



UNIVERSIDAD  
DE MÁLAGA



CURSOS ONLINE

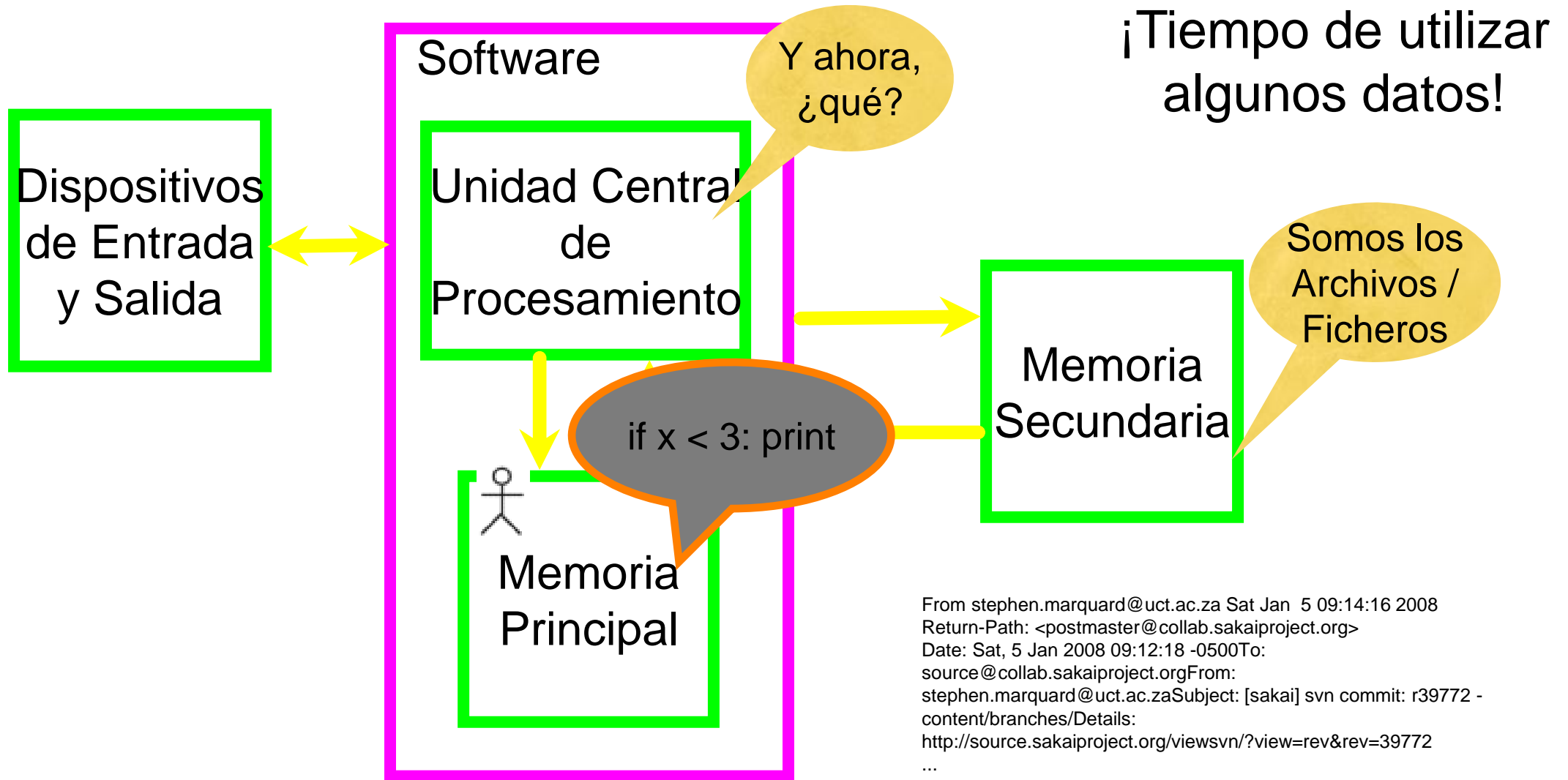
FUNDACIÓN GENERAL UNIVERSIDAD DE MÁLAGA

# Archivos

## Unidad 7

Python para principiantes  
[dgm@uma.es](mailto:dgm@uma.es)





# Procesamiento de Archivos

Un archivo de texto puede ser visto como una **secuencia de líneas**

From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008

Return-Path: <postmaster@collab.sakaiproject.org>

Date: Sat, 5 Jan 2008 09:12:18 -0500

To: source@collab.sakaiproject.org

From: stephen.marquard@uct.ac.za

Subject: [sakai] svn commit: r39772 - content/branches/

Details: <http://source.sakaiproject.org/viewsvn/?view=rev&rev=39772>

Archivo mbox-short.txt en campus virtual

# Abriendo un Archivo

- Antes de leer los contenidos de un archivo, debemos decirle a Python **con qué archivo** vamos a trabajar y **qué vamos a hacer** con él
- Esto se hace con la función **open()**
- **open()** retorna un “**manejador de archivo**” – una variable que permite hacer operaciones sobre el archivo
- Similar a “Archivo -> Abrir” en un procesador de textos

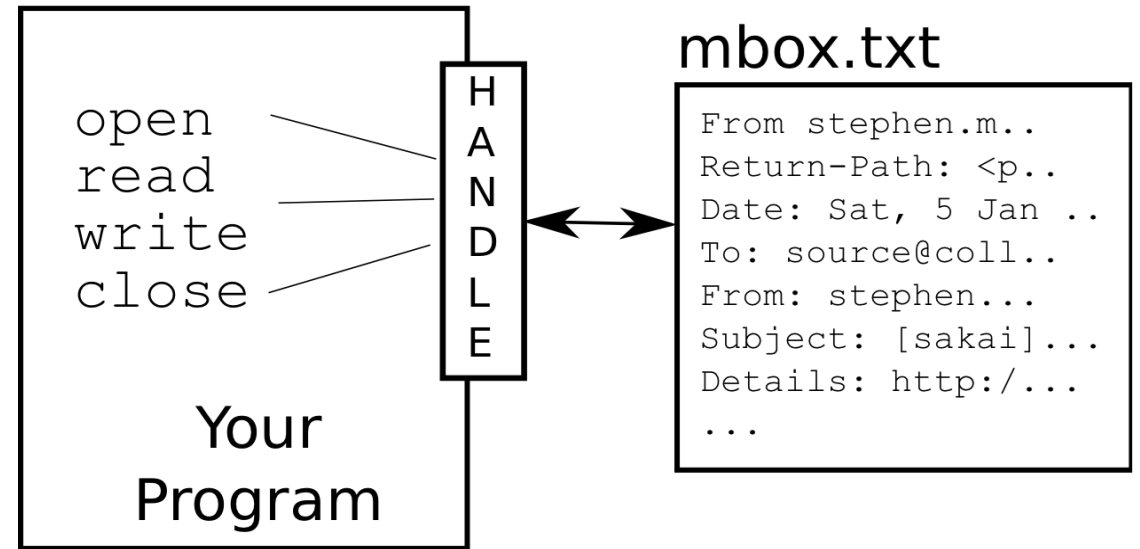
# Usando open()

```
fhand = open('mbox.txt', 'r')
```

- **handle = open(filename, mode)**
- Retorna un manejador para manipular el archivo
- filename es un string e indica el nombre del archivo
- mode es opcional y debería ser 'r' si vamos a leer del archivo y 'w' si vamos a escribir en él

# ¿Qué es un manejador?

```
>>> fhand = open('mbox.txt')
>>> print(fhand)
<_io.TextIOWrapper name='mbox.txt' mode='r' encoding='cp1252'>
```



# Cuando no se encuentra un archivo

```
>>> fhand = open('stuff.txt')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
FileNotFoundError: [Errno 2] No such file or
directory: 'stuff.txt'
```

# El carácter de nueva línea

- Se utiliza un carácter especial denominado de **nueva línea** (newline) para indicar cuándo acaba una línea y comienza otra
- Se representa en strings con **\n**
- Ojo! Sigue siendo un carácter – no dos

```
>>> stuff = 'Hello\nWorld!'
>>> stuff
'Hello\nWorld!'
>>> print(stuff)
Hello
World!
>>> stuff = 'X\nY'
>>> print(stuff)
X
Y
>>> len(stuff)
3
```



# Procesamiento de archivos

Un archivo de texto puede ser visto como una **secuencia de líneas**

From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008

Return-Path: <postmaster@collab.sakaiproject.org>

Date: Sat, 5 Jan 2008 09:12:18 -0500

To: source@collab.sakaiproject.org

From: stephen.marquard@uct.ac.za

Subject: [sakai] svn commit: r39772 - content/branches/

Details: <http://source.sakaiproject.org/viewsvn/?view=rev&rev=39772>

# Procesamiento de archivos

Un archivo de texto tiene `\n` (nueva línea) **al final de cada línea**

```
From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008\nReturn-Path: <postmaster@collab.sakaiproject.org>\nDate: Sat, 5 Jan 2008 09:12:18 -0500\nTo: source@collab.sakaiproject.org\nFrom: stephen.marquard@uct.ac.za\nSubject: [sakai] svn commit: r39772 - content/branches/\n\nDetails: http://source.sakaiproject.org/viewsvn/?view=rev&rev=39772\n
```

# Leyendo archivos en Python

# El manejador de archivo como una secuencia

- Un manejador de archivo abierto para lectura puede ser tratado como una **secuencia de strings** donde cada línea en el archivo es un string en la secuencia
- Podemos usar un **bucle for** para iterar a través de la secuencia
- Recuerda - una secuencia es un conjunto ordenado (obtendremos el contenido del archivo en el mismo orden en el que está almacenado)

```
xfile = open('mbox.txt')  
for cheese in xfile:  
    print(cheese)
```

Una vez leído el archivo, tenemos que cerrarlo (`xfile.close()`) y volver a abrirlo si queremos volver a procesarlo. Esto es, no podemos "retroceder" en la lectura.

# Contando líneas en un archivo

- Abre un archivo
- Usa un bucle for para leer cada línea
- Cuenta las líneas e imprime su número

```
fhand = open('mbox.txt')
count = 0
for line in fhand:
    count = count + 1
print('Número de líneas:', count)
```

```
$ python open.py
Número de líneas: 132045
```

# Leyendo el archivo \*completo\*

- Podemos leer el archivo completo (\n incluidos) en un string simple

```
>>> fhand = open('mbox-short.txt')
>>> inp = fhand.read()
>>> print(len(inp))
94626
>>> print(inp[:20])
From stephen.marquar
```

# Buscando en un archivo

- Podemos usar sentencias if en el bucle for para imprimir solamente las líneas que cumplan **cierto criterio**

```
fhand = open('mbox-short.txt')
for line in fhand:
    if line.startswith('From:') :
        print(line)
```

# OOPS!

¿Qué hacen estas líneas  
en blanco aquí?

From: `stephen.marquard@uct.ac.za`

From: `louis@media.berkeley.edu`

From: `zqian@umich.edu`

From: `rjlowe@iupui.edu`

...



# OOPS!

¿Qué hacen estas líneas en blanco aquí)

- Cada línea de archivo tiene un carácter de nueva línea al final
- La sentencia `print` añade otro newline al final de cada línea

```
From: stephen.marquard@uct.ac.za\n\nFrom: louis@media.berkeley.edu\n\nFrom: zqian@umich.edu\n\nFrom: rjlowe@iupui.edu\n\n...
```

# Buscando en un archivo (arreglado)

- Podemos eliminar los espacios en blanco al final de un string usando la función **rstrip()**
- El newline es considerado un “espacio en blanco” y es eliminado

```
fhand = open('mbox-short.txt')
for line in fhand:
    line = line.rstrip()
    if line.startswith('From:') :
        print(line)
```

```
From: stephen.marquard@uct.ac.za
From: louis@media.berkeley.edu
From: zqian@umich.edu
From: rjlowe@iupui.edu
....
```

# Saltando líneas con continue

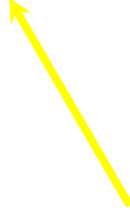
- Si una línea no nos interesa, podemos saltarla con la sentencia continue

```
fhand = open('mbox-short.txt')
for line in fhand:
    line = line.rstrip()
    if not line.startswith('From:') :
        continue
    print(line)
```

# Usando in para seleccionar líneas

Podemos utilizar **in** en una línea como nuestro **criterio de selección**


```
fhand = open('mbox-short.txt')
for line in fhand:
    line = line.rstrip()
    if not '@uct.ac.za' in line :
        continue
    print(line)
```




```
From stephen.marquard@uct.ac.za Sat Jan  5 09:14:16 2008
X-Authentication-Warning: set sender to stephen.marquard@uct.ac.za using -f
From: stephen.marquard@uct.ac.za
Author: stephen.marquard@uct.ac.za
From david.horwitz@uct.ac.za Fri Jan  4 07:02:32 2008
X-Authentication-Warning: set sender to david.horwitz@uct.ac.za using -f...
```

```
fname = input('Nombre de archivo: ')
fhand = open(fname)
count = 0
for line in fhand:
    if line.startswith('Subject:') :
        count = count + 1
print('Hubo', count, 'líneas con subject en', fname)
```

# Preguntar por el nombre de un archivo



Nombre de archivo: mbox.txt  
Hubo 1797 líneas con subject en mbox.txt



Nombre de archivo: mbox-short.txt  
Hubo 27 líneas con subject en mbox-short.txt

# Nombres de archivo incorrectos

```
fname = input('Nombre de archivo: ')
try:
    fhand = open(fname)
except:
    print('El archivo no pudo abrirse:', fname)
    quit()

count = 0
for line in fhand:
    if line.startswith('Subject:') :
        count = count + 1
print('Hubo', count, 'líneas con subject en', fname)
```

Nombre de archivo: mbox.txt

Hubo 1797 líneas con subject en mbox.txt

Nombre de archivo: na na boo boo

El archivo no pudo abrirse: na na boo boo

# Escribiendo en archivos

- Para escribir en un archivo, tenemos que abrirlo con el modo "w" al hacer open
- ¡¡¡Cuidado, el contenido previo **se borrará!!!**
- También Podemos usar el **modo "a"** si lo que queremos es **añadir** datos al final del archivo
- El método **write** del manejador del archivo, permite escribir datos en él – retorna el número de caracteres escritos
- Al final, tenemos que cerrar el archivo con **close**, para asegurarnos de que los contenidos quedan escritos

```
fout = open('output.txt', 'w')
```

```
>>> line1 = "Esto va a escribirse,\n"
```

```
>>> fout.write(line1)
```

```
22
```

```
fout.close()
```

```
fout = open('output.txt', 'a')
```

```
>>> line1 = "Y esto se añade.\n"
```

```
>>> fout.write(line1)
```

```
17
```

```
fout.close()
```

# Resumen

- Almacenamiento secundario
- Abriendo un archivo –
  - manejador de archivo
- Estructura de archivo –
  - el carácter newline
- Leyendo un archivo línea a línea con un bucle for
- Buscando líneas
- Leyendo nombres de archivos
- Tratando con nombres equivocados
- Escribiendo en archivos



## Ejercicio

Escribir un programa que pregunte por un nombre de archivo, lo abra, y lo lea mostrando las líneas de la forma:

X-DSPAM-Confidence: 0.8475

Contar las líneas y extraer los valores flotantes de cada una de ellas, calculando el promedio y mostrándolo al final.

Se pueden descargar datos de muestra desde el archivo mbox-short.txt en Campus Virtual. El valor promedio resultante debería ser:  
0.7507185185185187

# Acknowledgements / Contributions



These slides are Copyright 2010- Charles R. Severance ([www.dr-chuck.com](http://www.dr-chuck.com)) of the University of Michigan School of Information and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Continue...

Initial Development: Charles Severance, University of Michigan School of Information

... Insert new Contributors and Translators here  
Spanish Version: Daniel Garrido ([dgm@uma.es](mailto:dgm@uma.es))