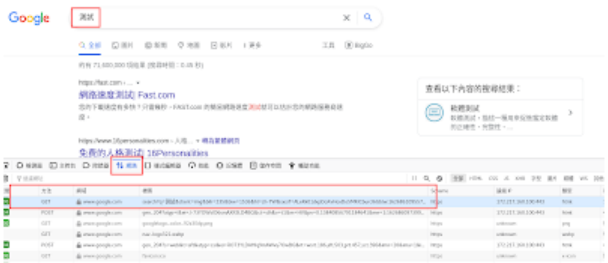
到網路上爬圖片吧！

學習目標：

- 寫出網路爬蟲程式，將 Google 上的圖檔回傳！

寫出網路爬蟲程式

1. 開啟瀏覽器，利用 Google 進行查詢！按下「F12」進行網址的觀察！



2. 在 python 文字介面中，進行分析與測試：

```
C:\workspace\LineBot> python3
>>>
```

3. 導入 urllib ，對 google 進行網站的連結：

```
>>> import urllib.request
>>> url = "https://www.google.com/"
>>> conn = urllib.request.urlopen(url)
>>> print(conn)
<http.client.HTTPResponse object at 0x7f0d76035550>
```

4. 將接收的物件，轉成資料印出來：

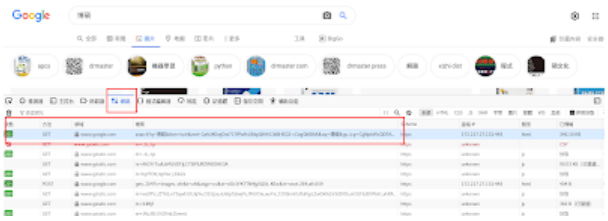
```
>>> data = conn.read()
>>> print(data)
(印出的資料太多了，省略一下...)
```

5. 修正一下，將 headers 的參數加上，限制資料印出的數量：

```
>>> header = { 'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0' }
>>> req = urllib.request.Request(url,headers=header)
>>> conn = urllib.request.urlopen(req)
>>> data = conn.read()
>>> print(data)
(印出的資料太多了，省略一下...)
```

PS: 你有個網頁好幫手「F12」！

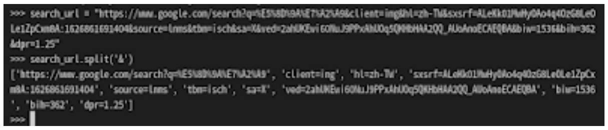
6. 回瀏覽器，在 google 查詢一下某家書商的名稱，再切換成圖片，觀察一下網址變化



PS: 注意其網址的組成！

7. 上圖中，按下滑鼠右鍵，可以「複製網址」

8. 將網址複製後，貼至 python 的文字介面視窗內，進行分析！



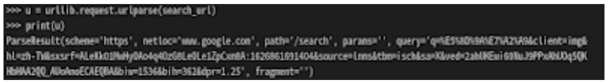
```
>>> search_url = "https://www.google.com/search?q=%E5%8D%9A%E7%A2%A9&client=img&hl=zh-TW&sxsrf=ALeKk01MwHy0Ao4q40zG8Le0
Le1ZpCxm8A:1626861691404&source=lnms&tbm=isch&sa=X&ved=2ahUKEwi60NuJ9PPxAhUOq5QKHbHAA2QQ_AUoAnoECAEQBA&biw=1536&bih=362
&dpr=1.25"
>>> search_url.split('&')
['https://www.google.com/search?q=%E5%8D%9A%E7%A2%A9', 'client=img', 'hl=zh-TW', 'sxsrf=ALeKk01MwHy0Ao4q40zG8Le0Le1ZpCx
m8A:1626861691404', 'source=lnms', 'tbm=isch', 'sa=X', 'ved=2ahUKEwi60NuJ9PPxAhUOq5QKHbHAA2QQ_AUoAnoECAEQBA', 'biw=1536
', 'bih=362', 'dpr=1.25']
>>> 
```

PS:觀察之後，可以猜測：

- q=%E5%8D%9A%E7%A2%A9: 代表查詢字串
- tbm=isch :指的是查詢圖片

9. 使用分析函式，進行相關網址分析：

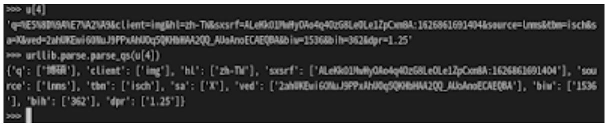
```
>>> u = urllib.request.urlparse(search_url)
>>> print(u)
```



```
>>> u = urllib.request.urlparse(search_url)
>>> print(u)
ParseResult(scheme='https', netloc='www.google.com', path='/search', params='', query='q=%E5%8D%9A%E7%A2%A9&client=img&
hl=zh-TW&sxsrf=ALeKk01MwHy0Ao4q40zG8Le0Le1ZpCxm8A:1626861691404&source=lnms&tbm=isch&sa=X&ved=2ahUKEwi60NuJ9PPxAhUOq5QK
HbHAA2QQ_AUoAnoECAEQBA&biw=1536&bih=362&dpr=1.25', fragment='')
```

10. 進行下一步的分析！

```
>>> u[4]
>>> urllib.parse.parse_qs(u[4])
```



```
>>> u[4]
'q=%E5%8D%9A%E7%A2%A9&client=img&hl=zh-TW&sxsrf=ALeKk01MwHy0Ao4q40zG8Le0Le1ZpCxm8A:1626861691404&source=lnms&tbm=isch&s
a=X&ved=2ahUKEwi60NuJ9PPxAhUOq5QKHbHAA2QQ_AUoAnoECAEQBA&biw=1536&bih=362&dpr=1.25'
>>> urllib.parse.parse_qs(u[4])
{'q': ['博碩'], 'client': ['img'], 'hl': ['zh-TW'], 'sxsrf': ['ALeKk01MwHy0Ao4q40zG8Le0Le1ZpCxm8A:1626861691404'], 'sou
rce': ['lnms'], 'tbm': ['isch'], 'sa': ['X'], 'ved': ['2ahUKEwi60NuJ9PPxAhUOq5QKHbHAA2QQ_AUoAnoECAEQBA'], 'biw': ['1536
'], 'bih': ['362'], 'dpr': ['1.25']}
>>> 
```

11. URL 分析列表，有助於組合回原來的查詢字串：

Attribute	Index	Value	Value if not present
scheme	0	URL scheme specifier	scheme parameter
netloc	1	Network location part	empty string
path	2	Hierarchical path	empty string
params	3	Parameters for last path element	empty string
query	4	Query component	empty string
fragment	5	Fragment identifier	empty string
username		User name	None
password		Password	None
hostname		Host name (lower case)	None
port		Port number as integer,if present	None

12. 大致上了解其組成結構後，可以進行測試：

```
>>> test = {'tbm': 'isch', 'q': '博碩'}
>>> urllib.parse.urlencode(test)
'tbm=isch&q=%E5%8D%9A%E7%A2%A9'
```

13. 將下列字串，放回瀏覽器的網址列，觀查結果是否相同：

```
https://www.google.com/search?tbm=isch&q=%E5%8D%9A%E7%A2%A9
```

14. 回到文字介面中，持續進行測試：

```
>>> url = f"https://www.google.com/search?{urllib.parse.urlencode(test)}/"
>>> req = urllib.request.Request(url, headers=header)
>>> conn = urllib.request.urlopen(req)
>>> data = conn.read()
>>> print(data)
(資料出現太多，省略過去....)
```

15. 從瀏覽器中，分析圖片位於 HTML 語法中的何處！提示：在「檢視原始碼中」，查詢關鍵字詞："img data-src"
16. 切回文字介面，設定關鍵字詞的樣板：正規化設定

    ```
    >>> import re
    >>> template = '"(https://encrypted-tbn0.gstatic.com[\S]*)"'
    >>> image_list = []
    >>> for i in re.finditer(template,str(data,'utf-8')):
    ...     image_list.append(i.group(1))
    >>> image_list[:5]
    ```

    PS: 語法注意事項

    - [\S]: 空白字元除外
    - * : 任意字數的字元
    - .group(1): 只擷取 template 字串中的有（）號的內容資料
    - [:5] : 取回前五行資料！
17. 整理下過的指令，可容易形成程式檔案：

    ```
    import urllib.request
    import re
    import random

    search_key_word = {'tbm': 'isch', 'q': event.message.text}
    url = f"https://www.google.com/search?{urllib.parse.urlencode(search_key_word)}/"
    header = { 'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0' }
    req = urllib.request.Request(url, headers=header)
    conn = urllib.request.urlopen(req)
    data = conn.read()
    template = '"(https://encrypted-tbn0.gstatic.com[\S]*)"'
    image_list = []
    for i in re.finditer(template,str(data,'utf-8')):
      image_list.append(i.group(1))

    random_image_url = image_list[random.randint(0, len(image_list)-1)]

    line_bot_api.reply_message(
      event.reply_token,
      ImageSendMessage(
        original_content_url=random_image_url,
        preview_image_url=random_image_url
      )
    )
    ```
18. 利用 line-bot-sdk-python 提供的 TemplateSendMessage 可以一次取得多張圖片：

    ```
    TemplateSendMessage(
      alt_text=alt_text
      template=ImageCarouselTemplate(
        columns=[ImageCarouselColumn(
          image_url='https://website/image.jpg',
          action=URIAction(uri='https://website',label='label'))]
      )
    )
    ```
19. 修改 LineBot/app/linebotmodules.py 檔案，將上面試過的指令，一一寫入檔案內！

    ```
    from linebot.models.send_messages import ImageSendMessage
    from app import line_bot_api, handler
    from linebot.models import MessageEvent, TextMessage, TextSendMessage

    import urllib.request
    import re
    import random

    # 查詢 google
    ```

```
@handler.add(MessageEvent, message=TextMessage)
def replyText(event):
    if event.source.user_id == "Uf4a596a6eb65eabf52c003ffe325a21d":

            search_key_word = {'tbm': 'isch', 'q': event.message.text}
            url = f"https://www.google.com/search?{urllib.parse.urlencode(search_key_word)}/"
            header = { 'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0' }
            req = urllib.request.Request(url, headers=header)
            conn = urllib.request.urlopen(req)
            data = conn.read()
            template = '"(https://encrypted-tbn0.gstatic.com[\S]*)"'
            image_list = []

            for i in re.finditer(template,str(data,'utf-8')):
                image_list.append(re.sub(r'\\u003d','=',i.group(1)))

            random_image_url = image_list[random.randint(0, len(image_list)+1)]

            line_bot_api.reply_message(
                event.reply_token,
                ImageSendMessage(
                    original_content_url=random_image_url,
                    preview_image_url=random_image_url
                )
            )
```

20. 將程式推上 Heroku 主機，並且進行測試！
21. 修改 LineBot/app/linebotmodules.py 檔案，加入 TemplateSendMessage 模組！

```
from linebot.models.send_messages import ImageSendMessage
from app import line_bot_api, handler
from linebot.models import MessageEvent, TextMessage, TextSendMessage

import urllib.request
import re
import random

# 查詢 google
@handler.add(MessageEvent, message=TextMessage)
def replyText(event):
    if event.source.user_id == "Uf4a596a6eb65eabf52c003ffe325a21d":
            search_key_word = {'tbm': 'isch', 'q': event.message.text, 'client': 'img'}
            url = f"https://www.google.com/search?{urllib.parse.urlencode(search_key_word)}/"
            header = { 'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0' }
            req = urllib.request.Request(url, headers=header)
            conn = urllib.request.urlopen(req)
            data = conn.read()
            template = '"(https://encrypted-tbn0.gstatic.com[\S]*)"'
            image_list = []
            for i in re.finditer(template,str(data,'utf-8')):
                image_list.append(re.sub(r'\\u003d','=',i.group(1)))

            #random_image_url = image_list[random.randint(0, len(image_list)-1)]
            random_image_list = random.sample(image_list,k=3)

            image_template = ImageCarouselTemplate(
                columns=[ImageCarouselColumn(image_url=urx,action=URIAction(label=f'image{j}',
                uri=urx)) for j,urx in enumerate(random_image_list)]
            )

            line_bot_api.reply_message(
                event.reply_token,
                TemplateSendMessage(
                    alt_text='Hello World',
                    template=image_template
                )
            )
```

PS: heroku 可能會當機，膽小者勿試！