

# 資料擷取與匯入

## Extracting and Importing Data

# Spreadsheet

- 讀取Excel、CSV

In [3]:

```
import pandas as pd

# 讀取CSV
df1 = pd.read_csv("Sample.csv", encoding = "big5")

# 讀取Excel: 須事先安裝xlrd (pip install xlrd)
df2 = pd.read_excel("Sample.xlsx", sheetname='sheet1')
```

一般文件的中文編碼使用big5



In [4]:

```
df1[:5]
```

Out[4]:

	year	peak_load	PRM
0	71	691.8	26.8
1	72	780.8	29.2
2	73	851.7	22.1
3	74	871.6	55.1
4	75	990.0	48.3

記得選擇sheet

## Notes

- ▶ 匯出成.CSV  
DataFrame.to\_csv('name.csv',  
encoding='big5')



## Database

- 範例：MySQL

安裝連接資料庫套件  
(e.g. mysql-connector-python, pymongo)

```
In [1]: import mysql.connector  
import pandas.io.sql as sql
```

```
#連接資料庫
```

```
config = {  
    'user': 'sc1387',  
    'password': '1234',  
    'host': '35.201.158.248',  
    'database': 'python_ds'
```

```
}
```

```
cnx = mysql.connector.connect(**config)
```

```
#readinto dataframe
```

```
df = sql.read_sql('select * from A_LVR_LAND;', cnx)  
df.head()
```

← 連接設定

← 連接資料庫

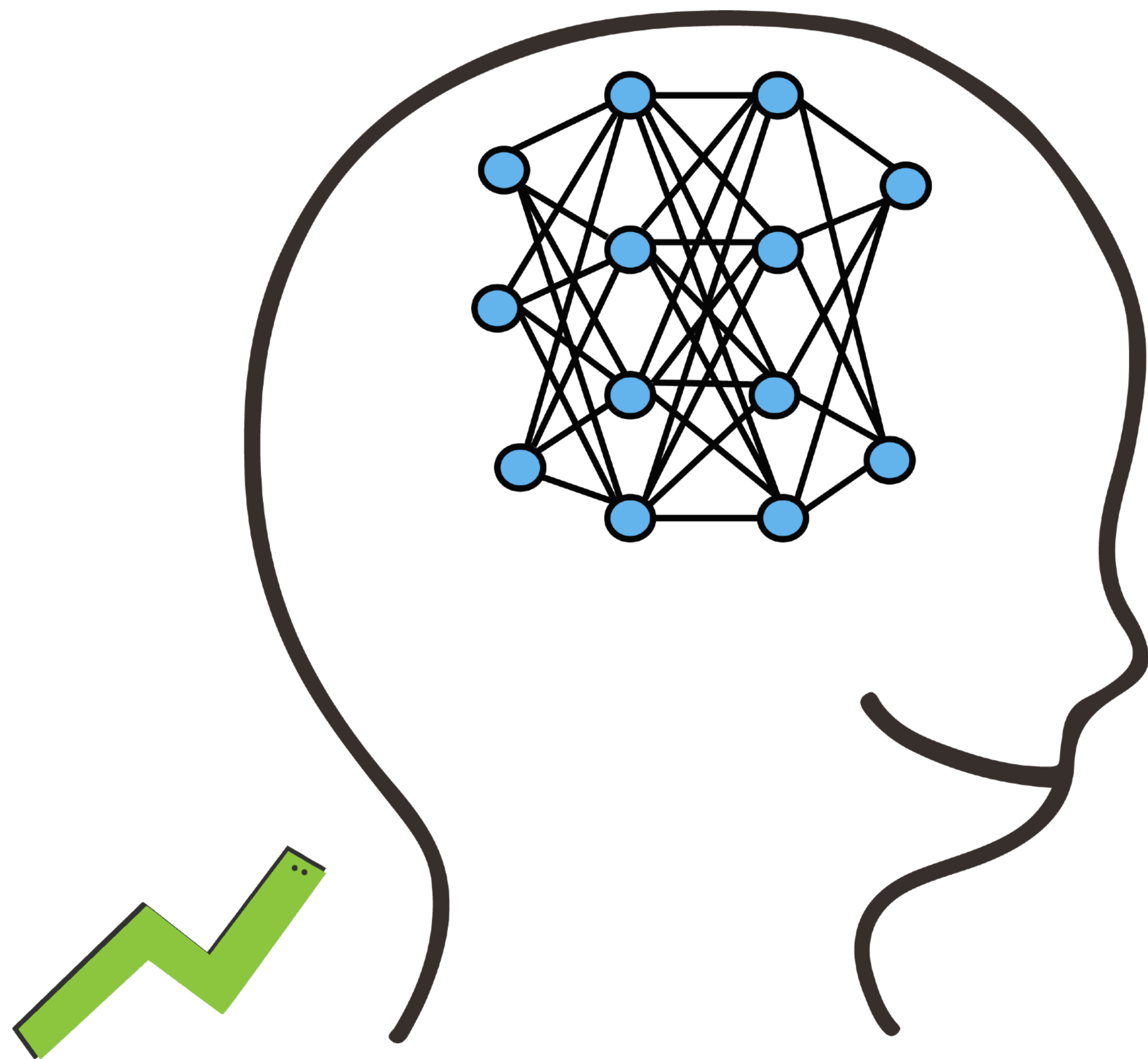
← 以SQL指令讀取DB資料  
存成DataFrame

```
Out[1]:
```

	鄉鎮市區	建物移轉總面積平方公尺	建物現況格局-房	建物現況格局-廳	總價元	單價每平方公尺
0	文山區	90.63	3	2	12000000	132406
1	北投區	164.59	5	2	11800000	71693
2	萬華區	34.15	1	1	8200000	240117
3	萬華區	43.40	1	1	9000000	207373
4	萬華區	40.95	0	1	9520000	232479

### Notes

- ▶ \*\* 是關鍵字引數 (Keyword Argument)，若想進一步了解，請參考講義最後的補充內容。



# 網路爬蟲

## Web Crawler

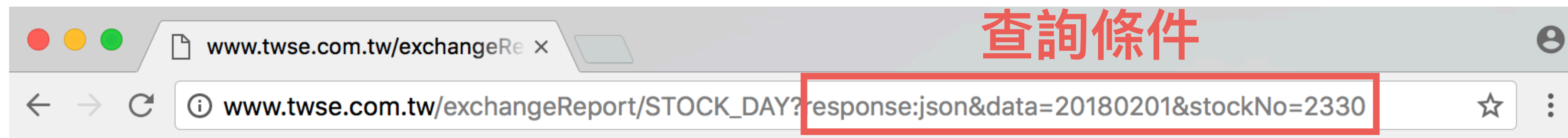


# 網頁傳值方式

- GET vs. POST
- GET將參數放在url之後傳遞
  - e.g. <https://www.google.com.tw/#q=python> (Google搜尋python)
- POST將參數隱藏起來，安全性較高、網址不變
  - e.g. 公開資訊觀測站



# GET 範例 - 台灣證券交易所



```
{"stat": "OK", "date": "20180313", "title": "107年03月 2330 台積電", "fields": ["日期", "成交股數", "成交金額", "開盤價", "最高價", "最低價", "收盤價", "漲跌價差", "成交筆數"], "data": [{"date": "107/03/01", "open": "43,847,984", "high": "10,669,194,561", "low": "244.00", "close": "245.00", "change": "-3.00", "volume": "11,589"}, {"date": "107/03/02", "open": "35,289,170", "high": "8,453,712,014", "low": "240.00", "close": "241.00", "change": "-3.00", "volume": "10,440"}, {"date": "107/03/05", "open": "27,337,846", "high": "6,607,367,732", "low": "242.50", "close": "243.00", "change": "+1.50", "volume": "9,321"}, {"date": "107/03/06", "open": "36,945,780", "high": "9,155,812,210", "low": "245.50", "close": "250.00", "change": "+8.50", "volume": "13,977"}, {"date": "107/03/07", "open": "30,391,219", "high": "7,525,601,638", "low": "248.00", "close": "248.50", "change": "-3.00", "volume": "10,521"}, {"date": "107/03/08", "open": "20,645,694", "high": "5,172,732,531", "low": "249.50", "close": "251.50", "change": "+2.50", "volume": "8,817"}, {"date": "107/03/09", "open": "22,887,063", "high": "5,723,277,170", "low": "250.00", "close": "251.00", "change": "+1.00", "volume": "7,050"}, {"date": "107/03/12", "open": "25,100,615", "high": "6,370,529,728", "low": "252.00", "close": "255.00", "change": "+3.50", "volume": "10,550"}, {"date": "107/03/13", "open": "34,264,883", "high": "8,816,553,606", "low": "255.50", "close": "259.00", "change": "+5.00", "volume": "11,700"}], "notes": ["符號說明: +/ - / x 表示漲/跌/不比價", "當日統計資訊含一般、零股、盤後定價、鉅額交易，不含拍賣、標購。", "ETF證券代號第六碼為K、M、S、C者，表示該ETF以外幣交易。"]}
```



# POST 範例 - 公開資訊觀測站

The screenshot shows a web browser window with the URL `mops.twse.com.tw/mops/web/t163sb04` highlighted in red. A red text overlay reads "查詢時網址不變". The website header includes a search bar and navigation links. The left sidebar lists various financial reports, with "綜合損益表" (Consolidated Income Statement) selected. The main content area displays the "綜合損益表" for a listed company, with filters for "市場別" (Market) set to "上市" (Listed), "年度" (Year) set to "106" (2010), and "季別" (Quarter) set to "4" (4th Quarter). Below the filters, there are buttons for "列印網頁" (Print Page), "開新視窗" (Open New Window), and "問題回報" (Report Problem). The title "上市公司第四季資料" (Listed Company 4th Quarter Data) is displayed. A note states: "註：依證券交易法第36條及證券期貨局相關函令規定，財務報告申報期限如下：" (Note: According to Article 36 of the Securities and Futures Act and related orders of the Securities and Futures Commission, the reporting deadlines for financial reports are as follows:). Two bullet points follow: "1.一般行業申報期限：第一季為5月15日，第二季為8月14日，第三季為11月14日，年度為3月31日。" (1. General industry reporting deadlines: 1st quarter is May 15, 2nd quarter is August 14, 3rd quarter is November 14, annual is March 31.) and "2.金控業申報期限：第一季為5月30日，第二季為8月31日，第三季為11月29日，年度為3月31日。" (2. Financial holding company reporting deadlines: 1st quarter is May 30, 2nd quarter is August 31, 3rd quarter is November 29, annual is March 31.).



# 快速抓取網頁中表格

- `pd.read_html()`
  - 回傳HTML中所有的表格
  - 回傳格式：DataFrame的list





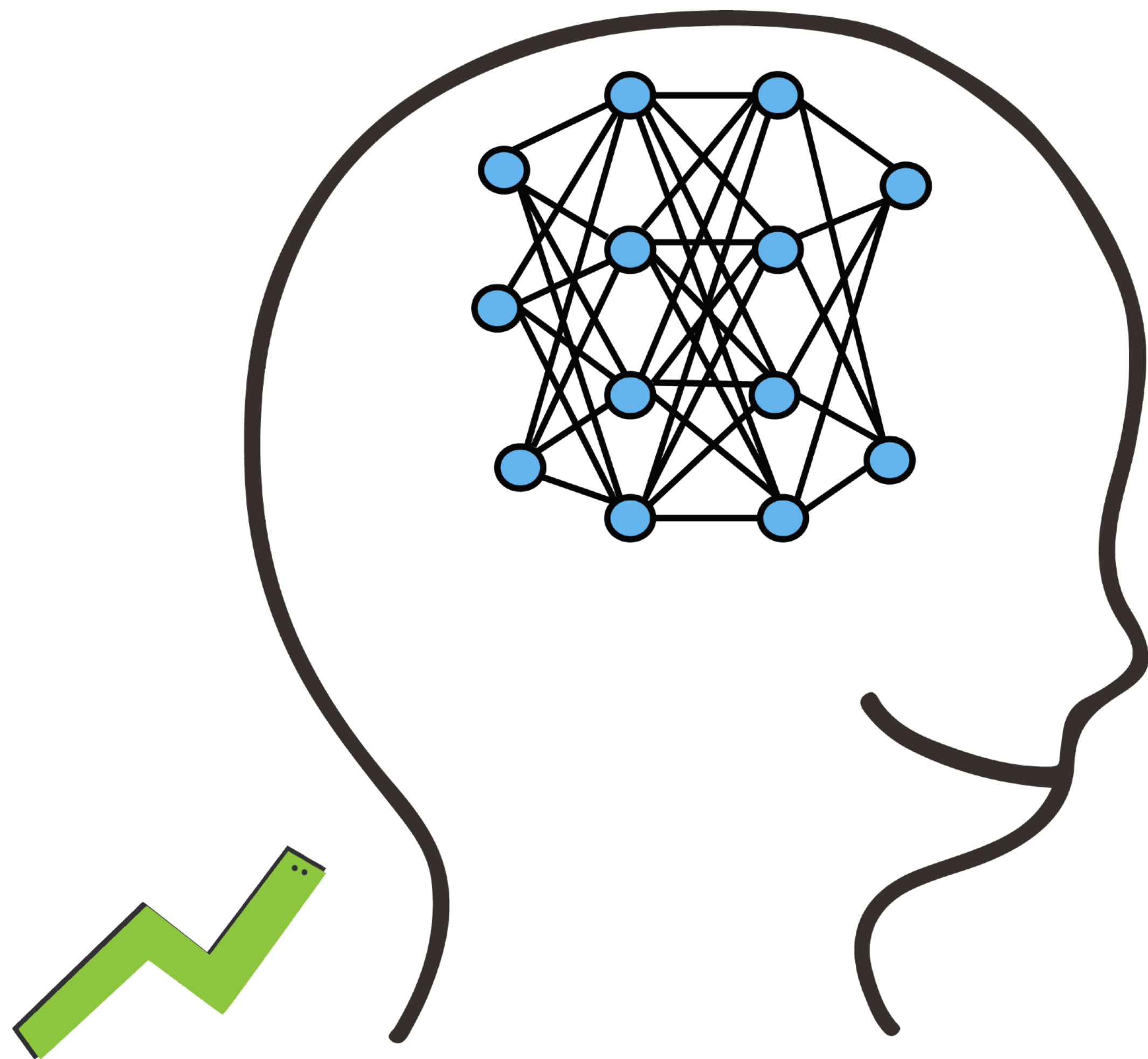
## JSON

- JSON (Javascript Object Notation)
- 源起於網頁Javascript表示物件的格式，後來變成廣受歡迎的常用資料格式，也是文件儲存的NoSQL資料庫使用的格式 (e.g. MongoDB)

- 物件{}、串列[]

- 範例(wiki)：

```
{
  "firstName": "John",
  "lastName": "Smith",
  "sex": "male",
  "age": 25,
  "address":
  {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021"
  },
  "phoneNumber":
  [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "fax",
      "number": "646 555-4567"
    }
  ]
}
```



**(optional) HTML剖析-  
使用BeautifulSoup**



## 什麼是HTML

- HTML (HyperText Markup Language) : 超文件標示語言
- 網頁最基本的語言 (常搭配CSS、Javascript等網頁語言)
- tag <> 、 attribute (e.g. bgcolor)

```
<html>
<head>
<title>網頁標題</title>
</head>
<body bgcolor="yellow">
您好！
</body>
</html>
```








## HTML 剖析

```
<img class="itemcov"
alt="Python 自動化的樂趣：搞定重複瑣碎&單調無聊的工作"
data-original="http://iml.book.com.tw/image/getImage?i=http://www.books.com.tw/img/001/073/93/0010739372.jpg&w=85&h=120&v=585a59ac"
width="85"
height="120">
```

☐



**Python 自動化的樂趣：搞定重複瑣碎&單調無聊的工作**

中文書， [Al Sweigart](#) H&C， [碁峰](#)，出版日期：2016-12-29

優惠價： **79 折**，**395元** [放入購物車](#) [試讀](#)

運用Python寫出程式，幫您在幾分鐘內搞定平常以人工手動處理需要花費數小時的工作。一旦掌握了程式設計的基礎知識，就能輕鬆使用Python編寫程式，把自動化的好用和效率應用在下列這些工作上：

- 在一個或多個檔案中搜尋文字
- 建立、更新..... [more](#)

# HTML 剖析

- html結構：tag、attribute
- 剖析套件：BeautifulSoup4

安裝 beautifulsoup4

```
In [120]: from bs4 import BeautifulSoup
          soup = BeautifulSoup(res.text, 'html.parser')
          print (soup.title.string)
```

← 將html格式的string傳進BeautifulSoup  
← 取得不同tag的內容 e.g.title, p, b...

博客來-目前您搜尋的關鍵字為:python

```
In [121]: #爬取書名
          books = pd.Series()
          for book in soup.select("img[class='itemcov']"):
              books = books.append(pd.Series([book['alt']])).reset_index(drop=True) #加到pd.Series
```

← 回傳<img class="itemcov"...>  
↑ 取得tag中的alt值

# 轉存為DataFrame

- Series to DataFrame

Series1

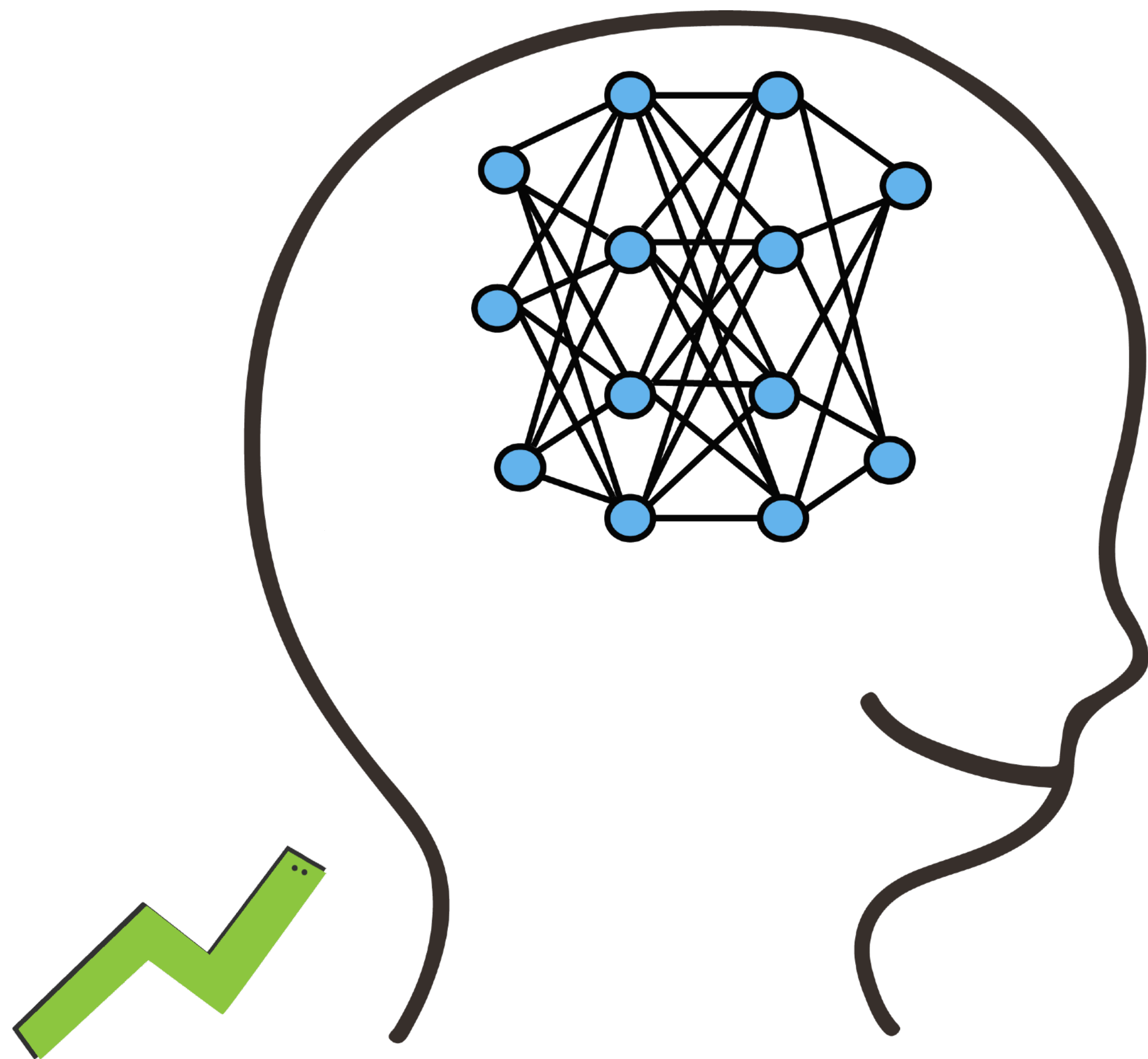
Series2

```
In [122]: #合併成DataFrame
df = pd.DataFrame({'書名':books, '價格': prices})
df[['書名','價格']]
```

Out[122]:

	書名	價格
0	Python+Spark 2.0+Hadoop機器學習與大數據分析實戰	537
1	網站擷取：使用Python	458
2	Python機器學習	458
3	Data Science from Scratch中文版：用Python學資料科學	458
4	機器學習：使用Python進行預測分析的基本技術	458
5	Python程式設計實務：從初學到活用Python開發技巧的16堂課	442
6	精通 Python：運用簡單的套件進行現代運算	616
7	Think Python：學習程式設計的思考概念 第二版	411
8	Python程式設計入門指南	411
9	比Hadoop+Python還強：Spark MLlib機器學習實作	432
10	Python 3.5 技術手冊	411
11	不止是測試：Python網路爬蟲王者Selenium	504
12	Python x Arduino物聯網整合開發實戰	387
13	Python 程式設計：從入門到進階應用	466





# 補充：函式引數

## Function Argument

# 參數 vs. 引數

- 參數 (Parameters) : 變數名稱
- 引數 (Arguments) : 實際傳入的值

參數

```
In [1]: def hello(name):  
        print('Hello ' + name)
```

引數

```
In [2]: hello('Andy')  
Hello Andy
```



# 兩種引數

- 位置引數 (Positional Arguments)
- 關鍵字引數 (Keyword Arguments)





# 位置引數 (Positional Arguments)

- 根據位置 (argument order matters) 依序將引數傳入函式中的參數

```
In [1]: def hello_all(first, second, third, fourth):  
        print('Hello ' + first)  
        print('Hello ' + second)  
        print('Hello ' + third)  
        print('Hello ' + fourth)
```

```
In [2]: hello_all('Andy', 'Ben', 'Cathy', 'David')
```

```
Hello Andy  
Hello Ben  
Hello Cathy  
Hello David
```



# 位置引數 (Positional Arguments)

- \* 可接受傳入任意長度的位置引數，傳入值將視為tuple

```
In [1]: def hello_all(first, *names):  
        print('Hello ' + first)  
        for name in names: #names is tuple  
            print('And Hello ' + name)
```

```
In [2]: hello_all('Andy', 'Ben', 'Cathy', 'David')
```

```
Hello Andy  
And Hello Ben  
And Hello Cathy  
And Hello David
```



# 關鍵字引數 (Keyword Arguments)

- 不管引數位置順序，將引數對應到參數名稱輸入函式

```
In [1]: def hello_all(first, second, third, fourth):  
        print('Hello ' + first)  
        print('Hello ' + second)  
        print('Hello ' + third)  
        print('Hello ' + fourth)
```

```
In [2]: hello_all(third='Andy', first='Ben', fourth='Cathy', second='David')
```

```
Hello Ben  
Hello David  
Hello Andy  
Hello Cathy
```





# 關鍵字引數 (Keyword Arguments)

- \*\* 可接受傳入任意長度的關鍵字引數，傳入值將視為dictionary

```
In [1]: def hello_all(**guests):  
        for key in guests.keys():  
            print('Hello ' + key + ' ' + guests[key])
```

```
In [2]: guests={'Andy': 'Chen', 'Ben': 'Wang', 'Cathy': 'Lin', 'David': 'Wu'}  
        hello_all(**guests)
```

```
Hello Ben Wang  
Hello Andy Chen  
Hello David Wu  
Hello Cathy Lin
```