



SCU LawTech

Class #10

Content

- ◆ 函式
- ◆ 本學期彙整
- ◆ 下學期的旅程

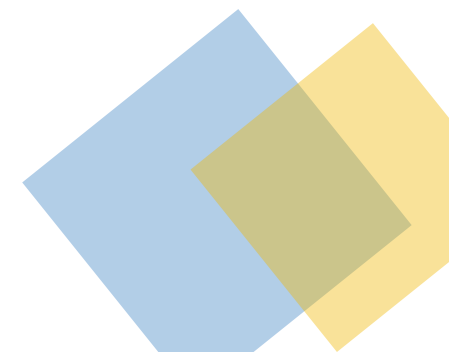
函式

函数 - 概念 - 1

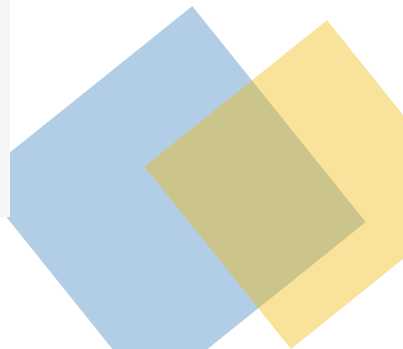
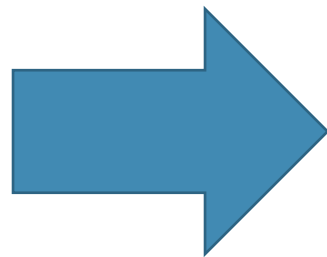
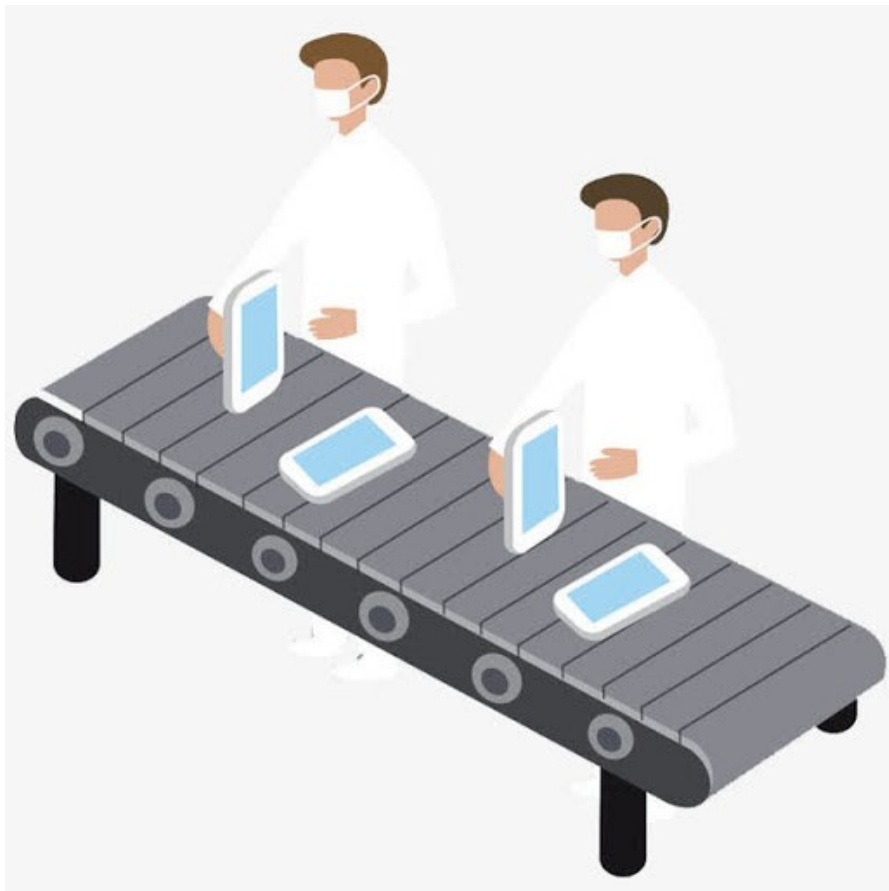
In Python, a function is a group of related statements that performs a specific task.

Functions help break our program into smaller and modular chunks. As our program grows larger and larger, functions make it more organized and manageable.

Furthermore, it avoids repetition and makes the code reusable.



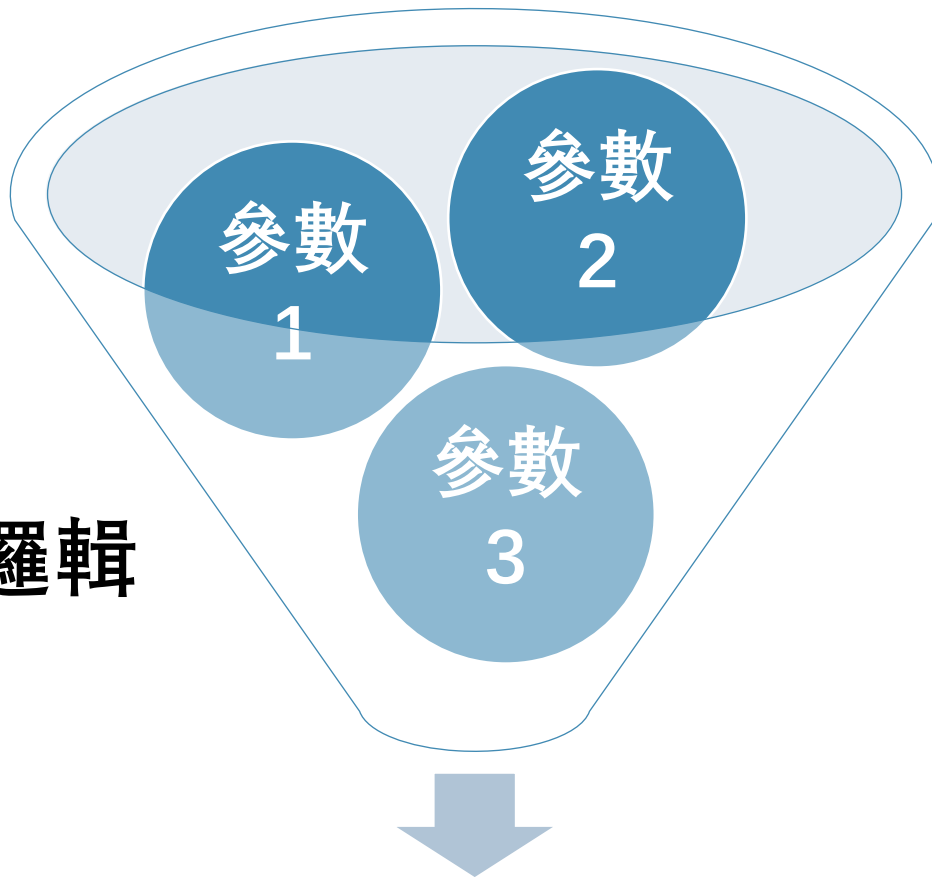
函式 – 概念 - 2



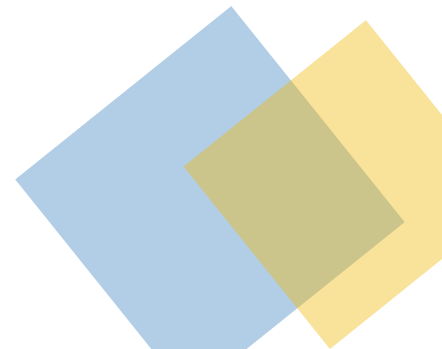
函式 – IPO

Input :
輸入，程序的輸入

Process :
處理，程序的主要邏輯



Output : 輸出，程序的輸出



函式 – 結構

函式定義

函式名稱*

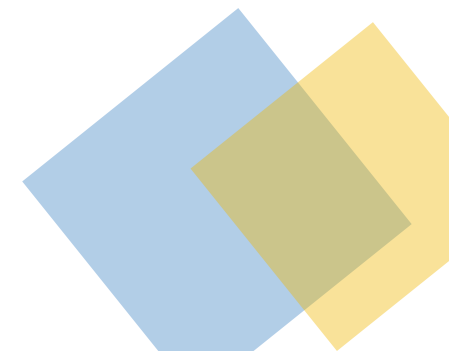
函式參數*

函式的執行任務*

```
def square_area(width,height):  
    return width*height  
  
print(square_area(2,2))
```

執行函式*

***函式參數接受不輸入參數值



函式 – 參數 - 1

關鍵字參數 Keyword Argument

```
def square_area(width,height):  
    return width*height  
# Solution 1  
square_area(width=2,height=2)  
# Solution 2  
square_area(2,2)
```

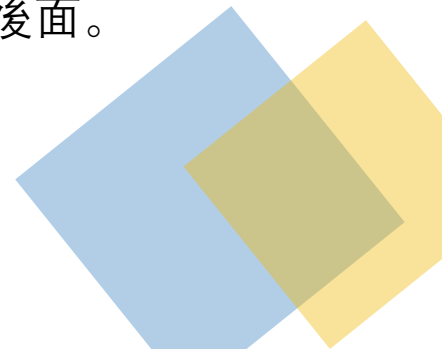
預設值參數 Default Argument

```
def square_area(width=2,height=2):  
    return width*height  
# Solution 1  
square_area(width=4,height=4)  
# Solution 2  
square_area(4,4)
```

預設值參數類型 Default Argument with Type

```
def square_area(width:int,height:int=2):  
    return width*height  
# Solution 1  
square_area(width=4,height=4)  
# Solution 2  
square_area(4,4)
```

***預設函式參數與關鍵字參數存在時，須將預設值參數排在後面。



函式 – 參數 - 2

Input: 【格式: *變數名稱】

```
def get_info_args(*args):  
    print(args)  
get_info_args('Hi', 'LawTech', 'Here')
```

Output: 【格式: Tuple】

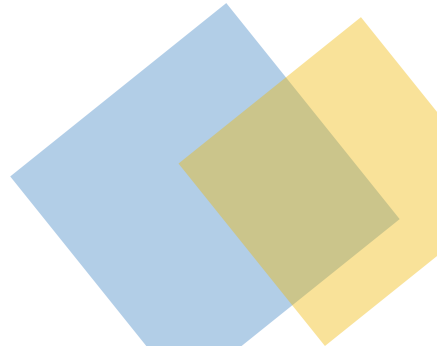
```
('Hi', 'LawTech', 'Here')
```

Input: 【格式: **變數名稱】

```
def get_info_kwargs(**kwargs):  
    print(kwargs)  
get_info_kwargs(width=2, height=2)
```

Output: 【格式: Dictionary】

```
{'width': 2, 'height': 2}
```



函式 – 種類

有回傳值

With Return Value(s)

```
def square_area_values(width,height):  
    return width*height  
x1 = square_area_values(2,2)  
print(f"回傳值: {x1}")
```

Output :

```
回傳值: 4
```

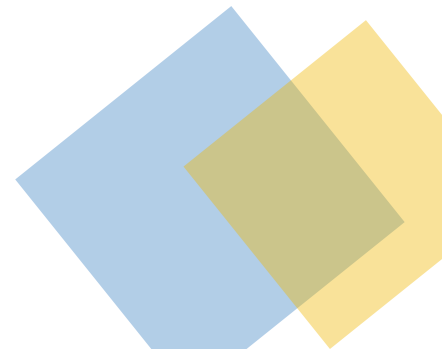
無回傳值

Without Return Value(s)

```
def square_area(width,height):  
    print(width*height)  
x2 = square_area(2,2)  
print(f"回傳值: {x2}")
```

Output :

```
4  
回傳值: None
```



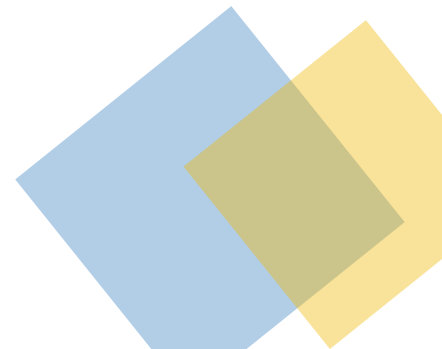
函式 – 變數範圍 – 區域變數

```
app.py x
app.py > calculate_number
1 def calculate_number():
2     # 區域變數
3     x = 100
4
5
6 print(x)
7
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL Code

```
[Running] python -u
"c:\Users\Shuan\Desktop\HelloPython\app.py"
Traceback (most recent call last):
  File "c:\Users\Shuan\Desktop\HelloPython\app.py", line 6,
    in <module>
      print(x)
NameError: name 'x' is not defined

[Done] exited with code=1 in 0.146 seconds
```



函式 – 變數範圍 – 全域變數

```
app.py x
app.py > calculate_number
1  # 全域變數
2  x = 100
3
4
5  def calculate_number():
6      global x
7      # 區域變數
8      x = 20
9
10
11 calculate_number()
12 print(x)
13
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Code

```
[Running] python -u
"c:\Users\Shuan\Desktop\HelloPython\app.py"
20

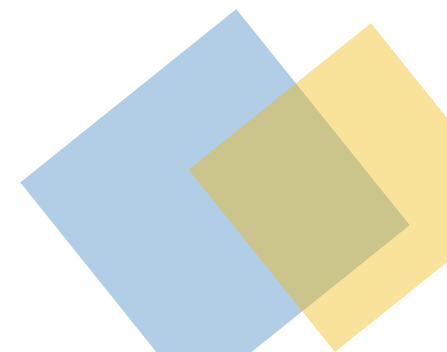
[Done] exited with code=0 in 0.258 seconds
```

函式－小考

請利用函式的概念，完成計算機之加、減、乘、除的功能，限時 10 分鐘！

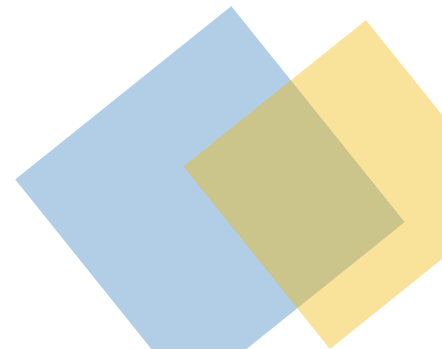
函式命名：

1. 加 > addition
2. 減 > subtraction
3. 乘 > multiplication
4. 除 > division



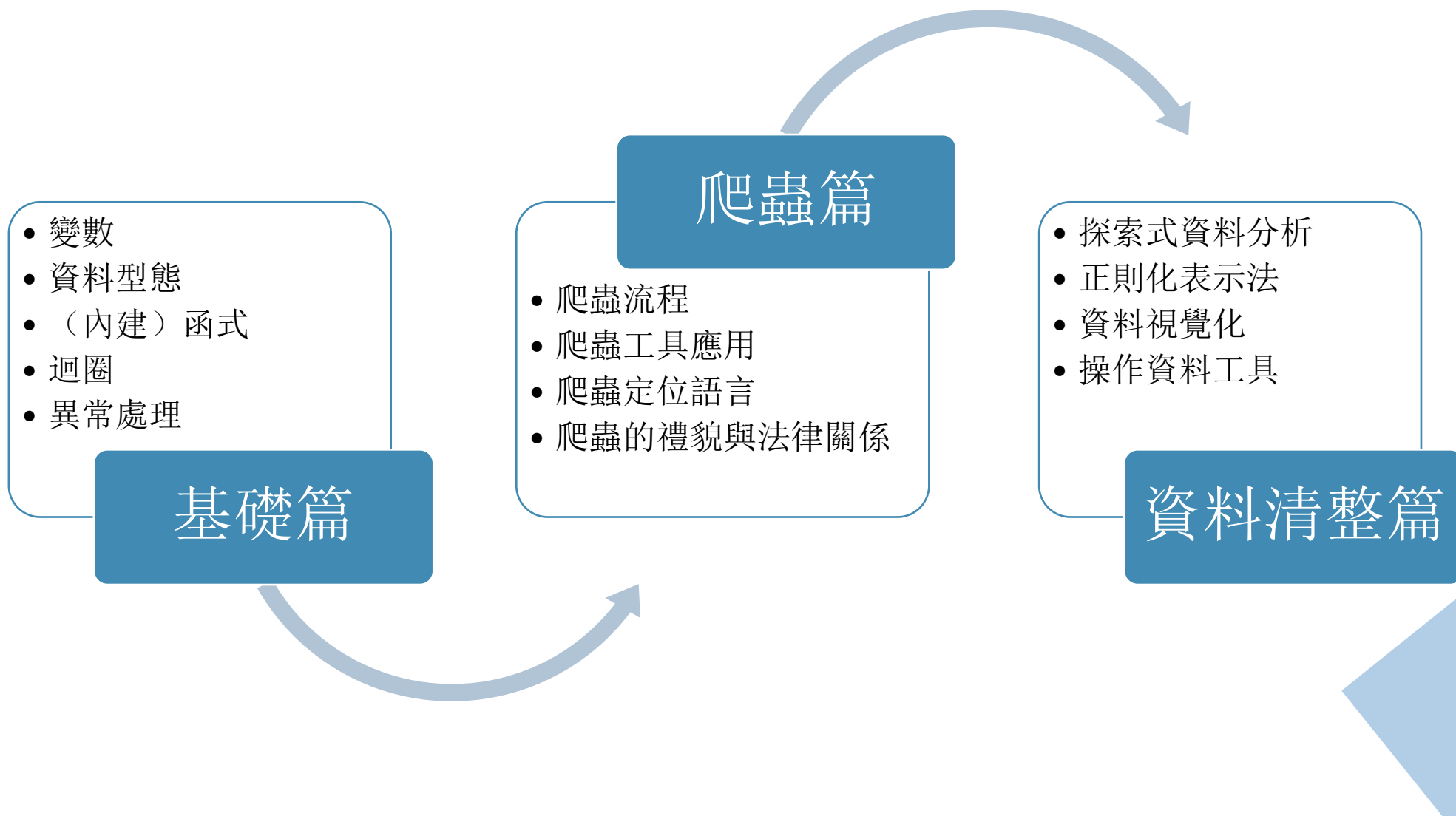
函式－小考

請利用加法函式，
完成在給予任意參數時，
皆可完成加總的任務，
限時 5 分鐘！



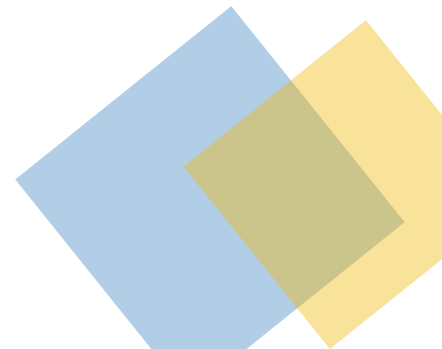
本學期 彙整

Python 學習歷程



專案題目

藉由證券詐欺歷史紀錄
建立損失賠償金額模型



專案歷程

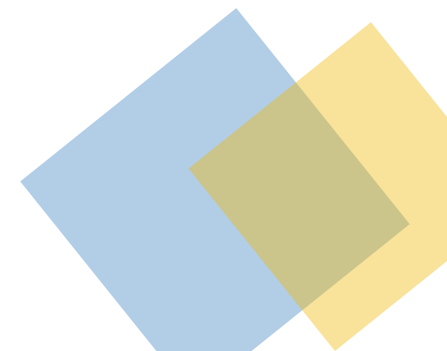
獲取資料



特征截取



模型建立

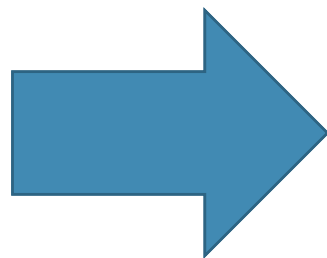


下學期的 旅程

活動時間

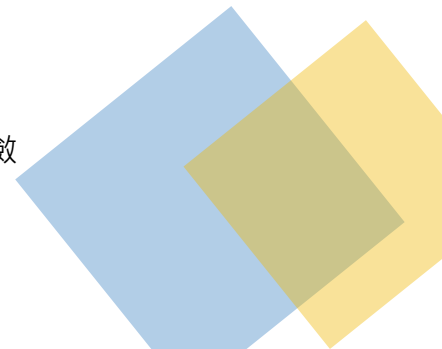


固定每週六上課



彈性上課

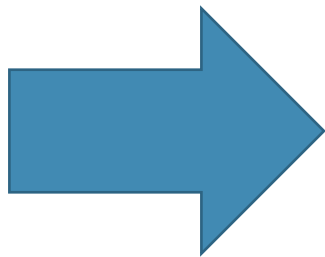
*** 前三週固定進行上學期專案收斂



教學方式

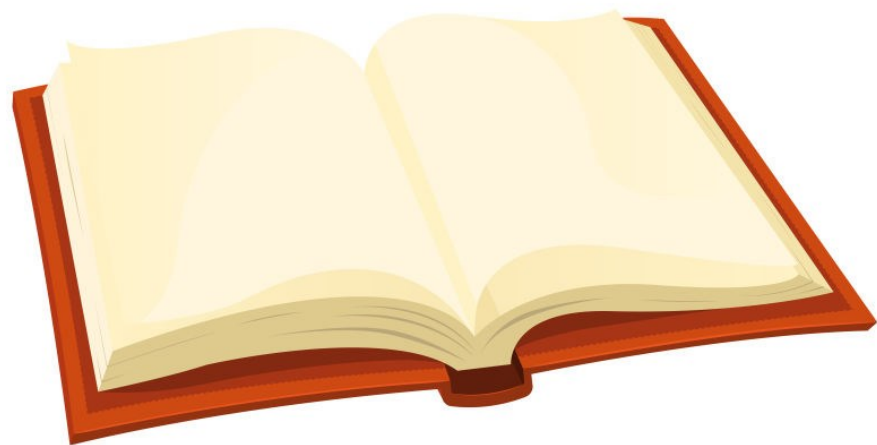


固定教學資源

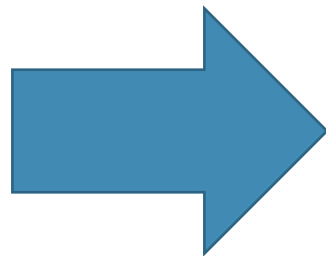


因材施教

專題形式



集體完成 1 個專題

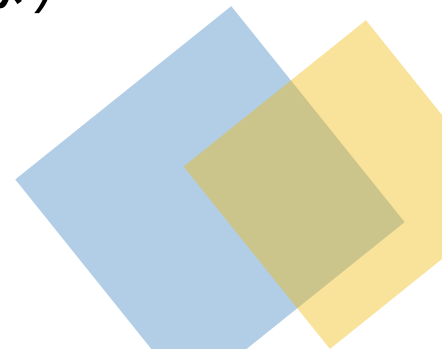


組別完成 1 個專題

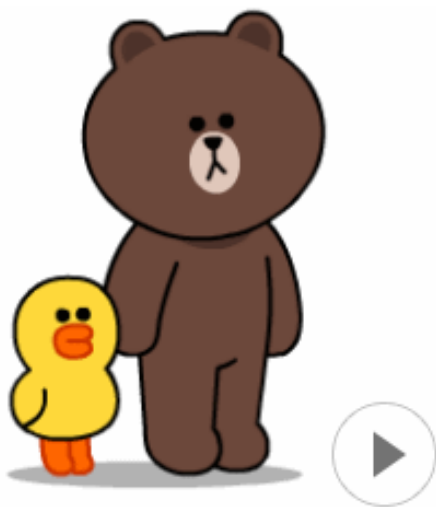
專題題目（待確認）

利用爬蟲歸納法律議題之多方見解

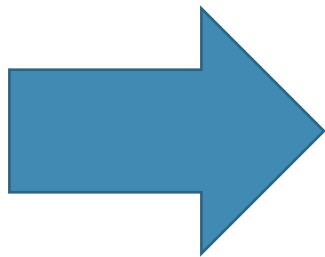
（例如：原因自由行為理論，可利用爬蟲去法學資料庫抓取相關文章整理學說見解；最後可對照爬蟲出來之結果能否與參考書之整理結果相對應）



呈現形式



成果默默無聞



成果發表

為何改變？

1. 程式學習需要的是**應用**場域，而不僅限在土法煉鋼。
2. 擁有程式能力是遠遠不足的，你需要的也包括**軟實力**。
3. 程式只是工具，若你不知道如何**定義**與**解決問題**，
最終問題還是石沉大海。
4. 程式須**熟能生巧**，才能發揮最大的效益。

P/S：其實是大家沒有共同的時間 🕒



還在等什麼？

專題視報名人數彈性調整分組，

若有任何疑問，

可私訊 SCU LawTech 粉專或 LINE 群組詢問！

！下個旅程是免費的~~~

SCU LawTech：<https://www.facebook.com/SCULawtech/>

掃描右側 QR Code，
加入我們下學期的專題實作團隊。



[109 年度 SCU LawTech 下個旅程團隊招募](#)



Q & A



THANKS