Introducció a la Programació
Grau en Estadística

Sessió1: Introducció i condicionals

Vicenç Soler

Índex

1. Introducció

Un dels objectius primordials d'aquesta assignatura és aprendre a fer un programa informàtic.

Per a fer-ho, podem tenir 2 possibilitats: o bé fem el programa directament en codi màquina (missió quasi impossible) o bé fem servir un dels llenguatges que tenim disponibles per a què els humans puguem programar màquines. Un dels més estesos és el Python, i aquest és el que es farà servir en aquesta assignatura.

Els llenguatges de programació com el **Python** permeten que un humà pugui desenvolupar un programa informàtic molt més ràpidament que en llenguatge màquina, ja que permeten que el puguem expressar d'una manera molt més intel·ligible.

Fins aquí, només ens queda saber un pas: com es transforma el que expressem en llenguatge Python per a què es transformi en codi màquina. Aquest pas el fa una eina anomenada **compilador**, que llegeix el que hem expressat en Python i el transforma a codi màquina. Per tant, cada llenguatge de programació tindrà el seu propi compilador, i aquest compilador traduirà el programa nostre escrit en Python en un programa executable per la màquina.

Quan escrivim un programa en Python i el volem compilar per executar-lo, el compilador el que primer mira és si entén el que hem escrit, repassant la sintaxi utilitzada amb la que ell entén. Si hi ha algun error, ens ho dirà i executarà el nostre programa. Si no en troba cap, l'executarà i farà el que hem expressat en el programa. Aquesta darrera frase és molt important, ja que el nostre programa farà el que li hem dit que faci, que potser no coincideix amb el que volem que faci, ja que potser ens hem equivocat a l'expressar-ho. En aquest punt haurem de repassar el que hem escrit, rectificar-ho i tornar-lo a compilar per a tornar a crear el programa executable.

De Python existeixen 2 versions: la versió 2 (o versió 2.7) i la versió 3. Ambdues difereixen una mica en la sintaxi d'algunes funcions, operadors, etc. i, en aquesta assignatura, farem servir la versió 3, ja que és més nova.

2. Estructura del Programa

En Python qualsevol programa que fem ha de tenir la següent estructura:

- Llistat d'imports de llibreries
- Definició de les nostres funcions
- Programa principal

Les dues primeres són opcionals, depenent si les necessitem pel nostre programa.

Notes sobre Python:

- Cadascuna de les instruccions que s'apliquin han d'estar escrites en una línia. Si volem posar 2 instruccions a una mateixa línia, les separarem amb el símbol ":" , així el compilador sap com estan delimitades les instruccions (una cada línia, o separades per ":").
- "print" és una funció i al posar el seu nom es crida per a què faci la seva escomesa, que és mostrar informació per pantalla. A diferència de R i altres sistemes, només es mostra informació per pantalla si ho fem a traves de la funció "print". Els paràmetres de la funció es passen entre parèntesi (en aquest cas el missatge, entre ").
- En Python les funcions poden o no retornar un valor. Un exemple de funció que ja està implementada al sistema i que retorna un valor seria la funció "sqrt" o "cos", que permeten calcular l'arrel quadrada i el cosinus d'un valor. Per altra banda, un exemple de funció que també està implementada al sistema i que no retorna cap valor és "print".
- El Python distingeix entre majúscules i minúscules, així que, per exemple, si enlloc d'escriure "print" escrivim "Print" no sap el que és i ens donarà error en aquella línia. Això val per a qualsevol instrucció del programa.

3. Comencem a programar

Un programa és una sèrie d'ordres que s'executen seqüencialment: una darrere l'altra.

Quan s'implementa un programa es necessita anar guardant valors en memòria, ja sigui resultats intermitjos, dades inicials (p.ex. dades d'una matriu), resultats finals o altres càlculs. Aquest emmagatzematge es fa via les variables.

Una **variable** correspon a un tros de memòria que identifiquem amb un nom (que li posem nosaltres els programadors) i que depèn del tipus de dada que guardarà, es reservarà més o menys memòria per a emmagatzemar-la. El **nom d'una variable** vàlid pot ser qualsevol combinació de nombres i lletres, sempre que comenci per una lletra. Així, noms vàlids poden ser "numero, gran1, var23e56" etc., i invàlids serien "14, 1numero, 1e3, etc.". A més, com el

Python distingeix entre majúscules i minúscules, no és la mateixa variable una que es digui "numero" i una altra "Numero".

Cada variable guarda un tipus de dada diferent, i això es defineix en la primera vegada que se li assigna un valor. Per exemple, si fem x=3, la variable "x" es defineix com que guarda enters (i això pot tenir algunes implicacions en càlculs que puguem fer amb aquesta variable), encara que és possible que en el futur pugui guardar nombres amb decimals. Per exemple, si després fem "x=2.3" la variable "x" passa ara a guardar nombres amb decimals i, per tant, ha canviat el tipus de dada que guarda.

Per exemple, en Python se'ns permet de fer:

```
x=1   #x és entera
print(x)

x=1.2 #x ara canvia a nombre amb decimals
print(x)

x="hola"     #ara x ja no guarda un nombre, sino un texte
x=x+1  #ERROR: com x ara és un texte, no se'ns permetrà sumar-lo.
print(x)

x="3"  #ara a 'x' hem guardat 3 com a texte
x=x+1  #ERROR: com x encara és un texte, no puc sumar-lo.
print(x)

x=4.3 #ara torna a ser un nombre
x=x-1 #i ara ja puc fer-li una operación aritmètica de nou
print(x)
```

Un petit programa que demana un nombre per teclat, calcula el seu doble i, finalment, el mostra per pantalla seria:

```
1.  numero=int(input("Entra un nombre:"))
2.  doble = numero * 2
3.  print("El seu doble és: ",doble)
```

Nota: els nombres que apareixen a la esquerra de cada línia, en color blau, fan referència al nombre de línia i en cap cas formen part del programa. Els col·loquem per a poder referir-nos en les explicacions a una línia concreta.

Comentaris a aquest programa:

- Hem posat un nom eloqüent a les variables: en lloc d'anomenar-les amb simples lletres x , y , z, etc. He preferit posar-li un nom que faciliti futures revisions, així ens enrecordarem amb més facilitat quina dada guardava cada variable.

- **input(...)** és la funció que permet a un usuari entrar una dada per teclat. Aquesta funció retorna la dada entrada en format texte (encara que entem un nombre).
- Per tant, necessitem convertir la dada entrada en un nombre amb el que operar. Això ho fem amb la funció **int(...)** . Si volguéssim entrar un nombre amb decimals faríem servir la funció **float(...)** .
- La línia número 2 calcula el doble del nombre entrat. En aquest cas, l'operador "=" assigna un resultat a una variable, i hi distingirem entre el que hi ha a l'esquerra i a la dreta del "=":
    - A la dreta hi haurà una expressió que el sistema avaluarà i calcularà el resultat. Si hem entrat el valor 3 a la variable **numero**, 3*2=6 => el resultat és 6.
    - A l'esquerra sempre hi haurà una variable, ja que estem dient on volem guardar el resultat.
    - Per tant, guardarem el valor 6 a la variable **doble**.
- A la línia 3 mostrarem el resultat per pantalla. La funció print accepta un nombre indeterminat de paràmetres, separats per "," , i va col·locant un paràmetre al costat de l'altre. El que hi ha entre "..." es mostrarà directament, i el que no, s'avaluarà el resultat i es mostrarà el resultat

    En aquest cas, mostraria per pantalla "El nombre és: 6".

En resum:

numero=int(input("Entra un nombre:"))	La funció **input(...)** demana una dada per teclat, que convertirà a un enter. Per tant, la dada que s'entri ha de ser un enter, i qualsevol altra dada donarà error quan fem la conversió amb la funció **int(...)**
doble = numero * 2	A **doble** guardarem el resultat d'agafar el valor guardat a la variable **numero** i multiplicar-lo per 2.
print("El seu doble és: ",doble)	Finalment, mostrarem per pantalla "El nombre és: ..."

4. Instruccions Condicionals

El tipus de programa que hem vist fins ara és un tipus de programa totalment lineal. Ara bé, podem incloure instruccions que ens permetin decidir què fer davant d'alguna condició.

4.1. Instrucció if

Aquest tipus d'instruccions s'anomenen instruccions condicionals i en Python tenim la instrucció **if**, que té la següent sintaxi:

If <condició> :
    BLOC CONDICIÓ CERTA

Aquesta instrucció executa les instruccions incloses a BLOC CONDICIÓ CERTA en cas que "condició" sigui certa, i és OBLIGATORI que la condició acabi amb el símbol ":", que té altres instruccions. Un exemple seria:

```
if  numero>9:
        print("El nombre és més gran que 9")
```

Tot el que s'executa si i només si la condició és certa ha d'estar indentat a la dreta, i amb la mateixa indentació (els mateixos espais per l'esquerra)

Només s'executa el print si numero>9. En cas que numero no fos >9, no executa el codi.

Si ara fem:

```
if  numero>9:
        print("El nombre és més gran que 9")   #2 línies que només s'executen si numero>9
        print("Ho sabies?")
print("Adéu")        #no pertany al "if" => sempre s'executa
```

Cas 1: Si "numero" és més gran que 9, mostraria per pantalla:

```
El nombre és més gran que 9
Ho sabies?
Adéu
```

Cas 2: Si "numero" NO és més gran que 9, mostraria per pantalla:

```
Adéu
```

És important que ens fixem en la indentació, ja que marquen el codi que s'executarà en cas que la condició sigui certa.

4.2. Instrucció if..else

De manera optativa, podem afegir a la instrucció if una instrucció **else**, on podrem especificar el tros de codi que s'executarà quan la condició de la instrucció if no sigui certa. Sintaxi:

If <condició> :
    BLOC CONDICIÓ CERTA
else:
    BLOC CONDICIÓ NO CERTA

Un exemple seria:

```
if   numero>9:
        print ("El nombre és més gran que 9")
else :
        print ("El nombre NO és més gran que 9")
```

És important remarcar que no podem posar la instrucció "else" sense haver posat abans la "if", ja que tot "else" va en combinació amb un "if", i s'ha de posar amb els mateixos espais per l'esquerra (indentació) que té el "if" al qual pertany.

Un exemple:

```
1.  numero=3
2.  if   numero>3 :
3.          suma=numero+2
4.          print ("La suma és: ",suma)
5.  else :
6.          doble = numero*2
7.          print ("El producte és: ",doble)
8.
```

Com ara "numero=3", no executarà el codi del "if", sinó el codi del "else" (mostrarà per pantalla el producte i no la suma), ja que "numero" no és >3 (numero=3).

Exercici proposat 1: Es proposa canviar la línia 2 per "if numero>=3:" per a comprovar que ara la condició sí és certa, ja que "numero" és més gran o igual que 3.

Exercici proposat 2: Es proposa canviar el valor de "numero" per a fer que la condició del "if" sigui certa, sense la modificació de l'exercici anterior.

Exercici proposat 3: Fer un programa que mostri 3 tipus de missatges respecte a un numero concret: si "numero>3", si "numero<3" i si numero és 3.

Les solucions estan al final del document.

4.3. Instrucció if..elif..else

Una altra variant és fer servir elif en lloc de l'else directament. Aquesta variant permet preguntar de nou en cas que volguem fer un altre if després d'un else, i va molt bé per a especificar casos excloents.

Sintaxi:

If <condició1> :
BLOC CONDICIÓ SI condició1 CERTA
elif <condició2>:
BLOC CONDICIÓ SI condició2 CERTA
...
elif <condició_n>:
BLOC CONDICIÓ SI condició_n CERTA
else:
BLOC CONDICIÓ SI cap de les condicions anteriors eren CERTES

Per exemple, en el cas de l'exercici proposat 3 d'abans, tenim 3 casos excloents: is >3, si <3 i si =3. Si implementéssim aquest exercici fent anar "elif" seria més fàcil:

```
1. numero=int(input("Entra un num.:"))
2. if   numero>3 :
3.         print ("El ",numero," és >3 ")
4.  elif  numero<3 :
5.         print ("El ",numero," és >3 ")
6. else:                    #el darrer else s'executa en cas que tot els anteriors no hagin estat certs.
7.         print ("El numero és 3 ")
```

En aquest cas mostraria per pantalla un dels 3 missatges.

4.4. Format comprimit 'x if y else z'

Una altra variant és fer servir el format comprimit <**valor> if <condició> else <valor2>** per a assignar un valor o un altre a una variable.

Exemple. Els 2 programes són equivalents.

	Format comprimit		**Format no comprimit**
1.	n = 5	1.	n = 5
2.	x = 3 if n>1 else 8 #posa valor 3 si n>1 i 8 sinó.	2.	if n > 1: x = 3 (*)
3.	print(x) -> mostrarà 3, ja que n>1	3.	else: x = 8
		4.	print(x)-> mostrarà 3, ja que n>1

(*)si només és 1 instrucció el que hi ha dins l'if, es pot posar a la mateixa línia

5. Solucions als exercicis proposats

Exercici proposat 2: Es proposa canviar el valor de "numero" per a fer que la condició del "if" sigui certa, sense la modificació de l'exercici anterior.

```
1.  numero=3
2.  if  numero>=3 :
3.          suma=numero+2
4.          print ("La suma és: ",suma)
5.  else :
6.          doble = numero*2
7.          print ("El producte és: ",doble)
```

Exercici proposat 3: Fer un programa que mostri 3 tipus de missatges respecte a un numero concret: si "numero>3", si "numero<3" i si numero és 3.

En aquest cas farem "ifs anidats" (ficarem instruccions "if" dins d'altres instruccions "if"). Primer comprovarem si numero>3. Si no és cert, vol dir que només podrà ser <3 o igual a 3. Aleshores, al "else", tornarem a preguntar si el numero, que sabem que no és >3, és <3. Si això fos fals, voldria dir que ni és >3 ni <3, per tant és 3.

```
1.  numero=int(input("Entra un num.:"))
2.  if  numero>3 :
3.      print ("El ",numero," és >3 ")
4.  else :
5.      if  numero<3 :          #tot aquest "if" està dins de l'else de la línia 4 (indentat).
6.          print ("El ",numero," és >3 ")
7.      else:           #el darrer else s'executa si les 2 condiciona anteriors eren falses.
8.          print ("El numero és 3 ")
```