Python 3.x 網頁資料擷取與分析特訓教材

勘誤表

最後更新日期：2024/3/14

勘誤表閱讀前準備事項：

1.請先確認本產品是否有提供系統更新檔下載並且已下載完畢進行手動更新。

2.或是直接開啟練習系統或模擬測驗系統執行「查詢系統最新版本」以完成線上自動更新。

3.本勘誤內容必須搭配下載系統更新檔手動更新或進行系統更新後，才會使系統內容與勘誤一致。

壹、內容勘誤

章節	頁次	行	勘誤前	勘誤後	更新日期
Chapter 0	0-23	4 ~ 6	安裝 Python 第三方函式庫建議使用 pip，此一程式在 Python 2（>=2.7.9 版本）或 Python 3（>=3.4 版本）安裝時就已經安裝完畢，可以直接使用。若無法使用…	安裝 Python 第三方函式庫建議使用 pip 或 pip3，如果系統只安裝了 Python 2，則只能使用 pip。如果系統只安裝了 Python 3，則既可以使用 pip 也可以使用 pip3，二者是等價的。如果系統中同時安裝了 Python 2 和 Python 3，則 pip 預設給 Python 2 用，pip3 指定給 Python 3 用。若無法使用…	20240314

<table>
<tr><th>章節</th><th>頁次</th><th>行</th><th>勘誤前</th><th>勘誤後</th><th>更新日期</th></tr>
<tr><td>Chapter 1</td><td>1-8</td><td>24
25</td><td>▶▶ 範例程式 E1-1-2-4.py
<pre>import PyPDF2
pdfFiles = ["E1-1-1-1-out0.pdf","E1-1-1-1-out1.pdf",
"E1-1-1-1-out2.pdf"]
pdfWriter = PyPDF2.PdfFileWriter()
pdfOutput = open(' E1-1-2-4-comb.pdf','wb')
for fileName in pdfFiles:
    pdfReader = PyPDF2.PdfFileReader(open(fileName,'rb'))
    for pageNum in range(pdfReader.numPages):
   print(pdfReader.getPage(pageNum))
   pdfWriter.addPage(pdfReader.getPage(pageNum))
pdfWriter.write(pdfOutput)
pdfOutput.close()</pre></td><td>▶▶ 範例程式 E1-1-2-4.py
<pre>import PyPDF2
pdfFiles = ["E1-1-1-1-out0.pdf","E1-1-1-1-out1.pdf",
"E1-1-1-1-out2.pdf"]
pdfWriter = PyPDF2.PdfFileWriter()
pdfOutput = open(' E1-1-2-4-comb.pdf','wb')
for fileName in pdfFiles:
    pdfReader = PyPDF2.PdfFileReader(open(fileName,'rb'))
    for pageNum in range(pdfReader.numPages):
        print(pdfReader.getPage(pageNum))
        pdfWriter.addPage(pdfReader.getPage(pageNum))
pdfWriter.write(pdfOutput)
pdfOutput.close()</pre></td><td>20240314</td></tr>
<tr><td>Chapter 1</td><td>1-9

1-10</td><td>26
~
31
1</td><td>PDFMiner 是一個可以從 PDF 檔案中提取資訊的工具，注重的是獲取和分析文字資料，允許獲取某一頁中文字的準確位置和一些諸如字體、行數的資訊。包括可以把 PDF 檔轉換成 HTML 等格式（但不能直接看）的 PDF 轉換器、可以用於除文字分析以外其他用途的 PDFParser（檔案分析器）、PDFDocument（檔案物件）、PDFResourceManager（資源管理器）、PDFPageInterpreter（解譯器）、PDFPageAggregator（聚合器）與 LAParams（參數分析器）。轉換的整體思維是先建構檔案物件，接著利用 PDFParser 解析檔案物件，然後提取所需內容。如下範例所示：</td><td>PDFMiner 是一個 Python 套件，主要用於從 PDF 文件中提取和分析文本資訊。如果沒有安裝過，使用前要用以下的命令在 Anaconda prompt 安裝這個套件：pip(或 pip3) install pdfminer.six 下面這段 Python 程式碼的主要目的是從一個 PDF 文件中提取文本並將其寫入到一個文本文件中。程式碼首先從 pdfminer 模組中導入 extract_text 函數和 LAParams 類。extract_text 函數用於從 PDF 文件中提取文本，而 LAParams 類用於設定佈局分析的參數。接著定義了一個名為 parse 的函數，該函數負責從輸入的 PDF 文件中提取文本並將其寫入到輸出的文本文件中。在 parse 函數中，程式碼使用 extract_text 函數從輸入的 PDF 文件中提取文本。這裡使用了 LAParams 來設定佈局分析的參數。提取的文本被寫入到輸出的文本文件中。如下範例所示：</td><td>20240314</td></tr>
</table>

章節	頁次	行	勘誤前	勘誤後	更新日期

Chapter 1	1-10	4 ~ 30	範例程式 **E1-1-3-1.py**
			`from pdfminer.pdfparser import PDFParser, PDFDocument`
			`from pdfminer.pdfinterp import PDFResourceManager, PDFPageInterpreter`
			`from pdfminer.layout import LAParams`
			`from pdfminer.converter import PDFPageAggregator`
			`from pdfminer.pdfinterp import PDFTextExtractionNotAllowed`
			`def parse():`
			`  fn = open('E1-1-2-2-input.pdf','rb')`
			`  parser = PDFParser(fn)`
			`  doc = PDFDocument()`
			`  parser.set_document(doc)`
			`  doc.set_parser(parser)`
			`  doc.initialize("")`
			`  if not doc.is_extractable:`
			`  raise PDFTextExtractionNotAllowed`
			`  else:`
			`  resource = PDFResourceManager()`
			`  laparams = LAParams()`
			`  device = PDFPageAggregator(resource,laparams=laparams)`
			`  interpreter = PDFPageInterpreter(resource,device)`
			`  for page in doc.get_pages():`
			`  interpreter.process_page(page)`
			`  layout = device.get_result()`
			`  for out in layout:`
			`  if hasattr(out,"get_text"):`
			`  with open('E1-1-3-1-output.txt','a') as f:`
			`  f.write(out.get_text()+'\n')`
			`if __name__ == '__main__':`
			`  parse()`
				範例程式 **E1-1-3-1.py**
				`from pdfminer.high_level import extract_text`
				`from pdfminer.layout import LAParams`
				`def parse():`
				`  input_pdf = 'E1-1-2-2-input.pdf'`
				`  output_txt = 'E1-1-3-1-output.txt'`
				`  # 使用pdfminer的高階函式來提取文本`
				`  text = extract_text(input_pdf, laparams=LAParams())`
				`  # 將提取的文本寫入輸出文件`
				`  with open(output_txt, 'a') as f:`
				`  f.write(text)`
				`if __name__ == '__main__':`
				`  parse()`
					20240314

Chapter 1	1-10 1-11	34 35 1 ~ 18	· 1-5 行 import 所需套件。 · 6 行開始解析與獲取所需資訊。 · 7 行設定輸入的'E1-1-2-2-input.pdf'，'rb'參數表示以二進位元讀取模式打開本機 PDF 檔案，此為衛福部的國人全民健康保險就醫疾病資訊。 · 8 行創建一個 PDF 檔案分析器。 · 9 行創建一個 PDF 檔案。 · 10-11 行連接分析器與檔案物件。 · 12 行提供初始化密碼，如果沒有密碼就創建一個空的字串。 · 13-15 行檢測檔案是否提供 txt 轉換，不提供就忽略，執行 15 行以後程式。 · 16 行創建 PDF 資源管理器。 · 17 行創建一個 PDF 參數分析器。 · 18 行創建聚合器，用於讀取檔案的物件。 · 19 行創建解譯器，對檔案編碼，解釋成 Python 能夠識別的格式。 · 20 行的 doc.get_pages()獲取 page 列表，然後用迴圈遍歷清單，每次處理一頁的內容。 · 21 行利用解譯器的 process_page()方法解析讀取單獨頁數。 · 22 行使用聚合器的 get_result()方法獲取內容。 · 23 行的 layout 是一個 LTPage 物件，裡面存放著這個 page 解析出的各種物件，利用迴圈來遍歷。 · 24-26 行判斷是否含有 get_text()方法，獲取我們想要的文字，然後採用附加方式寫入'E1-1-3-1-	· 1-2 行是從 pdfminer 模組中導入所需的函數和類。extract_text 是一個高階函數，用於從 PDF 文件中提取文本。LAParams 是一個類，用於設定佈局分析的參數。 · 3-5 行定義了一個名為 parse 的函數，並設定了輸入 PDF 文件和輸出文本文件的名稱。 · 7 行設定輸入的'E1-1-2-2-input.pdf'，'rb'參數表示以二進位元讀取模式打開本機 PDF 檔案，此為衛福部的國人全民健康保險就醫疾病資訊。 · 8 行呼叫 extract_text 函數來從輸入的 PDF 文件中提取文本，並將提取的文本存儲在變數 text 中。 · 11-12 行將提取的文本寫入到輸出的文本文件中。 · 14-15 行呼叫 parse()。	20240314

<table>
<tr><th>章節</th><th>頁次</th><th>行</th><th>勘誤前</th><th>勘誤後</th><th>更新日期</th></tr>
<tr><td></td><td></td><td></td><td>output.txt'。
· 27-28 行呼叫 parse()。</td><td></td><td></td></tr>
<tr><td>Chapter 1</td><td>1-20</td><td>6
~
10</td><td>範例程式 E1-1-5-1.py
<pre>#文字檔讀取與寫入：台積電存股規劃
import math
with open('data5.txt', 'r') as fin:
    with open('data5_w.txt', 'w') as fout:
   for line in fin:
  data=math.ceil(20/(float(line)*0.001425))
  print('每股價格:%5.2f, 每日需購股數:%5.0f' %(float(line), data))
  fout.write(str(data)+'\n')</pre></td><td>範例程式 E1-1-5-1.py
<pre>#文字檔讀取與寫入：台積電存股規劃
import math
with open('data5.txt', 'r') as fin:
    with open('data5_w.txt', 'w') as fout:
        for line in fin:
            data=math.ceil(20/(float(line)*0.001425))
            print('每股價格:%5.2f, 每日需購股數:%5.0f' %(float(line), data))
            fout.write(str(data)+'\n')</pre></td><td>20240314</td></tr>
<tr><td>Chapter 1</td><td>1-27</td><td>10
~
21</td><td>範例程式 E1-2-3-1.py
<pre>#開啟data6.csv 檔案，將日期字串中的'/ '改成'- '，並另外儲存成data6_out.csv
import csv
with open('data6.csv','r') as fin:
    with open('data6_out.csv','w') as fout:
   csvreader = csv.reader(fin, delimiter=',')
   csvwriter = csv.writer(fout, delimiter=',')
   header = next(csvreader)
   print(header)
   csvwriter.writerow(header)
   for row in csvreader:
  row[6] = row[6].replace('/','-')
  print(','.join(row))
  csvwriter.writerow(row)
  print(','.join(row))
  csvwriter.writerow(row)</pre></td><td>範例程式 E1-2-3-1.py
<pre>#開啟data6.csv 檔案，將日期字串中的'/ '改成'- '，並另外儲存成data6_out.csv
import csv
with open('data6.csv','r') as fin:
    with open('data6_out.csv','w',newline='') as fout:
        csvreader = csv.reader(fin, delimiter=',')
        csvwriter = csv.writer(fout, delimiter=',')
        header = next(csvreader)
        print(header)
        csvwriter.writerow(header)
        for row in csvreader:
            row[6] = row[6].replace('/','-')
            print(','.join(row))
            csvwriter.writerow(row)</pre></td><td>20240314</td></tr>
<tr><td>Chapter 1</td><td>1-33</td><td>25</td><td>範例程式 E1-3-3-1.py
<pre>import yaml
with open("E1-3-3-1-input.yaml",encoding="utf-8") as f:
     list_doc = yaml.load(f)
for sense in list_doc:
    if sense["name"] == "sense3":
         sense["value"] = 1234
with open("E1-3-3-1-output.yaml", "w",encoding="utf-8") as f:
    yaml.dump(list_doc, f)</pre></td><td>範例程式 E1-3-3-1.py
<pre>import yaml
with open("E1-3-3-1-input.yaml",encoding="utf-8") as f:
     list_doc = yaml.full_load(f)
for sense in list_doc:
    if sense["name"] == "sense3":
         sense["value"] = 1234
with open("E1-3-3-1-output.yaml", "w",encoding="utf-8") as f:
    yaml.dump(list_doc, f)</pre></td><td>20240314</td></tr>
</table>

章節	頁次	行	勘誤前	勘誤後	更新日期

Chapter 1	1-40	32	範例程式 **E1-4-3-2.py**
			import xml.etree.ElementTree as ET
			tree = ET.parse('country_data.xml')
			root = tree.getroot()
			for country in root.findall('country'):
			rank=int(country.find("rank").text)
			if rank>50:
			root.remove(country)
			tree.write("E1-4-3-2-output.xml",encoding="utf-8")
				範例程式 **E1-4-3-2.py**
				import xml.etree.ElementTree as ET
				tree = ET.parse('country_data.xml')
				root = tree.getroot()
				for country in root.findall('country'):
				rank=int(country.find("rank").text)
				if rank>50:
				root.remove(country)
				tree.write("E1-4-3-2-output.xml",encoding="utf-8")
					20240314

Chapter 1	1-46	16 26	範例程式 **E1-5-2-1.py** import sqlite3 conn = sqlite3.connect('test.db') c = conn.cursor() print "Opened database successfully"; c.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \ VALUES (1, 'Paul', 32, 'California', 20000.00)"); c.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \ VALUES (2, 'Allen', 25, 'Texas', 15000.00)"); c.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \ VALUES (3, 'Teddy', 23, 'Norway', 20000.00)"); c.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \ VALUES (4, 'Mark', 25, 'Rich-Mond ', 65000.00)"); conn.commit() print "Records created successfully"; conn.close()	範例程式 **E1-5-2-1.py** import sqlite3 conn = sqlite3.connect('test.db') c = conn.cursor() print("Opened database successfully") c.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \ VALUES (1, 'Paul', 32, 'California', 20000.00)"); c.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \ VALUES (2, 'Allen', 25, 'Texas', 15000.00)"); c.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \ VALUES (3, 'Teddy', 23, 'Norway', 20000.00)"); c.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY) \ VALUES (4, 'Mark', 25, 'Rich-Mond ', 65000.00)"); conn.commit() print("Records created successfully") conn.close()	20240314
Chapter 1	1-47	20 ~ 23	範例程式 **E1-5-2-2.py** import sqlite3 conn = sqlite3.connect('test.db') c = conn.cursor() print("Opened database successfully") cursor = c.execute("SELECT id, name, address, salary from COMPANY") for row in cursor: print "ID = ", row[0] print "NAME = ", row[1] print "ADDRESS = ", row[2] print "SALARY = ", row[3], "\n" print("Operation done successfully") conn.close()	範例程式 **E1-5-2-2.py** import sqlite3 conn = sqlite3.connect('test.db') c = conn.cursor() print("Opened database successfully") cursor = c.execute("SELECT id, name, address, salary from COMPANY") for row in cursor: print("ID = ", row[0]) print("NAME = ", row[1]) print("ADDRESS = ", row[2]) print("SALARY = ", row[3], "\n") print("Operation done successfully") conn.close()	20240314

<table>
<tr><th>章節</th><th>頁次</th><th>行</th><th>勘誤前</th><th>勘誤後</th><th>更新日期</th></tr>
<tr><td>Chapter 1</td><td>1-53</td><td>4
6
9</td><td>參考解答
<pre>import json
import csv
with open("GE1-1-input.json",encoding = 'utf8') as file:
    data = json.load(file)
with open("GE1-1-output.csv", "w",encoding = 'utf8') as file:
    csv_file = csv.writer(file)
    for item in data:
   csv_file.writerow([item['sno'], item['sna'],item['tot']])</pre></td><td>參考解答
<pre>import json
import csv
with open("GE1-1-input.json",encoding = 'utf-8') as file:
    data = json.load(file)
with open("GE1-1-output.csv", "w",encoding = 'utf-8-sig',newline='') as file:
    csv_file = csv.writer(file)
    for item in data:
        csv_file.writerow([item['sno'], item['sna'],item['tot']])</pre></td><td>20240314</td></tr>
<tr><td>Chapter 1</td><td>1-53</td><td>18</td><td>請撰寫程式上網路抓取農產品交易行情(此為農委會開放資料，網址：'http://data.coa.gov.tw/…</td><td>請撰寫程式上網路抓取農產品交易行情(此為農委會開放資料，網址：'https://data.coa.gov.tw/…</td><td>20221212</td></tr>
<tr><td>Chapter 1</td><td>1-54</td><td>23
29
32</td><td>參考解答
<pre>import urllib
import csv
import json
import urllib.request
url = 'http://data.coa.gov.tw/Service/OpenData/FromM/FarmTransData.aspx'
with urllib.request.urlopen(url) as response:
    jdata = response.read()
    data1 = json.loads(jdata)
data=sorted(data1, key=lambda k: k["交易量"],reverse=True)
with open("GE1-2-output.csv", "w",encoding = 'utf8') as file:
    csv_file = csv.writer(file)
    for item in data:
   csv_file.writerow([item['交易日期'], item['作物名稱'], item['市場名稱'], item['平均價'],item['交易量']])</pre></td><td>參考解答
<pre>import urllib
import csv
import json
import urllib.request
url = 'https://data.coa.gov.tw/Service/OpenData/FromM/FarmTransData.aspx'
with urllib.request.urlopen(url) as response:
    jdata = response.read()
    data1 = json.loads(jdata)
data=sorted(data1, key=lambda k: k["交易量"],reverse=True)
with open("GE1-2-output.csv", "w",encoding = 'utf-8-sig',newline='') as file:
    csv_file = csv.writer(file)
    for item in data:
        csv_file.writerow([item['交易日期'], item['作物名稱'], item['市場名稱'], item['平均價'],item['交易量']])</pre></td><td>20221212
20240314</td></tr>
</table>

<table>
<tr><th>章節</th><th>頁次</th><th>行</th><th>勘誤前</th><th>勘誤後</th><th>更新日期</th></tr>
<tr><td>Chapter 1</td><td>1-56</td><td>10
~
13</td><td>參考解答
<pre>import urllib
import gzip
import json
import urllib.request
url = 'https://tcgbusfs.blob.core.windows.net/blobyoubike/YouBikeTP.gz'
with urllib.request.urlopen(url) as response:
    with gzip.GzipFile(fileobj=response) as f:
   jdata = f.read()
   f.close()
   data = json.loads(jdata)
fout=open('GE1-3-output.txt', 'w', encoding = 'utf8')
for key,value in data["retVal"].items():
    sno = value["sno"]
    sna = value["sna"]
    mday = value["mday"]
    sbi = value["sbi"]
    print("NO." + sno + " " + sna + " " +mday + " " + sbi)
    data="NO." + sno + " " + sna + " " +mday + " " + sbi
    fout.write(str(data)+'\n')
fout.close()</pre></td><td>參考解答
<pre>import urllib
import gzip
import json
import urllib.request
url = 'https://tcgbusfs.blob.core.windows.net/blobyoubike/YouBikeTP.gz'
with urllib.request.urlopen(url) as response:
    with gzip.GzipFile(fileobj=response) as f:
        jdata = f.read()
        f.close()
        data = json.loads(jdata)
fout=open('GE1-3-output.txt', 'w', encoding = 'utf-8-sig',newline='')
for key,value in data["retVal"].items():
    sno = value["sno"]
    sna = value["sna"]
    mday = value["mday"]
    sbi = value["sbi"]
    print("NO." + sno + " " + sna + " " +mday + " " + sbi)
    data="NO." + sno + " " + sna + " " +mday + " " + sbi
    fout.write(str(data)+'\n')
fout.close()</pre></td><td>20240314</td></tr>
<tr><td>Chapter 1</td><td>1-57</td><td>3</td><td>請撰寫程式，讀取"GE1-4-input.json"(此為文化部展覽資訊 json 檔案)，將其中的 sno：站點代號、title(活動名稱)...</td><td>請撰寫程式，讀取"GE1-4-input.json"(此為文化部展覽資訊 json 檔案)，將其中的 title(活動名稱)...</td><td>20230814</td></tr>
<tr><td>Chapter 1</td><td>1-58</td><td>7
9
12
13</td><td>參考解答
<pre>import json
import csv
with open("GE1-4-input.json",encoding = 'utf8') as file:
    data = json.load(file)
with open("GE1-4-output.csv", "w",encoding = 'utf8') as file:
    csv_file = csv.writer(file)
    for item in data:
   csv_file.writerow([item['title'], item['showUnit'], item['startDate'],item['endDate']]</pre></td><td>參考解答
<pre>import json
import csv
with open("GE1-4-input.json",encoding = 'utf-8') as file:
    data = json.load(file)
with open("GE1-4-output.csv", "w",encoding = 'utf-8-sig',newline='') as file:
    csv_file = csv.writer(file)
    for item in data:
        csv_file.writerow([item['title'], item['showUnit'], item['startDate'],item['endDate']])</pre></td><td>20240314</td></tr>
</table>

<table>
<tr><th>章節</th><th>頁次</th><th>行</th><th>勘誤前</th><th>勘誤後</th><th>更新日期</th></tr>
<tr><td>Chapter 1</td><td>1-59</td><td>31</td><td>參考解答
<pre>import xml.etree.ElementTree as ET
import csv
tree = ET.parse("GE1-5-input.xml")
root = tree.getroot()
ubikefile = open("GE1-5-output.csv", "w",encoding = 'utf8')
csvwriter = csv.writer(ubikefile)
for row in root:
 ubike = []</pre></td><td>參考解答
<pre>import xml.etree.ElementTree as ET
import csv
tree = ET.parse("GE1-5-input.xml")
root = tree.getroot()
ubikefile = open("GE1-5-output.csv", "w",encoding =
'utf-8-sig',newline='')
csvwriter = csv.writer(ubikefile)
for row in root:
 ubike = []</pre></td><td>20240314</td></tr>
<tr><td>Chapter 1</td><td>1-61</td><td>15
18
~
25</td><td>參考解答
<pre>import xml.etree.ElementTree as ET
import csv
tree = ET.parse("GE1-6-input.xml")
root = tree.getroot()
Countyfile = open("GE1-6-output.csv", "w",encoding = 'utf8')
csvwriter = csv.writer(Countyfile)
for row in root:
 County = []
 pno = row.find('欄位1').text
 County.append(pno)
 cname = row.find('欄位2').text
 County.append(cname)
 ename = row.find('欄位3').text
 County.append(ename)
 csvwriter.writerow(County)
Countyfile.close()</pre></td><td>參考解答
<pre>import xml.etree.ElementTree as ET
import csv
tree = ET.parse("GE1-6-input.xml")
root = tree.getroot()
Countyfile = open("GE1-6-output.csv", "w",encoding =
'utf-8-sig',newline='')
csvwriter = csv.writer(Countyfile)
for row in root:
        County = []
        pno = row.find('欄位1').text
        County.append(pno)
        cname = row.find('欄位2').text
        County.append(cname)
        ename = row.find('欄位3').text
        County.append(ename)
        csvwriter.writerow(County)
Countyfile.close()</pre></td><td>20240314</td></tr>
</table>

<table>
<tr><th>章節</th><th>頁次</th><th>行</th><th>勘誤前</th><th>勘誤後</th><th>更新日期</th></tr>
<tr><td>Chapter 1</td><td>1-63</td><td>6
9
~
16</td><td>參考解答

```
import xml.etree.ElementTree as ET
import csv
tree = ET.parse("GE1-7-input.xml")
root = tree.getroot()
Hospitalfile = open("GE1-7-output.csv", "w",encoding = 'utf8')
csvwriter = csv.writer(Hospitalfile)
for row in root:
 Hospital = []
 pno = row.find('縣市別').text
 Hospital.append(pno)
 cname = row.find('醫療機構名稱').text
 Hospital.append(cname)
 ename = row.find('醫療機構地址').text
 Hospital.append(ename)
 csvwriter.writerow(Hospital)
Hospitalfile.close()
```

</td><td>參考解答

```
import xml.etree.ElementTree as ET
import csv
tree = ET.parse("GE1-7-input.xml")
root = tree.getroot()
Hospitalfile = open("GE1-7-output.csv", "w",encoding =
'utf-8-sig',newline='')
csvwriter = csv.writer(Hospitalfile)
for row in root:
        Hospital = []
        pno = row.find('縣市別').text
        Hospital.append(pno)
        cname = row.find('醫療機構名稱').text
        Hospital.append(cname)
        ename = row.find('醫療機構地址').text
        Hospital.append(ename)
        csvwriter.writerow(Hospital)
Hospitalfile.close()
```

</td><td>20240314</td></tr>
<tr><td>Chapter 1</td><td>1-65</td><td>29</td><td>參考解答

```
import yaml
import json
with open("GE1-8n-input.json",encoding = 'utf-8-sig') as file:
    data = json.load(file)
with open("GE1-8n-output.yaml", "w",encoding="utf-8") as f:
    yaml.dump(data, f,default_flow_style=False,
allow_unicode=True)
```

</td><td>參考解答

```
import yaml
import json
with open("GE1-8n-input.json",encoding = 'utf-8-sig') as file:
    data = json.load(file)
with open("GE1-8n-output.yaml",
"w",encoding='utf-8-sig',newline='') as f:
    yaml.dump(data, f,default_flow_style=False,
allow_unicode=True)
```

</td><td>20240314</td></tr>
<tr><td>Chapter 1</td><td>1-66</td><td>新增</td><td>2. 需要 pdfminer 將 PDF 檔案轉成文字檔。</td><td>2. 需要 pdfminer 將 PDF 檔案轉成文字檔。
3. 相關套件版本：
pdfkit 1.0.0
pdfminer 20191125
PyPDF2 2.11.2</td><td>20230814</td></tr>
<tr><td>Chapter 2</td><td>2-8</td><td>新增</td><td>· 13 行將 df2 合併到 newdf 物件中。</td><td>· 13 行將 df2 合併到 newdf 物件中。
· 14-15 行將 newdf 物件中 columns 為 0 欄位改為'股票代號'，為 1 欄位改為'股票名稱'，然後將'有價證券代號及名稱'欄位刪除。</td><td>20190220</td></tr>
</table>

章節	頁次	行	勘誤前	勘誤後	更新日期
Chapter 2	2-37	27 ~ 32	輸出結果 請輸入欲搜尋的字串：特獎 搜尋 特獎 成功 特獎 出現 10 次 請輸入欲搜尋的字串：頭獎 搜尋 頭獎 成功 頭獎 出現 18 次	輸出結果 縣市 區域 PM2.5 0 高雄市 橋頭 62 1 高雄市 林園 59 2 高雄市 楠梓 56 3 高雄市 大寮 50 4 高雄市 鳳山 49 5 屏東縣 屏東 48 6 高雄市 小港 47 …下略	20190220
Chapter 2	2-65	19 22	請撰寫一程式，讀取"file: GE2-2-input.html"(此為臺北市政府 交通即時資料開放資料專區網頁，本題需要從其資料集列表中的「公車即時資料」取得資料，包含資料集、城市、說明、頻率、檔案連結等 5 個欄位的欄位名稱)並將其轉存為"GE2-2-output.csv"。	請撰寫一程式，讀取"file: GE2-4-input.html"(此為臺北市政府 交通即時資料開放資料專區網頁，本題需要從其資料集列表中的「公車即時資料」取得資料，包含資料集、城市、說明、頻率、檔案連結等 5 個欄位的欄位名稱)並將其轉存為"GE2-4-output.csv"。	20230814
Chapter 2	2-67	18	參考解答 import csv from urllib.request import urlopen from bs4 import BeautifulSoup file_name = "GE2-4-output.csv" f = open(file_name, "w", encoding = 'utf8') w = csv.writer(f) htmlname="file:GE2-2-input.html" html = urlopen(htmlname) bsObj = BeautifulSoup(html, "lxml") cell=bsObj.find("table").find("thead").find("tr").findAll("th") F0 = cell[0].text F1 = cell[1].text F2 = cell[2].text F3 = cell[3].text F4 = cell[4].text F5 = cell[5].text data = [[F0,F1,F2,F3,F4,F5]] w.writerows(data) f.close()	參考解答 import csv from urllib.request import urlopen from bs4 import BeautifulSoup file_name = "GE2-4-output.csv" f = open(file_name, "w", encoding = 'utf8') w = csv.writer(f) htmlname="file:GE2-2-input.html" html = urlopen(htmlname) bsObj = BeautifulSoup(html, "lxml") cell=bsObj.find("table").find("thead").find("tr").findAll("th") F0 = cell[0].text F1 = cell[1].text F2 = cell[2].text F3 = cell[3].text F4 = cell[4].text F5 = cell[5].text data = [[F0,F1,F2,F3,F4]] w.writerows(data) f.close()	20190307
Chapter 2	2-68	4 14	http://data.ntpc.gov.tw/od/data/api/6DCFF24A-838C-40FB-A9DF-F1160AFAFE84?$format=json	https://data.ntpc.gov.tw/api/datasets/6DCFF24A-838C-40FB-A9DF-F1160AFAFE84/json?page=0&size=1000	20221212

<table>
<tr><th>章節</th><th>頁次</th><th>行</th><th>勘誤前</th><th>勘誤後</th><th>更新日期</th></tr>
<tr><td>Chapter 2</td><td>2-69</td><td>20
21
26
~
31</td><td>參考解答
import requests
import json
url = 'http://data.ntpc.gov.tw/od/data/api/6DCFF24A-838C-40FB-A9DF-F1160AFAFE84?$format=json'
res = requests.get(url)
data = json.loads(res.text)
print('新北市大專院校名單：\n')
for record in data:
    if record['type'] == '大專院校':
        print('名稱：%s' % record['name'])
        print('地址：%s' % record['address'])
        print('聯絡電話：%s' % record['tel'])
        print('網站：%s' % record['website'])
        print('資料更新時間：%s' % record['update_date'])
        print()</td><td>參考解答
import requests
import json
url = 'http://data.ntpc.gov.tw/api/datasets/6DCFF24A-838C-40FB-A9DF-F1160AFAFE84/json?page=0&size=1000'
res = requests.get(url)
data = json.loads(res.text)
print('新北市大專院校名單：\n')
for record in data:
    if record['地標類型'] == '大專院校':
        print('名稱：%s' % record['地標名稱'])
        print('地址：%s' % record['地址'])
        print('聯絡電話：%s' % record['電話'])
        print('網站：%s' % record['網址'])
        print('資料更新時間：%s' % record['更新日期'])
        print()</td><td>20221212</td></tr>
<tr><td>Chapter 2</td><td>2-70</td><td>6</td><td>請撰寫程式，抓取中央銀行新臺幣/美元 銀行間收盤匯率，網址為 https://www.cbc.gov.tw/lp.asp?CtNode=645&CtUnit=308&BaseDSD=32&mp=1</td><td>請撰寫程式，抓取中央銀行新臺幣/美元 銀行間收盤匯率，網址為 https://www.cbc.gov.tw/tw/lp-645-1.html</td><td>20221212</td></tr>
<tr><td>Chapter 2</td><td>2-72</td><td>8
9
14
23</td><td>參考解答
import csv
from urllib.request import urlopen
from bs4 import BeautifulSoup
file_name = "GE2-6-output.csv"
f = open(file_name, "w", encoding = 'utf8')
w = csv.writer(f)
htmlname="https://www.cbc.gov.tw/lp.asp?CtNode=645&CtUnit=308&BaseDSD=32&mp=1"
html = urlopen(htmlname)
bsObj = BeautifulSoup(html, "lxml")
count=0
for single_tr in bsObj.find("table",{"class":"DataTable2"}).findAll("tr"):
    if count==0:
        cell = single_tr.findAll("th")
    else:
        cell = single_tr.findAll("td")
    F0 = cell[0].text
    F1 = cell[1].text
    data = [[F0,F1]]
    w.writerows(data)
    count=count=1
f.close()</td><td>參考解答
import csv
from urllib.request import urlopen
from bs4 import BeautifulSoup
file_name = "GE2-6-output.csv"
f = open(file_name, "w", encoding = 'utf8')
w = csv.writer(f)
htmlname="https://www.cbc.gov.tw/tw/lp-645-1.html"
html = urlopen(htmlname)
bsObj = BeautifulSoup(html, "lxml")
count=0
for single_tr in bsObj.find("section",{"class":"lp"}).findAll("tr"):
    if count==0:
        cell = single_tr.findAll("th")
    else:
        cell = single_tr.findAll("td")
    F0 = cell[0].text
    F1 = cell[1].text
    data = [[F0,F1]]
    w.writerows(data)
    count=count+1
f.close()</td><td>20221212</td></tr>
</table>

章節	頁次	行	勘誤前	勘誤後	更新日期
Chapter 2	2-73	6	需要 import selenium 並搭配 PhantomJS 以模擬進入網站操作。	需要 import selenium 並搭配 ChromeDriver 以模擬進入網站操作。請確保在與你的腳本相同的目錄中有適用於你 Chrome 瀏覽器版本的 chromedriver.exe 可執行檔。（下載步驟請參照後面補充）	20230814
Chapter 2	2-73 2-74	11 ~ 20 1 ~ 3	輸出： [['July 13, 2018', 'https://www.paymentscardsandmobile.com/mastercard-and-visa-hit-with-setback-in-uk-interchange-fees-law-suit/', 'Mastercard and Visa hit with setback in UK interchange fees law suit', 'In a damaging set back, a British appeals court ruled in favour of retailers including\xa0some of the UK's biggest retailers,\xa0in massive a blow to the card networks which could now…'], ['July 13, 2018', 'https://www.paymentscardsandmobile.com/biometric-technology-use-set-to-rocket-for-payment-authentication/', 'Biometric technology use set to rocket for payment authentication', 'With new EU regulations coming into place next September, Mastercard is predicting a significant increase in the use of biometric technology to authenticate who is paying. With regards to card…'], …下略	輸出：（請留意以下為 2023/8/8 執行結果的前 3 筆，此結果會隨網頁更新） ['02 Aug 23 @ 10:45', 'How AI is changing fintech jobs', 'While a lot of the conversation surrounding AI is focusing on the threat automation poses to jobs—the World Economic Forum's Future of Jobs Report 2023 found that AI is expected... read more'] ['28 Jul 23 @ 09:42', 'Can blockchain transform the financial services industry?', 'The global blockchain market is set to grow $163.83 billion by 2029, up from $7.18 billion in 2022, exhibiting a CAGR of 56.3%. That being said, blockchain is still viewed... read more'] ['18 Jul 23 @ 11:30', 'The future of Point of Sale payments', 'Point of Sale (POS) payments have come a long way since the introduction of the first credit card in the 1950s. Today, consumers have a plethora of payment options to... read more'] …下略	20230814

章節	頁次	行	勘誤前	勘誤後	更新日期

Chapter 2	2-74	10 ~ 29	參考解答
			import os
			import sys
			from bs4 import BeautifulSoup
			from selenium import webdriver
			path_PhantomJS = os.getcwd()+"./PhantomJS.exe"
			browser = webdriver.PhantomJS(executable_path = path_PhantomJS)
			scraping_url = "http://www.paymentscardsandmobile.com/"
			browser.get(scraping_url)
			browser.maximize_window()
			browser.implicitly_wait(10)
			new_result = []
			html_soup = BeautifulSoup(browser.page_source, "lxml")
			for tag_article in html_soup.find("div", attrs={"class": "posts-list listing-alt"}).find_all("article"):
			date = tag_article.find("time", attrs={"itemprop": "datePublished"}).get_text().strip(' \t\n\r')
			url = tag_article.find("a", attrs={"itemprop": "name url"})['href']
			title = tag_article.find("a", attrs={"itemprop": "name url"}).get_text().strip(' \t\n\r')
			content = tag_article.find("p").get_text().strip(' \t\n\r')
			new_result.append([date, url, title, content])
			print(str(new_result).encode(sys.stdin.encoding, "replace").decode(sys.stdin.encoding))
			browser.close()
				import os
				import sys
				from bs4 import BeautifulSoup
				from selenium import webdriver
				from selenium.webdriver.chrome.service import Service
				path_ChromeDriver = os.getcwd() + "/chromedriver.exe"
				scraping_url = "http://www.paymentscardsandmobile.com/"
				service = Service(executable_path=path_ChromeDriver)
				chrome_options = webdriver.ChromeOptions()
				browser = webdriver.Chrome(service=service, options=chrome_options)
				browser.get(scraping_url)
				browser.maximize_window()
				browser.implicitly_wait(10)
				new_result = []
				html_soup = BeautifulSoup(browser.page_source, "lxml")
				for tag_article in html_soup.find("section", attrs={"class": "content clearfix pt30 pb30"}).find("div", attrs={"class": "wrap"}).find_all("div", attrs={"class": "news_infoinside single_inside div"}):
				date = tag_article.find("p", attrs={"class": "datetime mt10 tal"}).get_text().strip(' \t\n\r')
				url = tag_article.find("a")['href']
				title = tag_article.find("h2", attrs={"class": "tal"}).get_text().strip(' \t\n\r')
				content = tag_article.find("p", attrs={"class": "news_excerpt mt10 tal"}).get_text().strip(' \t\n\r')
				new_result.append([date, url, title, content])
				for item in new_result:
				print(item)
				browser.close()
					20230814

Chapter 2	2-74 2-75	32 ~ 35 1 ~ 6	· 1-4 行 import 所需套件。 · 5-6 行啟動 WebDriver（在背景執行 PhantomJS）。 · 7-9 行前往 www.paymentscardsandmobile.com 首頁，並最大化螢幕。 · 10 行如果網站在 10 秒內回應則繼續執行下一步，否則等待 10 秒。 · 11-13 行取得網頁原始碼並交由 BeautifulSoup 進行解析，找出所有的單篇新聞區塊。 · 14-18 行取得所需 date（日期）、url（網址）、title（標題）、content（新聞內容），加入 new_result 串列物件中。 · 19 行列印串列物件。 · 20 行最後關閉 browser 物件。	· 1-5 行 import 所需套件。os 模塊，用於操作文件和目錄；sys 模塊，用於與 Python 直譯器進行交互；從 bs4 庫導入 BeautifulSoup 類，用於解析 HTML 文檔；從 selenium 庫導入 webdriver 模塊，用於控制瀏覽器操作；從 selenium.webdriver.chrome.service 導入 Service 類，用於指定 Chrome Driver 服務。 · 6-10 行啟動 WebDriver。6 行創建變數 path_ChromeDriver，用於存儲 Chrome Driver 的路徑，os.getcwd()獲取當前工作目錄；7 行創建變量 scraping_url，儲存目標網頁的 URL；8 行創建 Service 實例，設置 Chrome Driver 的執行路徑；9 行創建 ChromeOptions 實例，用於設置 Chrome 瀏覽器的選項；10 行創建 Chrome 瀏覽器的實例，傳入 Chrome Driver 服務和選項。 · 11-12 行前往 www.paymentscardsandmobile.com 首頁，並最大化螢幕。 · 13 行如果網站在 10 秒內回應則繼續執行下一步，否則等待 10 秒。14 行建立 new_result 串列物件。 · 15-16 行取得網頁原始碼並交由 BeautifulSoup 進行解析，找出所有的單篇新聞區塊。 · 17-21 行取得所需 date（日期）、url（網址）、title（標題）、content（新聞內容），加入 new_result 串列物件中。 · 22-23 行列印串列物件。 · 24 行最後關閉 browser 物件。 你可以在 ChromeDriver 官方網站上下載適用於你的作業系統和	20230814

章節	頁次	行	勘誤前	勘誤後	更新日期
				Chrome 瀏覽器版本的 ChromeDriver。以下是下載的步驟： 1. 首先，確定你的 Chrome 瀏覽器的版本。在 Chrome 瀏覽器中，點擊右上角的功能表圖示（通常是三個豎點或水平線），然後選擇"說明">"關於 Google Chrome"。 2. 在"關於 Google Chrome"頁面中，你將看到當前安裝的 Chrome 瀏覽器的版本號。 3. 訪問 ChromeDriver 官方網站：https://sites.google.com/chromium.org/driver/ 4. 在頁面上找到與你的 Chrome 瀏覽器版本匹配的 ChromeDriver 版本。你可以在"下載"部分中找到各個版本的下載連結。 5. 點擊適用於你作業系統的連結，下載對應版本的 ChromeDriver。 6. 下載完成後，將 ChromeDriver 解壓到與 GE2-7_20230808.py 相同的資料夾中，然後在你的代碼中指定這個路徑作為 executable_path 參數。 確保下載的 ChromeDriver 版本與你的 Chrome 瀏覽器版本完全匹配，以獲得最佳的相容性和穩定性。如果你的 Chrome 瀏覽器更新了，可能需要下載新版本的 ChromeDriver。	
Chapter 3	3-84	9	最後三筆的西瓜/香瓜價雨量	最後三筆的西瓜/香瓜價與量	20230814
Chapter 3	3-89	16	E.列印全台灣測站資料。	E.列印全台灣測站資料（從 County 欄位到 Status 欄位）。	20230814
Chapter 3	3-95	28	24 get_day = []	24 #get_day = []	20190604

章節	頁次	行	勘誤前	勘誤後	更新日期
Chapter 3	3-96	8 27 31	45 get_day.append(today) 46 for t in weather[0].find_all("value"): 47 T.append(t.text) 48 for td in weather[1].find_all("value"): 49 TD.append(td.text) 50 for rh in weather[2].find_all("value"): 51 RH.append(rh.text) 52 for wd in weather[5].find_all("value"): 53 WD.append(wd.text) 54 ws = weather[6].find_all("value") 55 for k in range(0,len(ws),2): 56 WS.append(ws[k].text) 57 BF.append(ws[k+1].text) 58 for at in weather[8].find_all("value"): 59 AT.append(at.text) 60 wx = weather[9].find_all("value") 61 for w in range(0,len(wx),2): 62 Wx.append(wx[w].text) 63 Wx_n.append(wx[w+1].text) 64 rain1 = weather[3].find_all("value") 65 for l in range(0,len(rain1)): 66 pop6 = rain1[l].text 67 PoP6h.append(pop6) 68 PoP6h.append(pop6)	45 #get_day.append(today) 46 for t in weather[0].find_all("value"): 47 T.append(t.text) 48 for td in weather[1].find_all("value"): 49 TD.append(td.text) 50 for rh in weather[2].find_all("value"): 51 RH.append(rh.text) 52 for wd in weather[5].find_all("value"): 53 WD.append(wd.text) 54 ws = weather[6].find_all("value") 55 for k in range(0,len(ws),2): 56 WS.append(ws[k].text) 57 BF.append(ws[k+1].text) 58 for at in weather[8].find_all("value"): 59 AT.append(at.text) 60 wx = weather[9].find_all("value") 61 for w in range(0,len(wx),2): 62 Wx.append(wx[w].text) 63 Wx_n.append(wx[w+1].text) 64 """rain1 = weather[3].find_all("value") 65 for l in range(0,len(rain1)): 66 pop6 = rain1[l].text 67 PoP6h.append(pop6) 68 PoP6h.append(pop6)"""	20190604
Chapter 3	3-97	3 4 6 7	82 "DAY" : DAY,"TIME" : TIME,"T":T,"TD" : TD,"RH":RH, "WD" : WD,"WS" : WS,"BF":BF,"AT" : AT,"Wx": Wx,"Wx_n":Wx_n,"PoP6h" : PoP6h,"PoP12h" :PoP12h,"get_day":get_day} df = pd.DataFrame(data,columns=["CITY","DISTRICT","GEOCODE", "DAY","TIME","T","TD","RH","WD","WS","BF","AT","Wx","Wx_n","P oP6h","PoP12h","get_day"]) 83 df1=df.sort_values(by="T", ascending=False) 84 print('以溫度遞減排序') 85 print(df1['T']) 86 print("目前最高氣溫為"+str(df1["T"].max())) 87 print("目前最低氣溫為"+str(df1["T"].min())) 88 save_name = "GE3-4-output.csv" 89 df.to_csv(save_name,index=False,encoding="utf_8_sig")	82 "DAY" : DAY,"TIME" : TIME,"T":T,"TD" : TD,"RH":RH, "WD" : WD,"WS" : WS,"BF":BF,"AT" : AT,"Wx": Wx,"Wx_n":Wx_n,"PoP12h" :PoP12h} df = pd.DataFrame(data,columns=["CITY","DISTRICT","GEOCODE", "DAY","TIME","T","TD","RH","WD","WS","BF","AT","Wx","Wx_n","P oP12h"]) 83 df1=df.sort_values(by="T", ascending=False) 84 print('以溫度遞減排序') 85 print(df1['T']) 86 print("目前最高氣溫為"+str(df1["T"].max())) 87 print("目前最低氣溫為"+str(df1["T"].min())) 88 save_name = "GE3-4-output.csv" 89 df.to_csv(save_name,index=False,encoding="utf_8_sig")	20190604
Chapter 3	3-99	25	商品 B 最低成交量 54894	商品 B 最低成交量 900	201900220
Chapter 3	3-100	15 16	5 print("商品A最高成交價"+str(prices[0].max(axis=0))) 6 print("商品B最低成交量"+str(amounts[1].min(axis=0)))	5 print("商品A最高成交價"+str(prices[:,0].max(axis=0))) 6 print("商品B最低成交量"+str(amounts[:,1].min(axis=0)))	201900220
Chapter 3	3-101	12	D.輸出台北市最近病例。	D.輸出台北市最近病例（指"發病日"最近）。	20230814

章節	頁次	行	勘誤前	勘誤後	更新日期
Chapter 4	4-84	10	3.將 X 陣列與 Y 陣列相加產生 Z 陣列，將之列印出來。	將 X 陣列與 Y 陣列相加產生 Z 陣列，將之列印出來。	201900220

範例程式更新：

碁峯請至 http://books.gotop.com.tw/download/AEY040000 下載

全華請至 http://www.chwa.com.tw/mis/gNzKb2018117084820/93m62o9Cyy2018117.rar 下載

◆其他疑問解答◆

Q：請問如何安裝勘誤的系統更新檔？

A：勘誤的各系統更新檔均為 EXE 格式的自解檔，請先安裝 WinZip 或 WinRAR 等解壓縮軟體後直接安裝更新檔即可。

Q：如果發現書本上的答案與光碟練習題的答案不同，應以何者為準？

A：類似情形，均應以書本上的答案（勘誤後）為準。

Q：如果發現未列在勘誤表中的問題該如何反應？

A：若對題目或答案有疑義，可將作好的原始檔案以 E-mail 附檔方式寄至本會客服信箱 master@mail.csf.org.tw。並請在內容註明電腦所使用的作業系統及應用軟體版本，以方便本會人員處理您的問題。

※若有其他產品或認證之問題，可至以下網頁取得相關資訊：

產品 FAQ 資料庫

TQC 認證出版品網頁

TQC+認證出版品網頁

認證線上報名系統