

Labs_p11

December 13, 2014

1 Pràctica 11: Aplicació amb menús

1.1 Objectiu

Aquesta pràctica té per objectiu practicar tots els continguts vistos en el curs:

- llenguatge **python**: variables, conjunts de dades, control de flux, fitxers, etc.
- mòduls **scipy**: càlcul simbòlic, representació gràfica, etc. fitxers, etc.

i introduir-nos en la creació de programes amb menús que permetin una interacció més complexa amb l'usuari.

IMPORTANT: En aquesta pràctica treballareu sobre exemples de codi que realitzen part de les accions que es demanen. Abans de començar a programar, dediqueu temps a ENTENDRE què fan aquests exemples, de manera que els pogueu modificar i adaptar a les vostres necessitats. Teniu tots els exemples disponibles al Campus Virtual.

1.2 Calculadora d'enters

Al final d'aquesta secció es proporciona el codi d'una calculadora d'enters que funciona amb menús.

El seu funcionament és simple:

- El programa disposa de dues variables on guardar informació: A i B.
- L'usuari només pot introduir dades a A.
- El programa realitza operacions entre els valors de A i B però sempre guarda el resultat en B.
- Si es vol modificar B, cal introduir una dada en A i passar-la a B.
- Totes les opcions són accessibles prement una tecla específica a través d'un menú, que es mostra una vegada i un altre.
- Es reserva la lletra "e" per aturar el menú i sortir del programa.

Comproveu que el programa funciona i que enteneu el codi que l'implementa.

```
In [1]: def valida_enter(minim, maxim, missatge):  
        """ demana un enter per teclat, amb un missatge personalitzar,  
            verificant que sigui correcte i estigui comprès  
            entre un valor mínim i un màxim"""  
        while(True):  
            text = input("%s [%i,%i]: " % (missatge, minim, maxim))  
            try:  
                num = int(float(text))  
                if(minim<=num<=maxim):  
                    return num  
            else:  
                print("    Fora de rang")  
        except ValueError:
```

```

        print("    Saltem Valor")
        return None

# Accions del menú
def esborra():
    return 0

def crea():
    variable = valida_enter(-1000,1000,">Introdueix el valor de A")
    return variable

def intercambia(x,y):
    return (y,x)

def signe(x):
    return -x

def suma(x,y):
    return x+y

def resta(x,y):
    return x-y

def producte(x,y):
    return x*y

def divisio(x,y):
    return x/y

def residu(x,y):
    return x%y
def potencia(x,y):
    return x**y

# Control del menú
def menu(var1,var2):
    """ Presenta el menú i els resultats del programa,
    demana una i crida la funció corresponent
    fins que es tria "e" per sortir"""
    comanda = ""
    while(comanda != "e"):
        imprimeix_menu(var1,var2)
        comanda = input(">Que vols fer? ")
        if comanda == "e":
            print("    Sortim!")
        elif comanda == "0":
            var1 = esborra()
        elif comanda == "1":
            var1 = crea()
        elif comanda == "2":
            (var1,var2) = intercambia(var1,var2)
        elif comanda == "3":
            var2 = signe(var1)

```

```

elif comanda == "4":
    var2 = suma(var1,var2)
elif comanda == "5":
    var2 = resta(var1,var2)
elif comanda == "6":
    var2 = producte(var1,var2)
elif comanda == "7":
    var2 = divisio(var1,var2)
elif comanda == "8":
    var2 = residu(var1,var2)
elif comanda == "9":
    var2 = potencia(var1,var2)
else:
    comanda = ""
    print("    Opcio erronea!")
return (var1,var2)

# Impressió de menú i resultats actuals
def imprimeix_menu(var1, var2):
    print("##### CALCULADORA NÚMEROS #####")
    print("0 - esborra A")
    print("1 - crea A")
    print("2 - A <-> B")
    print("3 - B = -A")
    print("4 - B = A + B")
    print("5 - B = A - B")
    print("6 - B = A * B")
    print("7 - B = A / B")
    print("8 - B = A % B")
    print("9 - B = A ** B")
    print("e - SURT")
    print("#####")
    print("  A = ", var1)
    print("  B = ", var2)
    print("#####")

## Aquí comença el programa principal
(A,B) = (0,0) #creem les variables
menu(A,B)     #cridem al control del menú

##### CALCULADORA NÚMEROS #####
0 - esborra A
1 - crea A
2 - A <-> B
3 - B = -A
4 - B = A + B
5 - B = A - B
6 - B = A * B
7 - B = A / B
8 - B = A % B
9 - B = A ** B
e - SURT
#####
A = 0

```

```

B = 0
#####
>Que vols fer? e
Sortim!

```

Out[1]: (0, 0)

Exercici 11.1:

- Entregueu el codi d'exemple de la calculadora d'enters **totalment comentat**; demostrant que **enteneu perfectament** tot el que fa.

1.2.1 Còpia de dades a disc

Comprovareu que el programa anterior resulta incòmode si es vol usar diverses vegades: després de fer uns càlculs i tancar el programa, les dades es perden i s'han de recordar i introduir de nou. Per resoldre aquesta limitació farem que el programa sigui capaç de guardar el valor final de A i B a disc, abans de sortir; i de llegir-los de disc, abans de començar.

Afegiu les següents funcions al vostre programa. En aquest cas, farem servir les eines d'escriptura i lectura del mòdul json que permeten desar i recuperar objectes python a disc.

```

In [2]: def guarda_fitxer(x,y):
        import json
        with open('valors_backup.json', 'w') as sortida:
            json.dump({'x': x, 'y': y}, sortida, sort_keys = True)

        def llegeix_fitxer():
            import json
            try:
                with open('valors_backup.json', 'r') as entrada:
                    data = json.load(entrada)
                    x = data['x']
                    y = data['y']
            except (IOError, ValueError): # si no existeix el fitxer
                x = 0 # les variables valen 0
                y = 0
            return (x,y)

```

Per tal de fer-les servir, modifiqueu el programa principal tal i com s'indica a continuació:

```

In [3]: ## Nou programa principal amb lectura/escriptura a fitxer
        (A,B) = llegeix_fitxer() # llegim les variables de fitxer
        (A,B) = menu(A,B)       # executem el menú, quan acaba, retorna darrers valors de A i B
        guarda_fitxer(A,B)     # guardem a fitxer els darrers valors de les variables, abans de sort

##### CALCULADORA NÚMEROS #####
0 - esborra A
1 - crea A
2 - A <-> B
3 - B = -A
4 - B = A + B
5 - B = A - B
6 - B = A * B
7 - B = A / B
8 - B = A % B
9 - B = A ** B

```

```

e - SURT
#####
A = 0
B = 0
#####
>Que vols fer? e
Sortim!

```

Comproveu que el programa ara guarda automàticament els darrers valors de A i de B i que els recupera correctament quan el torneu a obrir. De nou, analitzeu el codi i enteneu que fa.

Exercici 11.2:

- Entregueu el codi d'exemple de la calculadora d'enters amb capacitat de còpia disc comentant els canvis efectuats; demostrant que **enteneu perfectament** tot el que fa.

1.3 Calculador de polinomis

Fem ara una programa, similar a l'anterior, però que permeti efectuar càlculs amb polinomis **sympy**.

El programa ha de ser capaç de treballar amb expressions polinòmiques **sympy** fins a *grau 20*.

En aquest cas, el programa també treballarà amb menús i dos objectes **sympy** on guardar els polinomis: P i Q.

- L'usuari només pot introduir nous polinomis en P.
- El programa realitza operacions entre els polinomis de P i Q però sempre guarda el resultat en Q.
- Si es vol modificar Q, cal introduir un polinomi en P i passar-lo a Q.
- Totes les opcions són accessibles prement una tecla específica a través d'un menú, que es mostra una vegada i un altre.
- Es reserva la lletra "e" per aturar el menú i sortir del programa.

Oberveu que, ara, l'entrada de dades és una mica més complexa: cal demanar primer el grau del polinomi i després, demanar els coeficients necessaris.

Per simplicitat, considerarem només coeficients enters $\in [-1000, 1000]$.

Volem també, que si no s'especifica un coeficient, se li assigni el valor simbòlic a_i .

Us facilitem la funció `crea_polinomi()` que fa tot això. Analitzeu-la amb detall fins entendre el seu funcionament.

```

In [4]: def crea_polinomi():
import sympy
grau = None
while(grau == None):
    grau = valida_enter(0,20,"Grau del Polinomi")
sympy.var("x")
coeficients = sympy.symbols("a0:%i" % (grau+1))
poli = sympy.sympify("0")
for n,a in enumerate(coeficients):
    poli = poli + a*x**n
for n,a in enumerate(coeficients):
    valor_coef = valida_enter(-1000, +1000, ">Introdueix coeficient a_%i" % n)
    if (valor_coef != None):
        poli = poli.subs(a,valor_coef)
return poli

```

Exercici 11.3:

- Entregueu el codi d'exemple de la funció `crea_polinomi()` totalment comentat; demostrant que **enteneu perfectament** tot el que fa.

1.3.1 Menu i opcions

Basant-vos en l'exemple de la calculadora d'enters, creeu el menú i les opcions necessàries per que la calculadora de polinomis sigui capaç de:

- Crear $P(x)$
- Esborrar $P(x)$
- Per consistència amb `sympy` utilitzeu l'expressió zero: `P = sympy.sympify("0")`
- Avaluar $P(a)$
- Demaneu un valor enter a i feu que $Q(x) = P(a)$. Si no s'introdueix a , el programa li assigna un valor simbòlic i segueix fent el càlcul.
- Resoldre $P(x) = 0$
- En aquest cas, imprimir les arrels del polinomi, no cal desar res en $Q(x)$
- Intercambiar $P(x) \leftrightarrow Q(x)$
- Sumar $Q(x) = P(x) + Q(x)$
- Restar $Q(x) = P(x) - Q(x)$
- Multiplicar $Q(x) = P(x) * Q(x)$
- Derivar $Q(x) = \frac{dP(x)}{dx}$
- Integrar $Q(x) = \int P(x)dx$
- Integrar amb extrems definits $Q(x) = \int_c^d P(x)dx$
- Demaneu valors enters per c i d , si no s'introdueixen, assigneu-los de forma simbòlica i prosseguiu amb el càlcul.
- Representar gràficament $P(x)$ per $x \in [e, f]$:
- Demaneu valors enters per e i f ; si no s'introdueixen, feu la representació gràfica utilitzant dos límits per defecte.
- Podeu fer servir la funció: `sympy.plot(exp, (var, lim_min, lim_max))` per representar l'expressió `exp`, en funció de la variable `var` entre els límits `lim_min` i `lim_max`.
- Compte! Representeu només polinomis que tinguin tots els coeficients numèrics!!
- Per simplicitat, en aquest exercici NO CAL GUARDAR DADES DISC.

Exercici 11.4:

- Implementeu una calculadora de polinòmis amb totes les opcions demanades.

1.3.2 Còpia de polinomis a disc:

Basant-vos en l'exemple de la calculadora d'enters amb còpia de dades a disc, milloreu la calculadora de polinomis per que també tingui aquesta opció.

Exercici 11.5:

- Afegiu a la calculadora de polinomis la capacitat de desar automaticament el resultat a disc quan finalitza, i de recuperar-lo quan es torni a obrir el programa.

Nota: json no sap gestionar objectes sympy, però podeu guardar la representació en string del polinomi fent servir `str` i recuperar el polinomi d'una cadena fent servir `sympify`.

1.4 Exercici a entregar

Construiu un Notebook que resolgui, amb les explicacions i comentaris adequats, els exercicis anteriors.

En acabar, entregueu el fitxer Notebook generat a través de [Campus Virtual](#)