



Projecte-Python

Documentació i desenvolupament



Grup-9

Modul Optatiu - Introducció a la Programació amb Python
Professors: Adrian Cardenas

Cicle formatiu grau mitjà

Sistemes microinformàtics i xarxes

Curs: 2024-2025



Índex

Documentació i desenvolupament.....	2
Introducció	2
Importació i lectura del CSV	2
Incorporació del menú	4
Incorporació de la funció, facturació mensual	6
Incorporació de “Mostrar estoc”	9
Incorporació de “mostrar_3_millors”	10
Testejos	13

Documentació i desenvolupament

Introducció

Com es demana a la pràctica, hem a crear una documentació del desenvolupament del programa per conèixer cada funció i els processos que té pel seu funcionament.

Importació i lectura del CSV

Per començar el programa, el primer que farem és importar al codi la possibilitat de llegir les extensions “.CSV” per llegir la llista proporcionada.

Utilitzem les dues primeres línies per comentar l'autor i la descripció de la part del codi i a la tercera línia introduïm:

```
import csv
```

Importem la possibilitat de llegir extensions “.CSV” com hem mencionat abans.

```
# Autor - Miguel Rico
# Importem CSV (Per llegir els arxius CSV)
import csv
```

Continuem amb la creació d'una funció que sigui possible aquesta lectura.

Definim la funció:

```
def llegir_dades_botiga(nom_csv):, "nom_csv" l'utilitzem de paràmetre
```

Fem un diccionari amb blanc perquè quan faci la lectura puguem quedar-nos amb la informació llegida.

```
dades=[]
```



Ara codifiquem la part de la lectura, quan es demani que s'obri un arxiu .csv, configurem els paràmetres de com es llegirà tot el contingut, utilitzem aquest codi.

```
with open(nom_csv,mode='r', encoding='utf-8') as fitxer:  
    lector=csv.DictReader(fitxer)
```

Quan s'obri un arxiu .csv, té un mode de lectura amb una codificació de caràcters de utf-8, tota la informació es concentrarà en “fitxer”, després es crea una variable a on es farà tota aquesta lectura (lector=), com hem dit a “fitxers”.

A continuació crearem un bucle per anar afegint tota la informació llegida al diccionari creat anteriorment “dades” per si en algun futur volem imprimir algun resultat de la lectura.

```
for i in lector:  
    dades.append(i)
```

Utilitzem un for in a “lector”, perquè lector és la variable de fitxers on està tota la informació llegida, i donem al diccionari “dades” un “.append” amb un increment perquè escrigui tota la informació rebuda al diccionari.

Amb això acabem aquesta funció fen un “return” de dades per guardar tota la informació recol·lectada.

```
return dades
```

```
# Funcio que fa llegir el CSV
def llegir_dades_botiga(nom_csv):
    dades=[]
    with open(nom_csv,mode='r', encoding='utf-8') as fitxer:
        lector=csv.DictReader(fitxer)
        for i in lector:
            dades.append(i)
    return dades
```

Incorporació del menú

En aquesta part crearem les funcions per mostrar i donar-li “mecanismes”, primer mostrarem de manera gràfica el menú, perquè l'usuari tingui una orientació del que vol que sigui el següent pas donant-li unes opcions per elegir.

Creem de forma senzilla una funció anomenada “mostrar_menu” la definim d'aquesta forma:

```
def mostrar_menu():
```

Amb una sèrie de “prints” per imprimir el que seria gràficament les opcions que veurà l'usuari, representa-ho d'aquesta forma.

```
# Autor - Miguel Rico
# Funció que ens mostra el menu visualment
def mostrar_menu():
    print("====Benvingut a la APP de facturacio TechPlay====")
    print("1 - Calcular la facturacio mensual")
    print("2 - Stock disponible")
    print("3 - Tres productes mes facturats")
    print("4 - Sortir de l'APP")
```



Ara per donar funcionalitat i opcions que funcionin de veritat, creem una funció anomenada “menu”, com volem que es repeteixi diversos cops creem un bucle amb “while” i cridem a la funció “mostrar_menu” perquè ens imprimeixi el contingut que hem creat.

```
def menu():  
    while True:  
        mostrar_menu()
```

Després creem un input perquè l'usuari pugui seleccionar una opció, en aquest pas és on crearem totes les opcions i depèn l'usuari demani, cridarà una funció o un altre, tot això amb una variable anomenada “opcio”

```
opcio=input("Tria una opció 1-4: ")
```

Fem un “match” de la variable “opcio” perquè depèn el que escrigui l'usuari faci una cosa o un altre, en aquest cas crearem 5 “case”, 4 “case” de l'1 al 4 i l'últim farem un “case _” que és l'equivalent a un “else”, perquè això no deixa de ser condicions. Per finalitzar aquest punt a l'última línia cridem la funció “menu”, perquè sigui la primera a iniciar el programa.

El resultat el mostrem en la imatge:

```
# Autor - Miguel Rico  
# Funció demana mostrar el "mostrar_menu" i dona la possibilitat de elegir una opció del 1-4  
def menu():  
    while True:  
        mostrar_menu()  
        opcio=input("Tria una opció 1-4: ")  
        match opcio:  
            case "1":  
                print("En construccio")  
            case "2":  
                print("En construccio")  
            case "3":  
                print("En construccio")  
            case "4":  
                print("Sortin del programa, fins un altre!")  
                break  
            case _:  
                print("Opció invalida, siusplau seleccioni del 1-4")  
menu()
```

Cicle formatiu grau mitjà

Sistemes microinformàtics i xarxes

Curs: 2024-2025



Nota: Totes les opcions de l'1-3 estan en construcció perquè no hi ha funcions a cridar, si volem provar el programa correctament sense les funcions encara no les podem cridar.

Incorporació de la funció, facturació mensual

Definim la funció amb el nom “calcular_facturacio”.

```
def calcular_facturacio()
```

Preparem els increments total, amb IVA o sense, perquè quan faci el càlcul, comenci des de 0, els representem d'aquesta forma:

```
total_sense_iva=0
```

```
total_amb_iva=0
```

Cridem a la funció de lectura de CSV

```
nom_arxiu_csv = "dades_botiga.csv"
```

```
lector=llegir_dades_botiga(nom_arxiu_csv)
```

Amb això el que fem es cridar la funció ja creada que el que fa es llegir el contingut csv, i li donem un arxiu que es “dades_botiga.csv”.

Creem un bucle, perquè llenguixi els camps que volem comprovar amb “for i in” dins del CSV, que son “Quantitat_Venuda”, “Preu_unitari” i “IVA”

```
for i in lector:
```

```
    quantitat=int(i["Quantitat_Venuda"])
```

Cicle formatiu grau mitjà

Sistemes microinformàtics i xarxes

Curs: 2024-2025



```
preu_unitari = float(i["Preu_Unitari"])
```

```
iva_percent = float(i["IVA"]) / 100
```

Fem les operacions aritmètiques per fer el subtotal, el càlcul de l'IVA, i el resultat total incloïen l'IVA i el subtotal.

```
subtotal = quantitat * preu_unitari
```

```
iva=subtotal * iva_percent
```

```
total=subtotal + iva
```

Ara crearem l'increment perquè se sumi totes les files i ens doni el total de tota la taula. Farem els increments amb IVA i sense.

```
total_sense_iva += subtotal
```

```
total_amb_iva += total
```

Abans d'imprimir els resultats, creem una variable amb un "round", això ho fem perquè ens doni 2 decimals en comptes de molts.

```
round_sense_iva=round(total_sense_iva,2)
```

```
round_amb_iva=round(total_amb_iva,2)
```

Ara imprimim els resultats, amb un títol per identificar els resultats d'on venen.

```
print("\n===Facturació mensual===")
```

```
print("Aquest mes s'ha facturat sense IVA:", round_sense_iva,  
"€")
```


Cicle formatiu grau mitjà

Sistemes microinformàtics i xarxes

Curs: 2024-2025



```
print("Total amb IVA", round_amb_iva, "€")
```

Seria representat d'aquesta forma amb el codi totalment.

```
def calcular_facturacio():
    #Preparem el increment del valor perquè crearem un "for", el valor començarà des de 0
    total_sense_iva=0
    total_amb_iva=0
    #Llegim la funció del csv, per llegir el arxiu
    nom_arxiu_csv = "dades_botiga.csv"
    lector=llegir_dades_botiga(nom_arxiu_csv)
    #Començem el bucle perquè faci la operació i sumi els valors introduïts al inici
    for i in lector:
        quantitat=int(i["Quantitat_Venuda"])
        preu_unitari = float(i["Preu_Unitari"])
        iva_percent = float(i["IVA"]) / 100
        subtotal = quantitat * preu_unitari
        iva=subtotal * iva_percent
        total=subtotal + iva
        #Fem el increment desde 0 amb subtotal i el total
        total_sense_iva += subtotal
        total_amb_iva += total
        round_sense_iva=round(total_sense_iva,2)
        round_amb_iva=round(total_amb_iva,2)
    #Mostra un títol i el valor total amb un "round" per a per arrodonir la xifra
    print("\n===Facturació mensual===")
    print("Aquest mes s'ha facturat sense IVA:", round_sense_iva, "€")
    print("Total amb IVA", round_amb_iva, "€")
```

Un cop acabat tot això, a l'opció del menú, cridem a la funció, ja la tenim llesta perquè funcioni.

```
# Autor - Miguel Rico
# Funció demana mostrar el "mostrar_menu" i dona la posib
def menu():
    while True:
        mostrar_menu()
        opcio=input("Tria una opció 1-4: ")
        match opcio:
            case "1":
                calcular_facturacio()
            case "2":
```

Incorporació de “Mostrar estoc”

Començem la definició de la funció anomenada “mostrar_stock”

```
def mostrar_stock():
```

Creem una tupla per guardar tota la informació per si volem fer alguna consulta.

```
stockd={}
```

Cridem a la funció de lectura del CSV.

```
nom_arxiu_csv = "dades_botiga.csv"
```

```
lector=lleguir_dades_botiga(nom_arxiu_csv)
```

Fem un bucle per crear un increment per consultar els camps “Producte” i “Estoc_Disponible” de l'arxiu CSV.

```
for i in lector:
```

```
    producte = i ["Producte"]
```

```
    stock = int(i["Estoc_Disponible"])
```

Creem una condició per si hi ha estoc en el producte, comença un increment, si no, no es fa l'increment.

```
if producte in stockd:
```

```
    stockd[producte] += stock
```

```
else:
```

```
    stockd[producte] =stock
```



Per acabar la funció imprimim els resultats, però creem un petit bucle perquè escrigui tot els productes amb les seves unitats d'estoc amb un títol per saber que s'ha cercat.

```
print("\n--- Stock Disponible per a cada producte ---\n")
for producte, stock in stockd.items():
    print(producte, stock, "unitats")
```

Amb tota la codificació quedaria d'aquesta forma.

```
def mostrar_stock():
    stockd= {}
    nom_arxiu_csv = "dades_botiga.csv"
    lector=llegir_dades_botiga(nom_arxiu_csv)
    for i in lector:
        producte = i ["Producte"]
        stock = int(i["Estoc_Disponible"])
        if producte in stockd:
            stockd[producte] += stock
        else:
            stockd[producte] =stock

    print("\n--- Stock Disponible per a cada producte ---\n")
    for producte, stock in stockd.items():
        print(producte, stock, "unitats")
```

Incorporació de “mostrar_3_millors”

Definim la funció per començar i creem una tupla per guardar tota la informació

```
def mostrar_3_millors():
    producte_facturacio = {}
```

Cicle formatiu grau mitjà

Sistemes microinformàtics i xarxes

Curs: 2024-2025



Cridem a la funció de llegir el CSV.

```
nom_arxiu_csv = "dades_botiga.csv"
```

```
lector = llegir_dades_botiga(nom_arxiu_csv)
```

Fem un bucle per llegir els camps "Producte",
"Quantitat_Venuda", "Preu_Unitari".

```
for i in lector:
```

```
    producte = i["Producte"]
```

```
    quantitat = int(i["Quantitat_Venuda"])
```

```
    preu_unitari = float(i["Preu_Unitari"])
```

Fem els càlculs per calcular les facturacions i les condicions
per fer els increments sí és necessari.

```
facturacio = quantitat * preu_unitari
```

```
    if producte in producte_facturacio:
```

```
        producte_facturacio[producte] += facturacio
```

```
    else:
```

```
        producte_facturacio[producte] = facturacio
```

Fem la codificació perquè ens registri els tres millors
productes amb facturació calculada. Creem un diccionari per
registrar els productes amb la seva facturació total, fem un
ordre descendent i una variable amb una posició de les 3
primeres així fent una classificació.

```
tres_millors = []
```

```
for producte, facturacio in producte_facturacio.items():
```

```
tres_millors.append((producte, facturacio))

tres_millors.sort(key=lambda x: x[1], reverse=True)

tres_millors = tres_millors[:3]
```

Imprimim el contingut amb un petit bucle perquè faci un increment dels càlculs i pugui trobar els productes amb més volum facturat.

Per acabar, li cridem la funció dins del “case 3:” del “match”.

```
def mostrar_3_millors():
    # Diccionari per acumular la facturació per cada producte
    producte_facturacio = {}
    nom_arxiu_csv = "dades_botiga.csv"
    lector = llegir_dades_botiga(nom_arxiu_csv)
    for i in lector:
        producte = i["Producte"]
        quantitat = int(i["Quantitat_Venuda"])
        preu_unitari = float(i["Preu_Unitari"])
        # Calcular la facturació per aquesta línia
        facturacio = quantitat * preu_unitari
        if producte in producte_facturacio:
            producte_facturacio[producte] += facturacio
        else:
            producte_facturacio[producte] = facturacio
    # Inicialitzar una llista per als tres millors productes
    tres_millors = []
    # Cerca el diccionari per trobar els tres productes amb més facturació
    for producte, facturacio in producte_facturacio.items():
        # Afegir el producte actual a la llista
        tres_millors.append((producte, facturacio))
    # Ordenar la llista en ordre descendent segons la facturació
    tres_millors.sort(key=lambda x: x[1], reverse=True)
    # Mantenir només els tres primers elements a la llista
    tres_millors = tres_millors[:3]

    print("\n--- Tres productes amb més facturació ---")
    for producte, facturacio in tres_millors:
        print(producte, facturacio, "€")
```

```
case "3":
    mostrar_3_millors()
case "4":
```

Cicle formatiu grau mitjà

Sistemes microinformàtics i xarxes

Curs: 2024-2025



Testejos

Provem la primera funció:

```
PS C:\Users\MaikR\Desktop\Practicas Insti\ABP\ABP Projecte> & C:/Users/MaikR/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/MaikR/Desktop/Practicas I
nsti/ABP/ABP Projecte/botiga.py"
====Benvingut a la APP de facturacio TechPlay====
1 - Calcular la facturacio mensual
2 - Stock disponible
3 - Tres productes mes facturats
4 - Sortir de l'APP
Tria una opció 1-4: 1

===Facturació mensual===
Aquest mes s'ha facturat sense IVA: 233435.0 €
Total amb IVA 282456.35 €
====Benvingut a la APP de facturacio TechPlay====
1 - Calcular la facturacio mensual
2 - Stock disponible
3 - Tres productes mes facturats
4 - Sortir de l'APP
Tria una opció 1-4: 
```

Provem la segona funció:

```
====Benvingut a la APP de facturacio TechPlay====
1 - Calcular la facturacio mensual
2 - Stock disponible
3 - Tres productes mes facturats
4 - Sortir de l'APP
Tria una opció 1-4: 2

--- Stock Disponible per a cada producte ---

Minecraft 190 unitats
Bateria externa Aukey 504 unitats
Super Mario Odyssey 474 unitats
Càmera de seguretat Arlo Pro 98 unitats
Assassin's Creed Valhalla 543 unitats
Mario Kart 8 Deluxe 12 unitats
Diablo IV 365 unitats
Llums LED Philips Hue 246 unitats
Hogwarts Legacy 205 unitats
Valorant 176 unitats
Volant Logitech G29 396 unitats
Splatoon 3 252 unitats
Powerbank Xiaomi 20000mAh 212 unitats
Auriculars Logitech G Pro 299 unitats
Animal Crossing: New Horizons 78 unitats
GTA V 144 unitats
Comandament DualSense PS5 171 unitats
Cyberpunk 2077 284 unitats
Adaptador USB Wi-Fi TP-Link 157 unitats
Protectors de pantalla per a Switch 243 unitats
Comandament Xbox Series X 227 unitats
```

Per acabar, fem l'últim test de la tercera funció:

Cicle formatiu grau mitjà

Sistemes microinformàtics i xarxes

Curs: 2024-2025



```
PS C:\Users\MaikR\Desktop\Practicas Insti\ABP\ABP Projecte> & C:/U:
nsti/ABP/ABP Projecte/botiga.py"
====Benvingut a la APP de facturacio TechPlay====
1 - Calcular la facturacio mensual
2 - Stock disponible
3 - Tres productes mes facturats
4 - Sortir de l'APP
Tria una opció 1-4: 3

--- Tres productes amb més facturació ---
Assassin's Creed Valhalla 12883.0 €
Cyberpunk 2077 10261.0 €
Bateria externa Aukey 9558.0 €
====Benvingut a la APP de facturacio TechPlay====
1 - Calcular la facturacio mensual
2 - Stock disponible
3 - Tres productes mes facturats
4 - Sortir de l'APP
Tria una opció 1-4: 
```