



程式設計與網路資料搜集 期末專案

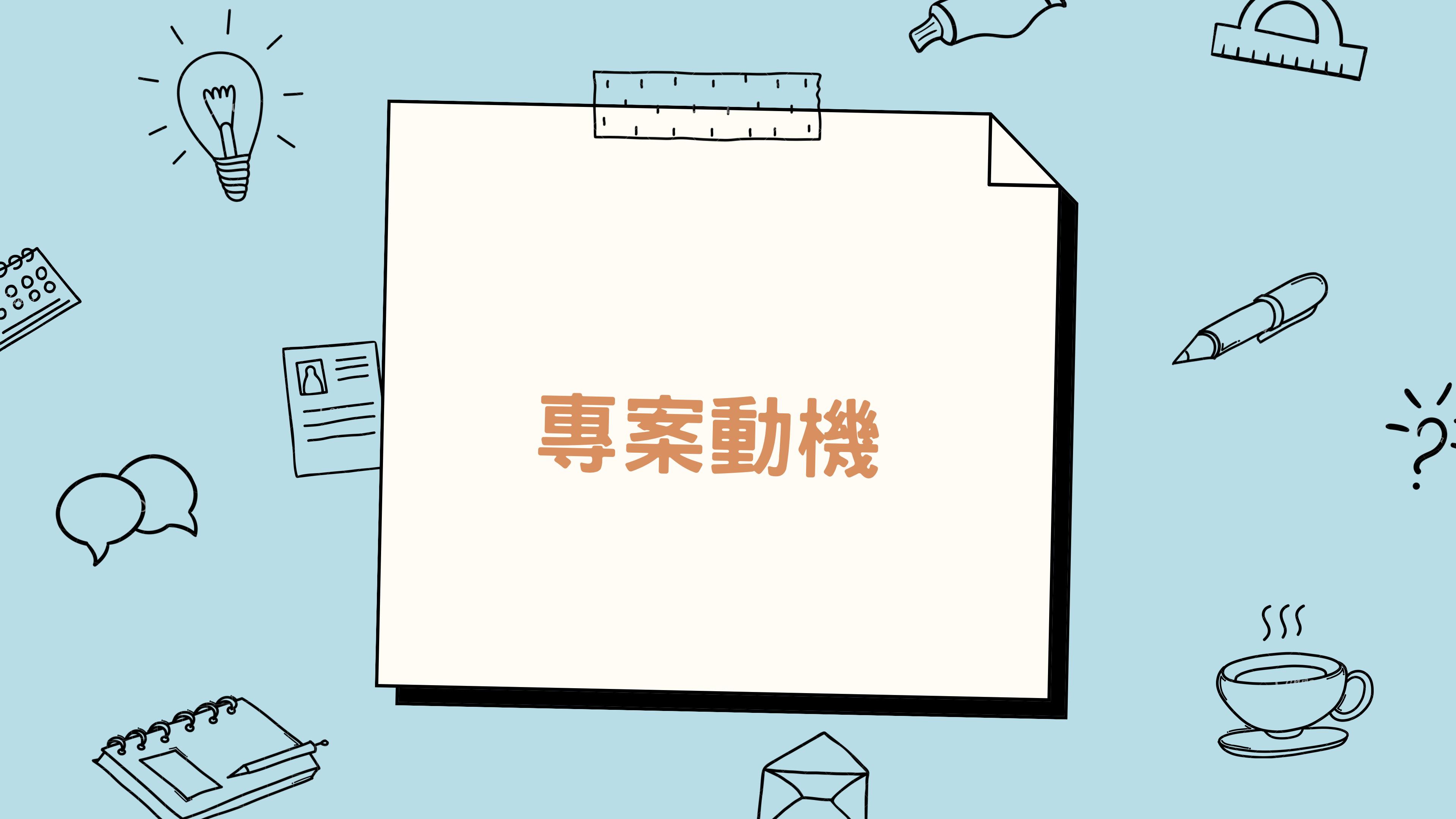
便宜的蝦皮哪裡買

蝦皮購物比價程式

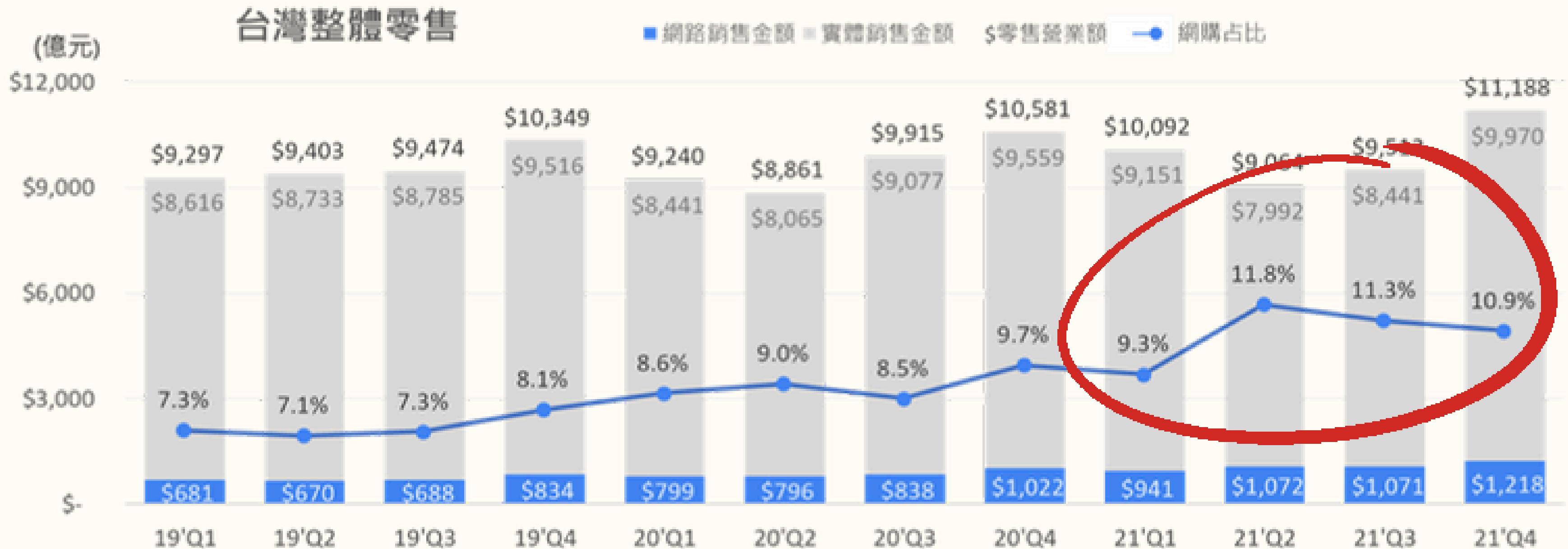
第 20 組

黃郁棻 呂昕玲 陳宛瑩 黃式傑

專案動機



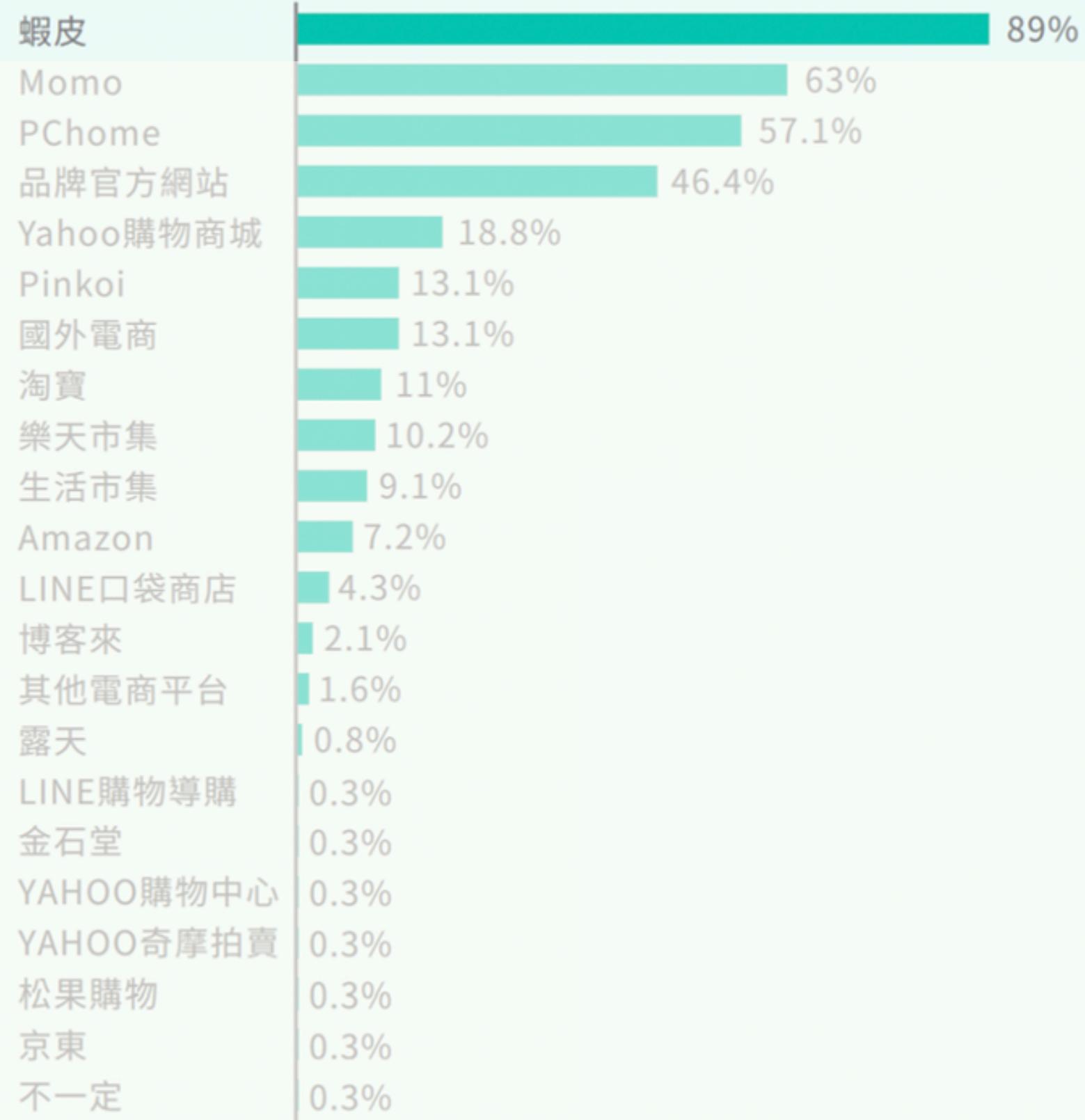
網路購物成為疫情/後疫情時代趨勢



資料來源：易普索市場研究〈後疫情時代，零售業如何應對？〉
經濟部統計處-批發零售餐飲業&零售業網路銷售額資料統計

年增24.5%

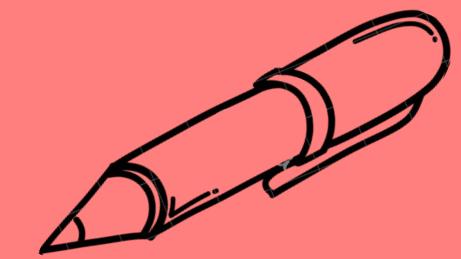
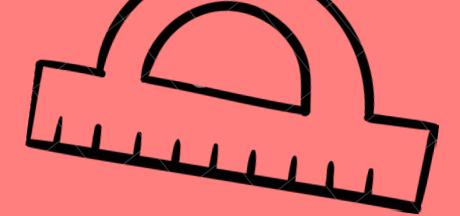
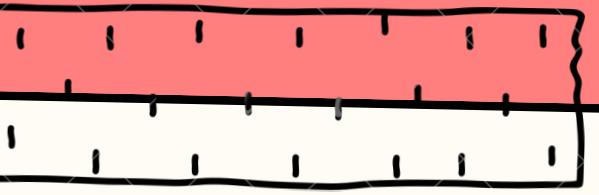
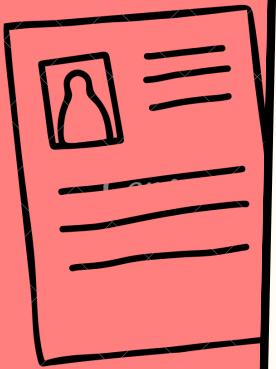
網購平台



蝦皮購物作為台灣
最主流網購平台之一

因此，
我們將以蝦皮購物為研究對象，
執行本次專題！

解決痛點





掃

搜尋 "掃" 寶場

掃地機器人



618 | 月月盛典

6/5早鳥搶購物金

最高折\$350



立即看▶

realme
所向披.me
登錄抽5G熱銷機
6/2 10:00 - 6/6 10:00



618 | 月月盛典 | 商城
SAMSUNG
指定品下單送豪禮
蝦幣享最高3,000

逛逛去▶

逛逛去▶



免運！蝦皮直送



蝦皮超市



宅家領券95折



蝦時尚



男人館



滿\$51折\$50



蝦皮海外直送



蝦皮超便宜



100%現折購物金



最強活動合集

品牌旗艦

今日爆殺



搶購去▶



滿\$10,000現金

天天搶9折券！

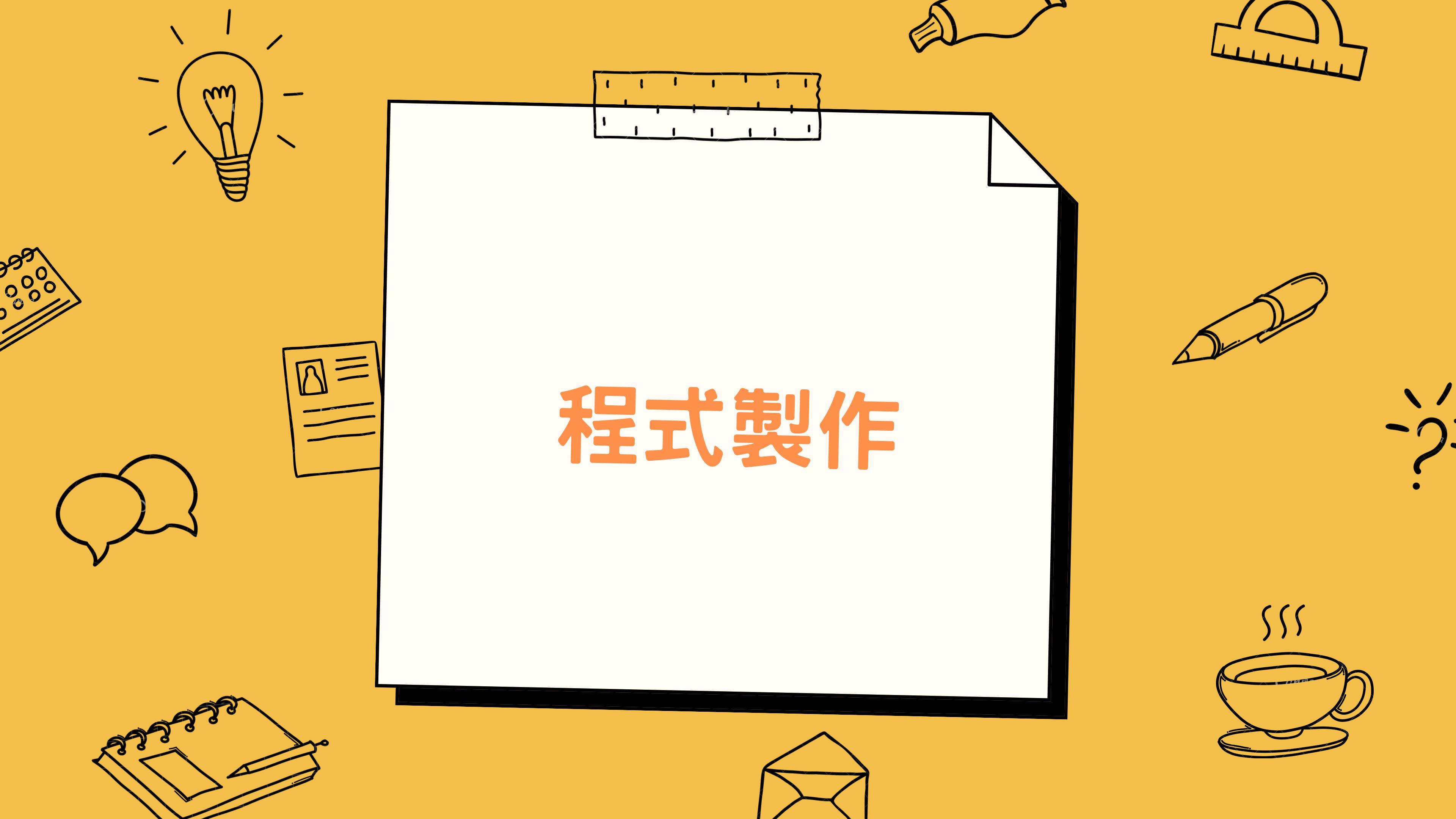
查看更多>

平台 缺點

- 搜尋結果出現大量廣告
- 折價券、免運優惠訊息不直觀
 - 👉 難以直接比價
 - 👉 優惠訊息隨時變動無從得知

結論：選擇商品曠日費時

程式製作



蝦皮優惠券 發放種類



運費抵用券

免運店家運費券

領券去▶



9折
低消\$599 最高折 \$70
即將失效: 剩下6小時

去逛逛



11/11限定 商城88折券
最高折\$5,000
使用期間：11/11 00:00-23:59

提醒我

優惠計算方式



商品價格



免運店家運費券



11/11限定 商城88折券

最高折\$5,000

使用期間：11/11 00:00-23:59



9折

低消\$599 最高折 \$70

即將失效: 剩下6小時

商品最低運費價格

商品最優折扣

=

最終購入價格

使用套件

網路爬蟲與
資料格式轉換



requests json
aiohttp urllib
asyncio

GUI介面架設



tkinter

程式製作(爬蟲)

```
1 import requests
2 from bs4 import BeautifulSoup
3 import json
4 # url編碼
5 import urllib
6 import aiohttp
7 import asyncio
8
9 class Shopee:
10     def __init__(self, Input):
11         self.Input = Input
12         self.Output = []
13
14     async def do(self):
15         async with aiohttp.ClientSession() as session:
16             price_lst = []
17             keyword = self.Input
18             limit = 50 # 回傳數量
19             max_price = 100000 # 最高價格
20             rate = 4 # 4星以上
21             headers = {
22                 'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36',
23                 'x-api-source': 'pc',
24                 'referer': f'https://shopee.tw/search?keyword={urllib.parse.quote(keyword)}'
25             }
26
27             # cookies
28             s = requests.Session()
29             url = 'https://shopee.tw/api/v4/search/product_labels'
30             r = s.get(url, headers=headers)
31
```

程式製作(爬蟲)

```
1 import requests
2 from bs4 import BeautifulSoup
3 import json
4 # url編碼
5 import urllib
6 import aiohttp
7 import asyncio
8
9 class Shopee:
10     def __init__(self, Input):
11         self.Input = Input
12         self.Output = []
13
14     async def do(self):
15         async with aiohttp.ClientSession() as session:
16             price_lst = []
17             keyword = self.Input
18             limit = 50 # 回傳數量
19             max_price = 100000 # 最高價格
20             rate = 4 # 4星以上
21             headers = {
22                 'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36',
23                 'x-api-source': 'pc',
24                 'referer': f'https://shopee.tw/search?keyword={urllib.parse.quote(keyword)}'
25             }
26
27             # cookies
28             s = requests.Session()
29             url = 'https://shopee.tw/api/v4/search/product_labels'
30             r = s.get(url, headers=headers)
31
```

使用asyncio去爬蟲

執行時間：21.589531 秒

原始程式碼

執行時間：71.522508 秒

```
32     # 搜尋頁網址
33     url = f"https://shopee.tw/api/v4/search/search_items/?by=relevancy&keyword={urllib.parse.quote(keyw
34     re = s.get(url, headers=headers)
35
36     # 確認資料無誤
37     if re.status_code == requests.codes.ok:
38         data = re.json()
39
40     # 商城優惠券
41     official_url = "https://shopee.tw/api/v2/voucher_wallet/batch_get_vouchers_by_promotion_ids"
42     async with session.get(official_url, headers = headers) as resp:
43         official = await resp.json()
44
45     # 總資料數
46     if data['query_rewrite'] == None:
47         note = "無查詢結果"
48         self.Output.append(note)
49         totalcount = 0
50         total_pg = 1
51         price_lst = [0]
52     else:
53         totalcount = data['query_rewrite']['ori_total_count']
54         # 分頁 (蝦皮每頁包含五十筆資料 ref:https://ithelp.ithome.com.tw/articles/10232930)
55         total_pg = (totalcount // 50) + 1
56         data['totalPage'] = total_pg
57         #print(f"total count: {totalcount}")
58         #print(f"total page: {total_pg}")
```

```
61 # 對每個分頁爬蟲
62 for num in range(0, 2):
63     #print("\nPage", num)
64     if totalcount == 0:
65         break
66     pg_url = f"https://shopee.tw/api/v4/search/search_items/?by=relevancy&keyword={urllib.parse.quote(
67     #page = num + 1
68     async with session.get(pg_url, headers = headers) as resp:
69         doc = await resp.json()
70         position = 0
71         count = 0
```

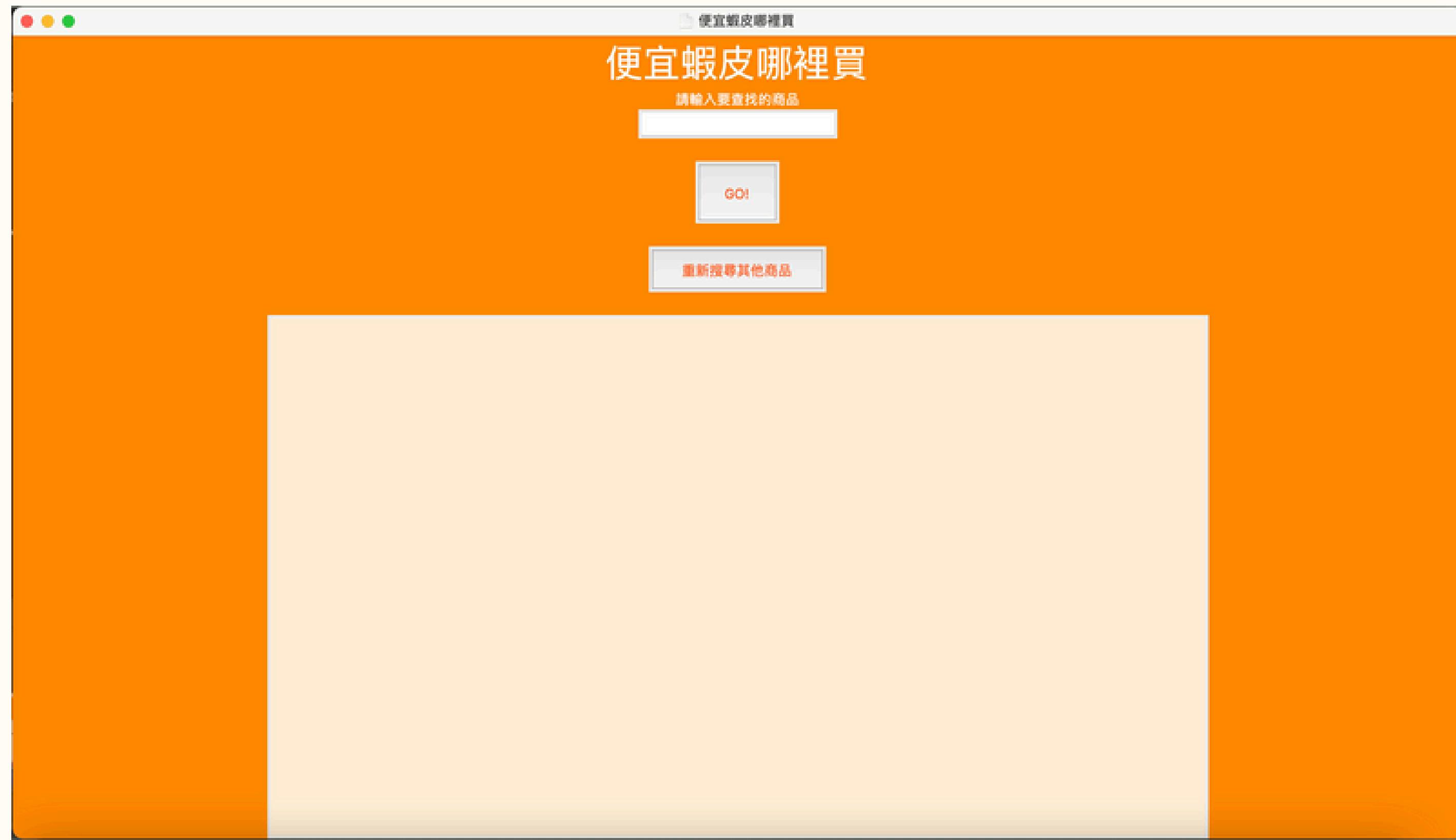
```
73     # 獲取分頁內每個商品的資訊
74     for d in doc['items']:
75         if totalcount == 0:
76             break
77         productid = d['item_basic']['itemid']
78         shopid = d['item_basic']['shopid']
79
80         # 商品名稱和網址
81         name = d['item_basic']['name']
82         # 商品網址
83         item_url = f"https://shopee.tw/product/{shopid}/{productid}"
84
85         # 商品內頁
86         # get的網址
87         re_item_url = f"https://shopee.tw/api/v4/item/get?itemid={productid}&shopid={shopid}"
88         async with session.get(re_item_url, headers = headers) as resp:
89             imf = await resp.json()
90
91         # 運費優惠內頁#
92         # get的網址
93         re_shipping_url = f"https://shopee.tw/api/v4/pdp/get_shipping?buyer_zipcode=&city=%E5%A4%A7"
94         async with session.get(re_shipping_url, headers = headers) as resp:
95             ims = await resp.json()
96
97         #每個細項名稱+運費#
98         ship = {} # ship = {運送方式: 運費}
99         if ims['data']['ungrouped_channel_infos'] == None:
100             ship['沒有適用的物流選項，請與賣家聯繫確認'] = 'N/A'
101         else:
102             for item in ims['data']['ungrouped_channel_infos']:
103                 ship[str(item['name'])] = int(item['min_price']/100000)
```

```
105     # 每個細項名稱+賣場優惠卷+價格
106     # voucher_discount = [(min_spend, discount_value)]
107     voucher_discount = []
108     # voucher_percent = [(min_spend, discount_percentage, max_discount)]
109     voucher_percent = []
110     for i, item in enumerate(imf['data']['shop_vouchers']):
111         if item['discount_value'] != None and item['discount_value'] > 0:
112             voucher_discount.append(
113                 (int(item['min_spend']/100000), int(item['discount_value']/100000)))
114             #print(f" 第{i+1}張折價券 滿{int(item['min_spend']/100000)} 折抵{int(item['discount_value'])}")
115         elif item['discount_percentage'] != None and item['discount_percentage'] > 0:
116             voucher_percent.append((int(item['min_spend']/100000), int(
117                 100 - item['discount_percentage']), int(item['discount_cap']/100000)))
118             #print(f" 第{i+1}張折價券 滿{int(item['min_spend']/100000)} 打{int(100 - item['discount_percentage'])}折")
119
120     # 組合優惠
121     bundle = []
122     if imf['data']['bundle_deal_info'] != None:
123         bundle.append(
124             imf['data']['bundle_deal_info']['bundle_deal_label'])
125             #print(f" 組合優惠:{imf['data']['bundle_deal_info']['bundle_deal_label']}")
126
127     # 商城
128     if imf['data']['is_official_shop'] == True:
129         official_shop = True
130     else:
131         official_shop = False
```

```
133     for i, item in enumerate(imf['data']['models']):
134         note = ""
135         # 過濾已經沒貨的
136         if item['stock'] == 0:
137             continue
138
139         # 算最低折扣價錢
140         price = int(item['price']/100000)
141         lowest_price = price
142         voucher_use = "無"
143         for voucher in voucher_discount:
144             if (price >= voucher[0]):
145                 price_dis = price - voucher[1]
146                 if (price_dis < lowest_price):
147                     lowest_price = price_dis
148                     voucher_use = f"滿{voucher[0]} 折抵{voucher[1]}元"
149         for voucher in voucher_percent:
150             if (price >= voucher[0]):
151                 if (round(price * voucher[1]/100) > price - voucher[2]):
152                     price_per = round(price * voucher[1]/100)
153                 else:
154                     price_per = price - voucher[2]
155                 if (price_per < lowest_price):
156                     lowest_price = price_per
157                     voucher_use = f"滿{voucher[0]} 打{voucher[1]}折 最高折抵{voucher[2]}元"
158
```

```
159     if len(imf['data']['models']) == 1:
160         note += f"商品名稱: {name}\n網址: {item_url}\n"
161     else:
162         note += f"商品名稱: {name}\n網址: {item_url}\n商品選項: {item['name']}\n"
163     note += f"原價: {price} 折扣後價格: {lowest_price} 庫存: {item['stock']} 所使用優惠
164     note += "運費: \n"
165     price_lst.append(lowest_price)
166
167     if (len(ship) != 0):
168         for k in ship.keys():
169             note += f"\t{k}: 運費{ship[k]}元\n"
170     void = True
171     note += "其他優惠: "
172     if (official_shop == True):
173         note += "\t商城優惠券"
174         void = False
175     if (len(bundle) != 0):
176         note += "\n\t".join(bundle)
177         void = False
178     if (void):
179         note += "\t無"
180     note += "\n\n"
181     self.Output.append(note)
```

```
183     rank = sorted(range(len(price_lst)), key=lambda p: price_lst[p])
184     #print(price_lst)
185     self.Output=[i for _,i in sorted(zip((price_lst), self.Output))]
186
```



程式製作(GUI)

```
1 import test
2 from tkinter import *
3 import asyncio
4 import nest_asyncio
5 import time
6
7 nest_asyncio.apply()
8 def printtxt():
9     A = en.get()
10    a = Shopee(str(A))
11    start = time.perf_counter() # 開始測量執行時間
12    asyncio.run(a.do())
13    elapsed = time.perf_counter() - start # 計算程式執行時間
14    print("執行時間 : %f 秒" % elapsed)
15    for i in range(len(a.Output)):
16        show.insert('insert',a.Output[i])
17
```

```
18 # 建立主視窗
19 shopee = Tk()
20 shopee.title("便宜蝦皮哪裡買")
21
22 # 標題\大小\ICON\顏色\透明度\置頂
23 # 設定視窗大小
24 shopee.geometry("1400x1200") # 寬*高 #+決定位置
25 shopee.minsize(width= 400,height = 200)
26 shopee.maxsize(width= 1400,height = 1200)
27 # 固定的話叫做resizable
28 # icon
29 shopee.iconbitmap("management.ico")
30 ## 背景顏色
31 shopee.config(background="#FF8800")
32 ## 透明度
33 shopee.attributes("-alpha",1) #1~0之間
34 # 置頂
35 shopee.attributes("-topmost",1)
36
```

```
38 v1 = StringVar()
39 en = Entry(textvariable=v1)
40
41 show = Text(bg="#FFEFD5", fg='black', font=("微軟正黑體", 16), width=90, height=60 ,relief='flat',wrap="word")
42
43 lb = Label(bg="#FF8800",text="便宜蝦皮哪裡買",fg="white",
44     font=('Arial Rounded MT', 36))
45 lb.pack()
46 lb2 = Label(bg="#FF8800",text="請輸入要查找的商品",fg="white")
47 lb2.pack()
```

```
49 # entry
50 en.pack()
51
52 lb3 = Label(text=" ",background="#FF8800")
53 lb3.pack()
54
55 btn = Button(text="GO!",fg="#FF5511",command = printtxt)
56 btn.config(width =5,height=3) #是以網格設計
57 btn.pack()
58
59 lb4 = Label(text=" ",background="#FF8800")
60 lb4.pack()
```

```
62 def restart():
63     v1.set("")
64     show.delete(1.0, "end")
65
66 btn1 = Button(text="重新搜尋其他商品", fg="#FF5511", command=restart)
67 btn1.config(width =15,height=2)
68 btn1.pack()
69 lb4 = Label(text=" ",background="#FF8800")
70 lb4.pack()
```

```
73 scroll = Scrollbar(command=show.yview)
74 show.pack()
75
76 shopee.mainloop()
77
```

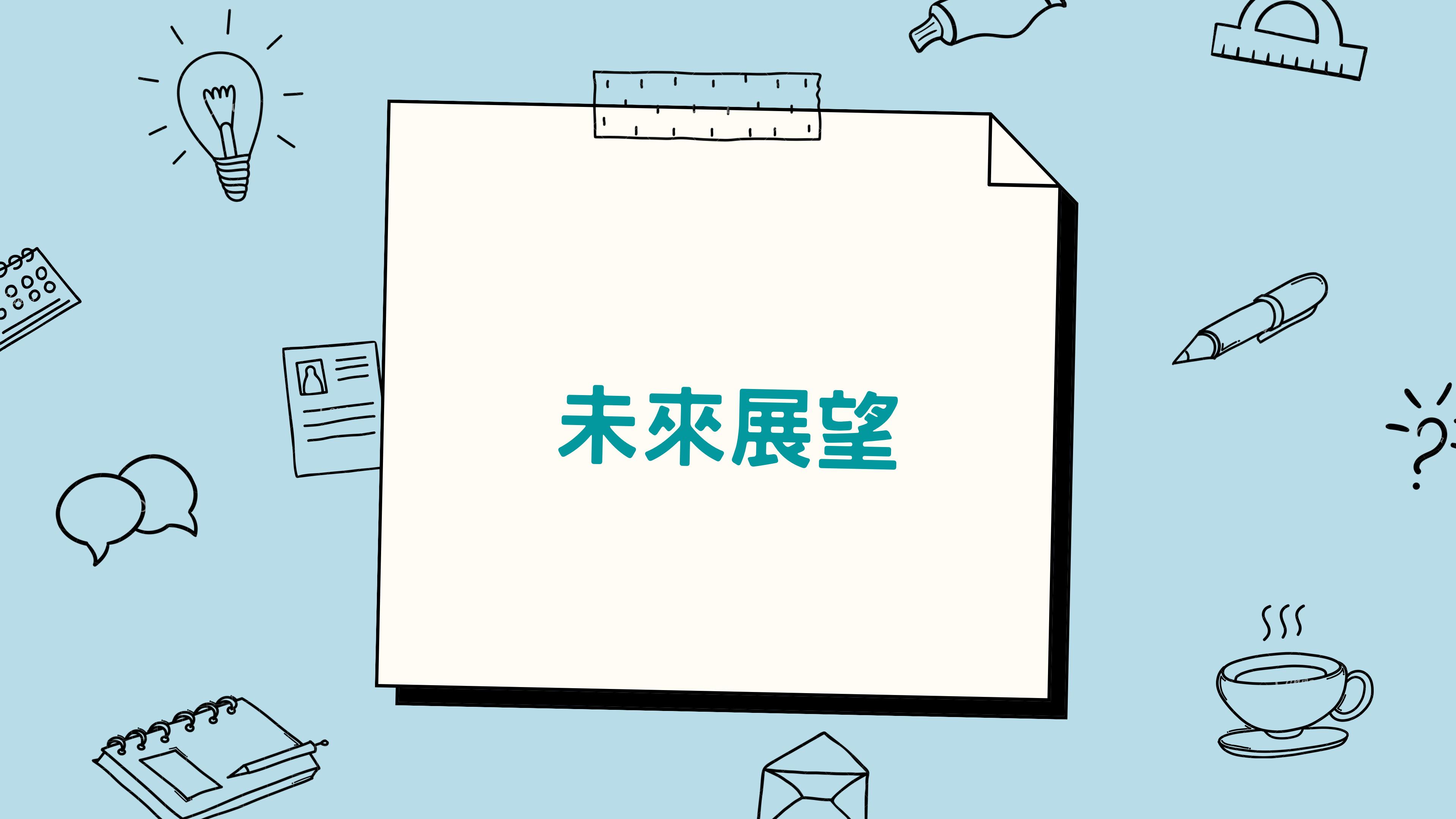
成品展示

蝦皮比價程式

比價程式優點

- 即時回傳使用者搜尋資料
 - 因應每日價格變動，完成使用者即時比價需求
- 價格加總直觀排序，提升比價效率
- 排除無庫存資訊，節省使用者比價時間
- 顯示其他資訊
 - 顯示庫存訊息，掌握商品存貨資訊
 - 提供所有運輸資訊，方便消費者選擇
 - 其他優惠資訊（例如：多件折扣），方便使用者參考
 - 直觀顯示所有商品內容物（包含加價購商品），避免價格誤導

未来展望



避免回傳時間過長以及技術問題，犧牲了一些蝦皮網站現有的功能
未來可以針對使用者需求

- 讓消費者選擇運輸方式
- 顯示商品圖片、出貨地、評價資料
- 增加其他種類折價券（如：蝦幣折價券、蝦皮商城優惠券）
- 優化GUI介面

分工方式

1. 爬蟲程式（主要由呂昕玲、黃郁棻負責）
2. GUI程式（主要由陳宛瑩、黃式傑負責）
3. 簡報製作（主要由陳宛瑩、黃式傑負責）
4. 影片製作（共同負責）

Thank You



Do you have any questions of us?