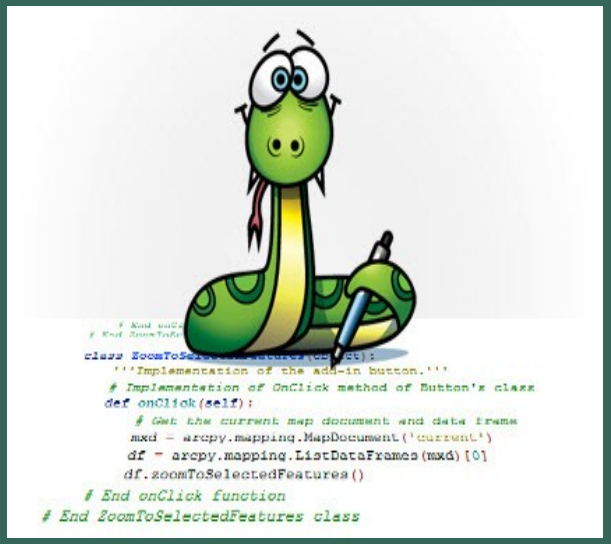
ARXIUS

LECTURA I ESCRIPTURA D'ARXIUS

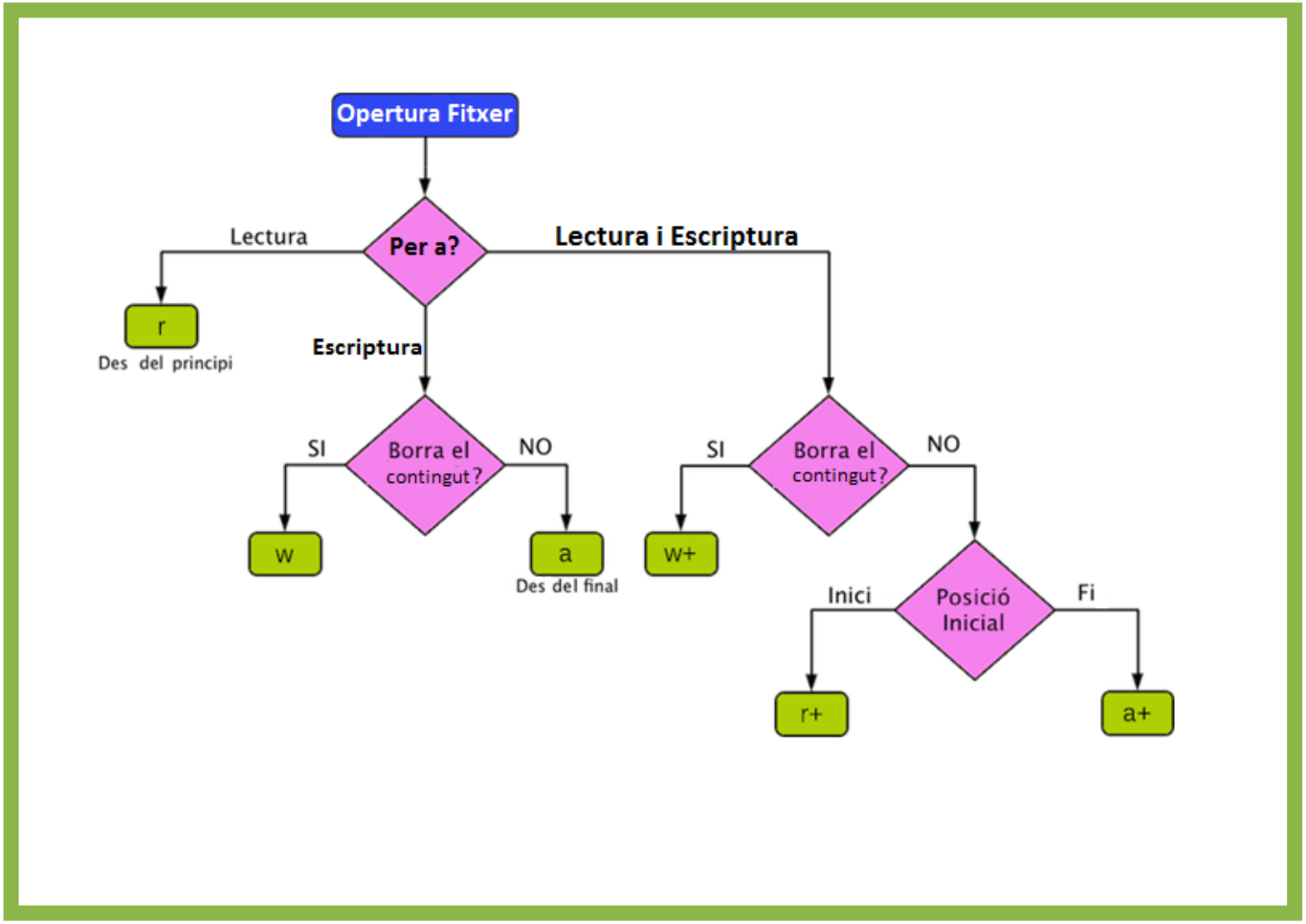


Mª Belén Tortosa Pedrón

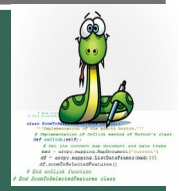
INTRODUCCIÓ

- Els fitxers en Python són objectes de tipus ***file*** creats mitjançant la funció ***open*** (obrir).
  - Aquesta funció pren com a paràmetres una cadena amb la ruta al fitxer a obrir, que pot ser relativa o absoluta.
  - Una cadena opcional indicant la manera d'accés (si no s'especifica s'accedeix en mode lectura).
    - La manera d'accés pot ser qualsevol combinació lògica de les següents maneres:
      - '***r***': read, lectura. Obre el fitxer tan sols lectura. L'arxiu ha d'existir prèviament, en cas contrari es llançarà una excepció de tipus ***IOError***.
      - '***w***': write, escriptura. Obre el fitxer tan sols escriptura. Si l'arxiu no existeix es crea. Si existeix, sobreescriu el contingut.
      - '***a***': append, afegir. Obre el fitxer tan en mode escriptura. Es diferencia de la manera 'w' en que en aquest cas no sobreescriu el contingut de l'arxiu, sinó que es comença a escriure al final de l'arxiu.
      - '***b***': binary, binari.
      - '+': Permet lectura i escriptura simultànies.
      - '***u***': universal newline, salts de línia universals. Permet treballar amb arxius que tinguin un format per als salts de línia que no coincideix amb el de la plataforma actual (en Windows s'utilitza el caràcter CR LF, en Unix LF i en Mac OS CR).
- Un enter opcional per especificar una mida de memòria intermèdia diferent de l'utilitzat per defecte.

FUNCIÓ *open*


Opertura Fitxer
Per a?
Lectura
Lectura i Escriptura
Escriptura
r
Des del principi
SI
Borra el contingut?
NO
w
a
Des del final
SI
Borra el contingut?
NO
w+
Inici
Posició Inicial
Fi
r+
a+

OBRIR UN ARXIU: *open()*



- En ***Python*** els fitxers s'obren amb la funció ***open()***.
  - Com a primer paràmetre es passa el nom del fitxer i com a segon paràmetre una cadena amb caràcters:

```
#Exemple f.open
f = open("arxiu.txt")
print(f.read())
f.close()
```

  - Exemples:
    - Obrir fitxer de lectura : *f = open("fichero.txt")*
    - Obrir fitxer de lectura : *f = open("fichero.txt", "r")*
    - Obrir fitxer de lectura en binari : *f = open("fichero.txt", "rb")*
    - Obrir fitxer per escriure des de zero : *f = open ("fichero.txt", "w")*
    - Obrir fitxer per afegir al final : *f = open ("fichero.txt", "a")*

DIRECTORIS

- Un problema relacionat amb la utilització de directoris és que els separadors de directoris en diferents sistemes són diferents, / en Unix i Macintosh, \ en Windows.
- La manera d'accedir a directoris independentment del sistema en el qual estem des de Python és usant el modulo **us**.

os.path.join("data","arxiu.txt")

```
import os
f=open(os.path.join("data","arxiu.txt"),"r")
print(f.read())
f.close()
```

LLEGIR FITXER: read() , readline () i readlines()

- Per llegir del fitxer, podem fer servir les funcions ***read ([bytes])*** i ***readline ([bytes])*** i ***readlines()***. El tipus de retorno sempre serà ***str*** sempre que s'obri el fitxer en mode text.
  - ***read()***: Lectura de tot el fitxer de cop. Exemple:
    - ***dada = f.read()***
  - ***read(numbytes):*** Si se li passa la longitud de bytes, llegirà només el contingut fins a la longitud indicada. Sabrem que em arribat al final del fitxer, si em trobat la cadena buida.
    - Lectura de 100 bytes: ***dada = f.read (100)***
  - ***readline()***: Lectura d'una línia complerta, inclou el '\n'. Exemple:
    - ***dada = f.readline ()***
  - ***readline(numbytes)***: Lectura dels bytes indicats. Exemple:
    - Lectura de 5 bytes: ***dada = f.readline (5)***
  - ***readlines()***: Llegeix totes les línies d'un arxiu. ***data=f.readlines()***

FITXERS EN PYTHON

- Els fitxers s'ofereixen al programa, en part, com llistes de línies, de tal manera que es poden recórrer utilitzant un bucle com en el programa, i no cal preocupar-se de quan s'arriba al final de fitxer.

```
fichero = open ('ejemplo.txt')

for linea in fichero:
    print(linea, ' Longitud:', len(linea));
```

- Si volem obtenir totes les línies del fitxer com una llista, podem executar ***fichero.readlines ()*** o ***list (fitxer)***.

```
fichero = open ('ejemplo2.txt','r')
print(list(fichero))
```

ESCRIURE EN UN FITXER: **write()**

- ***write (cadena):*** Escriu cadena dins de l'arxiu. Exemple:

```
f= open("arxiu.txt", "w")
f.write('Nueva linea')
```

- ***writelines (seqüència):*** seqüència serà qualsevol iterable els elements de la qual seran escrits un per línia.

```
f = open("arxiu.txt", "r+")

contingut = f.read()

lista = ['Línea 1\n', 'Línea 2']

f.writelines(lista)
```

- El que es fa a l'escriure és sobreescriure, i no inserir, la informació existent.

TANCAR UN ARXIU: *close()*

- ***close()***: Tanca l'arxiu. Exemp

```
f= open("arxiu.txt", "r")
contingut= f.read()
f.close()
print (contingut)
```

- Python incorpora una manera que permet que els fitxers es tanquin de forma automàtica sense necessitat d'invocar al mètode **close()**. Es tracta d'un

```
with open("arxiu.txt", "r") as f:
    contingut = f.read()

print (f.closed )
```

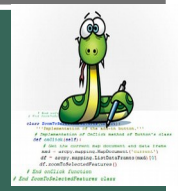
- Quan una estructura *with* finalitza, Python, automàticament invoca al mètode *close()*.
- La sentència *with* utilitza un àlies per a l'objecte *file*, la qual cosa permet accedir a l'objecte *file*, justament, per l'àlies indicat.

POSICIÓ DEL CURSOR. **tell()**

- ***tell()***: Ens indica en quina posició estem del fitxer.

```
f = open("arxiu.txt", "a")
linea1 = f.readline()
print(linea1)
print(f.tell())
mas = f.read(f.tell() * 2)
print(mas)
if f.tell() < 50:
    print(f.tell())
```

MOURE EL PUNTER: *seek()*



- **seek(byte)**: Mou el punter al byte indic

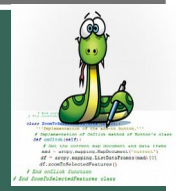
```
f = open("arxiu.txt", "r")
contingut = f.read()
# el punter està
# al final del document
print(f.tell())
f.seek(0) #Tornem a l'inici del document
print(f.read())
```

- Podem desplaçar-nos per ell, per llegir o escriure en una determinada posició.
  - El segon paràmetre és: **Cap o 0:** la posició és relativa al principi del fitxer **1:** la posició és relativa a la posició actual **2:** la posició és relativa al final del fitxer i cap enrere.
    - **f.seek (n)**: Anar al byte n del fitxer.
    - **f.seek (n, 0)**: Equivalent a l'anterior: Anar al byte n des del principi del fitxer.
    - **f.seek (n, 1)**: Desplaçar n bytes a partir de la posició actual del fitxer.
    - **f.seek (n, 2)**: Situar-nos n bytes abans del final de fitxer.

Només per a fitxers oberts en mode binari.

La funció **seek** només permet realitzar desplaçaments relatius a l'inici del fitxer quan aquest s'obre en mode text.

PROPIETATS DE *file*



- Es poden accedir a les següents propietats de l'objecte ***file***:
  - **closed**: retorna ***True*** si l'arxiu s'ha tancat. En cas contrari, ***False***.
  - **mode**: retorna la manera d'obertura.
  - **name**: retorna el nom de l'arxiu.
  - **encoding**: retorna la codificació de caràcters d'un arxiu de text.

```
f = open("arxiu.txt", "r+")
contingut = f.read()
nom = f.name
modo = f.mode
encoding = f.encoding
f.close()

if f.closed:
    print ("L'arxiu s'ha tancat correctament")
else:
    print ("L'arxiu està obert")

print(nom)
print(modo)
print(encoding)
```

EXERCICIS

1.1) Feu un programa que escrigui a l'arxiu "parells.txt" els nombres parells de l'1 al 100, i en "senars.txt" els senars de l'1 al 100.

1.2) Usant els arxius generats al problema anterior, feu un programa que fusioni "parells.txt" i "senars.txt" en un tercer arxiu "1a100.txt", intercalant una línia de senars, i una dels parells.

1.3) Feu un programa que demani dades de vàries persones , i les vagi desant a un arxiu "persones.txt". Les dades de cada persona seran (useu una llista):

- nom
- cognoms
- NIF
- edat
- alçada (en metres, p.ex: 1.63)

El procés s'atura quan l'usuari indica que no vol introduir més persones. El programa tindrà com a mínim les funcions llegirPersona i escriurePersonaADisc

1.4) Escriviu un programa que llegeixi l'arxiu "persones.txt" creat a l'exercici anterior, i mostri **totes les dades** de les persones **majors de 18 anys** per pantalla.

EXERCICI

- **1.5)** Farem un programa que treballi amb dos arxius d'entrada:
  - **Arxiu 1:** Trieu qualsevol arxiu que tingui dades.
  - **Arxiu 2:** Un arxiu que contingui una llista de números de línia en ordre creixent (el podeu crear manualment vosaltres mateixos). Per exemple:
    - 12
    - 15
    - 17
    - 26

- El vostre programa haurà d'obrir els dos arxius i mostrar per la pantalla les línies de l'arxiu 1 que s'indiquen a l'arxiu 2. Es a dir, la línia número 12 del arxiu 1, la línia número 15 de l'arxiu 1,etc.

EXERCICIS

1.6) Dins d'un mateix programa ...

a) Creeu un arxiu en mode w amb lectura i escriptura, i escriviu el següent contingut (a sac):

All paid jobs absorb and degrade the mind.

If Beethoven had been killed in a plane crash at the age of 22, it would have changed the history of music... and of aviation.

I do not have a psychiatrist and I do not want one, for the simple reason that if he listened to me long enough, he might become disturbed.

b) Torneu a l'inici de l'arxiu i mostreu la primera frase.

c) Fent servir un desplaçament des de la posició inicial, mostreu ara la paraula "mind" de la primera frase.

e) Fent servir un desplaçament des de la posició actual, mostreu ara la paraula "Beethoven" de la segona frase.

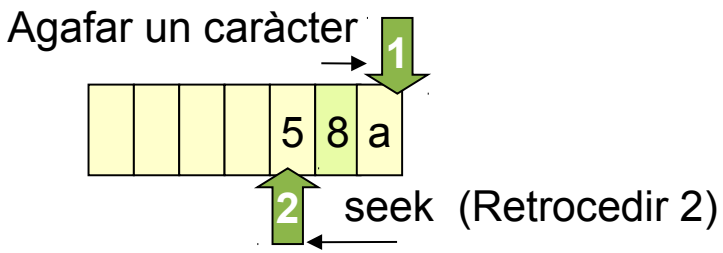
g) Fent servir un desplaçament, mostreu la última frase.

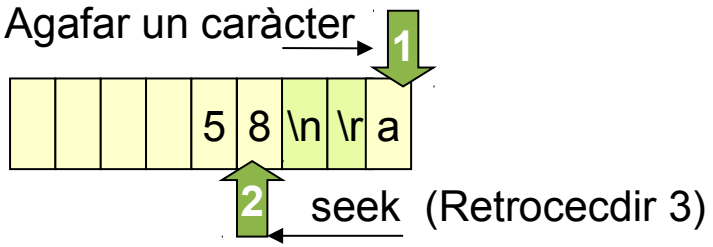
h) Calculeu la mida de l'arxiu en KBytes

EXERCICIS

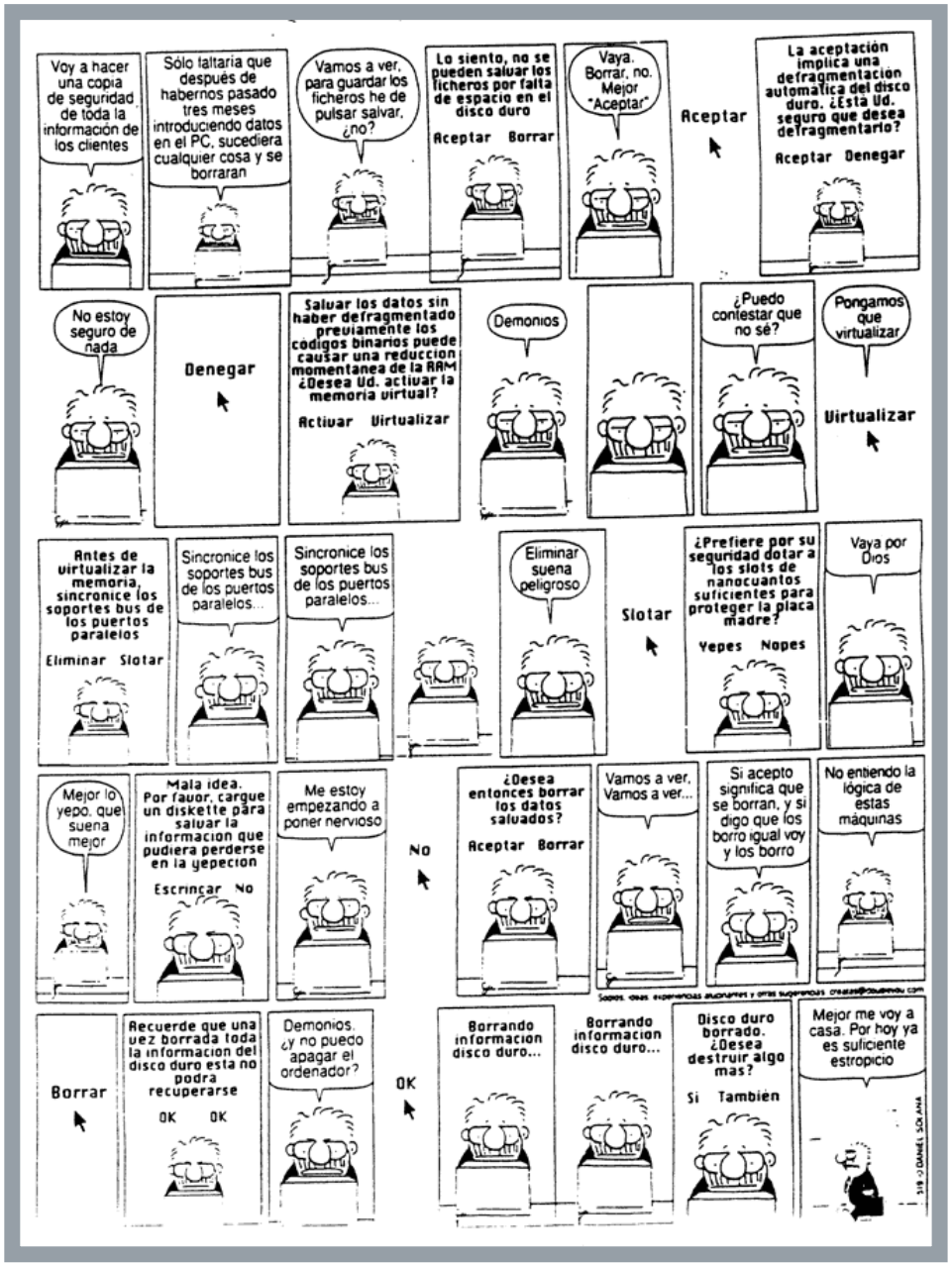
1.7) Feu un programa anomenat "**satanic.py**" que mostri per pantalla el contingut d'un arxiu de text totalment al revés (caràcter a caràcter).

- L'algorisme que es proposa és el següent:
  1. Situeu-vos a l'últim caràcter de l'arxiu
  2. Llegiu 1 CARACTER de l'arxiu, i mostreu-lo per pantalla.
  3. Si el caràcter llegit NO és un enter, retrocediu **2** posicions el cursor. Si ho és, retrocediu **3** posicions (l'espai ocupa 2 bytes!)





4. Torneu al punt 2 sempre i quan no estiguem ja a l'inici de l'arxiu (useu ***tell*** per a saber quan hi arribem)


Voy a hacer una copia de seguridad de toda la información de los clientes
Sólo faltaría que después de habernos pasado tres meses introduciendo datos en el PC, sucediera cualquier cosa y se borraran
Vamos a ver, para guardar los ficheros he de pulsar salvar, ¿no?
Lo siento, no se pueden salvar los ficheros por falta de espacio en el disco duro
Aceptar Borrar
Vaya. Borrar, no. Mejor "Aceptar"
Aceptar
La aceptación implica una defragmentación automática del disco duro. ¿Está Ud. seguro que desea defragmentarlo?
Aceptar Denegar
No estoy seguro de nada
Denegar
Salvar los datos sin haber defragmentado previamente los códigos binarios puede causar una reducción momentánea de la RAM ¿Desea Ud. activar la memoria virtual?
Activar Virtualizar
Demonios
¿Puedo contestar que no sé?
Pongamos que virtualizar
Virtualizar
Antes de virtualizar la memoria, sincronice los soportes bus de los puertos paralelos
Eliminar Slotar
Sincronice los soportes bus de los puertos paralelos...
Sincronice los soportes bus de los puertos paralelos...
Eliminar suena peligroso
Slotar
¿Prefiere por su seguridad dotar a los slots de nanocuantos suficientes para proteger la placa madre?
Yepes Nopes
Vaya por Dios
Mejor lo yepo, que suena mejor
Mala idea. Por favor, cargue un diskette para salvar la información que pudiera perderse en la yepecion
Escrincar No
Me estoy empezando a poner nervioso
No
¿Desea entonces borrar los datos salvados?
Aceptar Borrar
Vamos a ver, Vamos a ver...
Si acepto significa que se borran, y si digo que los borro igual voy y los borro
No entiendo la lógica de estas máquinas
Borrar
Recuerde que una vez borrada toda la información del disco duro esta no podrá recuperarse
OK OK
Demonios. ¿y no puedo apagar el ordenador?
OK
Borrando información disco duro...
Borrando información disco duro...
Disco duro borrado. ¿Desea destruir algo mas?
Si También
Mejor me voy a casa. Por hoy ya es suficiente estropicio
© DANIEL SOLANA