

Fonaments de Sistemes Operatius

Intèrpret Python

Carles Aliagas

carles.aliagas@urv.cat

M.Angels Moncusí

angels.moncusi@urv.cat

Departament d'Enginyeria Informàtica i Matemàtiques

UNIVERSITAT ROVIRA I VIRGILI

1. Intèrpret Python

- ◆ Llenguatge de programació interpretat
- ◆ Permet ampliar/mantenir les aplicacions fàcilment.
- ◆ Pot incorporar gran quantitat de funcions com a mòduls.
 - Import sys,so,re,glob
- ◆ Funcionalitat gràfica:
 - De tcl/tk -> tkinter
 - gtk2

2. Intèrpret Python

- ◆ Es crida des de la línia de comandes

```
bash-3.2$ python
```

```
Python 2.6.1 (r261:67515, Jun 24 2010, 21:47:49)
```

```
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for  
more information.
```

```
>>>
```

- A partir d'aquí es poden introduir sentències

```
>>> print "aloha"
```

```
aloha
```

```
>>> n = raw_input ("posa un numero :")
```

```
posa un numero :23
```

```
>>> print n
```

```
23
```

3. Programes Python

- ◆ Es poden crear fitxers scripts python:

```
#!/usr/bin/env python
print "Script de prova"
n=raw_input("Posa un nombre: ")
print n
```

Recordeu: `chmod u+x prova.py`

- En aquest cas la shell s'adona que l'script l'ha d'executar el programa 'python'

4. Variables

- ◆ Variables de qualsevol tipus
 - Simples:
 - Integer: `x=123`
 - Float: `y=12.90`
 - Caràcters: `a='b'`
 - Strings: `s="aloha"`
 - Compostos: llistes, tuples, sets, diccionaris
 - A mida: definits per l'usuari

5. expressions aritmètiques i strings

- ◆ Operació: '+'

```
a=23
b=4.5
print a+b
27.5
```

```
x="23"
y="4.5"
print x+y
234.5
```

- ◆ Substrings: s[i:j]

```
s="aloha baby"
print s[2:4]
oh
```

- ◆ Per tractar strings com a nombres:

- `int("23")`
- `float("4.5")`

- ◆ Per tractar nombres com a strings:

- `str(12345)`

```
print int(x)+b
27.5
```

6. Iteracions i condicionals

◆ Expressions condicionals:

<, >, <=, >=, !=, ==, in, not in, is, is not

◆ Bucle while:

```
i=0
while i<10:
    print i,i*i
    i+=1
```

◆ Bucle for:

```
for v in 1,5,7,8 :
    print v
```

- Assigna cada valor del conjunt d'elements a 'v' i executa el cos del bucle

◆ Conditionals if-then-else:

```
if a>=10:
    print "mes de deu"
elif a>=0:
    print "1 digit"
else:
    print "negatiu"
```

- *Vigilar amb les indentacions.*
- *No hi ha claus '{}'*
ni punts i coma ';'.

7. Dades Compostes

◆ Llistes: entre []

```
l=[12,'a',23,'b']
```

- Subllistes: `l[i:j]`
- Concatenació: `l+l`
- Funcions:

```
len,max,min,split,sort
```

```
len(l)    l.split(',')
```

◆ Tuples: entre ()

```
t=(1,'a',2,'b')
```

- Subtupla: `t[i:j]`
 - Estructura immutable
no es poden modificar
ni afegir dades

◆ Sets:

```
s=set("aloha")
```

```
set(['a','h','l','o'])
```

- Estructura amb dades úniques
- Operacions:
 - Unió: `|`
 - Intersecció: `&`
 - In A Not in B: `A-B`
 - In A or B not in both: `A^B`

◆ Diccionaris: entre { }

```
d={'a':1,'b':2,'c':3}
```

- Taules associatives
- ```
for i in d:
 print d[i]
```



# 8. Iteracions

## ◆ `for(i=ini;i<fi;i+=salt):`

```
for i in range(ini, fi, salt):
 print i
```

- Range genera una llista d'elements numèrics
- Itera el cos de bucle canviant el valor de la variable `i` per cadascun dels elements de la llista
- Versió més eficient: `xrange(ini, fi, salt)`
- No genera una llista sencera si no que la genera els elements a mesura que itera el bucle

# 9. Iteracions sobre elements

- ◆ Sobre elements de llista,tupla,sets,diccionaris:

```
l=[12,'a',23,'b']
for i in l:
 print i
```

```
t=(1,'a',2,'b')
for i in t:
 print i
```

```
s=set("aloha")
for i in s:
 print i
```

```
d={'a':1,'b':2,'c':3}
for i in d:
 print i,d[i]
```

# 10. Funcions

## ◆ Sobre elements de llista,tupla,sets,diccionaris:

```
y=1
def nom(a1,a2):
 global y
 x=0
 x += a1
 y += a2
 return x,y
```

- x: és una variable local que cada vegada s'inicialitza a 0
- y: és una variable global al mòdul, que manté el valor entre crides succesives

```
i,j = nom(3,4)
print i,j
```

- Executar varies vegades per veure els valors de retorn  
(veure que poden retornar tuples)

# 11. E/S sobre fitxers

- ♦ Python té funcions de tractaments de fitxers
- ♦ Primer s'ha d'obrir un fitxer i després llegir/escriure el seu contingut.

`f=open('nom',mode)`

- Per llegir

```
f=open('a.txt','r')
s=f.read() #Tot
l=f.readline() #1 linia
s=f.read(10) #10 characters
f.close()
```

- Per escriure

- main.py:

```
f=open('a.txt','w')
f.write("string \n")
f.close()
```

- Per iterar:

```
for s in f.read():
 print s,
```

# 12. Tractament d'errors

- ◆ El més habitual és afegir conditionals: if-elif-else
- ◆ Però es pot fer mitjançant tractament d'errors:

```
try:
 <sentencies>
except tipus:
 <tractament>
```

```
try:
 f=open('a.tmp','r')
except IOError:
 print "no es pot obrir"
...
```

# 13. Mòduls

- ◆ Els elements d'un mòdul python pot ser importat per un altre mòdul.
- ◆ Facilita la reutilització de codi

- A sub.py afegim:

```
def doble(a):
 return a*a
```

- Al programa principal:

- main.py:

```
#!/usr/bin/env python
import sub
print sub.doble(3)
```

- Les dades importades tenen el prefix del nom del mòdul

# 14. Mòdul os

- ◆ Permet accedir a comandes i al sistema de fitxers del Sistema operatiu host.
  - `os.system("comanda a executar")`  
`os.system("mv *.txt nou_dir")`
  - `os.path`: Permet manipular noms relatius/absoluts del sistema de fitxers
    - `os.path.abspath(fitxer)` -> retorna nom absolut
    - `os.path.dirname(nom_abs)` -> retorna el nom del directori
    - `os.path.basename(nom_abs)` -> retorna el nom del fitxer
    - `os.path.join(path1, path2)` -> uneix creant un únic nom absolut
    - `os.path.relpath(path, dir)` -> torna un nom relatiu des de dir
    - `os.path.isfile()`, `os.path.isdir()`, `os.path.islink()`, ...

# 15. Mòdul glob

- ◆ Retorna una llista d'element dels fitxers que troba en el path indicat a l'estil de la shell: []\*?.

- glob.glob(path)

```
print glob.glob("*.py")
```

```
['main.py', 'prova.py', 'sub.py']
```



# 16. Informació

- ◆ Hi ha molts punt d'informació:
  - <http://wiki.python.org/moin/BeginnersGuide/Programmers>
  - En aquesta pàgina es fa referencia a una bona colla de links de diverses menes: pdfs,wikis,....

# 17. Exercicis amb comandes

- ♦ Fer un script que cerquin uns fitxers per un arbre de directoris i canvii el nom del directori on es troben afegint una 'X' al començament del nom.