

# GithubAction與email

# 為甚麼要自動寄email?

如果我們需要大量的寄送郵件  
總不能一個一個複製

# 設定寄出郵件的google帳號



<https://myaccount.google.com/security?pli=1>

# 設定完成



首頁



個人資訊



資料和隱私權



安全性



使用者和分享內容

## 您登入 Google 的方式

請定期更新這些資訊，確保您隨時都能順利登入自己的 Google 帳戶



兩步驟驗證



啟用時間：清晨7:20



密碼金鑰

開始使用密碼金鑰



# 設定密碼

## ← 應用程式密碼

應用程式密碼可讓您在支援兩步驟驗證的裝置上，透過應用程式登入您的 Google 帳戶。應用程式密碼只要輸入一次即可，因此您不必記住這組密碼。 [瞭解詳情](#)

您沒有任何應用程式密碼。

選取您要產生應用程式密碼的應用程式和裝置。

選取應用程式

選取裝置 ▼

郵件

日曆

聯絡人

YouTube

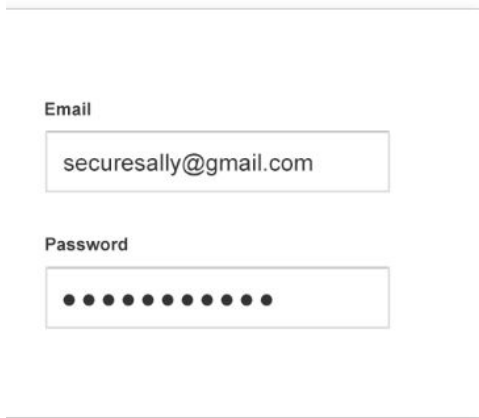
其他 (自訂名稱)

產生

<https://myaccount.google.com/apppasswords>

# 獲得密碼

## 系統產生的應用程式密碼



Email

securesally@gmail.com

Password

● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

您裝置專用的應用程式密碼



使用方式

在您想設定 Google 帳戶的應用程式或裝置上前往帳戶的「設定」頁面，然後將您的密碼替換成上方的 16 字元密碼。

這個應用程式密碼就如同您平常使用的密碼，可授予完整的 Google 帳戶存取權限。您不需要記住這組密碼，因此，請勿將密碼寫下或透露給任何人知道。

完成

## 接下來，送email的時候到了

```
import smtplib, ssl
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.header import Header

sender_email = "jufengwul02000@gmail.com"
receiver_email = "leowul02000@gmail.com"
password = " "

msg = MIMEMultipart()
msg['From'] = sender_email
msg['To'] = receiver_email
msg['Subject'] = Header('Test Send Email', 'utf-8').encode()

body = 'This is email from python'

msg_content = MIMEText(body)
msg.attach(msg_content)

# Create secure connection with server and send email
context = ssl.create_default_context()
with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as server:
    # 465是標準的SMTP-over-SSL端口
    server.login(sender_email, password)
    server.sendmail(sender_email, receiver_email, msg.as_string())

print("success send email")
```

# 小試身手的時間到了

小吳助理今天有一個名單

姓名,email

將姓名和email從文件中讀取，並且寫一個送email的程式自動送信

這必須要會讀檔案!!



## 將資料存檔 (write)

```
f = open("檔案名稱","w")
```

```
f.write("你的資料")
```

```
f.close()
```

這一個w是什麼意思？

# open mode

r : 只讀

w : 只寫

a : 追加

r+ : 讀寫->先讀後寫, 不清空內容,寫在開頭

w+ : 寫讀->先寫後讀, 打開後清空內容

a+ : 追加寫讀 -> 先寫後讀,不會清空內容

請大家嘗試看看

# 讀取檔案

我們可以用read和readline進行檔案的讀取

```
file1.read()
```

```
file1.readline()
```

```
file1.read(5)
```

[https://github.com/JuFengWu/basic\\_python/blob/master/read\\_file/read\\_file.py](https://github.com/JuFengWu/basic_python/blob/master/read_file/read_file.py)

## 讀取有逗號的檔案，並且分開

```
file1 = open("test.txt", "r",encoding="utf-8")

content = file1.read()

line = content.split("\n")
print(line)
result = line[0].split(",")
print(result)
```

# 我的email要漂亮

可以使用html格式送email

```
body = """  
<p>Python 測試email...</p>  
<p><a href="http://https://tw.fukuro-art-tech.com/">這是連結</a></p>  
"""  
  
msg_content = MIMEText(body, 'html')  
msg.attach(msg_content)
```

# 我的email需要附加檔案

```
pdf_path = "test_pdf_file.pdf"
with open(pdf_path, 'rb') as f:
    part = MIMEBase('application', 'octet-stream')
    part.set_payload(f.read())
    encoders.encode_base64(part)
    part.add_header('Content-Disposition', 'attachment; filename="pdf.pdf"')
    msg.attach(part)

txt_path = "test.txt"
with open(txt_path, 'rb') as f:
    part = MIMEBase('application', 'octet-stream')
    part.set_payload(f.read())
    encoders.encode_base64(part)
    part.add_header('Content-Disposition', 'attachment; filename="test.txt"')
    msg.attach(part)
```

# 如果要排時間送email ?

我們可以使用github 的github action當作伺服器排時程送email

// 每個星期一 13點 UTC (UTC 倫敦時間)

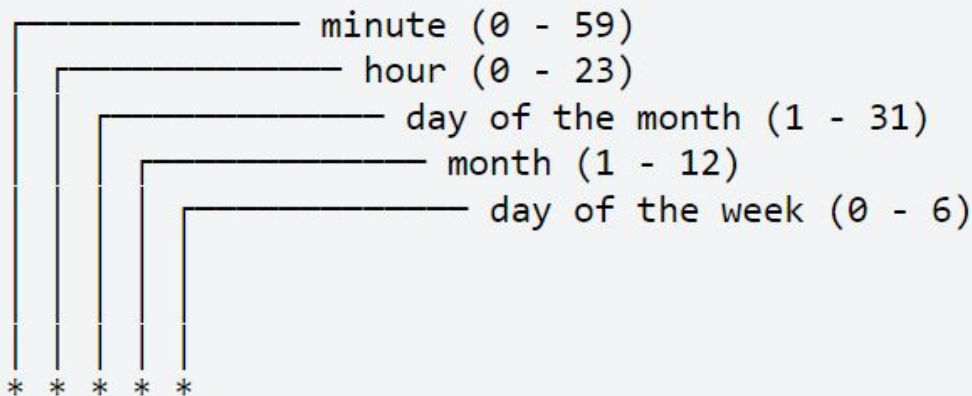
0 13 \* \* 1

// 每天的最後一刻

0 0 \* \* \*

// 每10分鐘

\*/10 \* \* \* \*



# 設定每5分鐘送email

使用情況：

每天早上利用爬蟲去找監督股價，如果低於某一個數值通知我

每周自動寫信通知所有人要記得開會

交給大家嘗試



# 利用github action送email

github action的使用回去看github action的講義

# 附錄：標籤語言

# 標記式語言是什麼？

文字與其他資訊結合一起的語法

但它不算是一種程式語言，而是透過標記方式將額外資訊進行標識

常見：html xml yaml

# html

HyperText Markup Language, 簡稱:HTML

HTML 是網頁上常使用的標記語言

# XML

Extensible Markup Language, 簡稱:XML

XML是照HTML的發展經驗, 產生出一套使用上規則嚴謹, 但是簡單的描述資料語言

# yaml

YAML是"YAML Ain't a Markup Language"

特色：使用空白符號縮排，相關的字符比較少，設計使人易讀

# 最簡單的Yaml

使用key value的方法來保存資料

name: John

age: 20

利用Yaml進行製作設定檔

[https://github.com/JuFengWu/basic\\_python/blob/master/yaml/example.yml](https://github.com/JuFengWu/basic_python/blob/master/yaml/example.yml)

# 利用Python讀取Yaml

---

```
import yaml

with open("example.yml", "r") as yamlData:
    data = yaml.load(yamlData, Loader=yaml.CLoader)

    print(data)

    for d in data:
        print(d)
        print(data[d])
```

[https://github.com/JuFengWu/basic\\_python/blob/master/yaml/read\\_yaml.py](https://github.com/JuFengWu/basic_python/blob/master/yaml/read_yaml.py)



# 小試身手時間到了

yaml內有一個 key： isDebug

如果isDebug設定為1，程式會出現 "這是debug模式"

如果isDebug設定為1，程式會出現 "這是運作模式"

## 把key value寫入yaml內

```
import yaml

data = {
    "isRun": "True",
    "MotorNumber": 5
}

# Save
with open('example.yml', 'w') as f:
    yaml.dump(data, f, Dumper=yaml.CDumper)
```

[https://github.com/JuFengWu/basic\\_python/blob/master/yaml/write\\_yaml.py](https://github.com/JuFengWu/basic_python/blob/master/yaml/write_yaml.py)