

The background of the image is a wide-angle photograph of a majestic mountain range. The mountains are rugged with dark, rocky slopes. In the distance, several peaks are partially obscured by a thick layer of white mist or low-hanging clouds, creating a sense of depth and scale. The sky above is a clear, pale blue.

登山訓練一把罩

王梓傑 李鈞冠 蔡孟羲 陳玖妃

PART 1

動機

鑑於登山新手不瞭解.....

■ 山岳難度資訊(如地形、耗時、坡度等統整資訊) ----以健行筆記(合歡東峰)為例

基本資料	交通方式	路線地圖	人文生態	相關GPX
所在縣市	南投縣仁愛鄉,花蓮縣秀林鄉	里程	單程1.1公里	
步道類型	高山步道	步道型態	必須折返	
海拔高度	3100~3421公尺	高度落差	321公尺	
路面狀況	石階、山徑、木棧道	所需時間	2小時 30分鐘(往返)	
難易度	低-中	所屬園區	太魯閣國家公園	
山系	合歡山系	最適季節	春、夏、秋	
申請入山	否	申請入園	否	
百岳 / 小百岳	合歡東峰			
附近步道	合歡尖山步道、合歡石門山步道、合歡主峰步道、合歡北峰步道			
相關山岳	合歡主峰、石門山、合歡西峰、合歡北峰、合歡東峰			
基點	無			

全部縣市 合歡東峰步道 搜尋 上傳 GPX

GPX列表 46 個GPX iOS 操作說明 Android 操作說明

合歡東峰 滑雪山莊上 松雪樓下
2.66 公里 2 小時 35 分鐘 388 公尺 378 公尺 2020-05-15
BOY 0 人收藏 140 次點閱 25 次下載 收藏 下載 掃描

2020-05-14 南投仁愛鄉滑雪山莊上合歡東峰
2.00 公里 1 小時 40 分鐘 377 公尺 333 公尺 2020-05-14
Chih Chun Wang 0 人收藏 132 次點閱 6 次下載 收藏 下載 掃描

合歡東峰原路來回
3.12 公里 3 小時 24 分鐘 681 公尺 692 公尺 2020-05-03
Ai-Han Chiang 0 人收藏 199 次點閱 20 次下載 收藏 下載 掃描

動機

鑑於登山新手不瞭解……

■ 要如何進行登山訓練？



概念

PART 2

1

整理熱門路線



2

依照該路線的各資訊項目(如高度、坡度等),
給予該項目在百岳中的相對難度



3

給出對應的訓練菜單

PART 3

過程

轉化資料成訓練菜單

將得到的資料劃分成不同難度，依據該難度給予訓練菜單

爬蟲程式

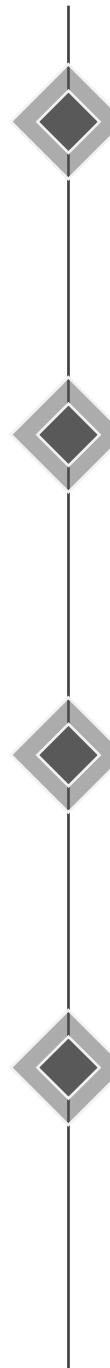
以健行筆記網站為準，
抓取百岳路線、登山路
線的熱門GPX(TOP3)

分析GPX資料

得出比健行筆記更多的
資訊，如坡度、實際路
程與連續陡升陡降

使用者介面

將上述資訊變成可與使
用者互動的介面



PART 3

過程

爬蟲程式

以健行筆記網站為準，
抓取百岳路線、登山路
線的熱門GPX(TOP3)

轉化資料成訓練菜單

將得到的資料劃分成不
同難度，依據該難度給
予訓練菜單

分析GPX資料

得出比健行筆記更多的
資訊，如坡度、實際路
程與連續陡升陡降

使用者介面

將上述資訊變成可與使
用者互動的介面

PART 4

撰寫 程式

Import

Module

```
from tkinter import *
from tkinter.ttk import Separator
import os
import requests
from bs4 import BeautifulSoup
import time
from svglib.svglib import svg2rlg
from reportlab.graphics import renderPM
from PIL import Image, ImageTk
import math
import gpxpy.parser as parser
import datetime
import matplotlib.pyplot as plt
```



PART 4

撰寫
程式

Web Crawler

類型 全部 百岳 小百岳 其他山岳 百大必訪步道

路線列表 63 筆路線資料

輸入路線名稱 搜尋 我要提供路線



合歡東峰步道

南投縣仁愛鄉,花蓮縣秀林鄉 **4.5** ★★★★★ 214人評分

難度 低-中 2.2公里 所需時間 2 小時 30 分鐘

想去 去過

1,921人想去 5,156人去過 162,370次點閱



合歡北峰步道

花蓮縣秀林鄉,南投縣仁愛鄉 **4.3** ★★★★★ 301人評分

難度 低-中 4公里 所需時間 3 小時

想去 去過

2,454人想去 5,144人去過 294,087次點閱



合歡石門山步道

花蓮縣秀林鄉,南投縣仁愛鄉 **4.5** ★★★★★ 246人評分

難度 低 1.568公里 所需時間 1 小時

想去 去過

1,793人想去 5,060人去過 112,951次點閱

PART 4

撰寫 程式

Web Crawler

輸入

用爬蟲將所有百岳路線及該路線所包含的山名預先存成dict，使用者輸入山名後，回傳該山包含在哪些登山路線，再讓使用者輸入想去哪個路線

```
import time

ur = 'https://hiking.biji.co/index.php?q=trail&part=%E5%85%A8%E9%83%A8&city=%E5%85%A8%E9%83%A8&zip='
    '%E5%85%A8%E9%83%A8&level=%E5%85%A8%E9%83%A8&time=%E5%85%A8%E9%83%A8&type=%E7%99%BE%E5%B2%B3&keyword=&page='
lst = ['1', '2', '3', '4']
|
urlhead = 'https://hiking.biji.co'
route_mtdict = dict() #裝步道資訊的字典[步道名:[網址, 包含的山A, B...]]
```

```
for i in lst:
    time.sleep(0.5)
    url = (ur+i)
    re = requests.get(url)
    soup = BeautifulSoup(re.text, 'html.parser')
    count = 0
    for route in soup.findAll('h2', {'class':'list-title'}):
        urltmp = urlhead + route.a['href']
        retmp = requests.get(urltmp)
        souptmp = BeautifulSoup(retmp.text, 'html.parser')
        for mtname in souptmp.select('div.flex.details-row'):
            if '百岳 / 小百岳' in mtname.text:
                cell = mtname.text.split()
                routeandcell = [route.a['title']]
                routeandcell.extend(cell[3:])
        route_mtdict[route.a['title']] = [urltmp, routeandcell]
        count += 1
        if count % 20 == 0:
            break
#rint(route_mtdict)
```

PART 4

撰寫 程式

Web Crawler

路線列表 63 筆路線資料

輸入路線名稱

合歡東峰步道

132 x 29.6 214人評分
Color #01200F
Font 22px "Helvetica Neue", Arial, "Heiti TC", "...
Contrast Aa 17.29 ✓

合歡北峰步道

花蓮縣秀林鄉,南投縣仁愛鄉 4.3 ★★★★★ 301人評分
難度 低-中 4公里 所需時間 3 小時
2,453人想去 5,143人去過 294,001次點閱

Elements Console Sources Network Performance Memory Application Security Lighthouse Adblock Plus

```
> a class="postMeta-img" href="/index.php?q=trail&act=detail&id=287" title="合歡東峰步道"></a>
<div class="postMeta-feedSummary">
  <h2 class="list-title">
    <a href="/index.php?q=trail&act=detail&id=287" title="合歡東峰步道">合歡東峰步道</a>
  </h2>
  <div class="location">南投縣仁愛鄉,花蓮縣秀林鄉</div>
</div>
```

路線名稱: 合歡東峰步道 所需時間: 2 小時 30 分鐘(往返)

難易度	低-中	所屬園區	太魯閣國家公園
div.flex-none.cell-item.title	110 x 81.2 Color #6F788A Font 16px "Helvetica Neue", Arial, "Heiti TC", "... Background #F7F7F7 Padding 15px	最適季節	春、夏、秋
百岳 / 小百岳	合歡東峰	申請入園	否
附近步道	合歡尖山步道、合歡石門山步道、合歡主峰步道、合歡北峰步道		
相關山丘	合歡東峰、石門山、合歡西峰、合歡北峰、合歡東峰		

Elements Console Sources Network Performance Memory Application Security Lighthouse Adblock Plus

```
> <div class="flex details-row"></div>
> <div class="flex none cell-item title">
  <div>百岳 / <br>小百岳</div>
  <div class="flex-1 cell-item">
    <ul class="flex flex-wrap divide-item"> -- 50
      <li>
        <a href="/index.php?q=mountain&act=detail&id=35" target="_blank">合歡東峰</a>
      </li>
    </ul>
  </div>
</div>
```

PART 4

撰寫 程式

Web Crawler

簡化

```
#找路線
route_mtddict = {'合歡東峰步道': ['https://hiking.biji.co/index.php?q=trail&act=detail&id=287', '合歡東峰步道'],
urlhead = 'https://hiking.biji.co'
def mt_rout():
    userinput = ans_mt.get()
    mtinroute = list() #回傳使用者輸入的山岳被包含在哪幾條步道
    for key,value in route_mtddict.items():
        if userinput in value[1]:
            mtinroute.append(key)
    mt.config(text = mtinroute)
    if len(mtinroute) == 0:
        mt.config(text = 'Oops，找不到你要的山唷，請重新輸入', fg = 'white', bg = '#323232', font = '微軟正黑體')
```

→輸入完接著就可以爬該路線的GPX囉！

PART 4

撰寫
程式

Web Crawler

爬GPX

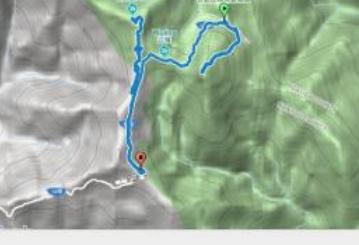
先爬出所有gpxcell

全部縣市 ▼ 合歡東峰步道

GPX列表 46 個GPX iOS 操作說明 Android 操作說明

**合歡東峰 滑雪山莊上 松雪樓下**
2.66 公里 ⏰ 2 小時 35 分鐘 ↗ 388 公尺 ↘ 378 公尺 ⏵ 2020-05-15
 BOY ❤ 0 人收藏 ⏷ 140 次點閱 ⏵ 25 次下載 收藏 下載 掃描

**2020-05-14 南投仁愛鄉滑雪山莊上合歡東峰**
2.00 公里 ⏰ 1 小時 40 分鐘 ↗ 377 公尺 ↘ 333 公尺 ⏵ 2020-05-14
 Chih Chun Wang ❤ 0 人收藏 ⏷ 132 次點閱 ⏵ 6 次下載 收藏 下載 掃描

**合歡主東峰單攻**
13.79 公里 ⏰ 7 小時 7 分鐘 ↗ 1,579 公尺 ↘ 1,613 公尺 ⏵ 2019-06-07
 Ethan Huang ❤ 1 人收藏 ⏷ 599 次點閱 ⏵ 89 次下載 收藏 下載 掃描

1 2 3 >

PART 4

撰寫 程式

Web Crawler

爬GPX

先爬出所有gpxcell

```
185     page = 1
186     gpxinf = list() #裝GPX資訊的list[[GPX名,網址,下載次數],...]
187     urlroute = value[0]
188     reroute = requests.get(urlroute)
189     souproute = BeautifulSoup(reroute.text, 'html.parser')
190     try:
191         for moregpx in souproute.select('div.more-gpx'):
192             urlmoregpx = urlhead + moregpx.a['href']
193             regpx = requests.get(urlmoregpx)
194             soupgrpx = BeautifulSoup(regpx.text, 'html.parser')
195             for totalgpxcell in soupgrpx.select('div.flex.items-baseline.gpx-title span'):
196                 cellist = totalgpxcell.text.split()
197                 totalgpxnum = int(cellist[0])
198                 #print(totalgpxnum)
199                 countgpx = 1
200
201             for gpxcell in soupgrpx.select('div.postMeta-feedSummery'):
202                 if page-1 == (totalgpxnum//20) and countgpx == totalgpxnum :
203                     findgpxinf(gpxcell)
204                     break
205                 elif countgpx % 20 == 1 and countgpx > 1:
206                     break
207                 findgpxinf(gpxcell)
208                 countgpx += 1
209
210             if not page-1 == (totalgpxnum//20) and countgpx == totalgpxnum :
211                 while True :
212                     page += 1
213                     urlnextpage = urlmoregpx + '&page=' + str(page)
214                     renextpage = requests.get(urlnextpage)
215                     soupnextpage = BeautifulSoup(renextpage.text, 'html.parser')
216
217                     for gpxcell in soupnextpage.select('div.postMeta-feedSummery'):
218                         if page-1 == (totalgpxnum//20) and countgpx == totalgpxnum :
219                             break
220                         elif countgpx % 20 == 1 and countgpx-1 == page*20:
221                             break
222                         findgpxinf(gpxcell)
223                         countgpx += 1
224                     if page-1 == (totalgpxnum//20) and countgpx == totalgpxnum :
225                         findgpxinf(gpxcell)
226                         break
227                         elif countgpx % 20 == 1 and countgpx-1 == page*20:
228                             continue
```

看更多GPX

不同頁數的GPX

PART 4

撰寫 程式

Web Crawler

爬GPX

依據gpxcell中的GPX資訊建立[GPX名,網址,下載次數]的lst

```
165     def findgpxinf(gpxcell):
166         li = 0
167         for download in gpxcell.select('ul.row-stack-list.info-list li'):
168             li +=1
169             if li == 4:
170                 gpxnameraw = gpxcell.h3.text
171                 gpxname = gpxnameraw.replace('\n', '')
172                 gpxname = gpxname.replace('/', '_') #修改gpx名變成跟下載的檔案一樣
173                 a = 0
174                 for gpxdownload in gpxcell.select('div.func-list a'):
175                     a += 1
176                     if a == 2:
177                         gpxurl = urlhead + gpxdownload['href']
178                         break
179                 downloadcell = download.text.split()
180                 #0603 將從數字拿掉 0611改用replace寫
181                 downloadnum = int(downloadcell[0].replace(',', '')) 
182                 gpxinf.append([gpxname, gpxurl, downloadnum])
183                 break
```

取得所有的GPX後，再做下載次數排序，並將下載次數前三名的GPX存到to_download_lst

```
250     gpxinfsorted = sorted(gpxinf, reverse = True, key = lambda d:d[2])
251     #print(gpxinfsorted)
252     #0611 把to_download_lst換到這
253     for g in gpxinfsorted[0:3]:
254         to_download_lst.append(g)
255     #print(to_download_lst)
```

→爬完接著就可以下載to_download_lst的GPX囉！

爬GPX

避免冷門路線沒有'更多gpx'按鈕的路線bug

```
230     except:
231         for nomoregpx in souproute.select('h3.list-title'):
232             try:
233                 urlnomoregpx = urlhead + nomoregpx.a['href']
234                 renomoregpx = requests.get(urlnomoregpx)
235                 soupnomoregpx = BeautifulSoup(renomoregpx.text, 'html.parser')
236                 for gpxpage in soupnomoregpx.select('div.primary-info'):
237                     gpxname = gpxpage.h1.text
238                     gpxname = gpxname.replace('/', '_')
239                     gpxurl = urlnomoregpx.replace('gpx_detail', 'download_gpx')
240                     span = 0
241                     for gpxdownload in gpxpage.select('div.flex.items-center.metrics-list span'):
242                         span += 1
243                         if span == 4:
244                             downloadcell = gpxdownload.text.split()
245                             downloadnum = int(downloadcell[0].replace(',', ''))
```

gpxinf.append([gpxname, gpxurl, downloadnum])

```
246             except:
247                 break
```

PART 4

撰寫
程式

Web Crawler

下載GPX

健行筆記會要求要登入才能下載，所以要有headers的資訊

The screenshot shows a browser window with multiple tabs open. The active tab is for 'hiking.biji.co/index.php?q=member&member=1166002#_'. The developer tools are open, specifically the Network tab. A large red box highlights the 'Request Headers' section, which lists the following headers:

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-TW,zh;q=0.9,en-US;q=0.8,en;q=0.7
Cache-Control: max-age=0
Connection: keep-alive
Cookie: _ga=GA1.2.793198669.1589982449; __auc=4219b9b71723256d6edb54ab191; __fbp=fb.1.1589982452087.605399371; Avivid_canv=0; Avivid_uuid=bf489202-2afb-4dc7-9129540180480%2fpicture%3Ftype%3Dlarge; EID=MV1LczHYAtK2P1RttIwMIQ%3D%3D; __gat=1; __lastv=1166002; __erEvtvId=1166002; page_view=6
Host: hiking.biji.co
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-site
Sec-Fetch-User: ?
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.138 Safari/537.36
```

Below the headers, another red box highlights the 'User-Agent' header value.

PART 4

撰寫 程式 分析 GPX

讀取GPX中的資訊

用gpx parser讀取GPX，得到所有記錄點的經緯度、高度及時間並存成lst，但有些含多個路段的GPX其時間紀錄並非按照順序排列，因此再用時間進行排序

```
306 for p in to_download_lst:#裝GPX資訊的list[[GPX名,網址,下載次數],...]
307     try :
308         # 開啟gpx檔
309         gpx_file = open(os.path.join(file_path,userinput,p[0]+'.gpx'), encoding="utf-8")
310         gpx_parser = parser.GPXParser( gpx_file ) # 抓取gpx檔中的trk, trkseg與trkpt標籤
311         gpx = gpx_parser.parse() # 讀取trk, trkseg與trkpt標籤
312
313         #trk = gpx.tracks # gpx檔trk中的內容，資料形式為list
314         #trkseg = trk[0].segments # 抓取 trk 中的 trkseg 標籤內容，資料形式為list
315         #trkpt = trkseg[0].points # 抓取 trkseg中的 trkpt 內容，資料形式為list
316         # trk[0], trkseg[0], trkpt[0] 的資料形式分別為(gpypy.gpx.)GPXTrack, GPXTrackSegment, 與GPXTr
317
318         latitude_x = []      #未整理座標緯度
319         longitude_x = []    #未整理座標經度
320         elevation_x = []    #未整理座標高度
321         time_x = []         #未整理座標時間
322
323         for trk in gpx.tracks:
324             for trkseg in trk.segments:
325                 for trkpt in trkseg.points :
326                     latitude_x.append(trkpt.latitude)
327                     longitude_x.append(trkpt.longitude)
328                     elevation_x.append(trkpt.elevation)
329                     time_x.append(trkpt.time)
330
331         latitude = []        # 排序後的座標緯度
332         longitude = []       # 排序後的座標經度
333         elevation = []       # 排序後的座標高度
334         time = []            # 排序後的座標時間
335
336         no_time_mark = 0    # 如果GPX沒有時間，則後面資料不要顯示時間
337
338         if None not in time_x:
339             no_time_mark += 1
340             tuples_lst = list(zip(latitude_x,longitude_x,elevation_x,time_x))
341             tuples = sorted(tuples_lst, key = lambda lst:lst[3])
342             for i in tuples:
343                 latitude.append(i[0])
344                 longitude.append(i[1])
345                 elevation.append(i[2])
346                 time.append(i[3])
347             else:
348                 latitude = latitude_x
349                 longitude = longitude_x
350                 elevation = elevation_x
351                 time = time_x
352
```

```
<trk>
<name>2020-05-15_09-20-05</name>
<trkseg>
<trkpt lat="24.141539" lon="121.28514">
<ele>3119.0</ele>
<time>2020-05-15T01:20:08Z</time>
</trkpt>
<trkpt lat="24.141468" lon="121.285164">
<ele>3121.0</ele>
<time>2020-05-15T01:20:17Z</time>
</trkpt>
<trkpt lat="24.14142" lon="121.28535">
<ele>3118.0</ele>
<time>2020-05-15T01:20:27Z</time>
</trkpt>
```

撰寫 程式

分析 GPX

分析路程資訊：平面距離、斜率、跟實際路程、垂直上下高度等

透過經緯度計算兩個點間的平面距離

```
289     def distance(x1, y1, x2, y2):  
290         R = 6373.0  
291         lat1 = math.radians(x1)  
292         lon1 = math.radians(y1)  
293         lat2 = math.radians(x2)  
294         lon2 = math.radians(y2)  
295         dlon = lon2 - lon1  
296         dlat = lat2 - lat1  
297         a = math.sin(dlat / 2)**2 + math.cos(lat1) * math.cos(lat2) * math.sin(dlon / 2)**2 #Haversine  
298         c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))  
299  
300     return (R * c)  
301  
302
```

得到平面距離後再透過兩個點高度差來計算出的斜率

```
356     dist = [] #點與點之間的距離差  
357     elev = [] #點與點之間的高度差  
358     slope = [] #點與點之間的斜率  
359  
360     for i in range(len(latitude)-1): # 算斜率  
361         x = distance(latitude[i], longitude[i], latitude[i+1], longitude[i+1])*1000  
362         y = elevation[i] - elevation[i+1]  
363         dist.append(x)  
364         elev.append(y)  
365         if x != 0:  
366             slope.append(y/x)  
367
```

分析路程資訊：平面距離、斜率、跟實際路程、垂直上下高度等

透過高度、斜率計算兩個點間的斜距(實際路程)，並算出垂直上下高度總和

```
372     slant_dist = [] #點與點之間的斜距，單位公尺
373
374     for i in range(len(dist)): # 總實際距離
375         slant_dist.append(math.sqrt(dist[i]**2 + elev[i]**2))
376
377     all_dist = round(sum(dist)/1000) # 總平面距離
378
379     uphill, downhill = 0, 0 # 垂直上升 / 下降高度
380
381     for i in elev:
382         if i > 0:
383             uphill += i
384         else:
385             downhill += i
386
```

撰寫 程式

分析 GPX

分析路程資訊：平面距離、斜率、跟實際路程、垂直上下高度等

計算坡度，並計算各種坡度比率

```

391     checkpoint = 0 # 記錄點 1
392     nextpoint = 1 # 記錄點 2
393     dis_slope = [] # 記錄每50公尺的斜率 (坡度)
394     up_count_lst = [] # >50公尺的連續陡上坡
395     down_count_lst = [] # >50公尺的連續陡下坡
396     up_count = 0
397     down_count = 0
398
399     while nextpoint < len(latitude): # 計算 dis_slope
400         if distance(latitude[checkpoint], longitude[checkpoint], latitude[nextpoint], longitude[nextpoint]) > 50:
401             nextpoint += 1
402         else:
403             x1, y1 = latitude[checkpoint], longitude[checkpoint] # 起始點經緯度
404             x2, y2 = latitude[nextpoint], longitude[nextpoint] # 相隔 50公尺 (以上) 的經緯度
405             dis_slope.append((elevation[nextpoint]-elevation[checkpoint])/(distance(x1, y1, x2, y2)))
406             checkpoint = nextpoint
407             if dis_slope[-1] > 0.37: # 如果角度大於20.3度
408                 up_count += 50 # 就把距離 x 算進連續陡上
409             elif dis_slope[-1] < 0.37 and up_count != 0:
410                 up_count_lst.append(up_count)
411                 up_count = 0
412             if dis_slope[-1] < -0.46: # 如果角度大於 24.7度
413                 down_count += 50
414             elif dis_slope[-1] > -0.46 and down_count != 0:
415                 down_count_lst.append(down_count)
416                 down_count = 0
417
418             if up_count_lst == []:
419                 up_count_lst.append(0)
420             if down_count_lst == []:
421                 down_count_lst.append(0)
422
423             degree = [] # 記錄距離間隔斜率 (坡度) 換算的角度
424
425             for i in dis_slope:
426                 angle = math.degrees(math.atan(i))
427                 degree.append(angle)
428
429             _0to10, _10to15, _15up = 0, 0, 0 # 計算坡角度比率
430
431             for i in degree:
432                 if 0 < abs(i) <= 10:
433                     _0to10 += 1
434                 if 10 < abs(i) <= 15:
435                     _10to15 += 1
436                 if abs(i) > 15:
437                     _15up += 1

```

PART 4

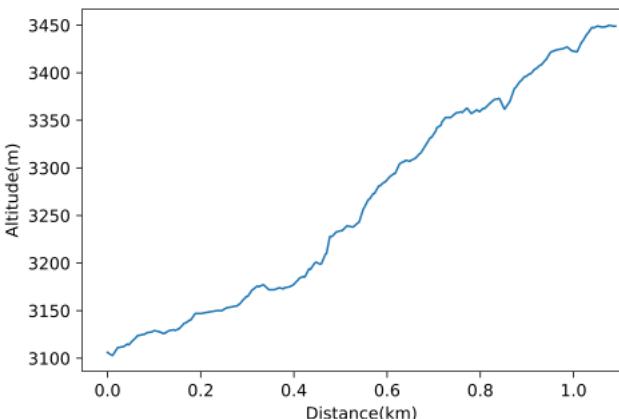
撰寫 程式

分析GPX

畫出水平距離高度圖

用matplotlib圖像化水平距離及高度資料，並存成png

```
442             dist_ = [0] #距離list
443             s = 0
444             for i in dist:
445                 s += i/1000 #把距離m換算成km
446                 dist_.append(s)
447
448             elev_ = [] #高度list
449             for i in elevation:
450                 elev_.append(i)
451
452             #畫高度距離表
453             plt.plot(dist_,elev_)
454             plt.ylabel('Altitude(m)')
455             plt.xlabel('Distance(km)')
456             plt.savefig(p[0]+'.svg')
457             plt.show()
458
459             drawing = svg2rlg(p[0]+'.svg')
460             renderPM.drawToFile(drawing, p[0]+'.png', fmt='PNG')
```



PART 4

撰寫 程式 訓練菜單

給予百岳分級

比較63條路線前三名的GPX的屬性(坡度比例、高度等等), 紿予難度分級, 分成Ultra、Hard、Medium、Easy、Rookie

```
484 difficulty = ['Ultra', 'Hard', 'Medium', 'Easy', 'Rookie']
485 climb_time = [datetime.timedelta(days=3, seconds=33536, microseconds=222000), datetime.timedelta(days=1,
486 highest = [3745, 3467, 3356, 3105, 0]
487 ave_hight = [3159, 2844, 2687, 2359, 0]
488 flat_rt = [58, 33, 18, 8, 0]
489 real_rt = [65, 38, 20, 8, 0]
490 ten_po = [63, 54, 48, 35, 0]
491 between_po = [18, 15, 13, 9, 0]
492 fifteen_po = [36, 29, 24, 10, 0]
493 go_up = [6371, 3606, 2199, 830, 0]
494 go_down = [6783, 3706, 2229, 830, 0]
495 steep_up_all = [2800, 1900, 1100, 250, 0]
496 steep_up = [450, 300, 200, 100, 0]
497 steep_down_all = [350, 250, 150, 50, 0]
498 steep_down = [1750, 1150, 650, 50, 0]
```

依據不同的資訊屬性, 將其難度分到不同的訓練組裡面, 其中耐力組包含高度、攀爬天數與實際路程、腿力組包括垂直升降距離、陡坡比例等等

```
500 endurance = [] # 耐力類
501 legs_train = [] # 腿部訓練
502
503 if no_time_mark != 0:
504     for i in range(len(difficulty)):
505         if time[-1]-time[0] >= climb_time[i]:
506             info1 = Label(z, text = '攀爬時間'+str(time[-1]-time[0])+' *難度: '+difficulty[i], fg = 'white',
507                         bg = '#32CD32', font = '標楷體 11')
508             info1.pack()
509             endurance.append(difficulty[i])
510             break
511         else:
512             info1 = Label(z, text = '該路線GPX無時間資料', fg = 'white', bg = '#32CD32', font = '標楷體 11')
513             info1.pack()
514
515         for i in range(len(difficulty)):
```

撰寫 程式 訓練 菜單

PART 4

設計訓練菜單

將各屬性的難度分別加入到耐力組與腿力組的菜單後，依照其難度，給予該訓練組對應內容的訓練強度

```
591 running_po = [7, 5, 3, 1, 0] 跑步
592 cross_jump = [90, 75, 60, 45, 30] 交互蹲跳
593 mt_climber = [90, 75, 60, 45, 30] 登山式
594 poppy = [11, 9, 7, 5, 3] 波比跳
595
596 info15 = Label(z, text = '建議訓練菜單（每週至少兩次）', fg = 'red', bg = 'yellow', font = '微軟正黑體 11')
597 info15.pack(pady = 9)
598
599 #跑步訓練
600 for i in difficulty:
601     if i == max(endurance, key = endurance.count):
602         info17 = Label(z, text = '跑 '+str(running_po[difficulty.index(i)])+' 度坡3000公尺')
603         info17.pack()
604
605 #登山訓練
606 for i in difficulty:
607     if i == max(endurance, key = endurance.count):
608         info18 = Label(z, text = '登山式 '+str(mt_climber[difficulty.index(i)])+' 秒 (四組)')
609         info18.pack()
610
611 #波比跳
612 for i in difficulty:
613     if i == max(endurance, key = endurance.count):
614         info19 = Label(z, text = '波比跳 '+str(poppy[difficulty.index(i)])+' 組 (12下 / 組)')
615         info19.pack()
616
617 #弓箭步交互蹲跳訓練
618 for i in difficulty:
619     if i == max(legs_train, key = legs_train.count):
620         info20 = Label(z, text = '弓箭步交互蹲跳 '+str(cross_jump[difficulty.index(i)])+' 秒')
621         info20.pack()
622
623 if legs_train[1] == 'Ultra' or legs_train[1] == 'Hard':
624     info21 = Label(z, text = '高腳杯深蹲 12 RM (四組)', fg = 'yellow', bg = '#323030', font = '微軟正黑體 11')
625     info21.pack() # 15度坡一定比例以上才有此訓練
626 else: # 要加不然排版跑掉
627     info21 = Label(z, text = '', fg = 'white', bg = '#323030', font = '微軟正黑體 11')
628     info21.pack()
629
630 except :
631     continue
```

耐力組訓練內容：跑步、登山式與波比跳

腿力訓練內容：弓箭步交互蹲跳與高腳杯深蹲

撰寫 程式

GUI Tkinter

PART 4

```
19 win = Tk()
20 win.title('登山訓練一把罩')
21 win.geometry('1920x1080')
22 win.config(bg = '#323232')
23
24 #scroll1 = Scrollbar(win)
25 #scroll1.pack(side = RIGHT, fill = Y)
26 #scroll2 = Scrollbar(fram, orient = HORIZONTAL)
27 #scroll2.pack(side = BOTTOM, fill = X)
28
29 # 建立標題「登山訓練一把罩」
30
31 title_text = Label(text = '登山訓練一把罩', fg = 'white', bg = '#323232', font = '華康粗圓體(P) 22')
32 title_text.pack(pady = 10, padx = 10)
33 sep = Separator(orient = HORIZONTAL)
34 sep.pack(fill = X, pady = 5)
35
36 =====
37
38 # 建立總體框架
39
40 fram = Frame(win, bg = '#323232')
41 fram.pack()
42
43 fram = Frame(win, bg = '#323232') # 資料顯示部分
44 fram.pack()
45
46 # 建立第三部分「今天想爬哪座山」
47
48 fram3 = Frame(fram, bg = '#323232')
49 fram3.pack()
50
51 ask_mt = Label(fram3, text = '今天想爬哪座山?', fg = 'white', bg = '#323232', font = '標楷體 14')
52 ask_mt.pack(side = LEFT, pady = 5, padx = 5)
53
54 =====
55
56 # 哪座百岳？
57
58 fram4 = Frame(fram3, bg = '#323232')
59 fram4.pack()
60
61 ans_mt = Entry(fram3)
62 ans_mt.pack(side = LEFT, pady = 9, padx = 5)
63
64 #找路線
65 route_mtdict = {'合歡東峰步道': ['https://hiking.biji.co/index.php?q=trail&act=detail&id=287', '合歡東峰步道']}
66 urlhead = 'https://hiking.biji.co'
67 def mt_route():
68     userinput = ans_mt.get()
69     mtinroute = list() #回傳使用者輸入的山岳被包含在哪幾條步道
70     for key,value in route_mtdict.items():
71         if userinput in value[1]:
72             mtinroute.append(key)
73     mt.config(text = mtinroute)
74     if len(mtinroute) == 0:
75         mt.config(text = 'Oops, 找不到你要的山嶺, 請重新輸入', fg = 'white', bg = '#323232', font = '微軟正黑體 10')
76
77 # 建立查詢路線按鈕
78 btn = Button(fram3, text = '找路線', fg = 'white', bg = '#323232', font = '微軟正黑體 10', command = mt_route)
79 btn.pack(side = LEFT, pady = 5, padx = 5)
80
81 # 顯示該百岳路線
82
83 fram_routes = Frame(fram, bg = '#323232')
84 fram_routes.pack()
85
86 mt = Label(fram_routes, text = '', fg = 'white', bg = '#323232', font = '微軟正黑體 11')
87 mt.pack(side = LEFT, pady = 9, padx = 5)
88
89 =====
90
91 # 哪條路線？
92
93 ask_rout = Label(fram5, text = '想要走哪條路線?', fg = 'white', bg = '#323232', font = '標楷體 14')
94 ask_rout.pack(side = LEFT, pady = 9, padx = 5)
95
96 =====
97
98 fram6 = Frame(fram, bg = '#323232')
99 fram6.pack()
100
101 rt = Label(fram6, text = '', fg = 'white', bg = '#323232', font = '微軟正黑體 11')
102 rt.pack(side = LEFT, pady = 9, padx = 5)
103
104 ans_rout = Entry(fram5)
105 ans_rout.pack(side = LEFT, pady = 9, padx = 5)
106 to_download_lst = []
```

建立Label,Entry,Button

PART 4

撰寫 程式

GUI Tkinter

將結果呈現給使用者

```
462 # 第二部分：呈現GPX的frame
463
464 z = Frame(farm, bg = '#323232')
465 z.pack(side = LEFT, pady = 15, padx = 40)
466
467 photo = Image.open(p[0] + '.png') # 開啟圖像位置
468 photo.thumbnail((240, 180)) # 依照比例重設大小，因為是照比例，所以他的長取決於高，主要看高多少，不要讓
469 photo.save(p[0] + 'new.png') # 後面輸入檔名，我不知道預設會存在哪，我的是存在我存py檔的地方
470 new = ImageTk.PhotoImage(file = p[0] + 'new.png') #開啟新存的圖像
471 z.new = new
472 lb = Label(z, image = z.new) #將新存圖像插入Label
473 lb.pack()
474
475 naaame = Label(z, text = p[0], fg = 'white', bg = '#323232', font = '微軟正黑體 11')
476 naaame.pack(pady = 5)
477
478
479 teeest = "、".join(route_mtdict[userinput][1][1:])
480
481 mtinside = Label(z, text = '這條路線有' + teeest, fg = 'white', bg = '#323232', font = '微軟正
482 mtinside.pack(pady = 5)
```

PART 5

成果 展示

The screenshot shows a search interface for mountain climbing routes. At the top left is a logo of a blue feather and the text "登山訓練一把罩". On the top right are window control buttons for minimize, maximize, and close. The main title "登山訓練一把罩" is centered at the top. Below it is a search bar with the placeholder "今天想爬哪座山？" containing the text "下油鍋上刀山". To the right of the search bar is a button labeled "找路線". Below this is an error message: "Oops, 找不到你要的山唷, 請重新輸入". Another search bar below it has the placeholder "想要走哪條路線？" containing the text "隔空打牛". To its right is a button labeled "找路線資訊". A final message below says "找不到這條路線唷, 請依上方顯示路線輸入".

登山訓練一把罩

今天想爬哪座山？ 找路線

Oops, 找不到你要的山唷, 請重新輸入

想要走哪條路線？ 找路線資訊

找不到這條路線唷, 請依上方顯示路線輸入

PART 5

成果展示

Ex: 南一段

登山訓練一把罩

登山訓練一把罩

南一段

這條路線有庫哈諾辛山、關山、海諾南山、小關山、卑南主山

攀爬時間5 days, 12:58:25 *難度: Ultra
最高點3655m *難度: Hard
平均高度2896m *難度: Hard
平面路程69km *難度: Ultra
實際路程93km *難度: Ultra
0~10度坡49% *難度: Easy
10~15度坡16% *難度: Ultra
15度以上坡31% *難度: Hard
總垂直上升20563公尺 *難度: Ultra
總垂直下降19814公尺 *難度: Ultra
陡升總距離3550m(陡升 : >20.3度) *難度: Ultra
最長陡升距離200m *難度: Medium
陡降總距離3000m(陡降 : >24.7度) *難度: Ultra
最長陡降距離150m *難度: Easy

建議訓練菜單 (每週至少兩次)

- 跑 7 度坡3000公尺
- 登山式 90 秒 (四組)
- 波比跳 11 組 (12下 / 組)
- 弓箭步交互蹲跳 90 秒 (四組)
- 高腳杯深蹲 12 RM (四組)

鐵本山 [上] 關山 (連走) 庫哈諾辛山

這條路線有庫哈諾辛山、關山、海諾南山、小關山、卑南主山

該路線GPX無時間資料
最高點3641m *難度: Hard
平均高度2781m *難度: Medium
平面路程29km *難度: Medium
實際路程30km *難度: Medium
0~10度坡39% *難度: Medium
10~15度坡18% *難度: Ultra
15度以上坡41% *難度: Ultra
總垂直上升3524公尺 *難度: Medium
總垂直下降3839公尺 *難度: Hard
陡升總距離1800m(陡升 : >20.3度) *難度: Medium
最長陡升距離400m *難度: Hard
陡降總距離1250m(陡降 : >24.7度) *難度: Ultra
最長陡降距離200m *難度: Easy

建議訓練菜單 (每週至少兩次)

- 跑 5 度坡3000公尺
- 登山式 75 秒 (四組)
- 波比跳 9 組 (12下 / 組)
- 弓箭步交互蹲跳 90 秒 (四組)
- 高腳杯深蹲 12 RM (四組)

南橫三星-庫哈諾辛山

這條路線有庫哈諾辛山、關山、海諾南山、小關山、卑南主山

攀爬時間10:57:22 *難度: Easy
最高點3105m *難度: Easy
平均高度2842m *難度: Medium
平面路程7km *難度: Rookie
實際路程8km *難度: Easy
0~10度坡19% *難度: Hard
10~15度坡21% *難度: Rookie
15度以上坡57% *難度: Ultra
總垂直上升982公尺 *難度: Easy
總垂直下降1077公尺 *難度: Easy
陡升總距離1250m(陡升 : >20.3度) *難度: Medium
最長陡升距離450m *難度: Ultra
陡降總距離500m(陡降 : >24.7度) *難度: Ultra
最長陡降距離300m *難度: Easy

建議訓練菜單 (每週至少兩次)

- 跑 1 度坡3000公尺
- 登山式 45 秒 (四組)
- 波比跳 5 組 (12下 / 組)
- 弓箭步交互蹲跳 90 秒 (四組)
- 高腳杯深蹲 12 RM (四組)

功能 檢討

1. 碴於時間關係， 還沒有做出Redo功能， 讓使用者輸入完路線資訊後， 需要再重開程式才能重新搜尋
2. 碴於版面配置， 還無法做出Scrollbar功能， 導致在部分螢幕較小的電腦， 會無法顯示完整的菜單資訊
3. 該.py檔程式應可打包為.exe檔， 讓使用者在無python作業環境下也可操作， 但目前執行上仍有bug
4. 程式運行中會下載3個GPX跟9張圖片， 可能會被判定為病毒軟體（誤）

未來 修正

1. 增訂讓使用者可以新增功能的內容
2. 百岳別名
3. 爬取健行筆記以外， 其他登山網站的資訊



THANKS