(1) 開發環境：MacOS version==11.2.3 using Python=3.7

(2) 程式邏輯：

Google search(可自行輸入關鍵字跟頁數) -> google news button -> 爬蟲標題以及連結做儲存

INSIDE news website -> 爬蟲所有大標題以及連結儲存

(3) 函式功能說明：

GoogleNews.py:

Google 這個 function 先去偽造 user_agent 以避免多次爬蟲被鎖 ip,

再來讓使用者可以自行在 GUI 中輸入關鍵字和所要爬的頁數，

我先將所有每一頁(搜尋頁)的 url 存在 urls 這個 list 當中，為了能讓我丟給 threads 時能夠平行運行

丟給 threads 幫我們運行

```
def google(path, keyW, pageN):
    start_time = time.time()
    user_agent = UserAgent()
    headers = {'User-Agent': user_agent.random}
    keyword = quote(keyW.encode('utf8'))
    pageNum = int(pageN)
    base_url = 'https://www.google.com.tw'
    search_url = base_url + f'/search?q={keyword}&gbv=2&tbm=nws'

    urls = []

    for _ in range(pageNum):
        urls.append(search_url)
        res = requests.get(search_url, headers=headers)
        soup = BeautifulSoup(res.text, 'lxml')
        next_page_html = soup.select('div.nMymef.MUxGbd.lyLwlc > a[aria-label^="下一頁"]')
        next_url = base_url + next_page_html[0]['href']
        search_url = next_url

    threads = []
    #lock = threading.Lock()
    for i in range(len(urls)):
        threads.append(googleNews(base_url, urls[i], path))
        threads[i].start()

    for i in range(len(urls)):
        threads[i].join()

    end_time = time.time()
    print(f"{end_time - start_time} 秒爬取 {pageNum} 頁的文章")
    print('-------End-------')
```

利用物件導向的方式寫 threads，
我將我的每一頁的 url 給傳入，
在這里利用 beautifulsoup 來爬我的標籤，
最後輸出 title 以及連結並且因為要連續寫入，所以利用 lock 來能夠一比一比鎖住，最後輸出給使用者想要的路徑。

```python
#google news
class googleNews(threading.Thread):
    def __init__(self, base_url, url, path):
        threading.Thread.__init__(self)
        self.base_url = base_url
        self.url = url
        self.lock = threading.Lock()
        self.path = path


    def run(self):
        self.res = requests.get(self.url)
        self.soup = BeautifulSoup(self.res.text, 'lxml')
        self.title_html = self.soup.select('h3.zBAuLc  div')
        self.website_html = self.soup.select('div.ZINbbc.xpd.O9g5cc.uUPGi > div.kCrYT > a[href^="/url"]')
        self.website_urls = [self.website_html[i]['href'].strip('/url?q=') for i in range(len(self.website_html))]
        self.file = self.path + '/google_Link.txt'



        self.lock.acquire()
        with open(self.file, 'w') as file:
            for title, link in zip(self.title_html, self.website_urls):
                #print('Title: ', title.text)
                #print('Website: ', link)
                file.write('Title: '+title.text+'\nWebsite: '+link+'\n')
        self.lock.release()

        time.sleep(random.randint(1,5))
```

INSIDENews.py:
這裡的 function 大同小異，一樣是透過 url 的方式，將多個 urls 丟給 threads 去做執行

```python
def Inside(path, keyword, page):

    thread = []
    base_url = "https://www.inside.com.tw/tag/AI"
    urls = [f"{base_url}?page={page}" for page in range(1, int(page)+1)]


    threads = []
    for i in range(int(page)):
        threads.append(Crawls([urls[i]], path, int(page)))
        threads[i].start()

    for i in range(int(page)):
        threads[i].join()
```

這裡和上面 google news 的也幾乎一樣，會去依照使用者選取的路徑儲存檔案，並且利用 beautifulsoup 來爬我的標籤，最後將我的 title 以及 website 給儲存。

```python
class Crawls(threading.Thread):
    def __init__(self, urls, path, pageNum):
        threading.Thread.__init__(self)
        self.urls = urls
        self.lock = threading.Lock()
        self.path = path
        self.pageNum = pageNum


    def run(self):
        self.file = self.path + '/INSIDE_News(AI)_Link.txt'

        for url in self.urls:
            response = requests.get(url)
            soup = BeautifulSoup(response.content, "lxml")

            titles = soup.find_all("h3", {"class": "post_title"})
            website = soup.select('div.post_list_item_content a[class^="js-auto_break_title"]')


            self.lock.acquire()
            with open(self.file, 'w') as file:
                for i in range(len(website)):
                    link = website[i]['href']
                    title = titles[i].getText().strip()
                    file.write('Title: ' + title + '\nWebsite: ' + link + '\n')
            self.lock.release()

            time.sleep(random.randint(1, 5))
```

GUI.ipynb:
我自行設計的 GUI
利用 tkinter 以及物件導向的概念去寫

在這些 functions 中，會依照點擊的地方而有反應，
如 print_selection 會在上方黃色地方顯示點擊了哪種新聞，
exitWindow 則是點擊 exit 時，會結束 GUI
searchPage 則會先需要使用者輸入完東西以後，並且跳出對話框，要選擇哪個路徑做儲存，並且依照所選的新聞網站來儲存．

```
def print_selection(self):
    self.labelSelect.config(text='Selected "{}"'.format(self.var_radio.get()))

def ExitWindow(self):
    try:
        self.window.destroy()
    except:
        pass

def searchPage(self):
    #self.window.withdraw()
    dir_path = filedialog.askdirectory(parent=self.window, initialdir='~/')
    self.radio = self.var_radio.get()
    self.Keyword = self.varKeyWord.get()
    self.Page = self.varPage.get()

    if self.radio == 'Google News':
        start_time = time.time()
        GoogleNews.google(dir_path, self.Keyword, self.Page)
        end_time = time.time()
        duration = end_time - start_time
        tk.messagebox.showinfo(title='reminder', message='共花費{}sec'.format(duration))

    if self.radio == 'INSIDE News(AI)':
        start_time = time.time()
        INSIDENews.Inside(dir_path, self.Keyword, self.Page)
        end_time = time.time()
        duration = end_time - start_time
        tk.messagebox.showinfo(title='reminder', message='共花費{}sec'.format(duration))
```