勞動部產業新尖兵計畫

人工智慧金融應用與實務培訓班

課程模組： AI 金融科技課程 - Python 程式設計

6. 檔案輸出入處理

葉建華 (Yeh, Jian-hua)

tdi.jhyeh@tdi.edu.tw
au4290@gmail.com

講次內容

- 何謂檔案?
- 檔案的分類
- 型別 str 、 bytes 、 bytearray
- 文字編碼系統

何謂檔案

- 檔案是一群相關資料的集合，是儲存在長期儲存體的一段資料流 (data stream)
- 在電腦中，通常檔案包括有程式檔（執行及系統檔案）及資料檔

何謂檔案

- 檔案是一群相關資料[illegible]長期儲存體的一段資料流 (da[illegible]
- 在電腦中，通常檔案[illegible]行及系統檔案）及資料檔

檔案管理

- 檔案還可以依據需求，分門別類進行管理
- 檔案系統中使用「資料夾」（或稱目錄）概念

檔案管理

- 檔案還可以依據需求，分門別類進行管理
- 檔案系 

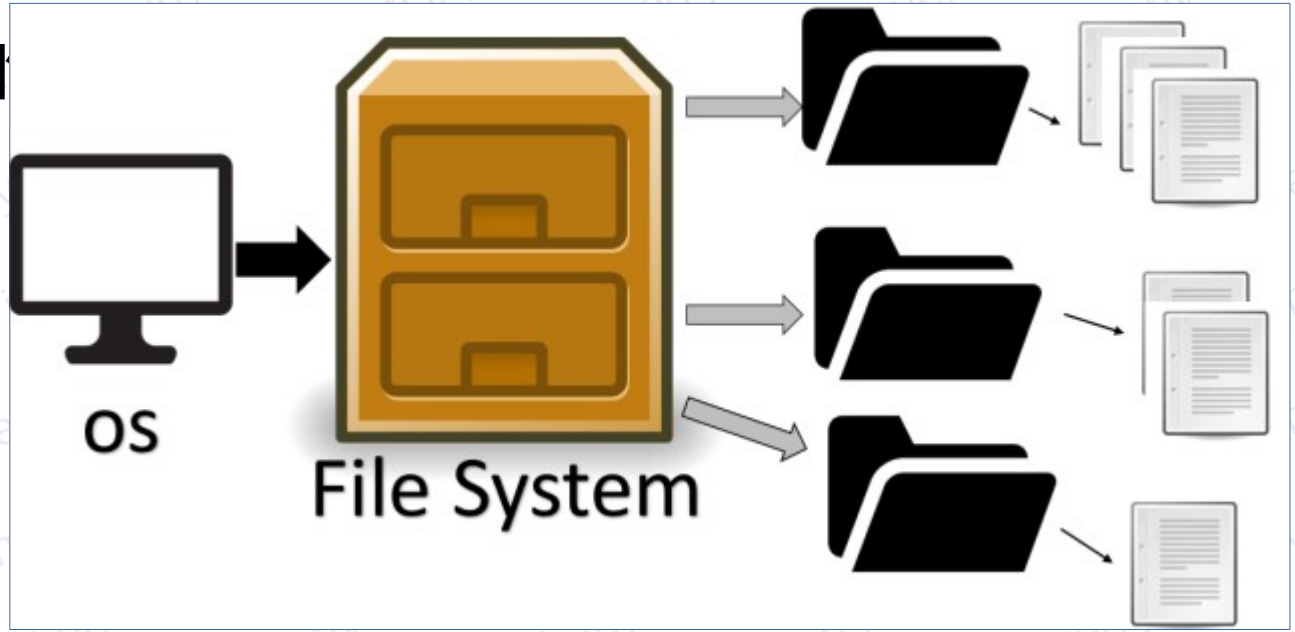
概念

電腦系統 - 檔案管理

- 電腦的作業系統透過檔案系統（子系統）來進行檔案的管理（透過資料夾組織檔案）

電腦系統 - 檔案管理

- 電腦的作業系統透過檔案系統（子系統）來進行檔案的



Python 的檔案管理功能

- 取得資料夾 / 檔案資訊
  - os.getcwd()：取得目前所在的資料夾
  - os.listdir()/os.scandir()：取得目前所在資料夾的內容
  - os.walk()：取得指定資料夾的「詳細」內容

Python 的檔案管理功能

- 取得資料夾 / 檔案資訊
  - 資料夾
  - 夾的內容
  - 「詳細」內容

```
import os

print('Current dir:', os.getcwd())

count = 0
for item in os.listdir():
    count += 1
    print(count, item)
print('Total', count, 'items in', os.getcwd())
```

```
Current dir: /home/jhyeh/Desktop/summer.course/python.basics/proj6
1 .ipynb_checkpoints
2 ch06.ipynb
Total 2 items in /home/jhyeh/Desktop/summer.course/python.basics/proj6
```

Python 的檔案管理功能

- 取得資料夾 / 檔案資訊
  - os.walk()：取得指定資料夾的「詳細」內容
  - iterator on tuples：

    (指定資料夾下的某子資料夾名稱（字串），

    其下所有子資料夾名稱（字串串列），

    其下所有所屬檔案名稱（字串串列）)

Python 的檔案管理功能

- 取得資料夾 / 檔案資訊
  - 的「詳細」內容
  - 名稱（字串），
    其下所有子資料夾名稱（字串串列），
    其下所有所屬檔案名稱（字串串列））

```
import os

path = '/opt/'
for item in os.walk(path):
    print('dir name:', item[0])
    print('sub-dir list:', item[1])
    print('file list:', item[2])
    print('='*80)
```

Python 的檔案管理功能

- 取得資料夾 /
  -
```
import os

path = '/opt/'
for item in os.
    print('dir
    print('sub-
    print('file
    print('='*8
```
  - 其下所有子
  - 其下所有所

```
dir name: /opt/
sub-dir list: ['idea']
file list: ['ideaIC-2020.3.2.tar.gz']
================================================================================
dir name: /opt/idea
sub-dir list: ['redist', 'license', 'jbr', 'plugins', 'lib', 'bin']
file list: ['Install-Linux-tar.txt', 'build.txt', 'LICENSE.txt', 'NOTICE.txt', 'icons.db',
'product-info.json']
================================================================================
dir name: /opt/idea/redist
sub-dir list: []
file list: ['annotations-java8.jar']
================================================================================
dir name: /opt/idea/license
sub-dir list: []
file list: ['jgoodies_forms_license.txt', 'javolution_license.txt', 'junit_license.txt', 'm
iglayout_swing_license.txt', 'trove4j_license.txt', 'microba_license.txt', 'growl.license',
'eclipse_license.txt', 'eclipse_license2.txt', 'picoContainer_license.txt', 'asm_license.tx
t', 'kryo-license.txt', 'jdom_license.txt', 'saxon-conditions.html', 'winp_license.txt', 'x
mlrpc_license.txt', 'javahelp_license.txt', 'swingx_license.txt', 'log4j_license.txt', 'nan
oxml_license.txt', 'oromatcher_license.txt', 'ant_license.txt', 'xerces_license.txt', 'jaxb
_license.txt', 'jaxen_license.txt', 'yourkit-license-redist.txt', 'imgscalr_license.txt', '
XStream_license.txt', 'third-party-libraries.html', 'gson_license.txt']
================================================================================
dir name: /opt/idea/jbr
sub-dir list: ['legal', 'include', 'conf', 'lib', 'bin']
```

Python 的檔案管理功能

- 資料夾名稱組合
  - os.path.join()： 將資料夾路徑與子資料夾名稱合併成路徑
  - os.path.exists()： 確認檔案 / 資料夾是否存在
  - os.path.isdir()： 確認是否為資料夾

Python 的檔案管理功能

- 資料夾名稱組合
  - 與子資料夾名稱合併成
  - 資料夾
  - 料夾

```
import os

path = '/opt'
subdir = 'idea'
# 注意! path2不一定存在
path2 = os.path.join(path, subdir)
print(path2)
print('real folder?', os.path.isdir(path2))
print('exists?', os.path.exists(path2))

subdir2 = 'noname'
path3 = os.path.join(path, subdir2)
print(path3)
print('real folder?', os.path.isdir(path3))
print('exists?', os.path.exists(path3))
```

```
/opt/idea
real folder? True
exists? True
/opt/noname
real folder? False
exists? False
```

Python 的檔案管理功能

- 資料夾建立與刪除
  - os.mkdir()： 建立單層資料夾，如果資料夾已存在則失敗
  - os.makedirs()： 建立指定的多層資料夾，如果資料夾已存在則失敗

Python 的檔案管理功能

- 資料夾建立與刪除
  - 料夾已存在則
  - os.makedirs()：建立指定的多層資料夾，如果資料夾已存在則失敗

```
import os

subdir1 = 'sub1'
subdir2 = 'sub2/sub3'
os.mkdir(subdir1)          # 如果'sub1'資料夾存在，失敗
os.makedirs(subdir2)       # 如果'sub2/sub3'資料夾存在，失敗
```

Python 的檔案管理功能

- 資料夾建立與刪除
  - 資料夾已存在則
  - os.mak
    夾已存

```
import os

subdir1 = 'sub1'
subdir2 = '
os.mkdir(su
os.makedirs
```

```
---------------------------------------------------------------------------
FileExistsError                           Traceback (most recent call last)
<ipython-input-16-e34d115dce33> in <module>
      4 subdir2 = 'sub2/sub3'
      5 os.mkdir(subdir1)          # 如果'sub1'資料夾存在，失敗
----> 6 os.makedirs(subdir2)      # 都會執行成功

~/anaconda3/lib/python3.8/os.py in makedirs(name, mode, exist_ok)
    221             return
    222     try:
--> 223         mkdir(name, mode)
    224     except OSError:
    225         # Cannot rely on checking for EEXIST, since the operating system

FileExistsError: [Errno 17] File exists: 'sub2/sub3'
```

Python 的檔案管理功能

- 檔案 / 資料夾更名
  - os.rename()： 參數為舊名稱、新名稱，檔案資料夾均有效

Python 的檔案管理功能

- 檔案 / 資料夾更名
  - os.rename()： 參數為舊名稱、新名稱，檔案資料夾均有效

```
import os

subdir1 = 'sub1'
subdir2 = 'newsub1'
os.rename(subdir1, subdir2)

fname1 = 'demo.txt'
fname2 = 'demo2.txt'
os.rename(fname1, fname2)
```

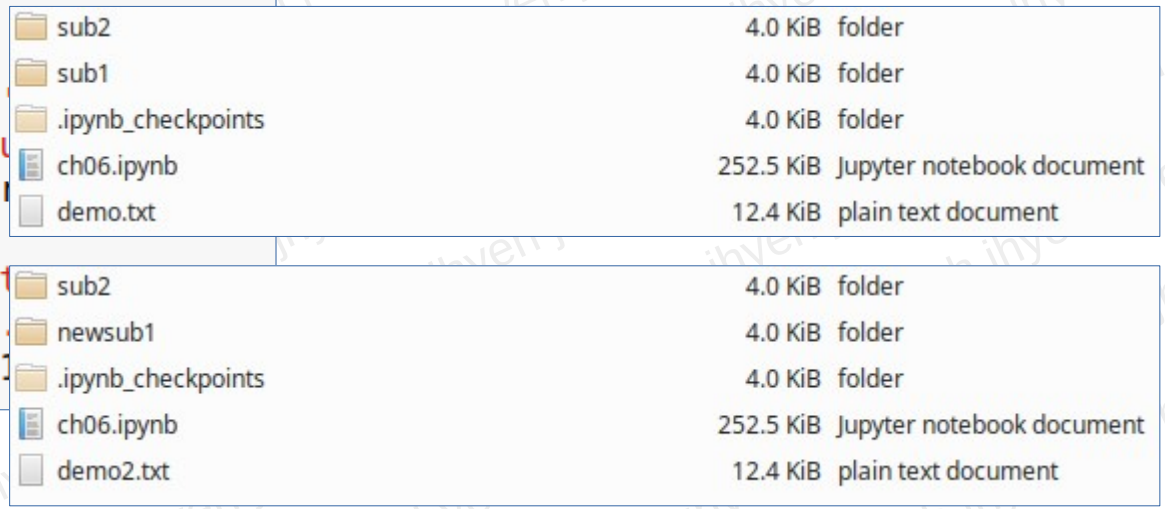
Python 的檔案管理功能

- 檔案 / 資料夾更名
  - os.rename()：參數為舊名稱、新名稱，檔案資料夾均有效

```
import os

subdir1 = 'sub1'
subdir2 = 'newsu
os.rename(subdi

fname1 = 'demo.t
fname2 = 'demo2.
os.rename(fname1
```



Python 的檔案管理功能

- 檔案 / 資料夾的描述屬性
  - os.scandir() 回傳的 ScandirIterator 物件會包含描述屬性，使用 iterator 數出 entry 的 stat() 方法可以取得， entry.name 可取得名稱
  - 描述屬性：檔案大小、修改時間、存取權限 ... 等等

Python 的檔案管理功能

- 檔案 / 資料夾的描述屬性
  - 件會包含描述
  - 方法可以取
  - 權限 ... 等等

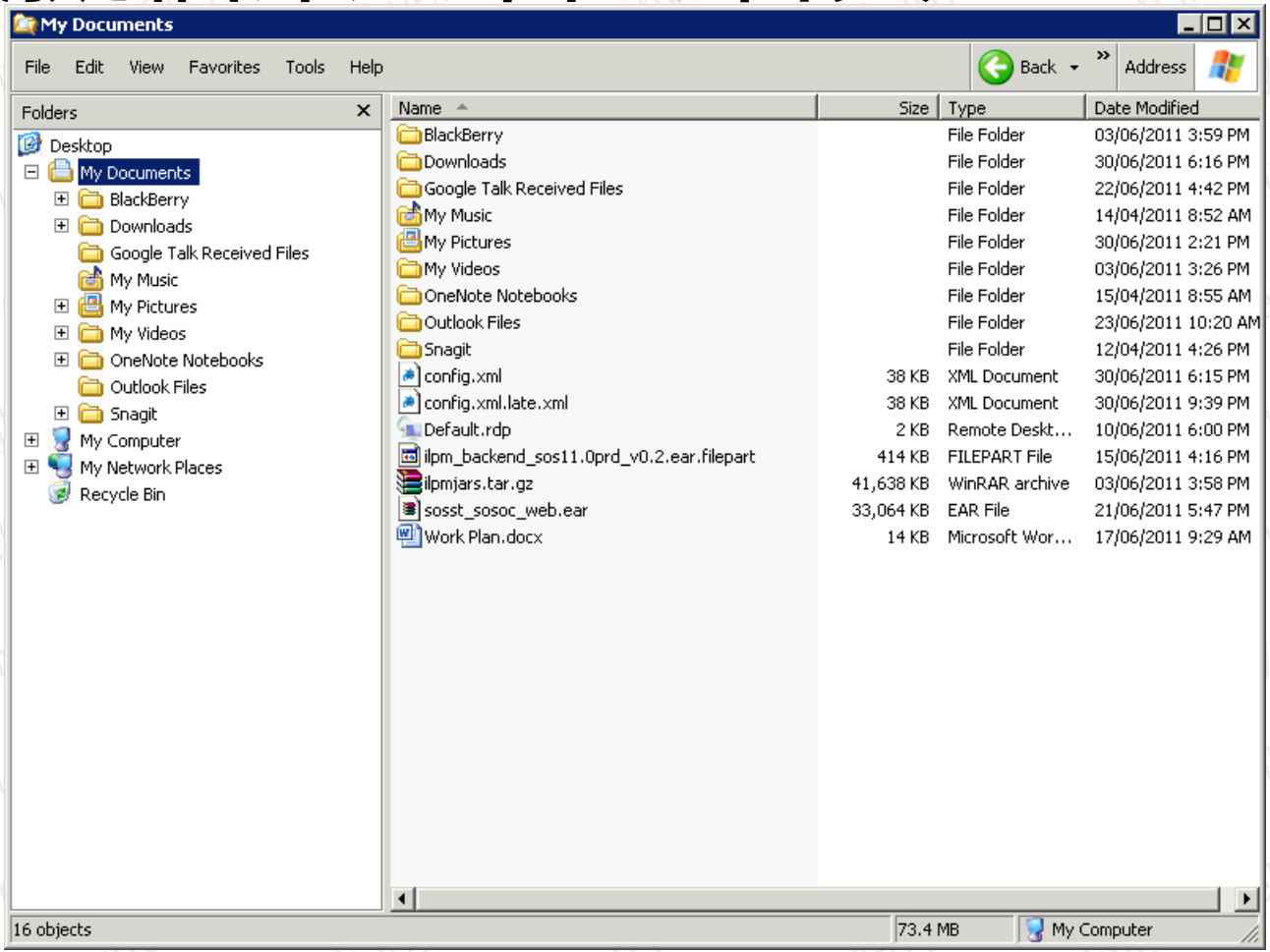
```python
import os
from datetime import datetime

path = '/opt/'
for entry in os.scandir(path):
    info = entry.stat()
    # epoch timestamp轉換成日期字串
    da = datetime.utcfromtimestamp(info.st_mtime)
    dstr = da.strftime('%Y/%m/%d')
    if entry.is_dir():
        print('資料夾: ', entry.name, '最後存取時間: ', dstr)
    elif entry.is_file():
        print('檔案: ', entry.name, '最後存取時間: ', dstr)
```

```
資料夾:  idea 最後存取時間:  2021/02/13
檔案:  ideaIC-2020.3.2.tar.gz 最後存取時間:  2021/02/13
```

練習：模擬檔案總管的右頁

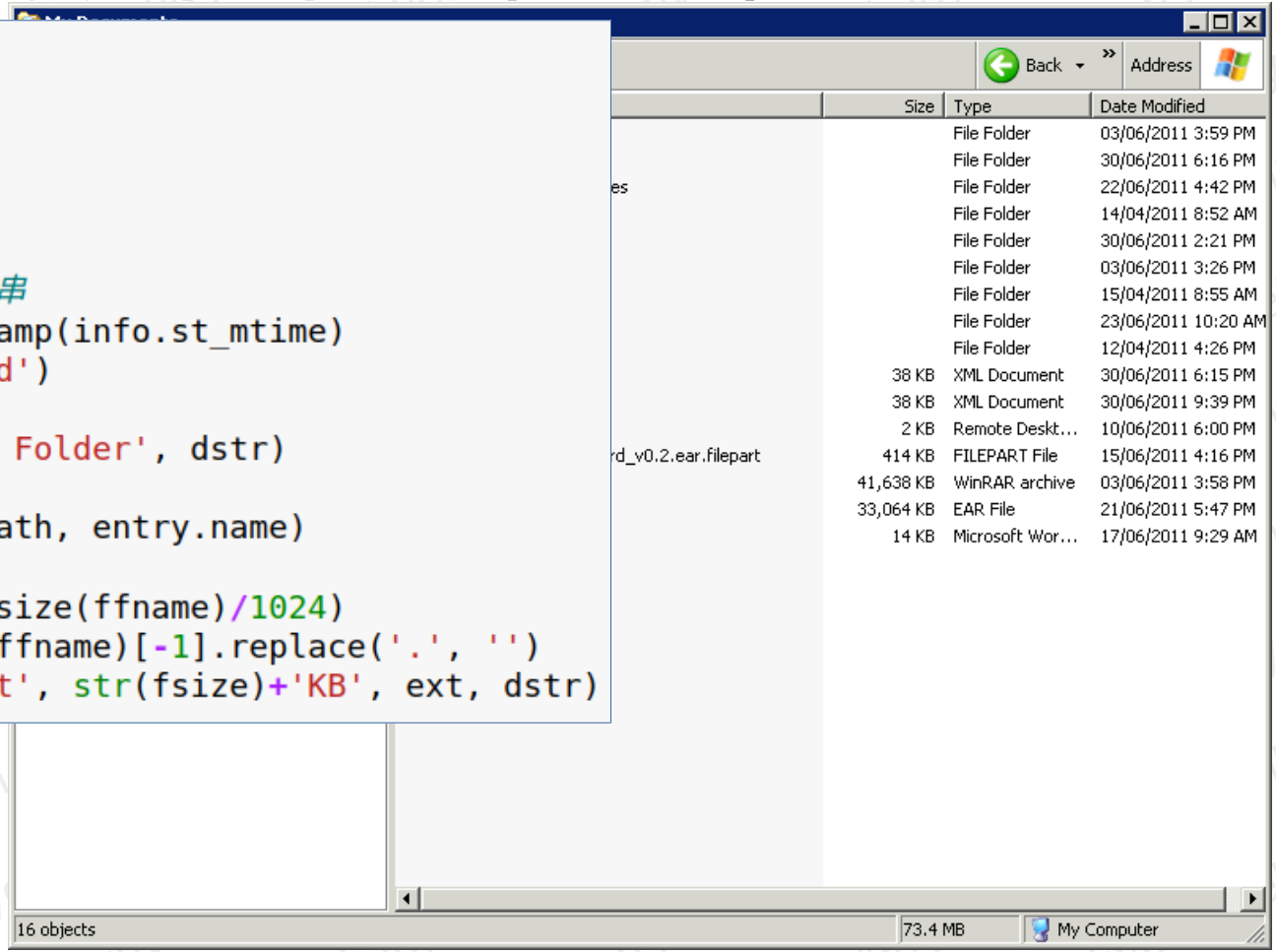
- 顯示資料夾和檔案
- 資料夾顯示最後修改時間
- 檔案顯示檔名、檔案型態、檔案大小、最後修改時間
- 怎麼做?



練習：模擬檔案總管的右頁

```python
import os
from datetime import datetime

path = '/boot'
for entry in os.scandir(path):
    info = entry.stat()
    # epoch timestamp轉換成日期字串
    da = datetime.utcfromtimestamp(info.st_mtime)
    dstr = da.strftime('%Y/%m/%d')
    if entry.is_dir():
        print(entry.name, 'File Folder', dstr)
    elif entry.is_file():
        ffname = os.path.join(path, entry.name)
        # size in byte
        fsize = int(os.path.getsize(ffname)/1024)
        ext = os.path.splitext(ffname)[-1].replace('.', '')
        print(entry.name+'\t\t\t', str(fsize)+'KB', ext, dstr)
```



- 怎麼做?

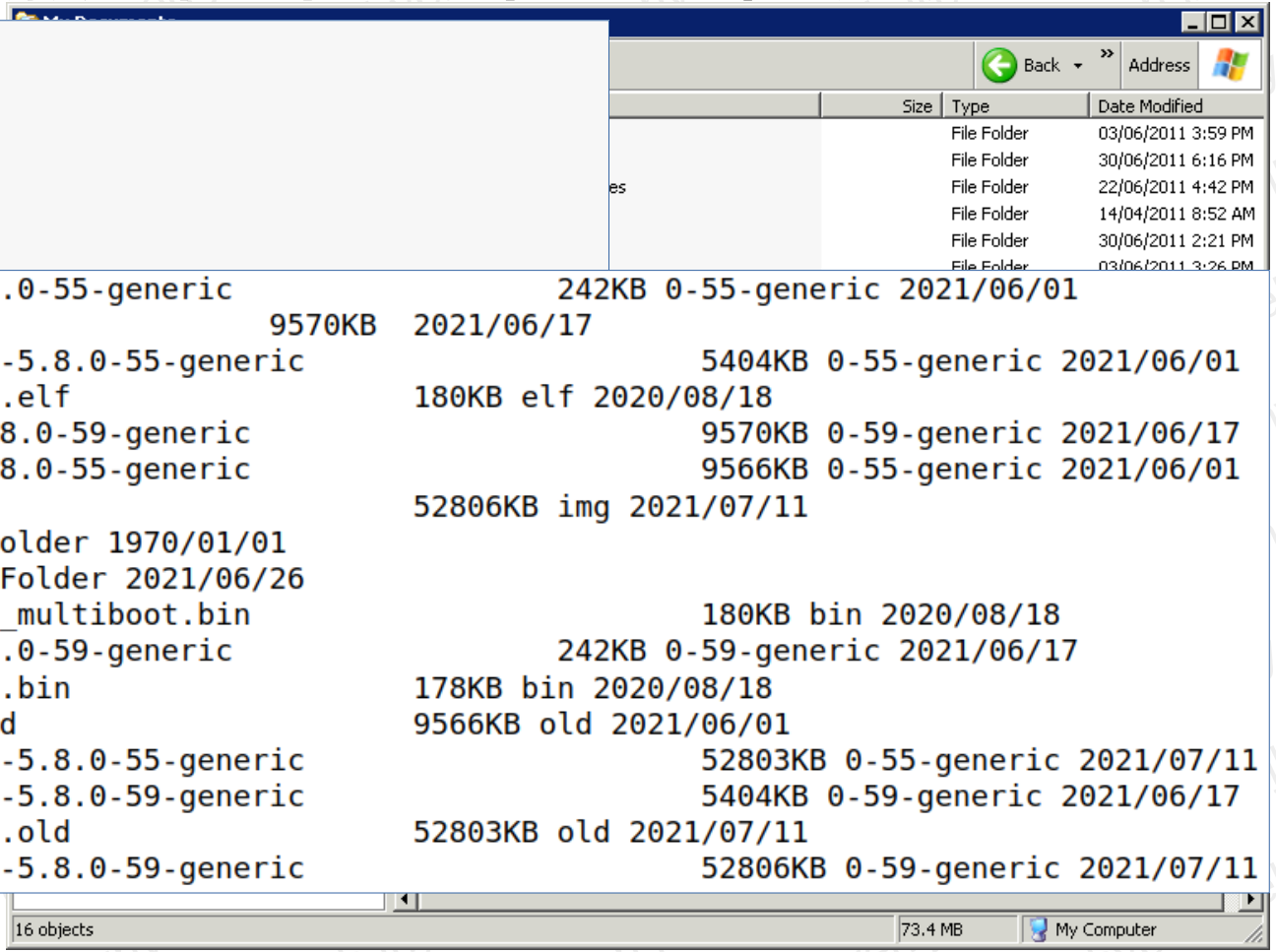
練習：模擬檔案總管的右頁

-
-
-
- 怎麼做?

```
import os
from datetime import datetime

path = '/boot'
for entry in os.scandir(path):
    info = entry.stat()
    # epoch timestamp
    da = datetime.utcf
    dstr = da.strftime
    if entry.is_dir():
        print(entry.na
    elif entry.is_file
        ffname = os.pa
        # size in byte
        fsize = int(os
        ext = os.path.
        print(entry.na
```

```
config-5.8.0-55-generic                    242KB 0-55-generic 2021/06/01
vmlinuz                    9570KB  2021/06/17
System.map-5.8.0-55-generic                    5404KB 0-55-generic 2021/06/01
memtest86+.elf                  180KB elf 2020/08/18
vmlinuz-5.8.0-59-generic                    9570KB 0-59-generic 2021/06/17
vmlinuz-5.8.0-55-generic                    9566KB 0-55-generic 2021/06/01
initrd.img                      52806KB img 2021/07/11
efi File Folder 1970/01/01
grub File Folder 2021/06/26
memtest86+_multiboot.bin                    180KB bin 2020/08/18
config-5.8.0-59-generic                  242KB 0-59-generic 2021/06/17
memtest86+.bin                  178KB bin 2020/08/18
vmlinuz.old                     9566KB old 2021/06/01
initrd.img-5.8.0-55-generic                    52803KB 0-55-generic 2021/07/11
System.map-5.8.0-59-generic                    5404KB 0-59-generic 2021/06/17
initrd.img.old                  52803KB old 2021/07/11
initrd.img-5.8.0-59-generic                    52806KB 0-59-generic 2021/07/11
```



講次內容

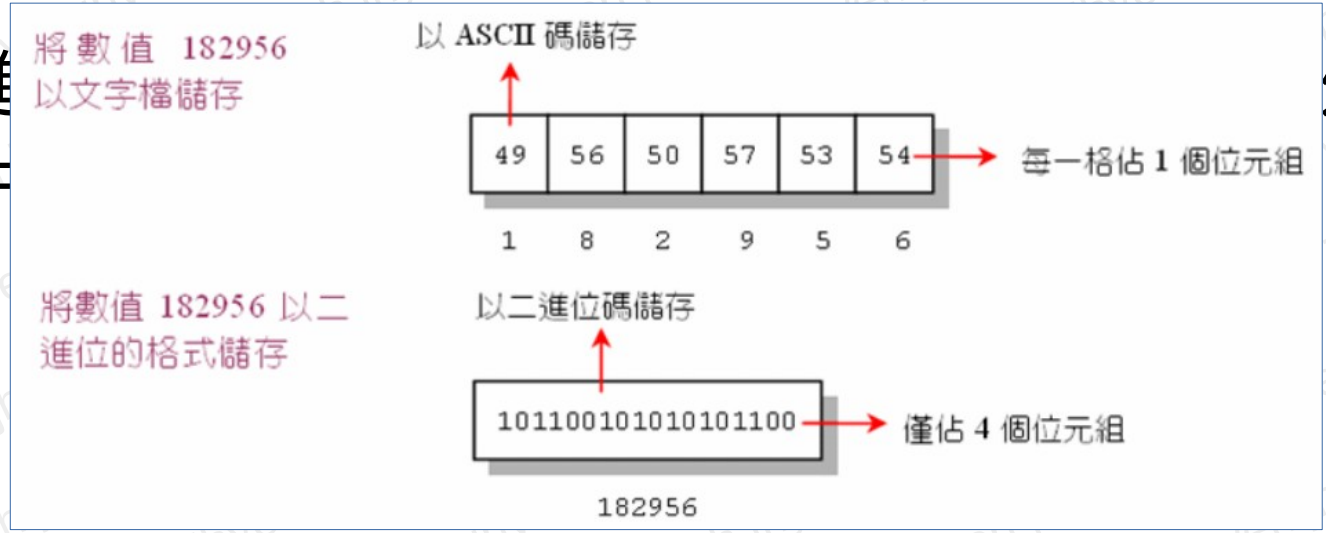
- 何謂檔案?
- 檔案的分類
- 型別 str、 bytes、 bytearray
- 文字編碼系統

檔案的分類

- 檔案依儲存方式，可分為
  - 文字檔：以列為單位，列與列之間為換行 ('\n') 字元
  - 二進位檔：以位元組 (byte) 為單位，沒有額外的 '\n' 字元

檔案的分類

- 檔案依儲存方式，可分為
  - 文字檔：以列為單位，列與列之間為換行 ('\n') 字元
  - 二進
    外的 '\n'
    字元



讀取文字檔案

- 使用內建的 open 函數來開啟
- 須指名開檔的「模式」
- 檔案是系統資源，須作開啟，結束後應關閉

```
infile = open('myfile.txt', 'r')     # 開檔，用內建的open函式
# 'r': 讀取模式

for line in infile:                  # 檔案物件支援迭代協定
    print(line, end='')              # 每次拿到一行資料，型別是str
infile.close()                       # 關閉
```

open 函數

- 函數介面： open(file, mode='r', buffering=-1, encoding=None, errors=None, newline=None)
  - file： 檔名、路徑
  - mode： 模式
  - buffering： 緩衝設定
  - encoding： 文字編碼
  - errors： 錯誤處理設定
  - newline： 換行判定規則

檔案開啟模式

- 分成文字 / 二進位、讀取 / 寫入 / 附加

模式	說明
r	讀取模式（預設）
w	寫入模式，開新檔案、或覆蓋舊檔（原舊檔內容消失）
a	附加（寫入）模式，開新檔案、或附加在舊檔尾端
x	寫入模式，若檔案不存在就開新檔案，若已存在則發生錯誤
t	文字模式（預設）
b	二進位模式
r+	更新模式，可讀可寫，檔案須已存在，從檔案開頭開始讀寫
w+	更新模式，可讀可寫，開新檔案、或覆蓋舊檔（原舊檔內容消失），從檔案開頭開始讀寫
a+	更新模式，可讀可寫，開新檔案、或從舊檔尾端開始讀寫

檔案讀寫：內容轉換

- 移除多餘的空白和空行

```
infname = 'myfile.txt'
inf = open(infname, 'r')
outfname = infname[:-4]+'2.txt'
outf = open(outfname, 'w')

for line in inf:              # 讀進來的line字串是有包含檔案內的換行字元哦!
    str1 = line.strip()       # 移除line的多餘空白
    if len(str1)>0:           # 如果移除完還有內容，寫進輸出檔
        outf.write(str1+'\n')
inf.close()
outf.close()
```

文字編碼

- open 的參數 encoding 是用來指定檔案使用的文字編碼
  - 若未指定，文字檔開啟時會使用作業系統的預設編碼
  - Windows： CP950 （或稱 MS950 、 Big5 ）
  - Linux： UTF-8

常見的支援編碼

編碼名稱	說明
ascii(us-ascii)	就是 ASCII 編碼
iso-8859-1(latin_1)	
cp1252	西歐 Windows 系統
big5(big5-tw)	台灣使用的繁體中文
cp950(ms950)	等同於 big5
big5hkscs	香港擴充 Big5
gb2312/gbk	簡體中文碼
utf8(utf_8)/utf_8_sig	
utf_16/utf_16_be/utf_16_le	

read、 readline、 readlines

- 檔案讀寫大都使用迭代法，這些函數不常用
  - read(size)： 最多讀取 size 個字元，若 size 為負數或 None，讀到檔案結尾 (EOF)
  - readline(size=-1)： 讀取一行或到 EOF，若指定 size, 最多讀取 size 個字元
  - readlines(hint=-1)： 讀取好幾行，放進 list，若指定 hint, 最多讀取 hint 個字元

read、 readline、 readlines

- 檔案讀寫大都使用迭代法，這些函數不常用

```
infname = 'myfile.txt'
inf = open(infname, 'r')

# 這樣寫不太優!
line = inf.readline()          # 讀進來的line字串是有包含檔案內的換行字元哦!
while line:
    print(line, end='')        # 避免連續輸出兩次換行字元
    line = inf.readline()

# 這樣寫不是比較整齊清潔嗎? !
for line in inf:               # 讀進來的line字串是有包含檔案內的換行字元哦!
    print(line, end='')        # 避免連續輸出兩次換行字元

inf.close()
```

- 讀到
- ize
- 取

二進位檔案讀寫

- 位元組 (byte) 資料是二進位的（0 與 1）
  - 8 個位元，通常以 16 進位表示
- 讀取二進位檔案

```
inf = open("myfile.jpg", "rb")     # jpeg圖檔就是二進位檔案

byte = inf.read(1)                 # 一次讀一個byte
while byte:
    print(byte)
    byte = inf.read(1)             # 繼續讀下一個byte

inf.close()
```

使用 with 敘述

- 可以確保檔案不再使用時的適當關閉

```
with open("myfile.txt",'r') as f:
    lines = f.readlines()
    for line in lines:
        print(line, end='')

# 不用關檔了！！
```

講次內容

- 何謂檔案?
- 檔案的分類
- 型別 str、 bytes、 bytearray
- 文字編碼系統

str 和 bytes

- str 型別：即 Unicode 字串，是一種序列型別，一個個字元組合而成
  - 'This is 中文 '

這樣有幾個字元?

- bytes 型別：位元組資料型別，是一種序列型別，當做一個個位元組
  - b'\x80\xAB\x12\xFF'

這樣有幾個位元組?

bytes

- bytes 型別：一個個位元組以 0~255 之間的 int 表示，且 bytes 內容不可變!

```
data = b'wxy\x7a'
print(data)                 # b'wxyz', 以ASCII字元輸出

print(type(data), type(data[0]))
# <class 'bytes'>, <class 'int'>

print(data[0], hex(data[0]))
# 'w' ASCII碼119, 十六進位'0x77'

print(b'\x7a' in data)      # 可以用 in 來判斷
print(data[2:])             # 可以切片
```

bytearray

- 就是可變的 bytes！

```
data = b'wxy\x7a'
print(data)                    # b'wxyz', 以ASCII字元輸出

ba = bytearray(data)
print(type(ba), type(ba[0]))
# <class 'bytearray'>, <class 'int'>

ba[3] = 0x70                   # 修改資料
print(ba)                      # 變成 bytearray(b'wxyp')
```

講次內容

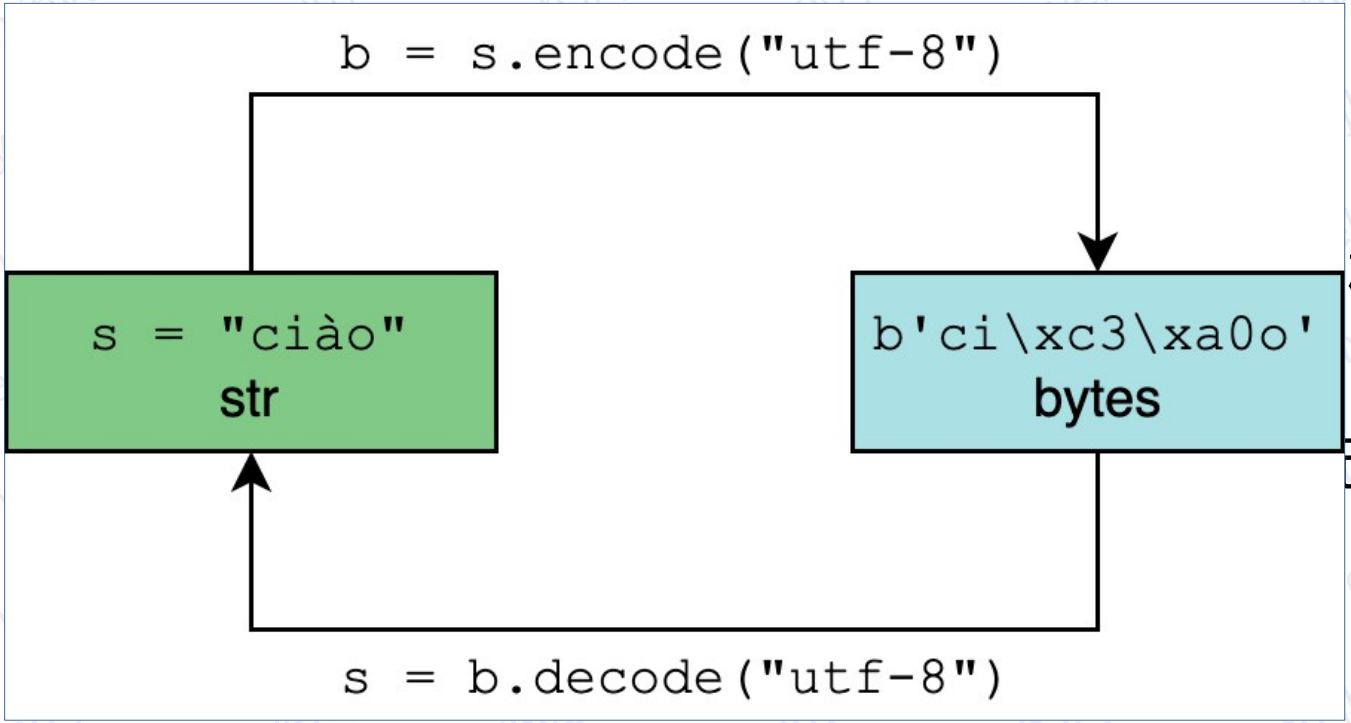
- 何謂檔案?
- 檔案的分類
- 型別 str、 bytes、 bytearray
- 文字編碼系統

文字編碼系統

- 就是配合各種字元集 (Character Set) 的編碼 / 解碼方法架構
  - 字元： 3、 A、空格 ' '、換行 '\n'、 電、ㄅ、ぬ、 é、☺
  - 字元碼：字元對應的碼 (數字)
  - 字元集 (character set)： 字元＋字元碼，也稱為字碼頁 (code page)
  - 編碼方法 (encoding scheme): 把一串字元轉成位元組的規則

文字編碼系統

- 就是配合各種字元集 (Character Set) 的編碼 / 解碼方法架構



- é、☺
- 碼頁 (code
- 元組的規則

編碼與解碼

- 相同字元，不同編碼

```
str1 = '葉'

print(str1.encode('big5'))
# 轉big-5編碼, b'\xb8\xad'

print(str1.encode('gbk'))
# 轉bgk編碼, b'\xc8~'

print(str1.encode('utf-8'))
# 轉utf-8編碼, b'\xe8\x91\x89'
```

- 相同位元，不同字元

```
data = b'\xB8\xAD'

print(data.decode('big5'))
# 將bytes轉為big5文字, '葉'

print(data.decode('gbk'))
# 將bytes轉為gbk文字, '腑'

print(data.decode('utf-8'))
# 將bytes轉為ytf-8文字, 解碼規則無法解碼
```

類檔案物件 stdin、 stdout

- 淺談標準輸出入

```
import sys                              # 要使用 stdout...

with open('stdout.txt', 'w', encoding='utf-8') as fout:
    backup = sys.stdout                 # 先將標準輸出物件「備份」起來
    sys.stdout = fout                   # 將標準輸出「重新導向」到檔案
    print('標準輸出重新導向')            # 螢幕上顯示不出來了！！
    sys.stdout = backup                 # 還原標準輸出物件
```

這個講次中，你應該學到了 ...

- 檔案系統的基本操作：資料夾、檔案屬性
- 文字檔案和二進位檔案的操作
- str 、 bytes 、 bytearray 的差異
- 文字編碼的轉換操作