

範例程式碼：

```
1 #Python 程式設計練習：投資方案 IRR、MIRR 及 NPV 計算(使用 pip 指令安裝 numpy-financial)
2 import numpy_financial as npf
3 print('#本程式為投資方案 IRR、MIRR 及 NPV 計算程式，由屏東大學周國華老師設計#')
4 print()
5 print('請輸入投資方案期初及各期期末淨現金流入(出)金額，數字間以空白間隔。')
6 print('例如，期初淨投資 50，四期期末各淨流入 30、20、20、10')
7 print('輸入範例：-50 30 20 20 10','請在下一行開始輸入：')
8 CashInOut = list(map(int, input().strip().split()))
9 # input().strip().split() 中的 strip()在去掉空格，split()在為各數值之間加上逗號分隔
10 # map(int, input().strip().split())是在把螢幕輸入的值轉成 int
11 # list()是在把內容轉成串列
12 financeR=eval(input('請輸入資金成本率，例如，15%，請輸入 0.15：'))
13 reinvestR=eval(input('請輸入各期淨現金流入再投資報酬率，例如，12%，請輸入 0.12：'))
14 IRR=npf.irr(CashInOut)
15 MIRR=npf.mirr(CashInOut,financeR,reinvestR)
16 NPV=npf.npv(financeR,CashInOut)
17 # irr()、mirr()及 npv()的內容必須是串列數字
18 IRR=IRR*100
19 MIRR=MIRR*100
20 print()
21 print('#按照您輸入的相關數據，本投資方案的 IRR 及 MIRR 是：',format(IRR,'.4f'),'%','及 ',
22 format(MIRR,'.4f'),'%',sep='')
23 print('#本投資方案的淨現值(NPV)是：$',format(NPV,'.2f'),sep='')
24 print()
25 print('謝謝使用本程式！')
```

程式碼解析：

1. 本範例使用 `numpy-financial` 這個套件的 `irr`、`mirr` 及 `npv` 這三個財務函數的功能，`numpy-financial` 套件並非 Python 執行環境的預設安裝套件，在使用前可透過 `pip` 指令進行安裝。在命令提示字元(CMD)下的安裝畫面：  
在 `C:\>` 下輸入 `pip install numpy-financial` 後，按下 `enter` 即可自動進行安裝。(同學畫面可能稍有不同)

```
C:\>pip install numpy-financial
Collecting numpy-financial
  Downloading numpy_financial-1.0.0-py3-none-any.whl (14 kB)
Collecting numpy>=1.15
  Downloading numpy-1.21.4-cp39-cp39-win_amd64.whl (14.0 MB)
    |#####| 14.0 MB 939 kB/s
Installing collected packages: numpy, numpy-financial
Successfully installed numpy-1.21.4 numpy-financial-1.0.0
WARNING: You are using pip version 20.2.3; however, version 21.3.1 is available.
You should consider upgrading via the 'c:\users\user\appdata\local\programs\python\python39\python.exe -m pip install
upgrade pip' command.
C:\>
```

2. 請注意安裝時和載入時的名稱差異：
  - 安裝時：`pip install numpy-financial` 套件名稱的短線位置在中間
  - 載入時：`import numpy_financial as npf` 套件名稱的短線位置在下方
3. 安裝 Python 時，記得把下面的選項打勾，在執行 `pip` 指令時才找得到。



(註：截至 2025/12/03，Python 最新版為 3.14.1 版)

4. 在 Python 程式碼中載入其他套件的方式(複習)：
  - `import moduleA`：在呼叫 `moduleA` 套件內的任何函數時，需寫成 **moduleA.函數名稱**
  - `import moduleA as mA`：在呼叫 `moduleA` 套件內的任何函數時，可簡寫成 **mA.函數名稱**
  - `from moduleA import *` 在呼叫 `moduleA` 套件內的任何函數時，可直接寫**函數名稱**(以上的說明，在呼叫套件內的**方法**時，亦同)
5. 上面關於使用 `pip` 安裝其他 Python 套件的說明，可能不適用在部分同學使用的 Mac 電腦上。用 Mac 電腦的同學，可在 Google 上搜尋「如何在 Mac 上安裝和使用 PIP」，即可找到許多詳細的說明。
6. 有在其他軟體課程學過 Python 的同學，可能使用過 Anaconda 這類 IDE 軟體(IDE：Integrated Development Environment)。學寫程式一開始要儘量用接近原生工具的方法，例如用 Python 的 Idle。當你熟悉語法且累積一些程式寫作經驗後，就要學習使用 IDE 軟體，可加快程式開發的效率。
7. 提醒：上述程式碼行號的第 21、22 行是同一行程式碼。