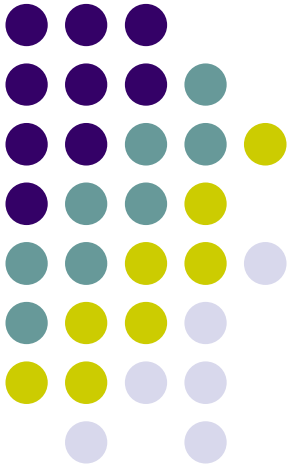
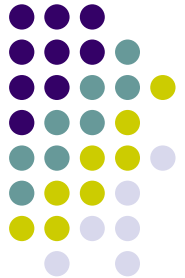
Python程式設計入門
檔案、位元組、文字

葉難

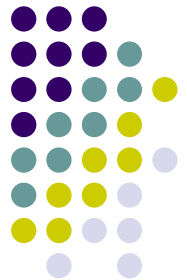


大綱



- 檔案
- 資料：位元組（byte）、文字
- 型別str、bytes、bytearray
- 文字編碼系統
- 檔案格式：二進位、CSV、JSON

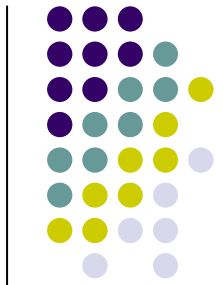
簡單範例：讀取文字檔並印出 file_basic.py



```
fin = open('file_basic.txt', 'r' )  # 開啓
        # 內建函式，開檔                  # 讀取模式
for line in fin:                  # 檔案物件支援迭代協定
    print(line, end='')           # 每次拿到一行
                                  # line的型別是str

fin.close()                                  # 關閉
```

- 檔案是系統資源，須作開啓，結束後應關閉

open

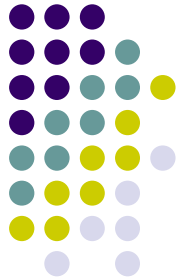


- 3.x版的內建函式open，就是模組io的函式open
- 2.x版的內建函式open，舊，沒有參數encoding；若想使用3.x版的open：

```
from io import open
```

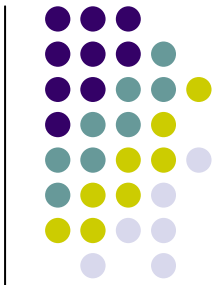
- 模組codecs的函式open：舊，不該使用

open的參數（部分）



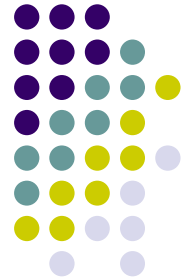
- open(file, mode='r', buffering=-1, encoding=None, errors=None, newline=None)
- file：檔名、路徑
- mode：模式
- buffering：緩衝設定
- encoding：文字編碼系統。應使用關鍵字參數
- errors：錯誤處理設定
- newline：換行判定規則

模式



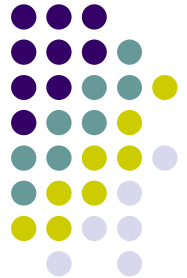
模式	說明
r	讀取模式（預設）
w	寫入模式，開新檔案、或覆蓋舊檔（原舊檔內容消失）
a	附加（寫入）模式，開新檔案、或附加在舊檔尾端
x	寫入模式，若檔案不存在就開新檔案，若已存在則發生錯誤
t	文字模式（預設）
b	二進位模式
r+	更新模式，可讀可寫，檔案須已存在，從檔案開頭開始讀寫
w+	更新模式，可讀可寫，開新檔案、或覆蓋舊檔（原舊檔內容消失），從檔案開頭開始讀寫
a+	更新模式，可讀可寫，開新檔案、或從舊檔尾端開始讀寫

移除空行（含TAB、空格等）



```
f1 = 'file_basic.txt'
fin = open(f1, 'r')
fout = open(f1[:-4] + '_removed.txt', 'w')
# ...省略...
for line in fin:
    s = line.strip()  # 移除左右兩端的空白字元
    if s:
        fout.write(line)
```

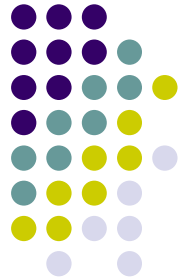
文字編碼系統



- open的參數「encoding」：指定編碼
- 若未指定，開啓文字檔時，系統預設使用模組locale的getpreferredencoding(False)的回傳值

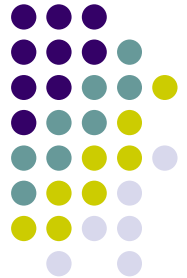
- Windows系統應會是'cp950'
- 若文字檔編碼爲ASCII，相容於CP950（Big5）
- 若文字檔編碼爲UTF-8或其他，出錯

Python支援的文字編碼系統



- ascii（us-ascii）
- iso-8859-1（latin_1）
- cp1252：西歐Windows系統
- big5（big5-tw）：台灣
- cp950（ms950）：幾乎等同於big5
- big5hkscs：香港擴充Big5
- gb2312、gbk：中國大陸
- utf8（utf_8）、utf_8_sig（加上BOM）
- utf_16，utf_16_be、utf_16_le

with述句



- 文脈管理協定（context management protocol）
- 文脈管理器：__enter__、__exit__
- 檔案物件支援文脈管理協定
- 語法：

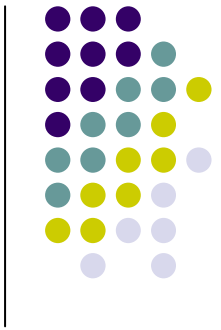
```
with open('file', 'r', encoding='utf-8') as f:
    ...使用檔案物件f...
```

read、readline、readlines

- read(size)：最多讀取size個字元；若size爲負數或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。

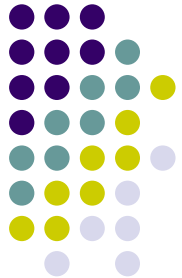
- 檔案物件支援迭代協定，幾乎不需要上述方法

不好的舊寫法



```
for line in fin.readlines():   # 含所有行的串列
    print(line, end='')

####

line = fin.readline()
while line:
    print(line, end='')
    line = fin.readline()
```

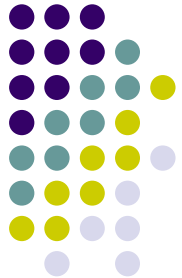
CSV



- knmi.py
- 自己解析處理
- 使用模組csv：reader、writer
- 範例檔內容如下：

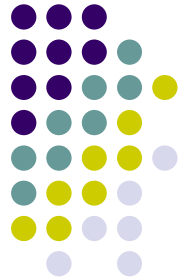
```
215,20160101,    49,    -5,    81
215,20160102,    62,    49,    73
215,20160103,    72,    53,    89
```

JSON



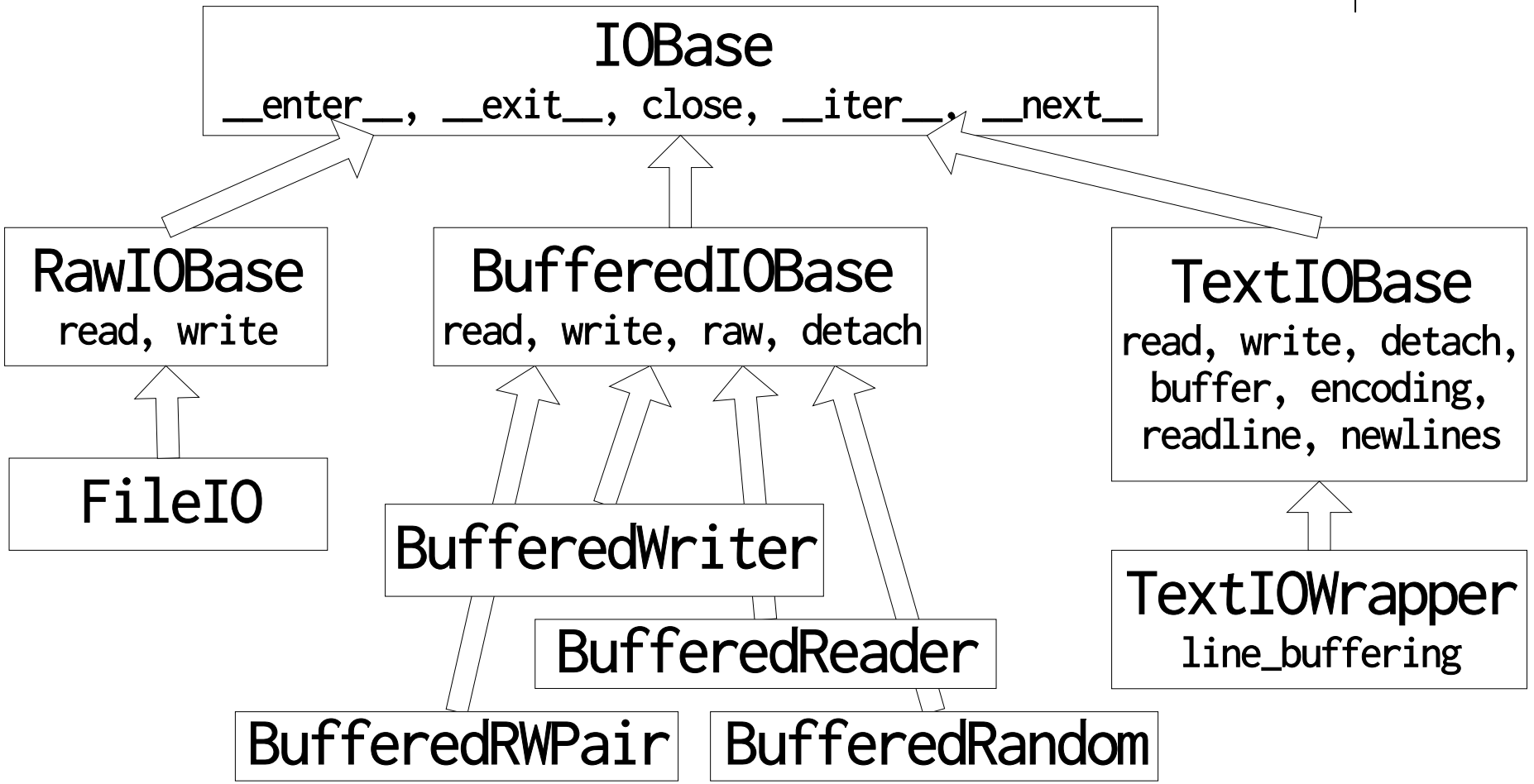
- 範例：USGS地震觀測資訊網站
  http://earthquake.usgs.gov/earthquakes/feed/v1.0/geojson.php
- 美化：https://jsonformatter.curiousconcept.com/
- 模組json：load、dump
- 範例earthquake.py、earthquake_data.json
- 了解每個項目放在哪裡
- 此範例只抓出地點、震度、時間

檔案



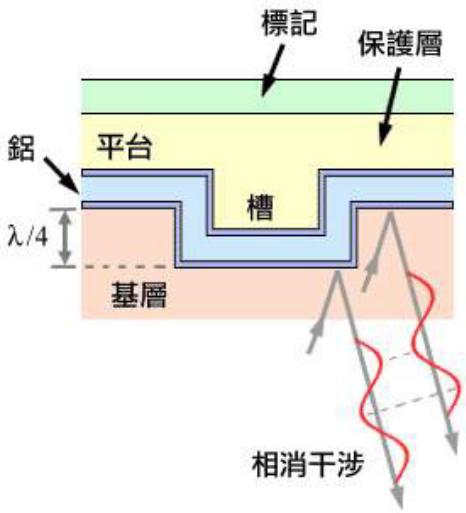
- 型別爲檔案的物件，提供存取背後儲存裝置或傳輸媒介的介面
- 資料來源各有不同特性：唯讀、任意位置讀寫、只能循序讀取、速度快慢等等
- 有些物件也提供類似於檔案的存取介面，但可能只有部分
- 或另稱爲類檔案（file-like）、串流（stream）

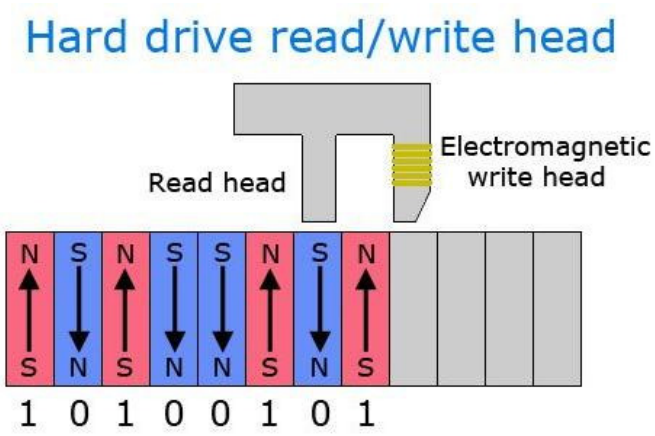
模組io的抽象型別與支援方法

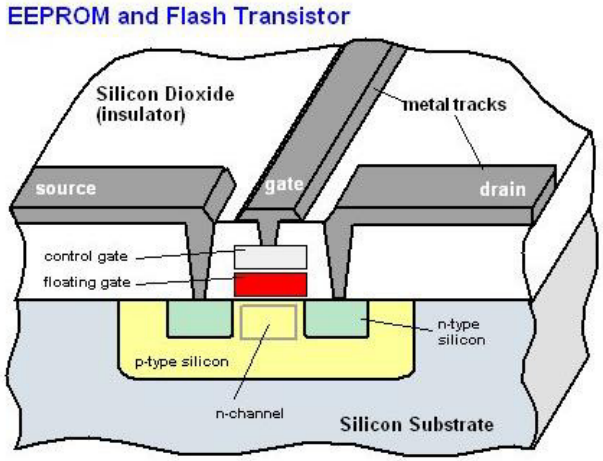


數位（二進位）資料

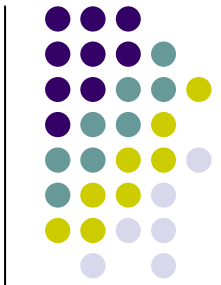
- 底層技術，抽象化成01010101...







位元組（**byte**）資料



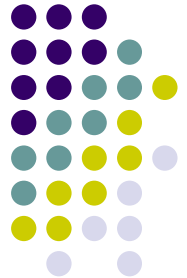
- 二進位：0與1
- 位元組（byte）：8個位元，通常以16進位表示

- HEX編輯器：HxD(Win)、0xED(Mac)、weHexEditor(Linux)、hexedit(Linux)
- 範例：hex.py

2.x與3.x版本差異

- 2.x版，型別str只能含有八位元字元（例ASCII），同時也擔任位元組（二進位）資料的型別
- 2.x版，Unicode字串由型別unicode負責
- 3.x版，型別str就是Unicode字串，無unicode型別
- 3.x版，以型別bytes負責位元組（二進位）資料

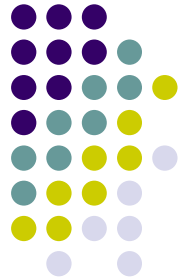
型別str與bytes



- 型別str，Unicode字串，裡頭是一個個字元，不要當做位元組
- 型別bytes，位元組資料型別，當做一個個位元組。字面值語法「b'\x00\xFF'」

```
>>> data = b'\x80\xAB\x12\xFF'
>>> data
b'\x80\xab\x12\xff'
```

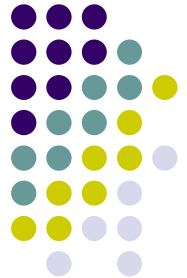
型別bytes



- 型別bytes，序列型別，位元組資料型別，裡頭是一個個int物件（介於0~255）

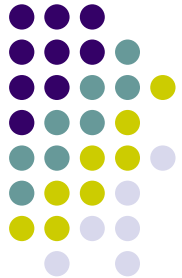
```
>>> data = b'abc\x64'
>>> data          # Python直譯器好心（雞婆）
b'abcd'           # 以ASCII字元輸出
>>> type(data), type(data[0])
(<class 'bytes'>, <class 'int'>)
>>> data[0], hex(data[0])        # 序列型別
(128, '0x80')
```

型別bytes是不可變序列型別



```
>>> data = b'\x03\x04\x05\x06\xff'

>>> b'\x03' in data          # 成員關係運算子
True
>>> data[2:]                 # 切片
b'\x05\x06\xff'

>>> b'0x03'      # 注意，這是4個位元組
b'0x03'
```

型別bytearray



- 可變的bytes

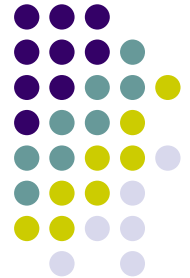
```
>>> data = b'\x03\x04\x05\x06\xff'
>>> ba = bytearray(data)
>>> ba
bytearray(b'\x03\x04\x05\x06\xff')
>>> ba[3]
6
>>> ba[3] = 0xAB    # 修改
>>> ba[3]
171
```

編碼（encode）、解碼（decode）

- 資訊表達形式的轉換
- 聲音、WAV、MP3、喇叭、麥克風
- 輪子轉動：累積里程數、瞬時速率

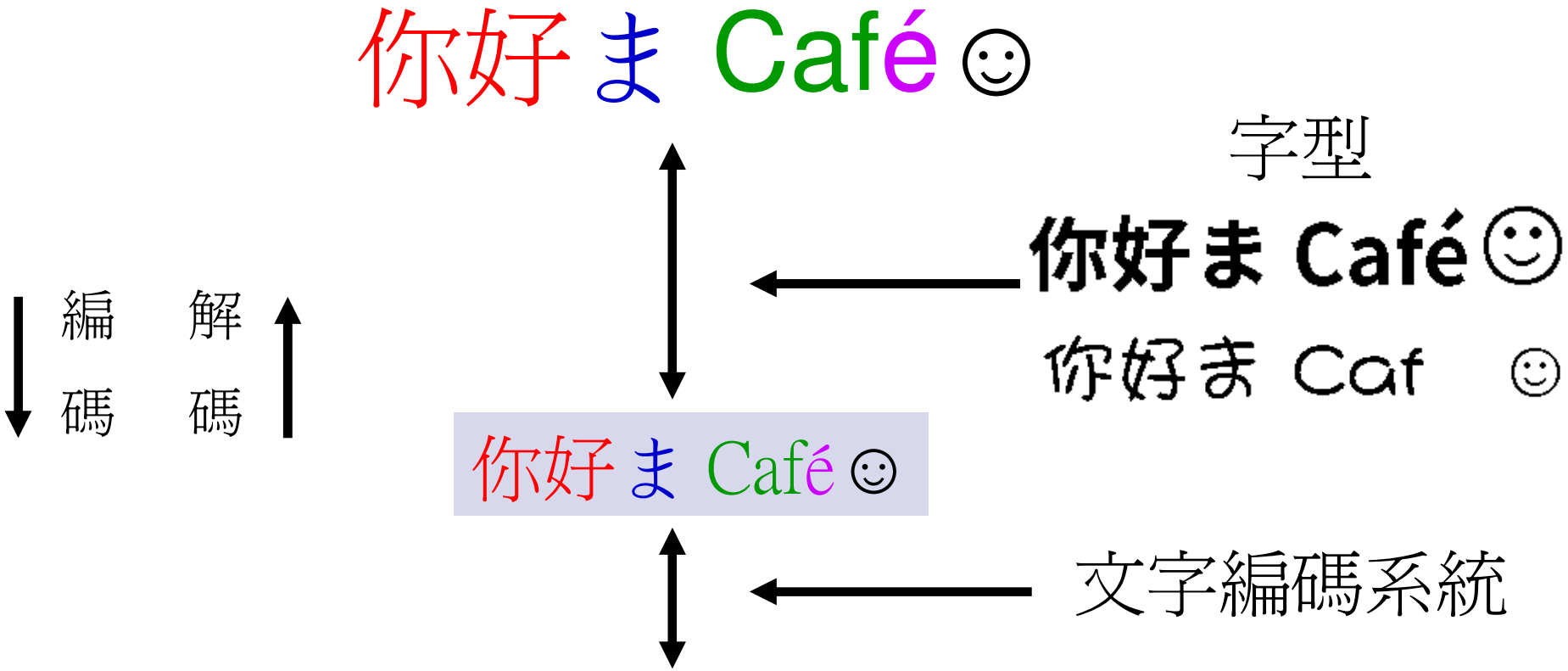
- 一般來說，朝向「人」的方向是解碼、遠離人的方向是編碼
- 文字編碼系統

文字編碼系統 ＝ 字元集 ＋ 編碼方法



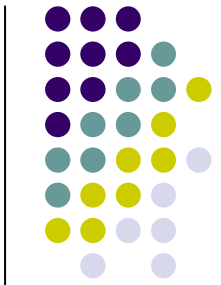
- 字元（character）：3、A、「空格」、「換行」、電、ㄅ、ぬ、é、☺、ʊ_ʊ
- 字元碼：字元對應的碼（數字）
- 字元集（character set）：字元和字元碼
  也可稱爲字碼頁（code page）
- 編碼方法（encoding scheme）：把一串字元轉成位元組的規則

處理文字需經過一堆程序



E4 BD A0 E5 A5 BD E3 81 BE 20 43 61 66 C3 A9 E2 98 BA

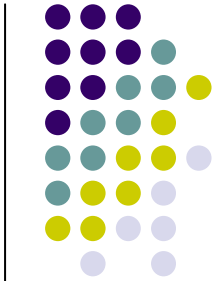
ASCII



- ASCII字元集（請看講義）
- 8位元編碼方法：一個ASCII字元對應到一個位元組（byte）
- 「A」編碼爲「0x61」（0b01100001）

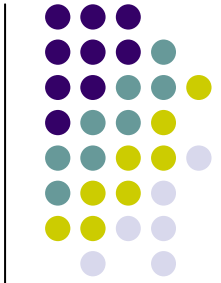
- ASCII字元集僅定義了128個字元，佔位元組的一半，其第7個位元皆爲0

Big5字元集



字元碼	字元	說明	字元碼	字元	說明
A140		全形空格	A148	？	全形問號
A1B4	●	實心圓	A1F8	↖	符號
A275	├	框線	A2A3	┘	框線
A2B2	3	全形數字	A2E8	Z	全形字母
A374	ㄅ	注音符號	A440	一	中文字
A441	乙	中文字	B344	蛇	中文字
C67E	籲	中文字	C940	乂	中文字
D662	虜	中文字	F9DC	嫺	中文字

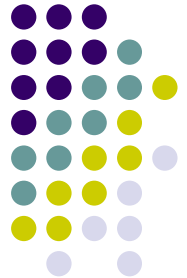
Big5編碼方法



- 相容於ASCII
- 一個字元可能編碼爲一或二個位元組

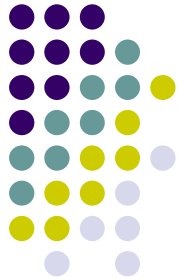
字元	a	b	蛇	ㄅ	c	嫺
十六進位	61	62	B3 44	A3 74	63	F9 DC

Unicode字元集



字元碼（code point）	字元	簡述
U+41	A	拉丁（英文）字母大寫
U+E9	é	有重音符號的拉丁字母
U+2523	┣	框線
U+3010	【	中文標點符號
U+3042	あ	日文平假名
U+6DA9	涩	簡體中文漢字
U+6E0B	渋	日文漢字
U+6F80	澀	繁體中文漢字
U+FEFF		BOM（Byte Order Mark）
U+FFFF		非有效字元

同一字元，不同編碼

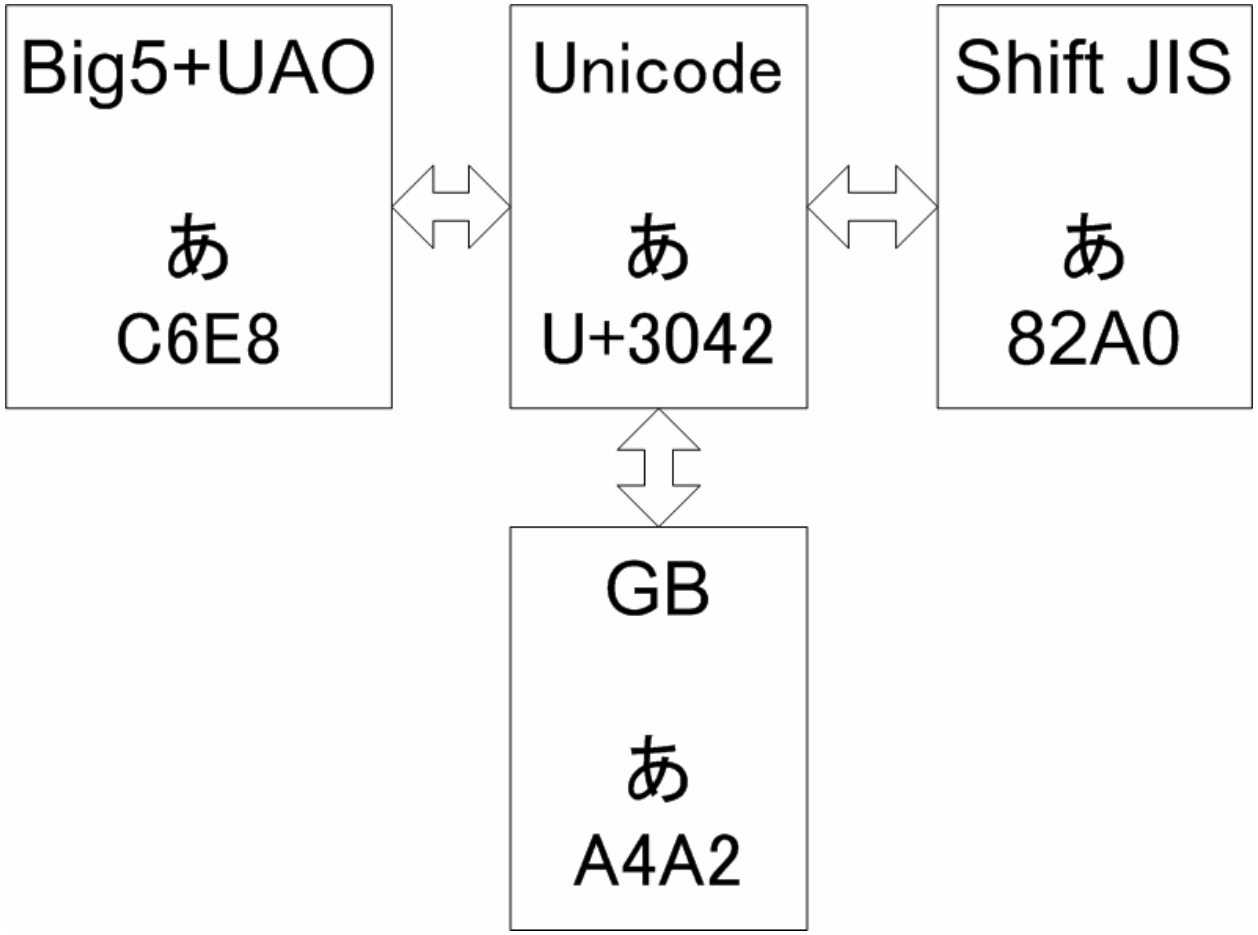


```
>>> s = '闡'
>>> s.encode('big5')
b'\xc4\xc4'
>>> s.encode('gbk')
b'\xeaU'
>>> s.encode('utf-8')
b'\xe9\x97\xa1'
```

同一編碼，不同字元

```
>>> data = b'\xB0\xA1'
>>> data.decode('big5')
'陛'
>>> data.decode('gbk')
'啊'
```

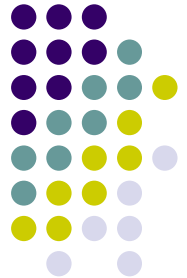
轉換



亂碼


三民網路書店
www.sanmin.com.tw
三民63週年慶
2016.7.10-8.8
我的確幸時光
購買三民東大出版品，滿399元送大象便條紙、800元送皮皮與波西書籤一組。
主題書展
進階查詢
繁體書查詢
簡體書查詢
當日宅配
簡體書
新書搶先報
好書新到貨
青春・原創
書展專區
原文書
新到貨 New arrivals
NYT 暢銷小說 Bestsellers
Movies 電影館
Bargains 特賣會
Secured By TWCA 臺灣認證 已認證
熱搜關鍵字
The Jungle Book硬格
Make Believe Ideas Phonics
Lulu 觸餘
Collins 摯
Usborne Lift-the-flap觸餘
Rainbow Magic
Usborne See Inside觸餘
Usborne Look Inside觸餘
2016
Alice in Wonderland

文字編碼轉換

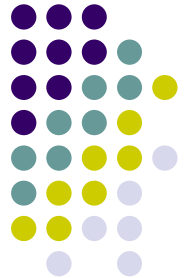


- GBK轉Big5：vase_tr.py

- 問題：test_hk.txt，使用香港的編碼big5hkscs，請轉成UTF-8，應正確顯示：

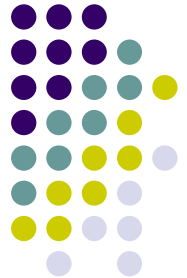
匯豐銀行警衛室
滙豐銀行警衛室

pickle醃製：物件轉位元組



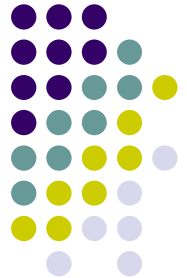
- serialization、marshalling、flattening、archiving
- 模組pickle，Python專屬格式
- 方法load、dump
- 醃製協定：已有許多版本
- 不具安全性檢查功能

pickle範例：pickle.py



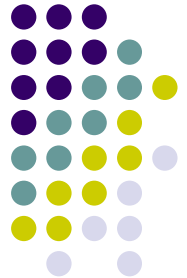
```
data = {'a': [1, 2, 3, 4],
        'John': 'abcdef',
        334455: 778899,   'default': None}

with open('pickle.bin', 'wb') as fout:
    pickle.dump(data, fout)

with open('pickle.bin', 'rb') as fin:
    d = pickle.load(fin)
    print(d['John'])
    print(d[334455])
```

模組**sys**



- sys.getdefaultencoding()，.py檔預設編碼，
  Win Py 2.x: 'ascii'，Win Py 3.x: 'utf-8'
- sys.stdin.encoding，標準輸入（鍵盤），
  Win: 'cp950'
- sys.stdout.encoding，標準輸出（螢幕），
  Win: 'cp950'

sys.stdin、sys.stdout



- 也是類檔案物件
- 轉向（redirection）
- print('Hello', file=fout)

```
import sys
with open('fout.txt', 'w', encoding='utf-8')
  as fout:
    backup = sys.stdout
    sys.stdout = fout
    print('Hello')
    sys.stdout = backup
```

Q&A