

寄信好麻煩



組員: 簡詩汶 游洧聖 張鈞量



問題

大量寄信給填google 表單者，需要一個一個人工操作，十分繁冗

目標

讓社團、團購主、活動組織可以很快速的寄客製化信件給每個填表單者。

方式

建造電腦應用程式，讓使用者匯入excel檔，編輯草稿、預覽及寄信。



使用步驟



1°

登入使用者Gmail帳號

2°

選擇csv / xlsx file

3°

編輯文字、主旨、收件人

4°

發送訊息，顯示成功寄送與否



使用工具



smtplib

Connect to gmail



Read csv / xlsx file
Extract columns



python

Webbrowser

Link to user guide web



GUI builder



pyinstaller

Pack python to exe



Design Logo





01

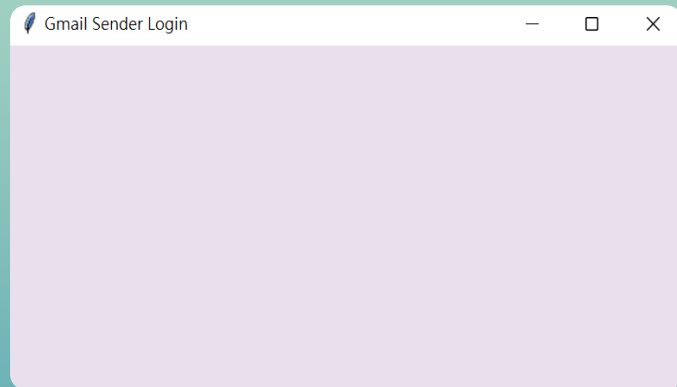
Login 視窗

連結寄件者的gmail發送權

1. Login 視窗

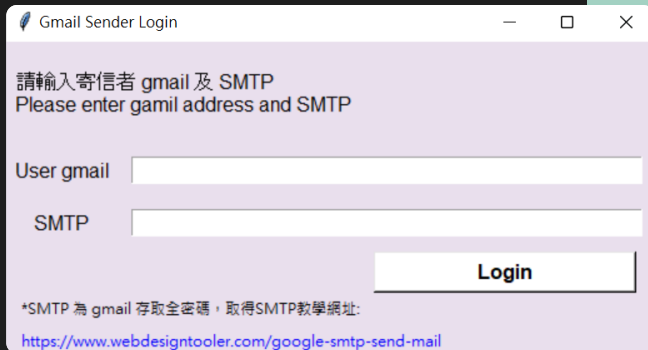
```
1 import tkinter as tk
```

```
32  ✓ if __name__ == "__main__":  
33      root = tk.Tk()  
34      root.title('Gmail Sender Login')  
35      root.configure(background='#e9dfed')  
36      root.geometry('+360+200')  
37      root.minsize(520, 250)  
38      root.maxsize(520, 250)
```



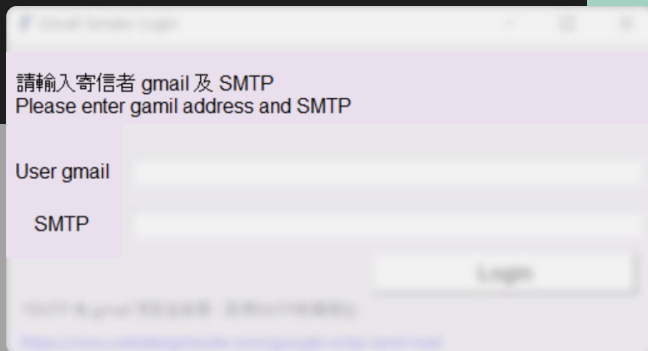
1. Login 視窗

```
40 # Label
41 label0 = tk.Label(text="請輸入寄信者 gmail 及 SMTP" \nPlease enter gamil address and SMTP", fg='black', bg="#e9dfed", font=("Helvetica", '12'))
42 label1 = tk.Label(text="User gmail", fg='black', bg="#e9dfed", font=("Helvetica", '12'))
43 label2 = tk.Label(text="SMTP", fg='black', bg="#e9dfed", font=("Helvetica", '12'))
44
45 label0.grid(column=0, row=0, columnspan=2, pady=20, padx=5, sticky='w')
46 label1.grid(column=0, row=1, padx=5)
47 label2.grid(column=0, row=2, padx=5)
48
49 # Entry
50 entry1 = tk.Entry(width=45, font=("Helvetica", '12'))
51 entry2 = tk.Entry(width=45, font=("Helvetica", '12'), show='*')
52
53 entry1.grid(column=1, row=1, padx=10, pady=10)
54 entry2.grid(column=1, row=2, padx=10, pady=10)
55 entry1.bind('<Return>', getLoginInput)
56 entry2.bind('<Return>', getLoginInput)
57
58 # Button
59 button1 = tk.Button(text='Login', width=20, bg="white", font=("Helvetica", '12', 'bold'), command=getLoginInput)
60 button1.grid(column=1, row=3, pady=2, padx=15, sticky='e')
61
62 # Lable, bg="#e9dfed"
63 label_smtp = tk.Label(text="*SMTP 為 gmail 存取全密碼，取得SMTP教學網址:", fg='black', bg="#e9dfed")
64 link1 = tk.Label(root, text="https://www.webdesigntooler.com/google-smtp-send-mail", fg="blue", bg="#e9dfed", cursor="hand2")
65 link1.bind("<Button-1>", lambda e: callback("https://www.webdesigntooler.com/google-smtp-send-mail"))
66
67 label_smtp.grid(column=0, row=4, columnspan=2, padx=10, sticky='w')
68 link1.grid(column=0, row=5, columnspan=2, padx=10, pady=5, sticky='w')
69
70 root.mainloop()
```



1. Login 視窗

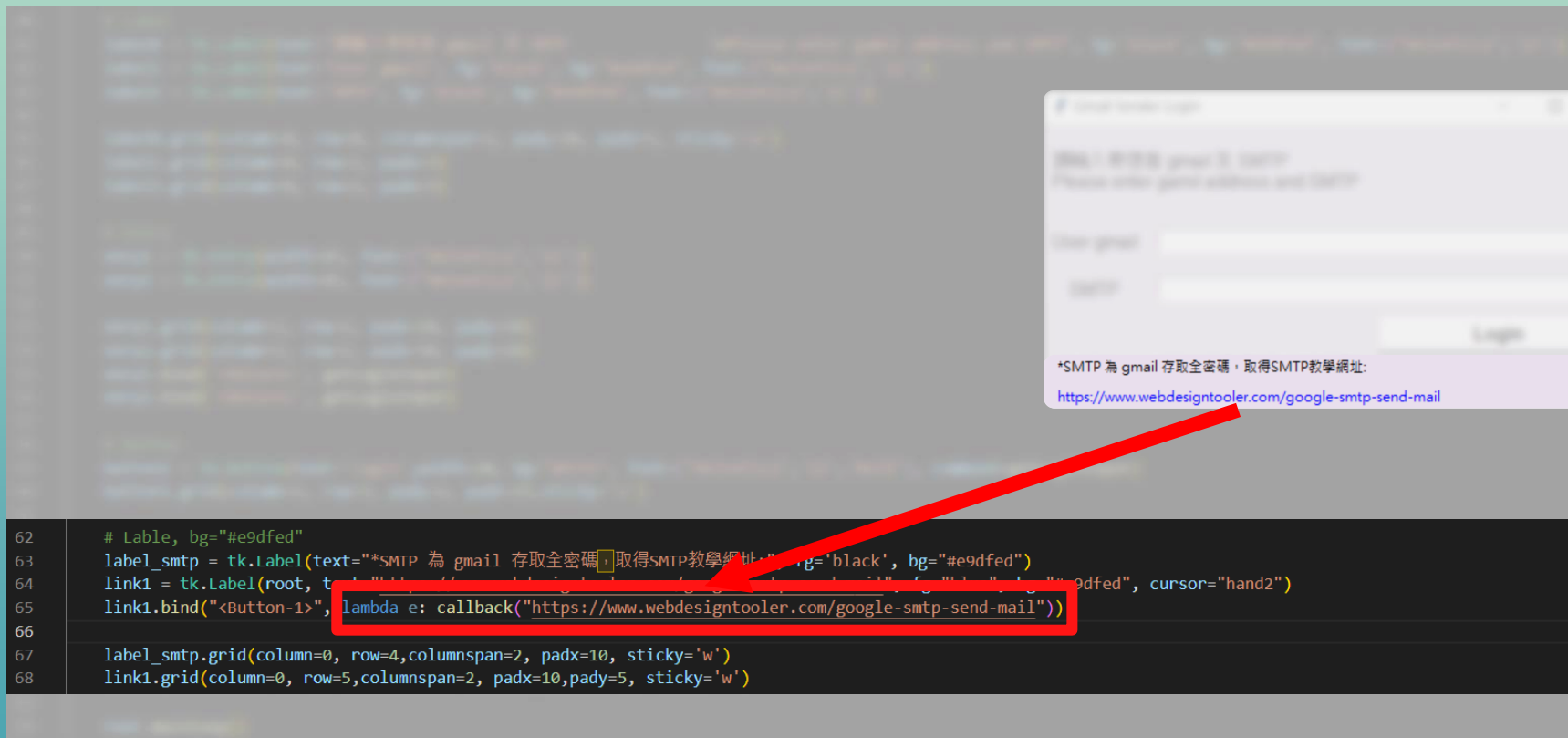
```
40 # Label
41 label0 = tk.Label(text="請輸入寄信者 gmail 及 SMTP", fg='black', bg="#e9dfed", font=("Helvetica", '12'))
42 label1 = tk.Label(text="User gmail", fg='black', bg="#e9dfed", font=("Helvetica", '12'))
43 label2 = tk.Label(text="SMTP", fg='black', bg="#e9dfed", font=("Helvetica", '12'))
44
45 label0.grid(column=0, row=0, columnspan=2, pady=20, padx=5, sticky='w')
46 label1.grid(column=0, row=1, padx=5)
47 label2.grid(column=0, row=2, padx=5)
```



1. Login 視窗

```
49 # Entry
50 entry1 = tk.Entry(width=45, font=("Helvetica", '12'))
51 entry2 = tk.Entry(width=45, font=("Helvetica", '12'), show='*')
52
53 entry1.grid(column=1, row=1, padx=10, pady=10)
54 entry2.grid(column=1, row=2, padx=10, pady=10)
55 entry1.bind('<Return>', getLoginInput)
56 entry2.bind('<Return>', getLoginInput)
```

1. Login 視窗



1. Login 視窗

```
2 import webbrowser
```

```
28 # 超連結
29 def callback(url):
30     webbrowser.open_new(url)
```

```
62 # Lable, bg="#e9dfed"
63 label_smtp = tk.Label(text="*SMTP 為 gmail 存取全密碼, 取得SMTP教學網址:", fg='black', bg="#e9dfed")
64 link1 = tk.Label(root, text="https://www.webdesigntooler.com/google-smtp-send-mail", fg="#e9dfed", cursor="hand2")
65 link1.bind("<Button-1>", lambda e: callback("https://www.webdesigntooler.com/google-smtp-send-mail"))
66
67 label_smtp.grid(column=0, row=4, columnspan=2, padx=10, sticky='w')
68 link1.grid(column=0, row=5, columnspan=2, padx=10, pady=5, sticky='w')
```

WEBDESIGN TOOLER

精選 圖片處理 網頁設計 網站架設 網頁設計軟體 免費軟體 ALL



【2021最新版】如何使用Google SMTP寄信(兩段式驗證+應用程式密碼) - 主機架站寄信教學

2021-05-25

相關文章

3個步驟加入Google Meet 會議、分享螢幕畫面、免安裝下載、免註冊的遠端桌面軟體

[網頁教學] 如何在網頁加入Vimeo影片(影音平台整合)

不會設計海報? 推薦一款免費海報設計線上工具 DesignCap

客制化網站架站費用高達30萬以上, 做客制網站到底值不值得? 4個為甚麼選擇客制化網站的原因。

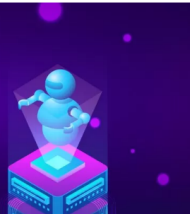
推薦一款清除瀏覽器快捷的好工具: Clear Cache + Google瀏覽器外掛

[網頁教學] 什麼是網頁網格(Grid System)?

如何解決網站縮放縮小的Cross-Frame Scripting問題?

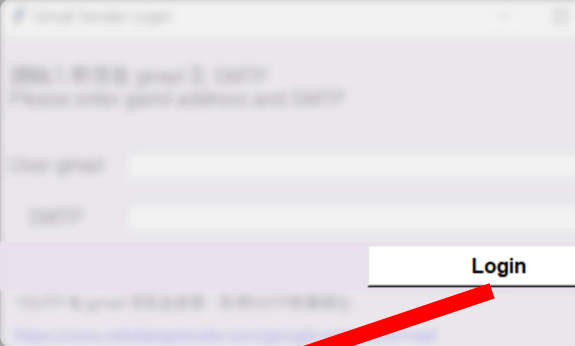
精選UE4的宣傳或是很難用嗎?

如何使用 Google SMTP寄信



1. Login 視窗

```
40 # Login window
41 login = tk.Toplevel(master=root, title='Login')
42 login.geometry('300x200')
43 login.title('Login')
44 login.configure(bg='white')
45 login.grid_columnconfigure(0, weight=1)
46 login.grid_rowconfigure(0, weight=1)
47 login.grid_rowconfigure(1, weight=1)
48 login.grid_rowconfigure(2, weight=1)
49 login.grid_rowconfigure(3, weight=1)
50 login.grid_rowconfigure(4, weight=1)
51 login.grid_rowconfigure(5, weight=1)
52 login.grid_rowconfigure(6, weight=1)
53 login.grid_rowconfigure(7, weight=1)
54 login.grid_rowconfigure(8, weight=1)
55 login.grid_rowconfigure(9, weight=1)
56 login.grid_rowconfigure(10, weight=1)
57 login.grid_rowconfigure(11, weight=1)
58 # Button
59 button1 = tk.Button(text='Login',width=20, bg="white", font=("Helvetica",'12','bold'), command=getLoginInput)
60 button1.grid(column=1, row=3, pady=2, padx=15,sticky='e')
```



1. Login 視窗

```
12 # login smtp
13 def loginSMTP(account,password):
14     try:
15         with smtplib.SMTP(host="smtp.gmail.com", port="587") as smtp: # 設定SMTP伺服器
16             try:
17                 smtp.ehlo() # 驗證SMTP伺服器
18                 smtp.starttls() # 建立加密傳輸
19                 smtp.login(account,password) # 登入寄件者gmail
20                 tk.messagebox.showinfo("^_^", '登入成功, 選擇檔案\nLogin successful. Please select a file')
21                 root.destroy()
22                 sender_tkGUI.exe_choose_file(account,password)
23             except Exception as e:
24                 tk.messagebox.showinfo("*_*", '信箱或SMTP輸入錯誤, 請重新輸入\nWrong User Gamil or Password')
25     except:
26         tk.messagebox.showinfo("*_*", '沒有網路連線')
```

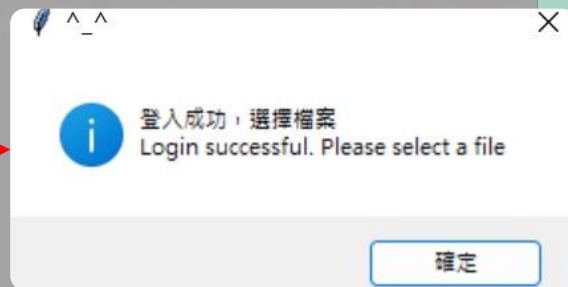
3 import smtplib

```
6 # 取得 input
7 def getLoginInput(event=None):
8     account = enry1.get()
9     password = enry2.get()
10    loginSMTP(account,password)
```

```
58 # Button
59 button1 = tk.Button(text='Login',width=20, bg="white", font=("Helvetica",'12','bold'), command=getLoginInput)
60 button1.grid(column=1, row=3, pady=2, padx=15,sticky='e')
```

1. Login 視窗

```
12 # login smtp
13 def loginSMTP(account,password):
14     try:
15         with smtplib.SMTP(host="smtp.gmail.com", port="587") as smtp: # 設定SMTP伺服器
16             try:
17                 smtp.ehlo() # 驗證SMTP伺服器
18                 smtp.starttls() # 建立加密傳輸
19                 # 登入信箱
20                 tk.messagebox.showinfo("^_^", '登入成功，選擇檔案\nLogin successful. Please select a file')
21                 root.destroy()
22                 sender.tkGUI_exe.choose_file(account,password)
23             except:
24                 tk.messagebox.showinfo("** **", '信箱或SMTP輸入錯誤，請重新輸入\nWrong User Gamil or Password')
25     except:
26         tk.messagebox.showinfo("** **", '沒有網路連線')
```



```
4 import senderTkGUI
```

進入下個python檔

02

選擇excel檔視窗

連結使用者的csv / xlsx檔

2. 選擇excel檔視窗

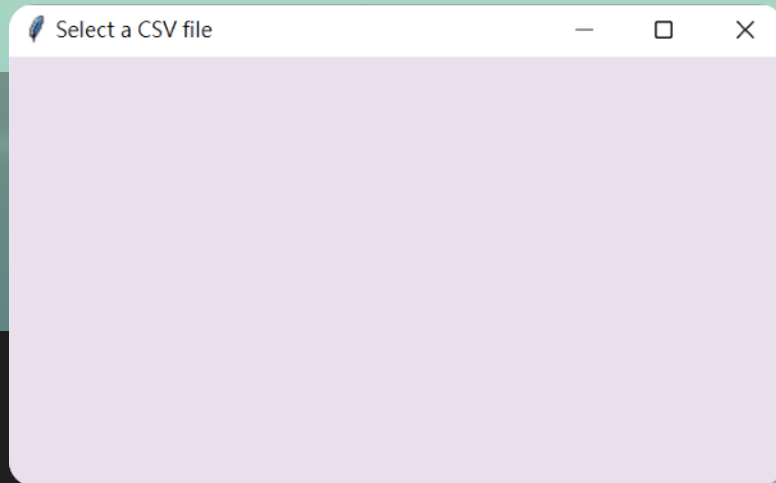
```
1 import tkinter as tk
2 from tkinter import ttk
3 import pandas as pd
4 from tkinter import filedialog
```

```
25 def exe_choose_file(user_account, user_password):
26     def loadFile():
27         file_path = filedialog.askopenfilename(title = "Select file",filetypes = (("CSV Files","*.csv"),("Excel files", "*.xlsx")))
28         if file_path[-5:]=="*.xlsx":
29             df = pd.read_excel(file_path)
30         elif file_path[-4:]=='*.csv':
31             df = pd.read_csv(file_path)
32         window0.destroy()
33         exe(df, user_account, user_password)
34     window0 = tk.Tk()
35     window0.title('Select file')
36     window0.minsize(450, 250)
37     window0.maxsize(450, 250)
38     window0.geometry('+400+150')
39     window0.config(bg="#e9dfed")
40
41     lb1 = tk.Label(window0, text='請選擇 csv 或 xlsx 檔\nPlease select a csv or xlsx file', bg="#e9dfed", font=("Helvetica",'12'))
42     lb1.pack(anchor=tk.N, pady=30)
43
44     # 選取檔案 (command 為選取檔案並把檔案中的 col 名加入到 div1 裡)
45     bt1 = tk.Button(window0, height=4, width=17, text='選取檔案\nSelect a file', command=loadFile, font=("Helvetica",'10'))
46     # 按下後會取得 df
47     bt1.pack(pady=10)
48
49     window0.mainloop()
```

2. 選擇excel檔視窗

```
import tkinter as tk
from tkinter import filedialog
import pandas as pd
import os
```

```
34 window0 = tk.Tk()
35 window0.title('Select file')
36 window0.minsize(450, 250)
37 window0.maxsize(450, 250)
38 window0.geometry('+400+150')
39 window0.config(bg="#e9dfed")
```



2. 選擇excel檔視窗

```
import tkinter as tk
from tkinter import messagebox
import pandas as pd
import os
```

```
def loadFile():
    # 選擇檔案
    file = filedialog.askopenfilename()
    if file:
        df = pd.read_csv(file)
        # 顯示檔案內容
        df.to_string()
```



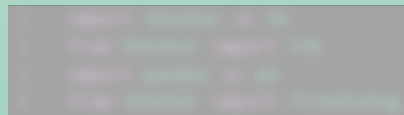
```
41 lb1 = tk.Label(window0, text='請選擇 csv 或 xlsx 檔\nPlease select a csv or xlsx file', bg="#e9dfed", font=("Helvetica",'12'))
42 lb1.pack(anchor=tk.N, pady=30)
43
44 # 選取檔案 (command 為選取檔案並把檔案中的 col 名加入到 div1 裡)
45 bt1 = tk.Button(window0, height=4, width=17, text='選取檔案\nSelect a file', command=loadFile, font=("Helvetica",'10'))
46 # 按下後會取得 df
47 bt1.pack(pady=10)
```

2. 選擇excel檔視窗



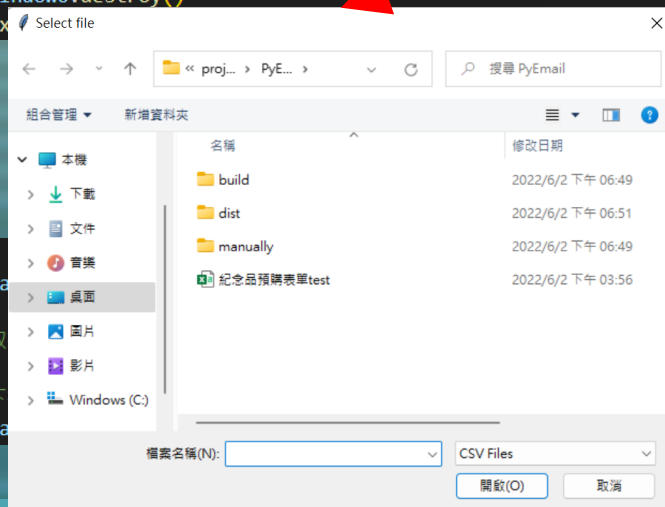
```
41 lb1 = tk.Label(window0, text='請選擇 csv 或 xlsx 檔\nPlease select a csv or xlsx file', bg='#e9dfed', font=("Helvetica", '12'))
42 lb1.pack(anchor=tk.N, pady=30)
43
44 # 選取檔案 (command 為選取檔案並把檔案中的 col 名加入到 div1 裡)
45 bt1 = tk.Button(window0, height=4, width=17, text='選取檔案\nSelect a file', command=loadFile, font=("Helvetica", '10'))
46 # 按下後會取得 df
47 bt1.pack(pady=10)
```

2. 選擇excel檔視窗



```
26 def loadFile():  
27     file_path = filedialog.askopenfilename(title = "Select file",filetypes = (("CSV Files","*.csv"),("Excel files", "*.xlsx")))  
28     if file_path[0] != "":  
29         df = pd.read_excel(file_path)  
30     elif file_path[-4:] == '.csv':  
31         df = pd.read_csv(file_path)  
32     window0.destroy()  
33     ex
```

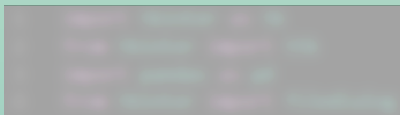
```
41 lb1 =  
42 lb1.pa  
43  
44 # 選取  
45 bt1 =  
46 # 按下  
47 bt1.pa
```



```
ect a csv or xlsx file', bg="#e9dfed", font=("Helvetica",'12'))
```

```
ect a file' command=loadFile, font=("Helvetica",'10'))
```

2. 選擇excel檔視窗



```
26  def loadFile():
27      file_path = filedialog.askopenfilename(title = "Select file",filetypes = (("CSV Files","*.csv"),("Excel files", "*.xlsx")))
28      if file_path[-5:]=="*.xlsx":
29          df = pd.read_excel(file_path)
30      elif file_path[-4:]=="*.csv":
31          df = pd.read_csv(file_path)
32      window.destroy()
33      exe(df, user_account, user_password)
```

進入下個視窗

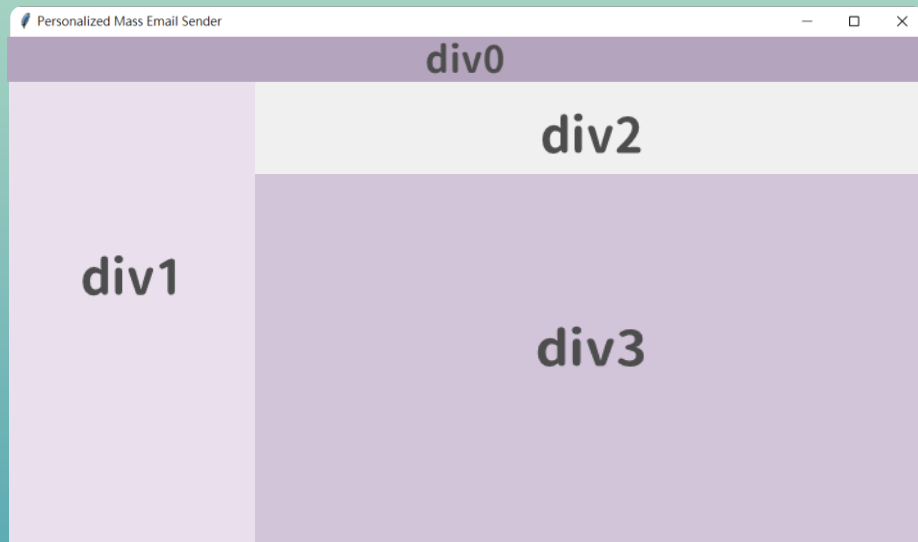
03

編輯信件草稿

+選擇收件人欄位、編輯標題、編輯草稿

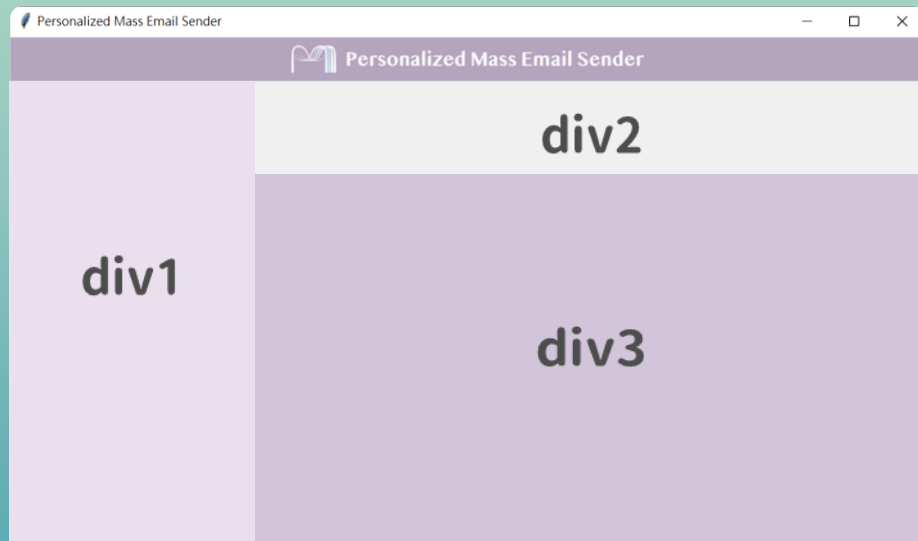
3. 編輯信件草稿

```
46 def exe(df, account, smtp):
47     window = tk.Tk()
48     window.title('大量郵件發送')
49     window.geometry('900x500+160+70')
50     window.minsize(900, 500)
51     align_mode = 'nswe'
52
53     """Frame (版面配置)"""
54     div_size = 200
55     img_size = 50
56     div0 = tk.Frame(window, width=900, height=30, bg='#b5a4bd')
57     div1 = tk.Frame(window, width=400, height=500)
58     div2 = tk.Frame(window, width=500, height=200)
59     div3 = tk.Frame(window, width=500, height=300, bg='#d2c5d9')
60
61     div0.grid(column=0, row=0, colspan=2, sticky=align_mode)
62     div1.grid(column=0, row=1, rowspan=2, sticky=align_mode)
63     div2.grid(column=1, row=1)
64     div3.grid(column=1, row=2, sticky=align_mode)
65
66     define_layout(window, cols=2, rows=3)
67     define_layout([div0, div1, div2, div3])
```



3. 編輯信件草稿-div0

```
71     '''---div0--- Logo列'''
72
73     image=Image.open('logo+mark+background.png')
74     img=image.resize((350,30),Image.ANTIALIAS)
75     img=ImageTk.PhotoImage(img)
76     panel = Label(div0, image = img, height=30,width=350, bg="#b5a4bd")
77     panel.grid(column=0, row=0)
```



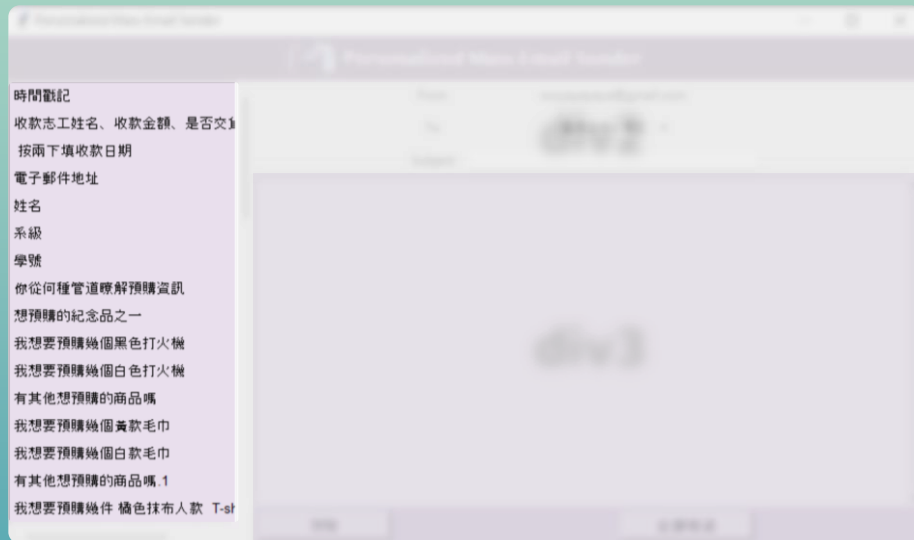
3. 編輯信件草稿-div1

```
79  '''---div1--- 檔案col顯示'''
80
81  replace_lst=[]
82  def cb(event):
83      index = div1_lb.curselection()
84      div3_text.insert('insert',"<<{'', 'iterating')
85      div3_text.insert(tk.INSERT,div1_lb.get(index), 'iterating')
86      div3_text.insert('insert',">>"}", 'iterating')
87      if ('<<{' + div1_lb.get(index) + ">>"}") not in replace_lst:
88          replace_lst.append('<<{' + div1_lb.get(index) + ">>"}')
89
90  # 右邊卷軸
91  div1_sb1 = tk.Scrollbar(div1)
92  div1_sb1.grid(column=1, sticky=align_mode)
93  # 下面的卷軸
94  div1_sb2= tk.Scrollbar(div1, orient=tk.HORIZONTAL)
95  div1_sb2.grid(column=0, sticky=align_mode)
96  # 左邊 col 的 Listbox
97  div1_lb = tk.Listbox(div1, yscrollcommand=div1_sb1.set, xscrollcommand=div1_sb2.set,\
98      selectmode=tk.SINGLE, bg="#e9dfed", font=("Helvetica", '10'), activestyle="dotbox",\
99      exportselection=0, selectborderwidth=5, selectbackground="#483d4d",\
100      selectforeground='white', relief="sunken")
101  div1_lb.grid(column=0, row=0, sticky=align_mode)
102  div1_lb.bind('<<ListboxSelect>>', cb)
103
104  # 結合右邊(垂直)卷軸
105  div1_sb1.config(command=div1_lb.yview)
106  # 結合下面(水平)捲軸
107  div1_sb2.config(command=div1_lb.xview)
108
109  # 把 csv 的 columns 放到 listbox 裡
110  for col in list(df.columns):
111      div1_lb.insert(tk.END, col)
```



3. 編輯信件草稿-div1

```
96 # 左邊 col 的 Listbox
97 div1_lb = tk.Listbox(div1, yscrollcommand=div1_sb1.set, xscrollcommand=div1_sb2.set,\
98     selectmode=tk.SINGLE, bg="#e9dfed", font=("Helvetica", '10'), activestyle="dotbox",\
99     exportselection=0, selectborderwidth=5, selectbackground="#483d4d",\
100     selectforeground='white', relief="sunken")
101 div1_lb.grid(column=0, row=0, sticky=align_mode)
102 div1_lb.bind('<<ListboxSelect>>', cb)
```



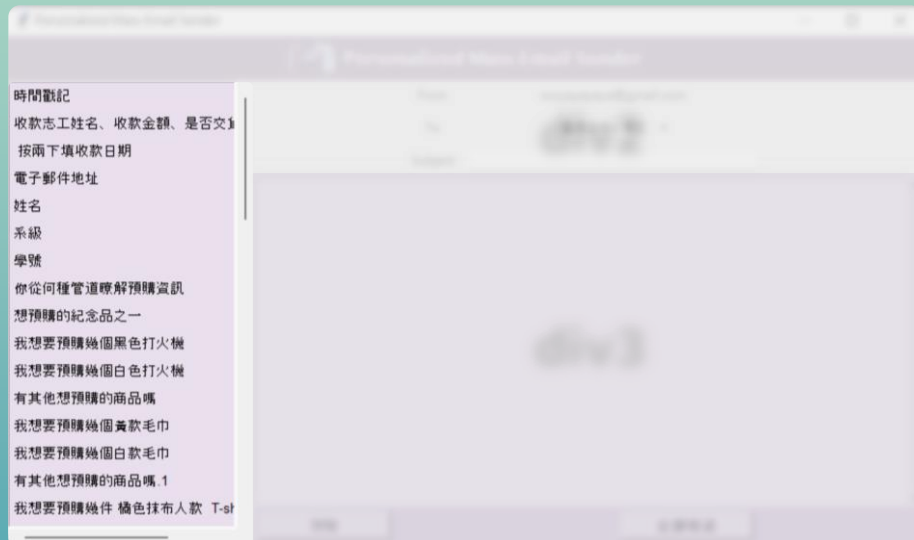
3. 編輯信件草稿-div1

```
90 # 右邊卷軸
91 div1_sb1 = tk.Scrollbar(div1)
92 div1_sb1.grid(column=1, sticky=align_mode)
93 # 下面的卷軸
94 div1_sb2= tk.Scrollbar(div1, orient=tk.HORIZONTAL)
95 div1_sb2.grid(column=0, sticky=align_mode)
```



3. 編輯信件草稿-div1

```
104 # 結合右邊(垂直)捲軸
105 div1_sb1.config(command=div1_lb.yview)
106 # 結合下面(水平)捲軸
107 div1_sb2.config(command=div1_lb.xview)
```



3. 編輯信件草稿-div1

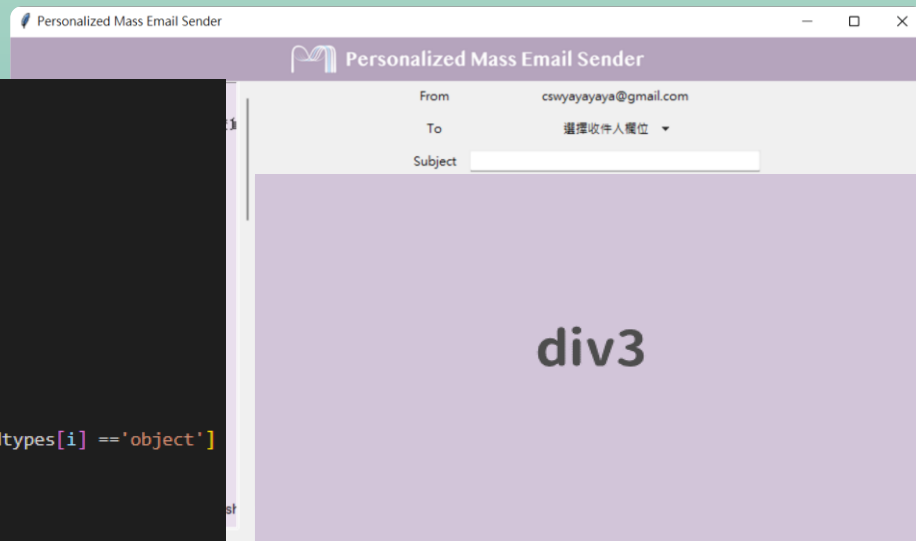
```
81 replace_lst=[]
82 def cb(event):
83     index = div1_lb.curselection()
84     div3_text.insert('insert',"<<{'", 'iterating')
85     div3_text.insert(tk.INSERT,div1_lb.get(index), 'iterating')
86     div3_text.insert('insert',">>"}", 'iterating')
87     if ('<<{' + div1_lb.get(index) + ">>"}") not in replace_lst:
88         replace_lst.append('<<{' + div1_lb.get(index) + ">>"}')
```

```
109 # 把 csv 的 columns 放到 listbox 裡
110 for col in list(df.columns):
111     div1_lb.insert(tk.END, col)
```



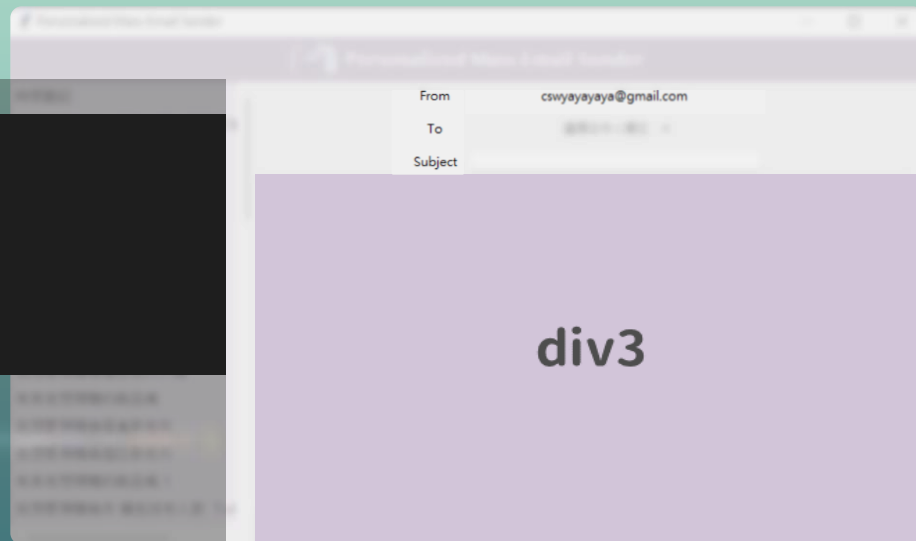
3. 編輯信件草稿-div2

```
113 '''---div2--- 寄信資訊'''
114 # label (from, to ,subject)
115 div2_title1 = tk.Label(div2, text="From", fg='black')
116 div2_title2 = tk.Label(div2, text="To", fg='black')
117 div2_title3 = tk.Label(div2, text="Subject", fg='black')
118 div2_title4 = tk.Label(div2, text=account, fg='black')
119
120 div2_title1.grid(column=0, row=0, sticky=align_mode)
121 div2_title2.grid(column=0, row=1, sticky=align_mode)
122 div2_title3.grid(column=0, row=2, sticky=align_mode)
123 div2_title4.grid(column=1, row=0, sticky=align_mode)
124
125 # optionlist 選擇收件者
126 optionList = [df.dtypes.index[i] for i in range(len(df.dtypes)) if df.dtypes[i] == 'object']
127 var= tk.StringVar()
128 div2_om1 = ttk.OptionMenu(div2, var, '選擇收件人欄位', *optionList)
129 div2_om1.grid(column=1, row=1, padx=10, pady=10)
130
131 # entry 填入信件標題subject
132 div2_entry1 = ttk.Entry(div2, width=40)
133 div2_entry1.grid(column=1, row=2, padx=10, sticky=align_mode)
```



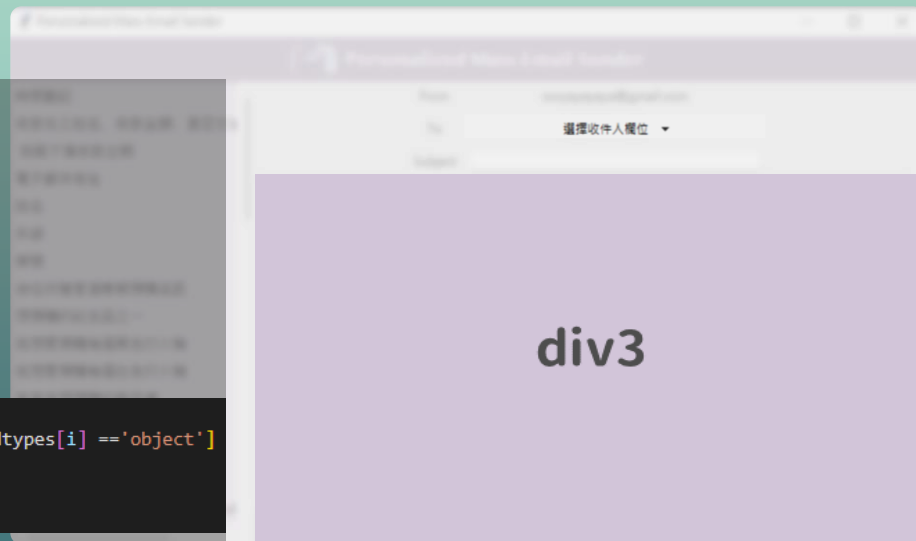
3. 編輯信件草稿-div2

```
114 # label (from, to ,subject)
115 div2_title1 = tk.Label(div2, text="From", fg='black')
116 div2_title2 = tk.Label(div2, text="To", fg='black')
117 div2_title3 = tk.Label(div2, text="Subject", fg='black')
118 div2_title4 = tk.Label(div2, text=account, fg='black')
119
120 div2_title1.grid(column=0, row=0, sticky=align_mode)
121 div2_title2.grid(column=0, row=1, sticky=align_mode)
122 div2_title3.grid(column=0, row=2, sticky=align_mode)
123 div2_title4.grid(column=1, row=0, sticky=align_mode)
```



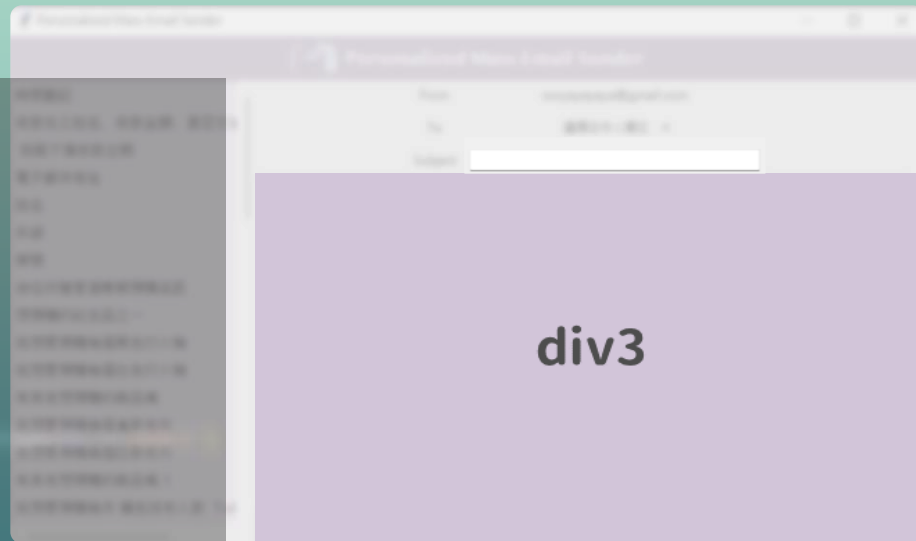
3. 編輯信件草稿-div2

```
125 # optionlist 選擇收件者
126 optionList = [df.dtypes.index[i] for i in range(len(df.dtypes)) if df.dtypes[i] == 'object']
127 var= tk.StringVar()
128 div2_om1 = ttk.OptionMenu(div2, var, '選擇收件人欄位', *optionList)
129 div2_om1.grid(column=1, row=1, padx=10, pady=10)
```



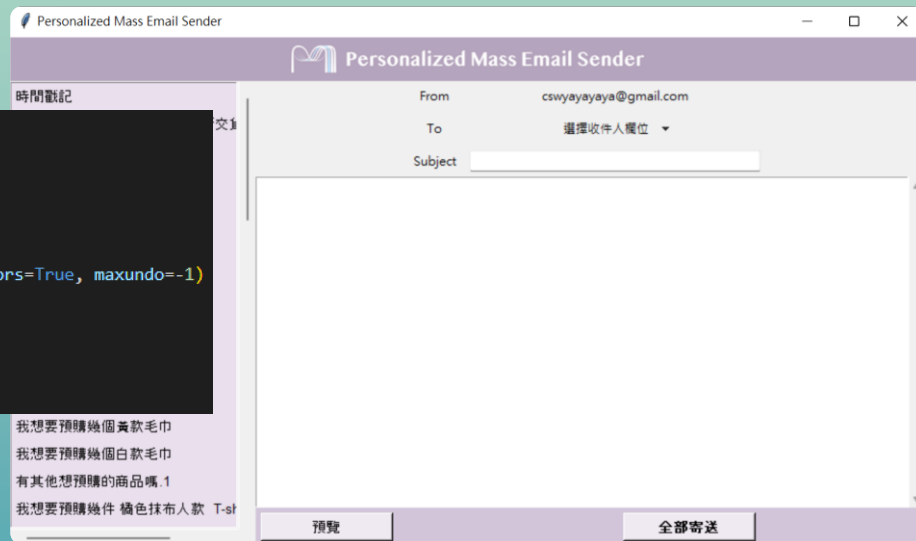
3. 編輯信件草稿-div2

```
131 # entry 填入信件標題subject
132 div2_entry1 = ttk.Entry(div2, width=40)
133 div2_entry1.grid(column=1, row=2, padx=10, sticky=align_mode)
```



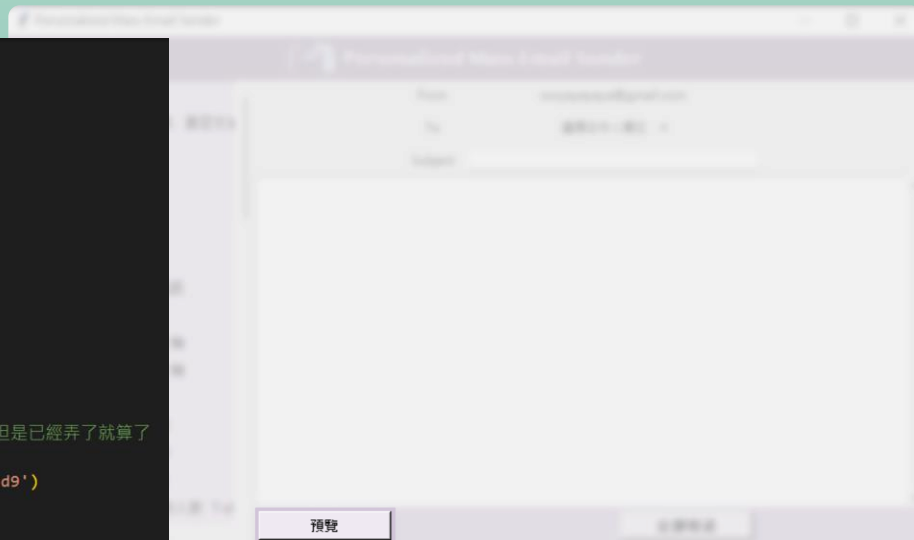
3. 編輯信件草稿-div3

```
136 '''---div3--- 信件內容'''
137 # Scrollbar
138 div3_sb = tk.Scrollbar(div3)
139 div3_sb.grid(column=3, row=0, sticky=align_mode)
140
141 # text 郵件內容文字框
142 div3_text = tk.Text(div3, yscrollcommand=div3_sb.set, undo=True, autoseparators=True, maxundo=-1)
143 div3_text.grid(column=0, row=0, columnspan=2, sticky=align_mode)
144 div3_text.tag_config('iterating', foreground="#a3466e")
145
146 # 結合 Scrollbar 和 text
147 div3_sb.config(command=div3_text.yview)
```



3. 編輯信件草稿-預覽

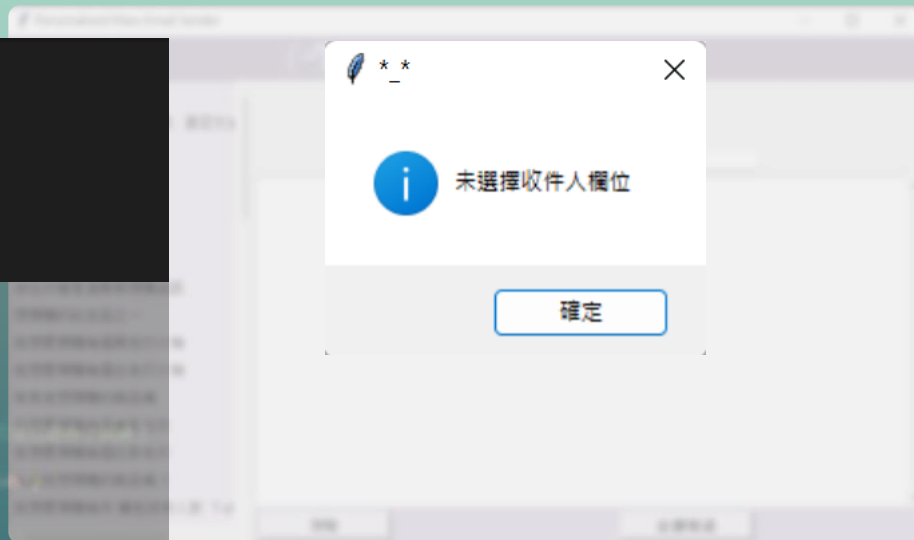
```
152 # button 預覽第一篇視窗
153 def preview():
154     show = True
155     try: # 得到剛剛使用者輸入的標題和接收者
156         subject = div2_entry1.get()
157         recipients = df[var.get()]
158         raw_content = div3_text.get("1.0","end")
159     except:
160         show = False
161         tk.messagebox.showinfo("*_*", '未選擇收件人欄位')
162
163     if show == True:
164         # Frame 預覽視窗
165         window_previes = tk.Tk()
166         window_previes.title('Preview')
167         window_previes.geometry('500x550+150+30')
168         window_previes.minsize(580, 500) # 這裡應該要用 .resizable(0, 0) 比較好，但是已經弄了就算了
169         window_previes.maxsize(580, 500)
170         div0_preview = tk.Frame(window_previes, width=550, height=50, bg='#d2c5d9')
171         div1_preview = tk.Frame(window_previes, width=550, height=450)
172         div0_preview.grid(column=0, row=0, sticky=align_mode)
173         div1_preview.grid(column=0, row=1, sticky=align_mode)
174         define_layout(window_previes, cols=0, rows=2)
175         define_layout([div0_preview, div1_preview])
176
177         # label 資訊欄
178         div0_lb1 = tk.Label(div0_preview, text='From:', bg='#d2c5d9', fg='black')
179         div0_lb2 = tk.Label(div0_preview, text='To:', bg='#d2c5d9', fg='black')
180         div0_lb3 = tk.Label(div0_preview, text='Subject:', bg='#d2c5d9', fg='black')
181         div0_lb1.grid(column=0, row=0, sticky='w', padx=50)
182         div0_lb2.grid(column=0, row=1, sticky='w', padx=50)
```



3. 編輯信件草稿-預覽

```
152 # button 預覽第一篇視窗
153 def preview():
154     show = True
155     try: # 得到剛剛使用者輸入的標題和接收者
156         subject = div2_entry1.get()
157         recipients = df[var.get()]
158         raw_content = div3_text.get("1.0","end")
159     except:
160         show = False
161         tk.messagebox.showinfo("*_*", '未選擇收件人欄位')
```

```
162
163 if show == True:
164     # 預覽視窗
165     window_preview = tk.Tk()
166     window_preview.title("預覽")
167     window_preview.geometry("600x400")
168     window_preview.resizable(100, 100) # 預覽視窗大小可調整
169     window_preview.maxsize(1000, 1000)
170     div1_preview = tk.Frame(window_preview, width=100, height=10, bg="white")
171     div1_preview.grid(column=0, row=0, sticky="n", height=10)
172     div2_preview.grid(column=0, row=1, sticky="n", height=10)
173     div3_preview.grid(column=0, row=2, sticky="n", height=10)
174     define_layout(window_preview, column=0, row=3)
175     define_layout(div1_preview, div2_preview)
176     # 顯示預覽
177     div1_label = tk.Label(div1_preview, text="From:", bg="white", fg="black")
178     div2_label = tk.Label(div2_preview, text="To:", bg="white", fg="black")
179     div3_label = tk.Label(div3_preview, text="Content:", bg="white", fg="black")
180     div1_label.grid(column=0, row=0, sticky="n", padx=5)
181     div2_label.grid(column=0, row=1, sticky="n", padx=5)
182     div3_label.grid(column=0, row=2, sticky="n", padx=5)
```



3. 編輯信件草稿-預覽

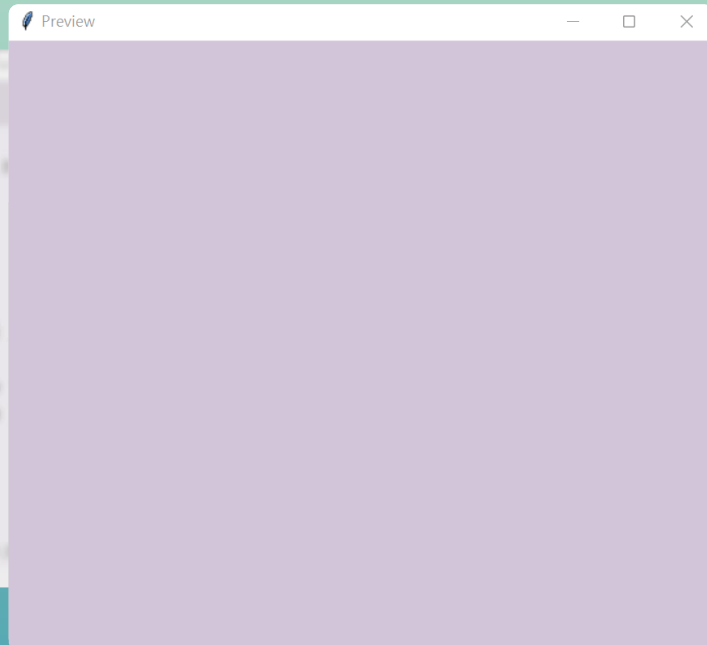
```
154 def preview():
155     show = True
156     # 顯示預覽視窗
157     subject = div0_entry.get()
158     recipients = div1_var.get()
159     msg_content = div0_text.get("1.0", "end")
160
161     except:
162         show = False
163         tk.messagebox.showinfo("提示", "不能預覽空人郵件")
164
```

```
163 if show == True:
164     # Frame 預覽視窗
165     window_previes = tk.Tk()
166     window_previes.title('Preview')
167     window_previes.geometry('500x550+150+30')
168     window_previes.minsize(580, 500) # 這裡應該要用 .resizable(0, 0) 比較好，但是已經弄了就算了
169     window_previes.maxsize(580, 500)
170     div0_preview = tk.Frame(window_previes, width=550, height=50, bg='#d2c5d9')
171     div1_preview = tk.Frame(window_previes, width=550, height=450)
172     div0_preview.grid(column=0, row=0, sticky=align_mode)
173     div1_preview.grid(column=0, row=1, sticky=align_mode)
174     define_layout(window_previes, cols=0, rows=2)
175     define_layout([div0_preview, div1_preview])
176
```

```
177 # 標題 顯示
178 div0_label = tk.Label(div0_preview, text="From:", bg="#d2c5d9", fg="black")
179 div0_label = tk.Label(div0_preview, text="To:", bg="#d2c5d9", fg="black")
180 div0_label = tk.Label(div0_preview, text="Subject:", bg="#d2c5d9", fg="black")
181 div0_label.grid(column=0, row=0, sticky="w", padx=5)
182 div0_label.grid(column=0, row=1, sticky="w", padx=5)
183 div0_label.grid(column=0, row=2, sticky="w", padx=5)
184
```

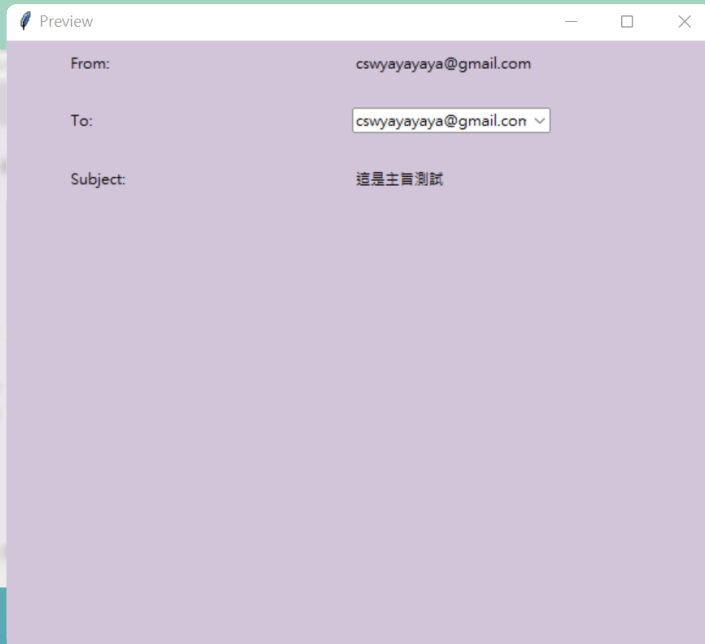
```
185 div0_label = tk.Label(div0_preview, text="Content:", bg="#d2c5d9", fg="black")
186 div0_label = tk.Label(div0_preview, text="Subject:", bg="#d2c5d9", fg="black")
187 div0_label.grid(column=0, row=3, sticky="w", padx=5)
188 div0_label.grid(column=0, row=4, sticky="w", padx=5)
189
```

```
190 # 內容 顯示 (發送信件時，發送) 內容
191 def content_selected(event): # 發送信件時，發送內容
192     content = msg_content.get()
```



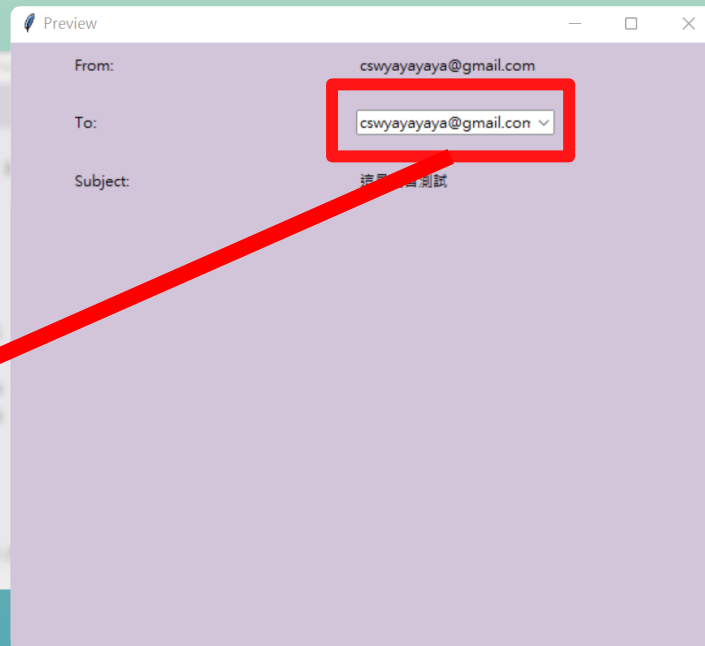
3. 編輯信件草稿-預覽

```
177 # label 資訊欄
178 div0_lb1 = tk.Label(div0_preview, text='From:', bg='#d2c5d9', fg='black')
179 div0_lb2 = tk.Label(div0_preview, text='To:', bg='#d2c5d9', fg='black')
180 div0_lb3 = tk.Label(div0_preview, text='Subject:', bg='#d2c5d9', fg='black')
181 div0_lb1.grid(column=0, row=0, sticky='w', padx=50)
182 div0_lb2.grid(column=0, row=1, sticky='w', padx=50)
183 div0_lb3.grid(column=0, row=2, sticky='w', padx=50)
184
185 div0_lb4 = tk.Label(div0_preview, text=account, bg='#d2c5d9', fg='black')
186 div0_lb6 = tk.Label(div0_preview, text=subject, bg='#d2c5d9', fg='black')
187 div0_lb4.grid(column=1, row=0, sticky='w', padx=50)
188 div0_lb6.grid(column=1, row=2, sticky='w', padx=50)
```



3. 編輯信件草稿-預覽

```
190 # combobox 選擇預覽的 gmail 收件人
191 def combobox_selected(event): # 改變選擇後預覽變化
192     content = raw_content[:]
193     for i in replace_lst:
194         try:
195             content = content.replace(i, df[i[3:-3]][div0_om1.current()])
196         except:
197             content = content.replace(i, "None")
198     div1_text.delete("1.0", "end")
199     div1_text.insert("insert", content)
201 previewList = [i for i in recipients]
202 div0_om1 = ttk.Combobox(div0_preview, values=previewList)
203 div0_om1.grid(column=1, row=1, sticky='w', padx=50, pady=10)
204
205 div0_om1.current(0) # 預設為第一個信箱
206 div0_om1.bind('<<ComboboxSelected>>', combobox_selected)
207 define_layout(div0_preview, cols=2, rows=5)
208 define_layout([div0_lb1, div0_lb2, div0_lb3, div0_lb4, div0_om1, div0_lb6])
```



3. 編輯信件草稿-預覽

```
210 # scrollbar & text 內文欄
211 div1_sb = tk.Scrollbar(div1_preview, orient='vertical')
212 div1_sb.grid(column=1, row=0, sticky=align_mode)
213
214 div1_text = tk.Text(div1_preview, yscrollcommand=div1_sb.set)
215 div1_text.grid(column=0, row=0, sticky=align_mode)
216
217 div1_sb.config(command=div1_text.yview)
218
219 # 預設內文為第一個 gmail (df[i[3:-3]][0])
220 content = raw_content[:]
221 for i in replace_lst:
222     try:
223         content = content.replace(i, df[i[3:-3]][0])
224     except:
225         content = content.replace(i, "None")
226
227 # 插入剛剛使用者輸入的內文
228 div1_text.insert("insert", content)
```

Preview

From: cswyayayaya@gmail.com

To:

Subject: 這是主旨測試

你好，你是

學號: B08201000

系級: 材料三

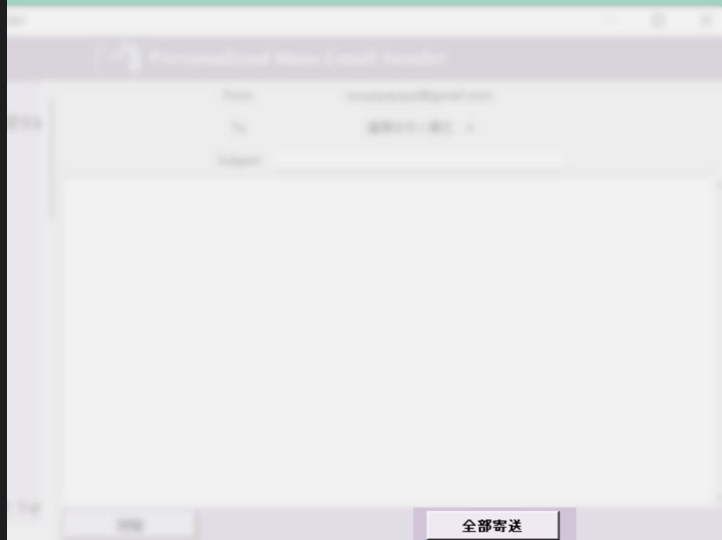
姓名: 王小明

3. 編輯信件草稿-預覽

```
234     # 預覽第一篇
235     div3_bt1 = tk.Button(div3, command=preview, width=15, text='預覽', bg='#efe9f2', font=("Helvetica", 12))
236     div3_bt1.grid(column=0, row=1, pady=5, padx=5, sticky='w')
```

3. 編輯信件草稿-寄送

```
287 # 寄出信
288 def send_all_mail():
289     show = True
290     try: # 得到剛剛使用者輸入的標題和接收者
291         subject = div2_entry1.get()
292         recipients = df[var.get()]
293         raw_content = div3_text.get("1.0", "end")
294     except:
295         show = False
296         tk.messagebox.showinfo("**_", '未選擇收件人欄位')
297
298     if show == True:
299         question = tk.messagebox.askyesno(message='確認寄出所有信件?')
300         if question == True:
301             lst_ok, lst_not_ok = [], []
302             for i in range(len(recipients)):
303                 content = raw_content[:]
304                 for j in replace_lst:
305                     content = content.replace(j, str(df[j[3:-3]][i]))
306                 re = sender_send.send(subject, account, smtp, recipients[i], content)
307                 if re == True: lst_ok.append(recipients[i])
308                 elif re == False: lst_not_ok.append(recipients[i])
309             show_result(lst_ok, lst_not_ok)
310         elif question == False:
311             pass
312
313 # 全部寄送
314 div3_bt2 = tk.Button(div3, command=send_all_mail, width=15, text='全部寄送', bg='#efe9f2')
315 div3_bt2.grid(column=1, row=1, padx=150, pady=5)
```



3. 編輯信件草稿-寄送

```
287 # 寄出信
288 def send_all_mail():
289     show = True
290     try: # 得到剛剛使用者輸入的標題和接收者
291         subject = div2_entry1.get()
292         recipients = df[var.get()]
293         raw_content = div3_text.get("1.0","end")
294     except:
295         show = False
296         tk.messagebox.showinfo("**_", '未選擇收件人欄位')
```



3. 編輯信件草稿-寄送

```
298 ∨ if show == True:
299     question = tk.messagebox.askyesno(message='確認寄出所有信件?')
300     if question == True:
301         lst_ok, lst_not_ok = [], []
302         for i in range(len(recipients)):
303             content = raw_content[:]
304             for j in replace_lst:
305                 content = content.replace(j, str(df[j[3:-3]][i]))
306             re = sender_send.send(subject, account, smtp, recipients[i], content)
307             if re == True: lst_ok.append(recipients[i])
308             elif re == False: lst_not_ok.append(recipients[i])
309         show_result(lst_ok, lst_not_ok)
310     elif question == False:
311         pass
```



3. 編輯信件草稿-寄送

```
298 if show == True:
299     question = tk.messagebox.askyesno(message='確認寄出所有信件?')
300     if question == True:
301         lst_ok, lst_not_ok = [], []
302         for i in range(len(recipients)):
303             content = raw_content[:]
304             for j in replace_lst:
305                 content = content.replace(j, str(lst[j][2:3][1]))
306             re = sender_send.send(subject, account, smtp, recipients[i], content)
307             if re == True: lst_ok.append(recipients[i])
308             elif re == False: lst_not_ok.append(recipients[i])
309             show_result(lst_ok, lst_not_ok)
310     elif question == False:
311         pass
```

```
5 import sender_send
```

進入下個python檔

3. 編輯信件草稿-寄送

```
1  from email.mime.multipart import MIMEMultipart
2  from email.mime.text import MIMEText
3  import smtplib
4
5  def send(send_subject, send_from, send_smtp, send_to, send_content):
6      content = MIMEMultipart() #建立MIMEMultipart物件
7      content["subject"] = send_subject #郵件標題
8      content["from"] = send_from #寄件者
9      content["to"] = send_to #收件者
10     content.attach(MIMEText(send_content)) #郵件內容
11
12     with smtplib.SMTP(host="smtp.gmail.com", port="587") as smtp: # 設定SMTP伺服器
13         try:
14             smtp.ehlo() # 驗證SMTP伺服器
15             smtp.starttls() # 建立加密傳輸
16             smtp.login(send_from, send_smtp) # 登入寄件者
17             smtp.send_message(content) # 發送郵件
18             return True
19         except Exception as e: # mail格式不對
20             return False
```

回到上個python檔

3. 編輯信件草稿-寄送

```
298 if show == True:
299     question = tk.messagebox.askyesno(message='確認寄出所有信件?')
300     if question == True:
301         lst_ok, lst_not_ok = [], []
302         for i in range(len(recipients)):
303             content = raw_content[:]
304             for j in replace_lst:
305                 content = content.replace(j, str(df[i][j].tolist()))
306             re = sender_send.send(subject, account, smtp, recipients[i], content)
307             if re == True: lst_ok.append(recipients[i])
308             elif re == False: lst_not_ok.append(recipients[i])
309             show_result(lst_ok, lst_not_ok)
310     elif question == False:
311         pass
```

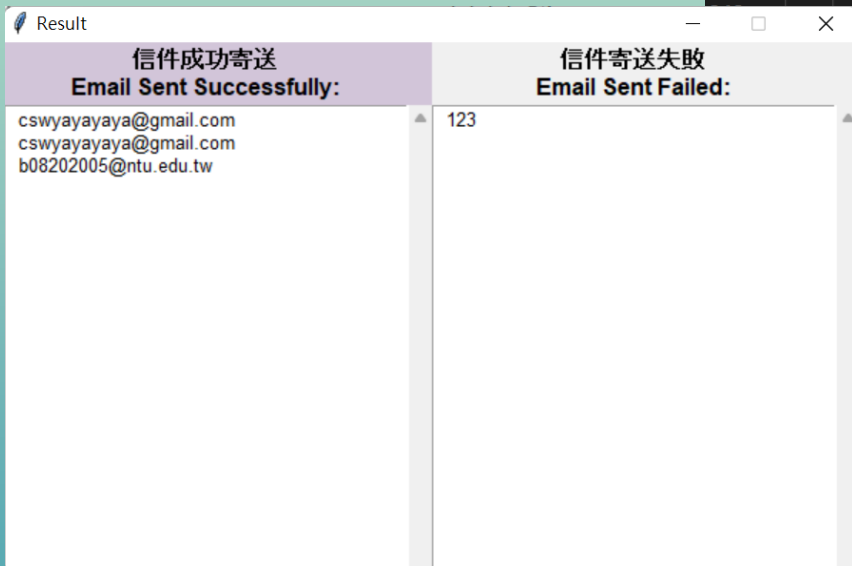

3. 編輯信件草稿-寄送

```
298 > if show == True:
299     question = tk.messagebox.askyesno(message='確認')
300 >     if question == True:
301         lst_ok, lst_not_ok = [], []
302 >         for i in range(len(recipients)):
303             content = raw_content[:]
304 >             for j in replace_lst:
305                 content = content.replace(j, str(i))
306             re = sender_send.send(subject, account, content)
307             if re == True: lst_ok.append(recipients[i])
308             elif re == False: lst_not_ok.append(recipients[i])
309             show_result(lst_ok, lst_not_ok)
310 >         elif question == False:
311             pass
```

show_result(lst_ok, lst_not_ok)

```
238 # 寄出信後的通知
239 > def show_result(sucess_lst, failed_lst):
240     # frame
241     window_show = tk.Tk()
242     window_show.title('Result')
243     window_show.geometry('600x370+300+90')
244     window_show.resizable(0, 0)
245     div0_show = tk.Frame(window_show, width=300, height=400, bg='#d2c5d9')
246     div1_show = tk.Frame(window_show, width=300, height=400)
247     div0_show.grid(column=0, row=0)
248     div1_show.grid(column=1, row=0)
249     define_layout(window_show, cols=2, rows=0)
250     define_layout([div0_show, div1_show])
251
252     # div0
253     div0_show_sb = tk.Scrollbar(div0_show, orient='vertical')
254     div0_show_lb = tk.Label(div0_show, text='信件成功寄送\nEmail Sent Successfully:',
255                             font=('Helvetica', 12))
256     div0_show_text = tk.Text(div0_show, yscrollcommand=div0_show_sb.set, font=('Helvetica', 12))
257
258     div0_show_sb.grid(column=2, row=1, sticky=align_mode)
259     div0_show_lb.grid(column=0, row=0, columnspan=2)
260     div0_show_text.grid(column=0, row=1, ipadx=15)
261     div0_show_sb.config(command=div0_show_text.yview)
262     define_layout(div0_show, cols=2, rows=2)
263     define_layout([div0_show_lb, div0_show_sb, div0_show_text])
264
265     for i in sucess_lst:
266         div0_show_text.insert("end", ' ' + i + '\n')
267         div0_show_text.configure(state='disabled')
```

3. 編輯信件草稿-寄送



```
238 # 寄出信後的通知
239 def show_result(sucess_lst, failed_lst):
240     # frame
241     window_show = tk.Tk()
    window_show.title('Result')
    window_show.geometry('600x370+300+90')
    window_show.resizable(0, 0)
    div0_show = tk.Frame(window_show, width=300, height=400, bg='#d2c5d9')
    div1_show = tk.Frame(window_show, width=300, height=400)
    div0_show.grid(column=0, row=0)
    div1_show.grid(column=1, row=0)
    define_layout(window_show, cols=2, rows=0)
    define_layout([div0_show, div1_show])

    # div0
    div0_show_sb = tk.Scrollbar(div0_show, orient='vertical')
    div0_show_lb = tk.Label(div0_show, text='信件成功寄送\nEmail Sent Successfully:',
    div0_show_text = tk.Text(div0_show, yscrollcommand=div0_show_sb.set, font=("Helv

    div0_show_sb.grid(column=2, row=1, sticky=align_mode)
    div0_show_lb.grid(column=0, row=0, columnspan=2)
    div0_show_text.grid(column=0, row=1, ipadx=15)
    div0_show_sb.config(command=div0_show_text.yview)
    define_layout(div0_show, cols=2, rows=2)
    define_layout([div0_show_lb, div0_show_sb, div0_show_text])

    for i in sucess_lst:
        div0_show_text.insert("end", ' ' + i + '\n')
    div0_show_text.configure(state='disabled')
```



包裝成應用程式



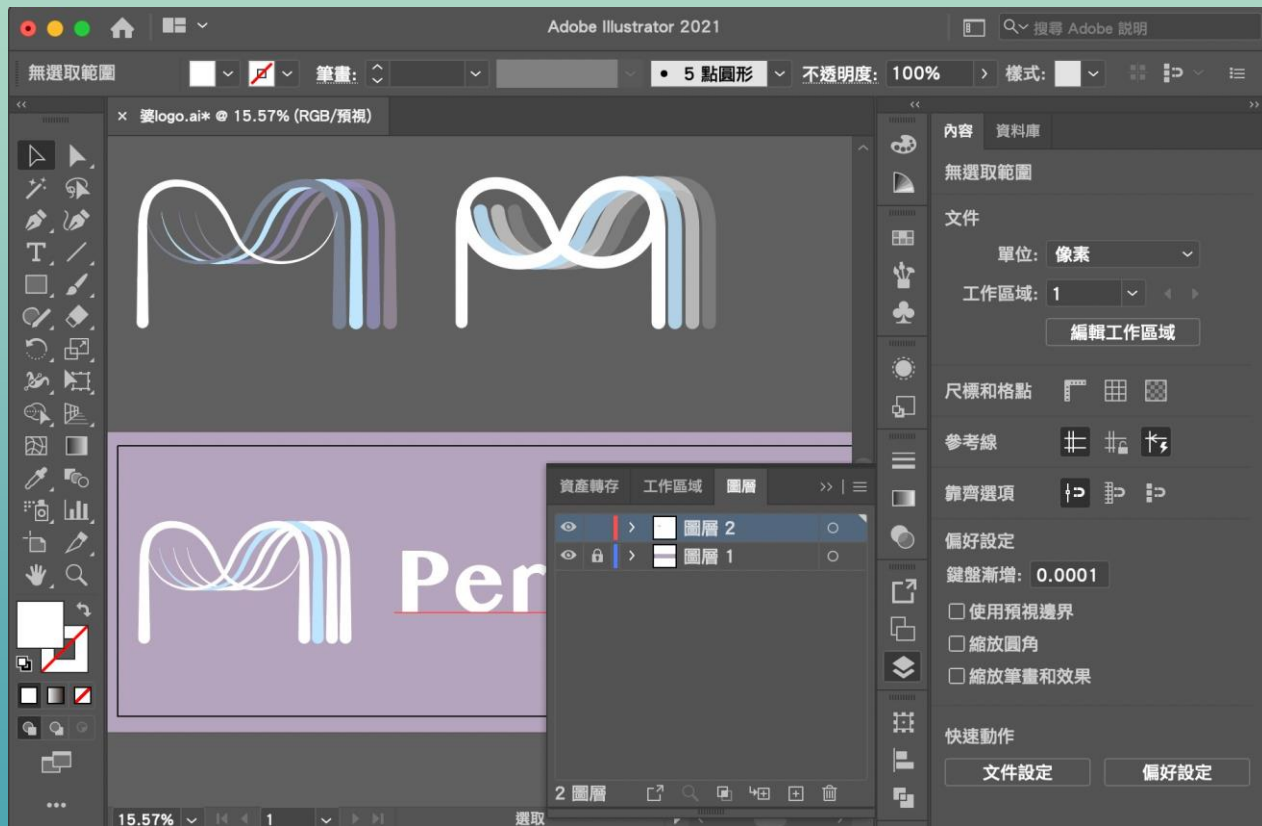
```
pyinstaller --onefile -w sender_login.py
```

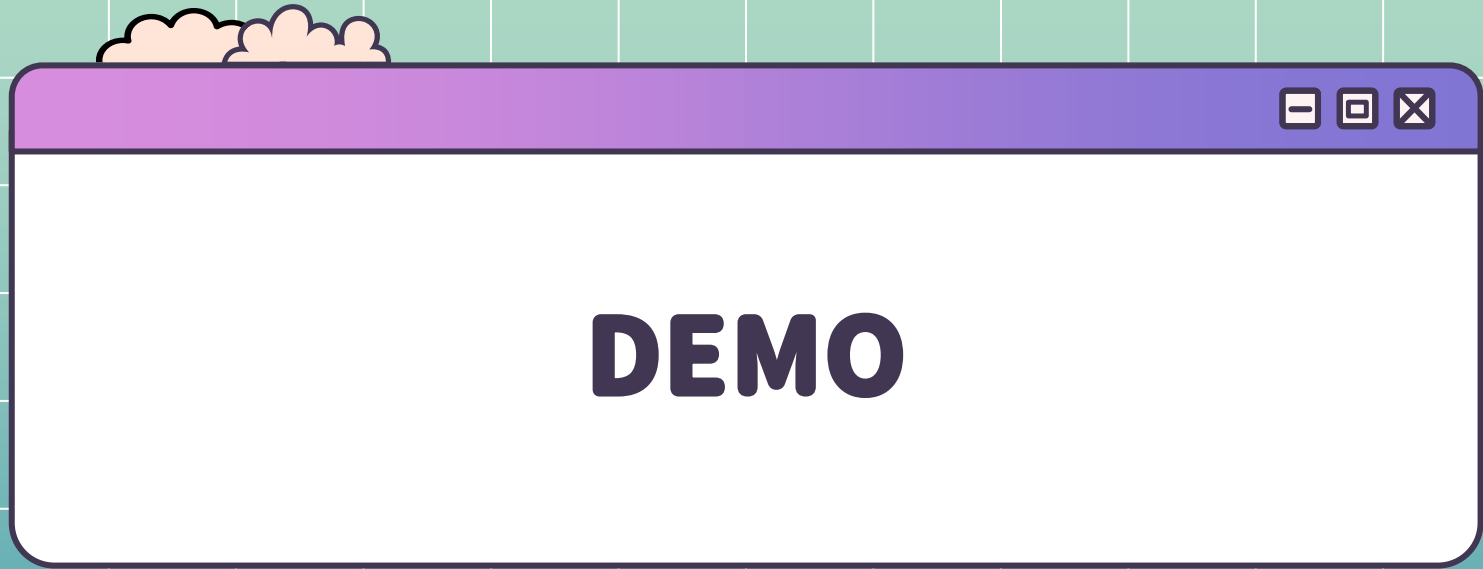
三個檔案合成一個

取消dos窗口

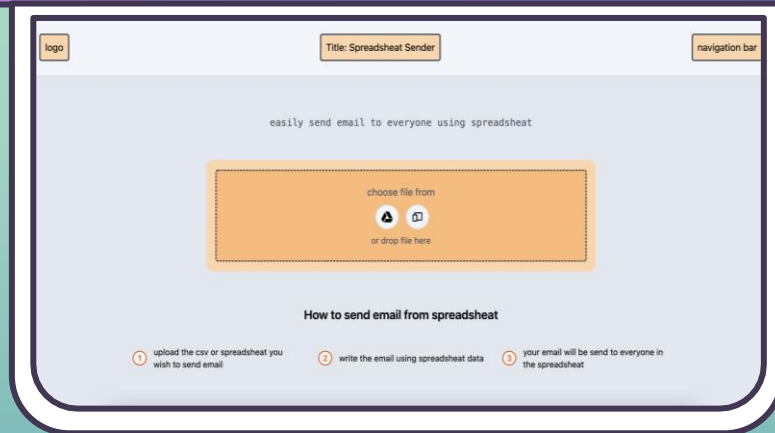
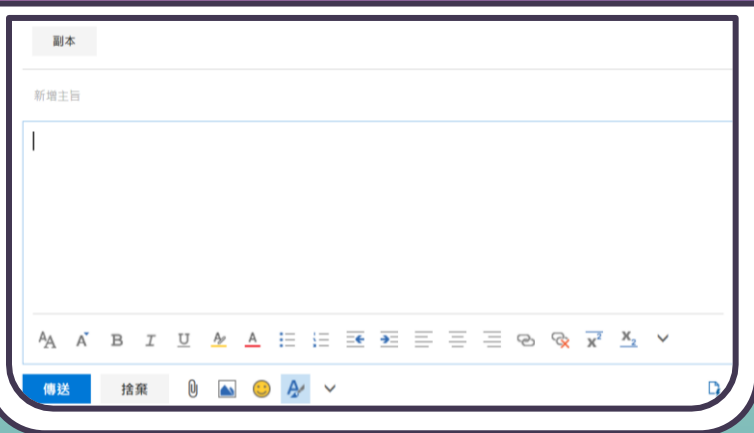
主程式

設計Logo





發展中...



字樣、插入附檔...

像 NTU mail一樣，
可以更改字體大小、**粗細**、*斜體*、底線、
插入圖檔、表格、附檔等

網頁版

做成網頁版讓大家免下載即可用；
連結google取得權限的方法得到寄 gmail
的權限。使用者就不用特別辦SMTP



組員分工



簡詩汶

Programmer

游涓聖

Programmer

張鈞量

Project Manager

