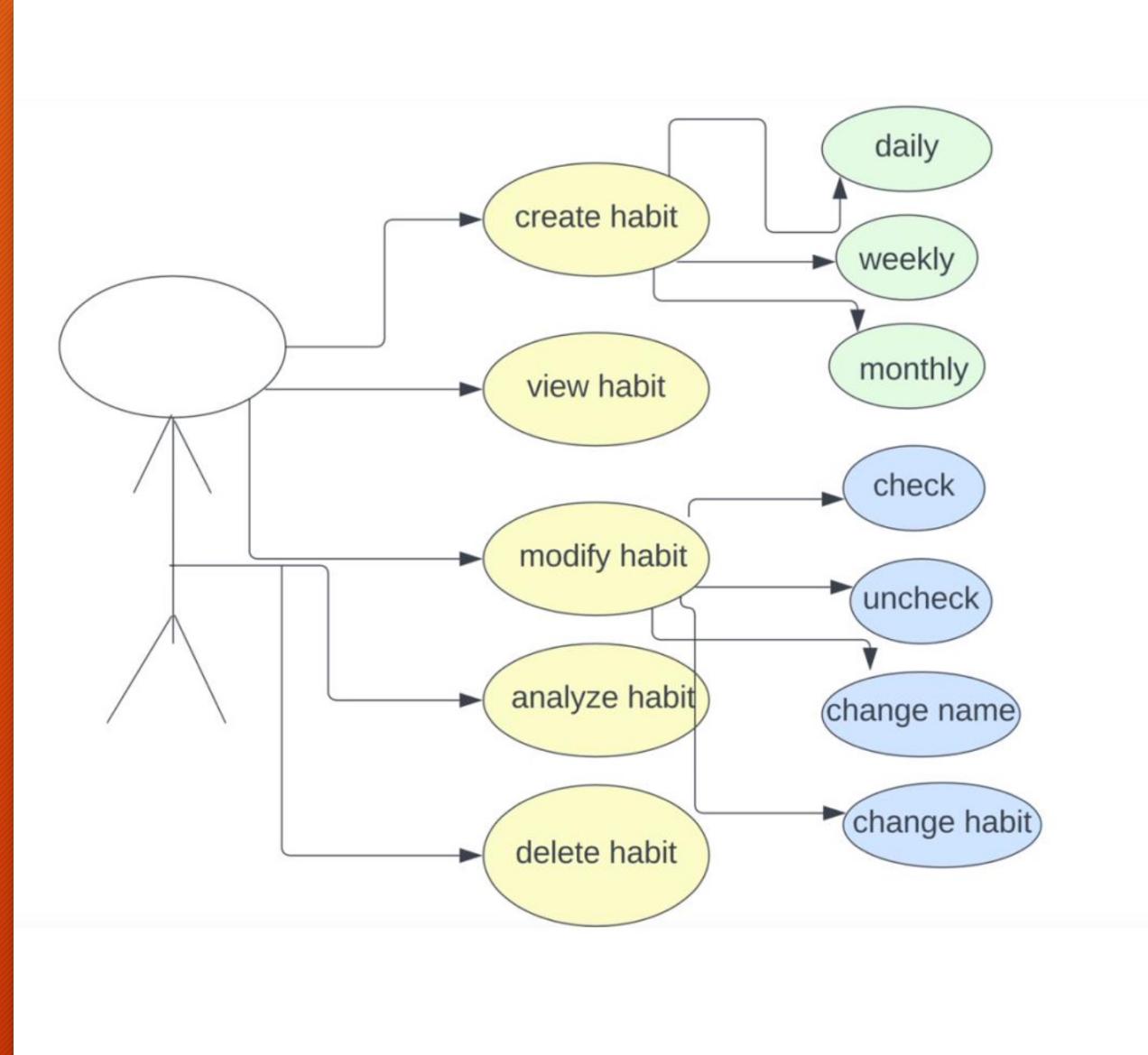
# selection at the end -add me
if ob.select= 1
    mirr_ob.select=1
    context.scene.objects.active = eval(
        ("Selected" + str(modifier)))
    mirror_ob.select = 0
    bpy.context.selected_objects.append(
        data.objects[one.name].select)
    print("please select exactly one object")
    return{'FINISHED'}
```

- OPERATOR CLASSES ---

```
types.Operator):
    X mirror to the selected
    object.mirror_mirror_x"
    or X"
context):
    ext.active_object is not
```

The Case Model of the Habit Tracker App:

- Shows how the user can use the app
- Asks the user to create, view, modify, analyze and delete habits.
- Shows how the user can perform actions and the path the user can choose in a brief way.



Main Menu Codes :

- In capture 1 # The menu will display the main menu options and will call the respective functions of that selected
 - # option by the user. however, if the user selects an invalid option error will be displayed.
 - # the menu function remains running till the 'QUIT' option is not selected by the user
-
- In capture 2 # this function is another menu to select the option between daily weekly and monthly
 - # we get to select the type of habit we want to create and finally pass the type as a parameter to
 - # other function that creates that type

Capture 1

```
def menu():
    while True:
        print("Select an Option:\n \n1.Create Habit")
        print("2.View Habit")
        print("3.Modify Habit")
        print("4.Analyze Habit")
        print("5.Delete Habit")
        print("6.Exit")
        try:
            x=int(input("Enter Your Choice... "))
        except:
            print("Invalid Input, Try Again")
        if x==1:
            create_habit_menu()
        elif x==2:
            view_habit()
        elif x==3:
            modify_habit()
        elif x==4:
            analyze_habit()
        elif x==5:
            delete_habit()
        elif x==6:
            print('Quiting...')
            break
```

Capture 2

```
def create_habit_menu():
    print("Select an Option:\n \n1.Daily")
    print("2.Weekly")
    print("3.Monthly")
    try:
        x=int(input("Enter Your Choice... "))
    except:
        print("Invalid Input, Try Again")
    if x==1:
        create_habit('Daily')
    elif x==2:
        create_habit('Weekly')
    elif x==3:
        create_habit('Monthly')
```

Create Habit

- # takes type as a parameter and asks user to fill the other required information and finally saves it in the
- # file by using file handling. it also displays the confirmation notification and also displays error if no such
- # type exists

```
def create_habit(types):  
    f=open('tasks.txt', 'a')  
    if types=='Daily':  
        x=input("Please enter the name of the task: ")  
        y=input("Please enter starting date in YYYY-MM-DD format: ")  
        z=input("Please enter ending date in YYYY-MM-DD format: ")  
        w=input("Please enter Daily Period: ")  
        li=x+" "+"Daily"+" "+y+" "+z+" "+str(0)+' '+w+"\n"  
        f.write(li)  
        print("Task Created Successfully")  
    elif types=='Weekly':  
        x=input("Please enter the name of the task: ")  
        y=input("Please enter starting date in YYYY-MM-DD format: ")  
        z=input("Please enter ending date in YYYY-MM-DD format: ")  
        w=input("Please enter Weekly Period: ")  
        li=x+" "+"Weekly"+" "+y+" "+z+" "+str(0)+' '+w+"\n"  
        f.write(li)  
        print("Task Created Successfully")  
  
    elif types=='Monthly':  
        x=input("Please enter the name of the task: ")  
        y=input("Please enter starting date in YYYY-MM-DD format: ")  
        z=input("Please enter ending date in YYYY-MM-DD format: ")  
        w=input("Please enter Monthly Period: ")  
        li=x+" "+"Monthly"+" "+y+" "+z+" "+str(0)+' '+w+"\n"  
        f.write(li)  
        print("Task Created Successfully")  
    else:  
        print("Invalid Type! ")
```

Viewing habits

- # this functions helps in viewing the habits that the user has previously created and in a simple manner for user to
- # understand it also displays all the related data to that habit and whether it is checked or unchecked

```
def view_habit():
    f=open("tasks.txt",'r')
    while True:
        x=f.readline()
        if x=='':
            break
        else:
            x=x.split(' ')
            print("\n")
            print("Habit Name:",x[0])
            print("Habit Type:",x[1])
            print("Starting Date:",x[2])
            print("Ending Date:",x[3])
            if int(x[4])>0:
                print("Status: Task Checked with",x[4]," Streak")
            else:
                print("Status: Task Unchecked")
            x[5]=x[5].split('\n')
            print("Period: ",x[5][0])
            print("\n")
```

Modifying Habits

- # Modify habit provides user options to modify various functionalities or check or uncheck a task
- # when the user makes their choice, this function calls the mark habit function and passes its selection as a parameter to it

```
def modify_habit():
    y=[]
    f=open("tasks.txt",'r')
    print("Select an Option to Modify:\n \n1.Check Task")
    print("2.Uncheck Task")
    print("3.Modify Habit Name")
    try:
        sel=int(input("Your Choice... "))
    except:
        print("Invalid Choice, Returning to Main Menu\n")
        return
    if sel==1:
        mark_habit('Check')
    elif sel==2:
        mark_habit('Uncheck')
    elif sel==3:
        l=input("Enter New Habit Name: ")
        mark_habit(l)
    else:
        print("Invalid Choice, Returning to Main Menu\n")
        return
    while True:
        x=f.readline()
        if x=='':
            break
        y.append(x)
```

Features Added:

- (capture 1) # this function is used to open the file and then search for the name of the habit the user wants to edit and finally
- # changes the habits data according to the user requirement provided to it as a parameter
- # at the end saves the updated data in the file again for viewing and further uses
- (capture2) # this function asks user for the name of the habit and then deletes that data as whole from the file

Capture 1

```
def mark_habit(x):  
    y=[]  
    f=open("tasks.txt",'r')  
    i=input("Enter the name of the habit to Modify: ")  
    while True:  
        o=f.readline()  
        if o=='':  
            f2=open("tasks.txt",'w')  
            for j in y:  
                f2.write(j)  
            print("Modification Complete!")  
            return  
        o=o.split(' ')  
        if o[0]==i:  
            if x=='Check':  
                o[4]=int(o[4])+1  
                o[4]=str(o[4])  
            elif x=='Uncheck':  
                o[4]=str(0)  
            else:  
                o[0]=x  
        li=o[0]+ " "+o[1]+ " "+o[2]+ " "+o[3]+ " "+o[4]+ " "+o[5]  
        y.append(li)
```

Capture 2

```
def delete_habit():  
    y=[]  
    f=open("tasks.txt",'r')  
    i=input("Enter the name of the habit to Delete: ")  
    while True:  
        o=f.readline()  
        if o=='':  
            f2=open("tasks.txt",'w')  
            for j in y:  
                f2.write(j) %  
            print("Deletion Complete!")  
            return  
        o=o.split(' ')  
        if o[0]!=i:  
            li=o[0]+ " "+o[1]+ " "+o[2]+ " "+o[3]+ " "+o[4]+ " "+o[5]  
            y.append(li)
```

Analyze Habits:

- # this function is used to analyze the habits and define if they are due or days remaining to their due
- # from the current date timestamp, it also displays other important functionalities that
- # helps in defining and analyzing the habits to the user.

```
def analyze_habit():
    today = date.today()
    today.strftime("%y-%m-%d")
    print("\n\n")
    print("Today's Date is: ",today)
    f=open("tasks.txt",'r')
    while True:
        x=f.readline()
        if x=='':
            break
        else:
            x=x.split(' ')
            print("\n")
            print("Habit Name:",x[0])
            print("Habit Type:",x[1])
            if int(x[4])>0:
                print("Status: Task Checked with",x[4]," Streak")
            else:
                print("Status: Task Unchecked")
            xx=datetime.datetime.strptime(today,"%y-%m-%d")
            yy=datetime.datetime.strptime(x[3],"%Y-%m-%d")
            d=yy-xx
            if int(d.days)>0:
                print("Ending in: ",d.days,' day(s)')
            else:
                print("Was due ",abs(d.days)," day(s) ago")
            x[5]=x[5].split('\n')
    print("\n\n")
```

Visualization of Streak Action:

```
1 Select an Option:  
2  
3 1.Create Habit  
4 2.View Habit  
5 3.Modify Habit  
6 4.Analyze Habit  
7 5.Delete Habit  
8 6.Exit  
9 Enter Your Choice... 4  
10  
11  
12 Today's Date is: 22-08-20  
13  
14  
15  
16 Habit Name: brushing  
17 Habit Type: Daily  
18 Status: Task Checked with 4 Streak  
19 Ending in: 20 day(s)  
20  
21  
22 Habit Name: bathing  
23 Habit Type: Daily  
24 Status: Task Checked with 2 Streak  
25 Ending in: 20 day(s)  
26  
27  
28 Habit Name: sleeping  
29 Habit Type: Daily  
30 Status: Task Unchecked  
31 Was due 2 day(s) ago  
32  
33  
34  
35 Select an Option:  
36  
37 1.Create Habit  
38 2.View Habit  
39 3.Modify Habit  
40 4.Analyze Habit  
41 5.Delete Habit  
42 6.Exit  
43 Enter Your Choice... 6  
44 Quiting...  
45 |
```