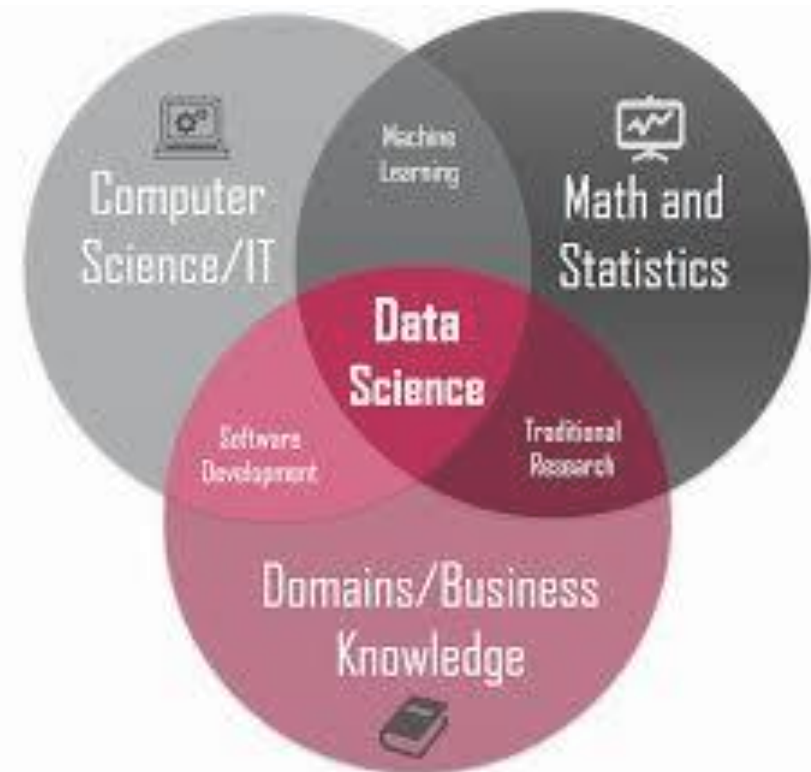


AI & DS



U03A-資料收集與網路爬蟲

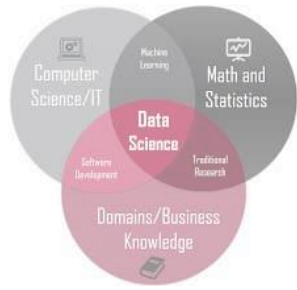
Data
Science

Artificial
Intelligence

2023.10_V1.1
Machine
Learning

Deep
Learning

Statistics



單元大綱

- 認識網路爬蟲: 文件檔案的爬取與儲存
- 網頁資料解析
- 文字與檔案資料的收集

[資料引用] 文淵閣工作室

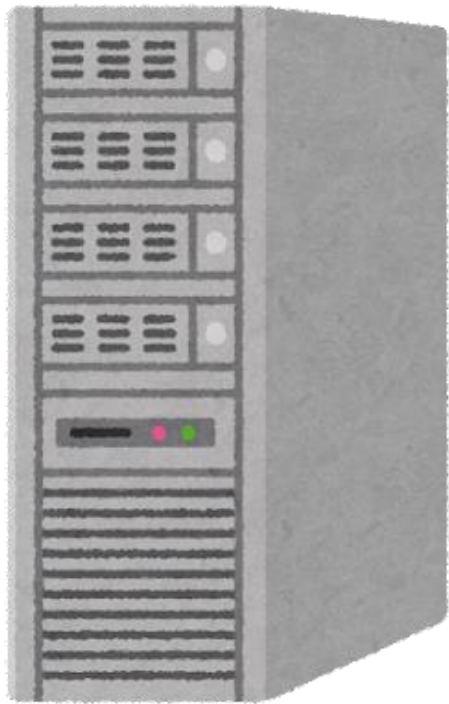
Part 1

認識網路爬蟲： 文件檔案的爬取與儲存



網路爬蟲(crawler)運作流程

網頁伺服器



請求
get post



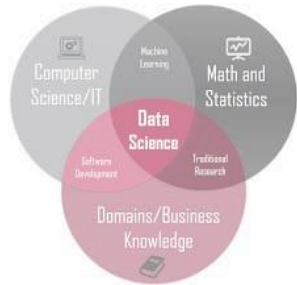
回應HTML

瀏覽器



資料爬取
requests

網頁解析
BeautifulS
oup



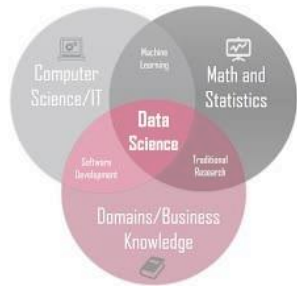
資料的爬取：requests模組

1. requests 模組 能發送 HTTP的請求取得伺服器的回應資料。
2. 常見請求方式 **get**、**post**
3. **基本語法**

```
import requests
```

```
回應內容 = requests.請求方式("網址", 參數)
```

4. 請求資料為**文字**時，可用 **回應內容.text** 取得值
5. 請求資料為**二進位檔案**時，可用 **回應內容.content** 取得值



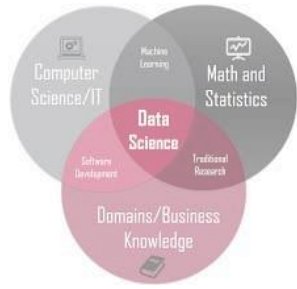
使用 requests 讀取網頁原始碼

語法

```
import requests  
回應內容 = requests.get("網址")
```

例如：以 utf-8 編碼讀取網頁的原始碼

```
import requests  
url = 'http://ehappy.tw/bsdemo1.htm'  
html = requests.get(url)  
html.encoding="utf-8"  
print(html.text)
```



資料檔案的儲存：open()函式

open()函式是用來進行檔案操作，返回值是檔案物件。

語法

with open(檔案, 模式) as 物件:

物件.read()

#可讀取文件內容，返回值是**字串**。

物件.readline()

#讀取文件指針處下一行，返回是**字串**。

物件.readlines()

#讀取文件每一行成**串列**返回。

物件.write()

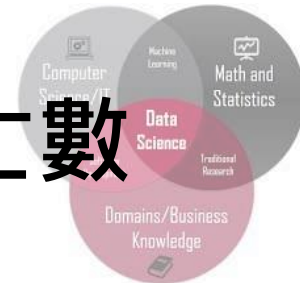
#可將字串寫入文件。

特殊設定：
+：開放為可讀寫
B：二進位檔案

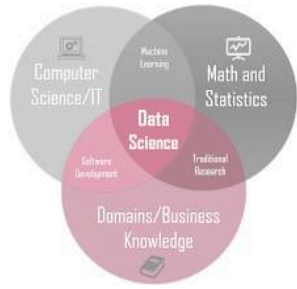
模式

模式	說明	模式	說明	模式	說明
r	唯讀，文件指針在 檔頭 ，預設模式。	a	附加，沒有檔案時會新增檔案，新增內容會放在 檔尾 。	w	寫入，沒有檔案時會新增檔案，寫入時會 複蓋 原內容。

範例：COVID-19各國家地區累積病例數與死亡數



- **範例說明**：讀取遠端公開資料CSV檔並顯示在畫面上。
- **資料來源**：政府資料公開平台-COVID-19各國家地區累積病例數與死亡數
<https://data.gov.tw/dataset/120449>
- **任務一**
讀取檔案來源並顯示在畫面上
- **任務二**
讀取檔案來源並下載儲存到本機中



範例：九天玄女「降肉」

- 範例說明：讀取Youtube影片縮圖檔並下載到本機。
- 資料來源：YouTube影片



```
import requests  
url = 'https://img.youtube.com/vi/qo3H-VgXmHE/sddefault.jpg'  
r = requests.get(url)
```

```
with open('ng.jpg', 'wb') as f:  
    f.write(r.content)
```

<https://youtu.be/qo3H-VgXmHE>

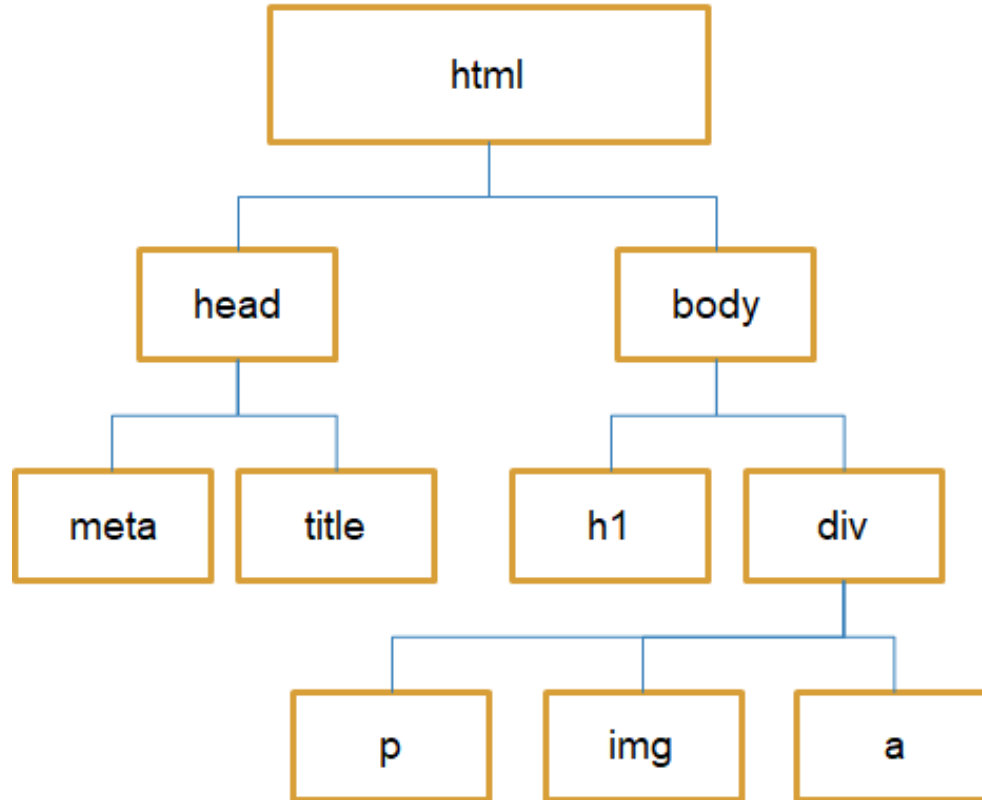
<https://img.youtube.com/vi/qo3H-VgXmHE/sddefault.jpg>

Part 2

網頁資料解析



HTML 網頁架構



```
<html>
```

```
<head>
```

```
<meta 文件屬性>
```

```
<title>標題</title>
```

```
</head>
```

```
<body>
```

```
<h1>標題</h1>
```

```
<div>
```

```
<p>內文段落</p>
```

```

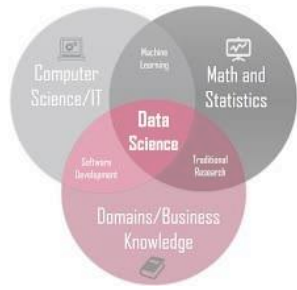
```

```
<a href="...">連結文字</a>
```

```
</div>
```

```
</body>
```

```
</html>
```

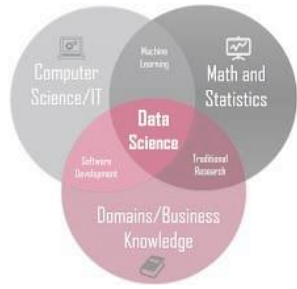


BeautifulSoup模組

- BeautifulSoup 模組能快速解析HTML的文件結構。
- BeautifulSoup 模組能將拆解的結果形成 BeautifulSoup物件，接著即能使用DOM結構名稱、標籤、id 名稱、class名稱，取得網頁元素的內容及屬性。
- 基本語法

```
from bs4 import BeautifulSoup
```

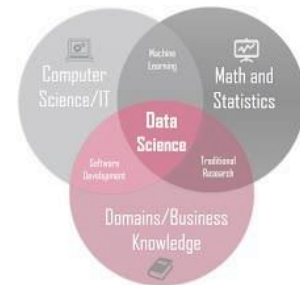
```
bs物件 = BeautifulSoup(HTML原始碼, "html.parser")
```



BeautifulSoup模組範例

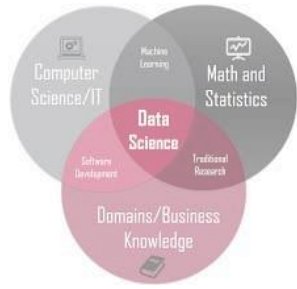
```
import requests
from bs4 import BeautifulSoup
url= 'http://ehappy.tw/bsdemo1.htm'
html = requests.get(url)
html.encoding="utf-8"
sp = BeautifulSoup(html.text,'html.parser')
# print(sp.text) 顯示內容，不包含html標籤
# print(sp.title) 顯示指定標籤內容
# print(sp.title.text) 顯示指定標籤內容, 不包含html標籤
```

BeautifulSoup模組的常用方法



- 以 **標籤(tag)** 為尋找標的時使用。
- **find()**：傳回第一個符合條件的標籤，找到會傳回一個**字串**，如果找不到則傳回 **None**。
- **find_all()**：傳回所有符合條件的標籤，找到會傳回一個**串列**，如果找不到則傳回空的串列。

<http://ehappy.tw/bsdemo2.htm>



BeautifulSoup模組取標籤屬性

- 語法

bsobj.get(屬性名稱)

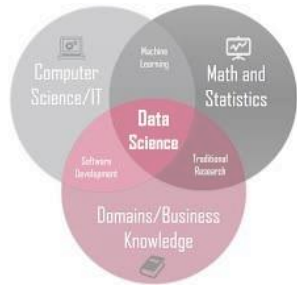
- 例如：讀取 <a> 標籤中的 href 的超連結

```
<a href="http://example.com/elsie" class="example"
```

```
id="link1">Elsie</a>
```

```
data=sp.find("a", id="link1")
```

```
print(data.get("href"))
```



BeautifulSoup模組取標籤屬性

- 語法

bsobj['屬性名稱']

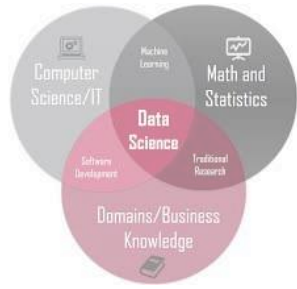
- 例如：讀取 `<a>` 標籤中的 href 的超連結

```
<a href="http://example.com/elsie" class="example" id="link1">Elsie</a>
```

```
data=sp.find("a", id="link1")
```

```
print(data["href"])
```

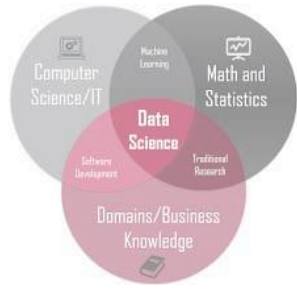

Lab3-1 : PTT 美食版爬蟲



PTT 美食版為例，希望從網頁中取得討論主題標題及連結：

```
import requests
from bs4 import BeautifulSoup
url= 'https://www.ptt.cc/bbs/Food/index.html'
html = requests.get(url)
html.encoding="utf-8"
sp = BeautifulSoup(html.text,'html.parser')
```

Lab3-1 : PTT 美食版爬蟲

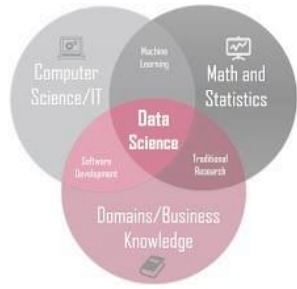


使用 find() 及 find_all() 分析主題區的HTML結構進行讀取：

```
import requests
from bs4 import BeautifulSoup
url= 'https://www.ptt.cc/bbs/Food/index.html'
html = requests.get(url)
html.encoding="utf-8"
sp = BeautifulSoup(html.text,'html.parser')


links = sp.find_all("div", class_="title")
for link in links:
    adata = link.find("a")
    print("http://www.ptt.cc" + adata['href'], adata.text)
```

Lab3-2 : 影城電影資料爬蟲





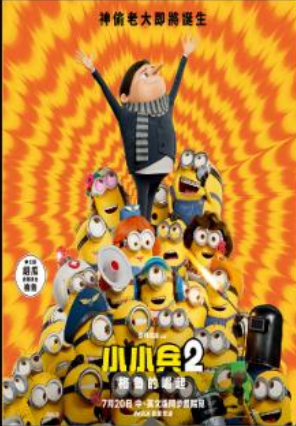





```
import requests, os
from bs4 import BeautifulSoup
url =
'https://srm.com.tw/%e9%9b%bb%e5%bd%b1%e7%b0%a1%e4%bb%8b/%e7%86%b1%e6%98%a0%e4%b
8%ad/'
r = requests.get(url)
sp = BeautifulSoup(r.text, 'html.parser')
datas = sp.find_all('article', class_='main_color inner-entry')
for i in range(len(datas)):
    print(f"【{datas[i].find('header').a['title']}】")
    print(datas[i].find('div', itemprop='text').text)
    print(datas[i].find('header').a['href'])
    print(datas[i].img['src'])
    if not os.path.exists('photos'):
        os.mkdir('photos')
    with open('photos/'+datas[i].find('header').a['title']+'.jpg', 'wb') as f:
        f.write(requests.get(datas[i].img['src']).content)
    print()
```

日日新影城-熱映電影



關於影城 電影簡介 場次查詢 影城服務 會員專區

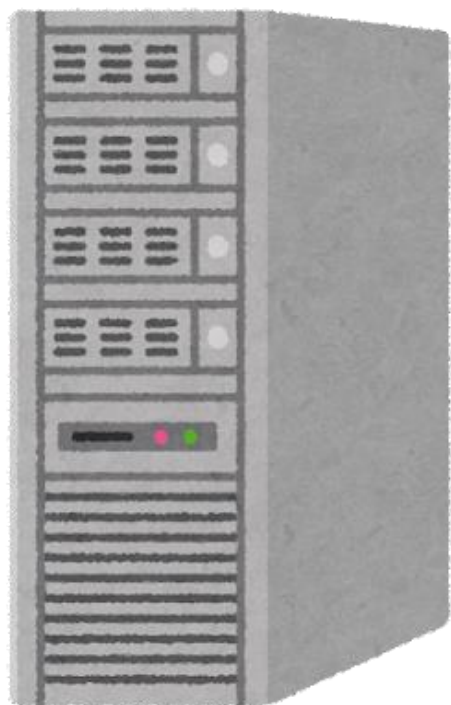


 <p>小小兵2：格魯的崛起</p> <p>上映日期： 2022.07.20</p>	 <p>邪門</p> <p>上映日期： 2022.07.15</p>	 <p>雷神索爾：愛與雷霆</p> <p>上映日期： 2022.07.06</p>	 <p>魔女二部曲：另一個她</p> <p>上映日期： 2022.07.01</p>	 <p>名偵探柯南：萬聖節的新娘</p> <p>上映日期： 2022.06.30</p>	 <p>侏羅紀世界：統霸天下</p> <p>上映日期： 2022.06.08</p>
---	---	---	---	---	---

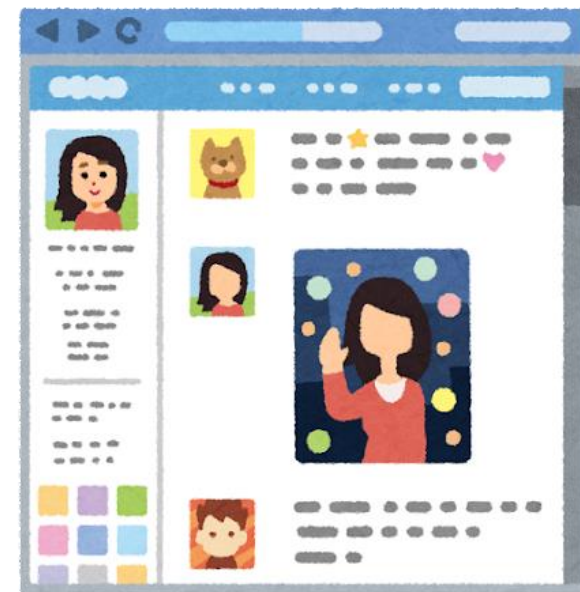
<https://srm.com.tw/電影簡介/熱映中/>

非同步載入(Ajax)網頁運作方式

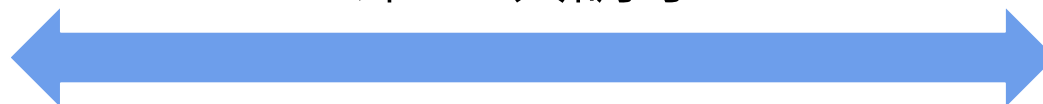
網頁伺服器



瀏覽器

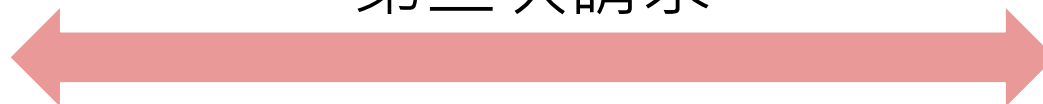


第一次請求



不帶資料的HTML

第二次請求



取得資料更新HTML

認識CDC Covid-19 每日數據網站



<https://sites.google.com/cdc.gov.tw/2019ncov/taiwan>

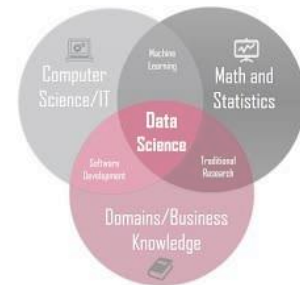
擷取非同步載入的資料來源

The screenshot displays a web browser window with the URL `https://sites.google.com/cdc.gov.tw/2019ncov/taiwan`. The page title is "COVID-19" and it features a "Global" / "Taiwan" toggle. The main content area, titled "國內通報總計", shows four statistics: "通報數" (2,361,186), "排除" (2,343,668), "確診" (15,862), and "死亡" (821). The "Network" tab in the Chrome DevTools is open, showing a list of requests. The selected request is a "Fetch/XHR" request, and its "Preview" tab is active, displaying a JSON response with the following data:

```
{...}
0: {
  確診: "15,862",
  死亡: 821,
  送驗: "2,361,186",
  排除: "2,343,668",
  昨日確診: 9,
  昨日排除: "12,658",
  昨日送驗: "12,972"
}
```

The "Console" tab is also visible at the bottom of the DevTools window.

Lab3-3 : Covid-19資料爬取



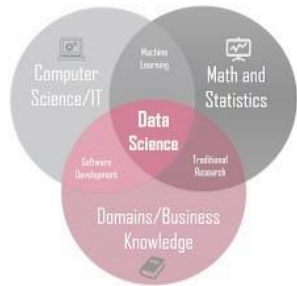
```
import requests
from bs4 import BeautifulSoup
url = "https://covid19dashboard.cdc.gov.tw/dash3"
html = requests.get(url)

# 將文字資料轉為字典格式
datas = eval(html.text)
# 取出每個欄位及值
for k, v in datas["0"].items():
    print(k + ":" + str(v))
```


Part 3

文字與檔案資料 的收集





檔案操作

- os 模組

- 匯入模組

- ```
import os
```

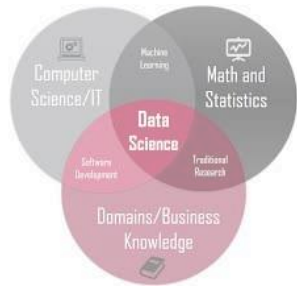
- os 函式

- `getcwd()`、`remove()`、`mkdir()`、`rmdir()`、
    - `makedirs()`

- 例如：取得目前的路徑。

- ```
import os
```

- ```
os.getcwd()
```



# Shutil 模組操作

- 匯入模組

```
import shutil
```

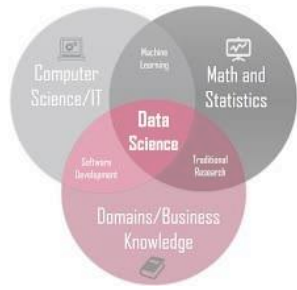
- shutil 函式

```
rmtree()
```

- 例如：刪除指定目錄(非空目錄)

```
import shutil
```

```
shutil.rmtree("第一層目錄/第二層目錄")
```



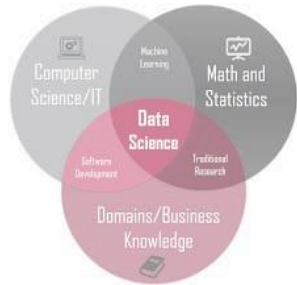
# os.path 模組操作

- 匯入模組

```
import os
```

- os.path 函式

```
abspath()、basename()、exists()、
getsize() split()、join()
```



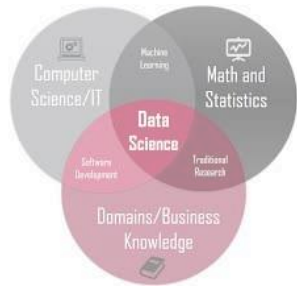
# os.path 函式

| 函式         | 說明                                            |
|------------|-----------------------------------------------|
| abspath()  | 傳回檔案完整的路徑名稱。                                  |
| basename() | 傳回檔案路徑名稱最後的檔案或路徑名稱。如果測試的是檔案會傳回檔名，測試的是路徑會傳回路徑。 |
| exists()   | 檢查指定的檔案或路徑是否存在。                               |
| getsize()  | 取得指定檔案的大小 (Bytes)。                            |
| split()    | 分割檔案路徑名稱為目錄路徑和檔案。                             |
| join()     | 將路徑和檔案名稱結合為完整路徑。                              |

□ 例如:顯示檔案的完整路徑

```
import os
file = "hello.py"
os.path.abspath(file)
```

# os.walk 函式操作



- 匯入模組

```
import os
```

- os.walk 函式

```
walk(path)
```

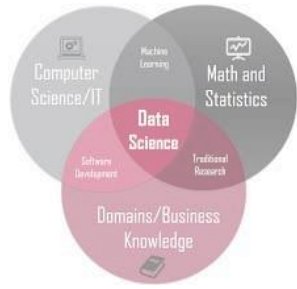
- 例如:取得 oswalk 路徑的檔案串列

```
import os
```

```
sample_tree=os.walk("/content/oswalk")
```

```
for dirname,subdir,files in sample_tree:
```

```
 print("檔案串列",files)
```



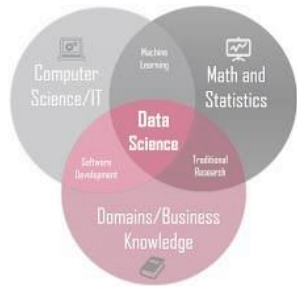
# 檔案的讀寫

- 用途：
  - 開啟指定的檔案，包括文字檔案和二進位檔案，以便進行檔案內容的讀取、寫入或修改。
- 語法：
  - `open(檔案名稱[, 模式][, 編碼])`
    - 檔案名稱：設定檔案的名稱，它是字串型態
    - 模式：
      - R 讀取模式，此為預設模式。
      - W 寫入模式，若檔案已存在，內容將會被覆蓋。
      - A 附加模式，若檔案已存在，內容會被附加至檔案尾端。
      - 編碼：指定檔案的編碼模式，一般可設定cp950或UTF-8。

# 查看檔案的編碼

```
import locale
locale.getpreferredencoding()
```





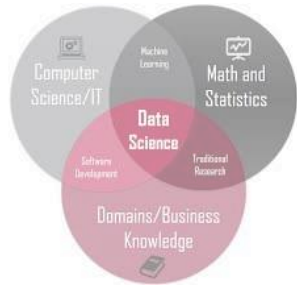
# 文字檔案的讀寫

- 寫入檔案:

```
f=open("file1.txt","w",encoding="utf-8")
f.write(content)
f.close()
```

- 讀取檔案

```
f=open("file1.txt","r",encoding="utf-8")
for line in f:
 print(line)
f.close()
```



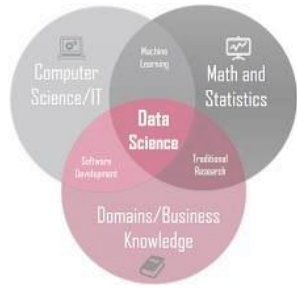
# 讀取文字檔求數字總和

- 使用 with open:

```
With open("file1.txt", "r") as f:
 lines=f.readlines()
```

- 範例：求各列數字的總和

以 `f.write("2\n4\n6")` 儲存數字字串，再以 `lines=f.readlines()` 讀取數字字串求總和。



# 讀取部份字元

□ 範例：移動到指定的位置，讀取一系列或n個字元。

```
f=open("file1.txt","r",encoding="utf-8")
```

```
f.seek(0) # 移至開始的位置
```

```
print(f.tell())
```

```
line1=f.readline() # 讀取一行
```

```
print(line1)
```

```
print(f.tell())
```

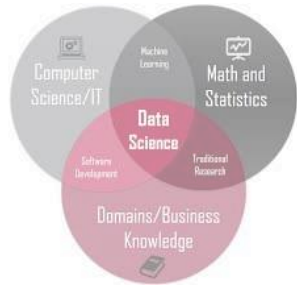
```
data=f.read(5) # 讀取5個字元
```

```
print(data)
```

```
data=f.read() # 讀取剩下的內容
```

```
print(data)
```

# 二進位檔案的讀寫



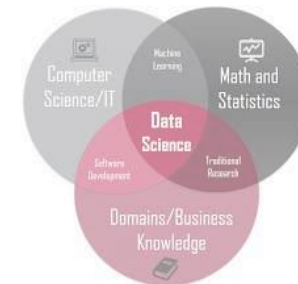
- 寫入二進位檔案:

```
content=content.encode("utf-8") #轉為 bytes
f=open("file1.bin","wb")
f.write(content) # content 必須是 bytes
f.close()
```

- 讀取二進位檔案

```
with open("file.bin","rb") as f: # 不可加 encoding 參數
 content=f.read().decode("utf-8") # 轉為字串
 print(content)
```

# 二進位檔案的應用



- 範例：讀取下列 url 的圖片並存檔。

[https://www.google.com/images/branding/googlelogo/2x/\\_googlelogo\\_color\\_272x92dp.png](https://www.google.com/images/branding/googlelogo/2x/_googlelogo_color_272x92dp.png)

