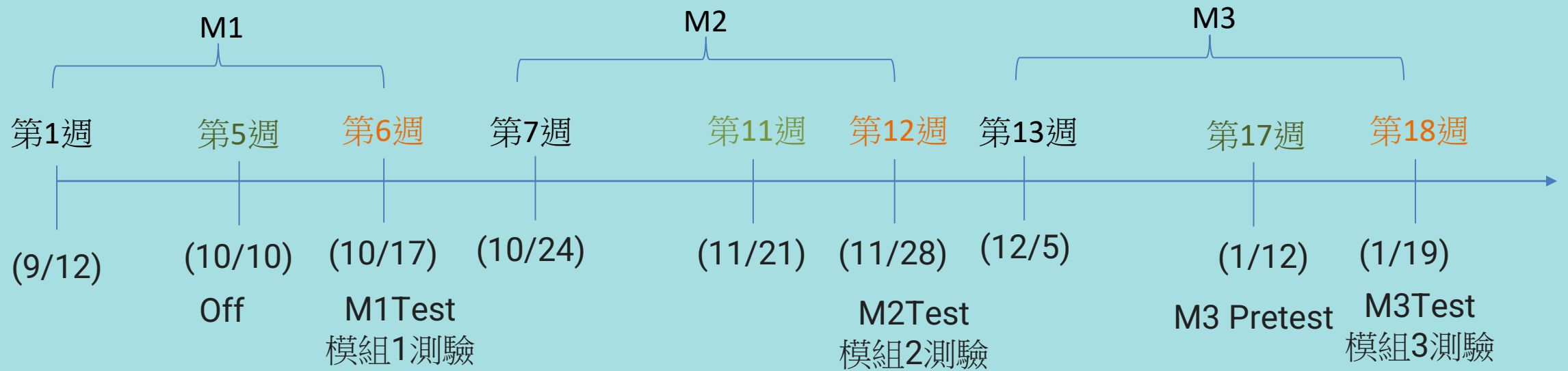




# 112-1基礎程式設計(3)

亞大資工系

# 課程進度



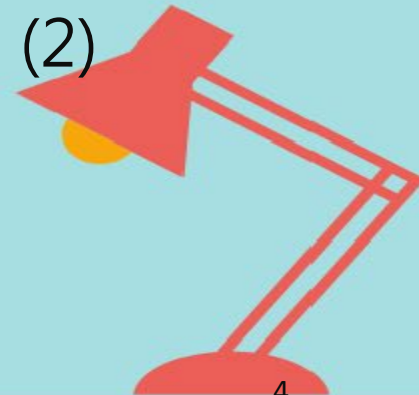
# 課程大綱

- W1-Python簡介及程式工具
- W2-變數和運算
- W3-迴圈和格式化輸出
- W4-判斷式和容器
- W5-字串處理和輸出入
- W6-M1測驗
- W07-字典容器
- W08-檔案處理
- W09-函數
- W10-進階流程控制
- W11-進階運算和生成器
- W12-M2測驗
- W13-進階函數
- W14-類別
- W15-進階類別
- W16-模組和套件
- W17-進階設計
- W18-M3測驗

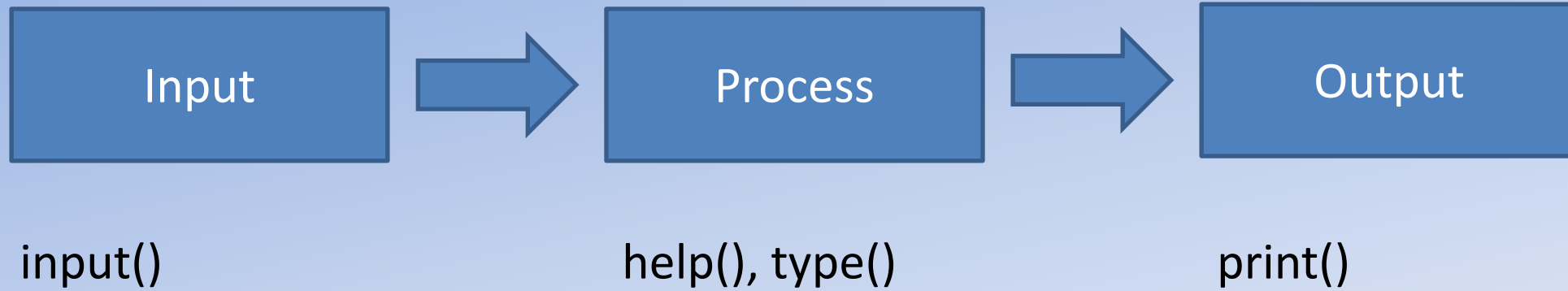


# 本週主題-迴圈和格式化輸出

- Essential-基本的
  - IPO model: input-process-output (輸入-處理-輸出)
    - Input: 序列(List), 字串切割(str.split()), len()函數
    - Process: for-loop
    - Process: 多重指定敘述(multiple assignment)
    - Output: f-格式化字串, print()的格式化輸出
  - 列印99乘法表
  - range()函數
- Advanced-進階的
  - 舊式字串格式化 (1) 百分比符號字串格式化(% string formatting) (2) str.format() 函數
  - 序列(List)的初始化和插入append()函數



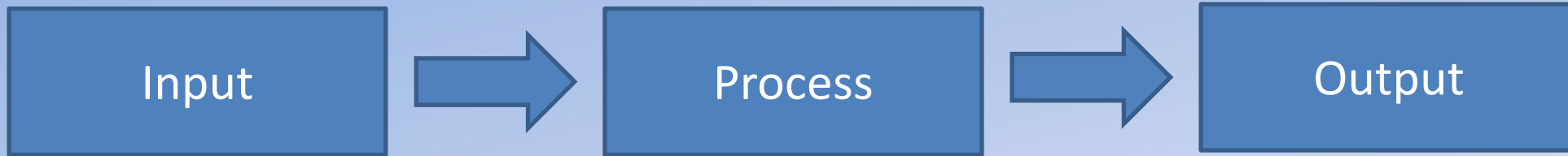
# IPO Model(1)



本章基本觀念是同學要知道：  
輸入用, 輸出用print()  
help()可以查看函數或類別的說明  
type()可以查看變數的型別



# IPO Model (2)



`input()`輸入一個變數  
使用`int()`轉換成整數變數  
使用`float()`轉換成浮點數變數

算術運算子  
運算子優先序  
程式中的字串表示  
註解

`print()`函數的參數`sep`和`end`  
Escape Sequence (逸出序列)

本章基本觀念是同學要知道：

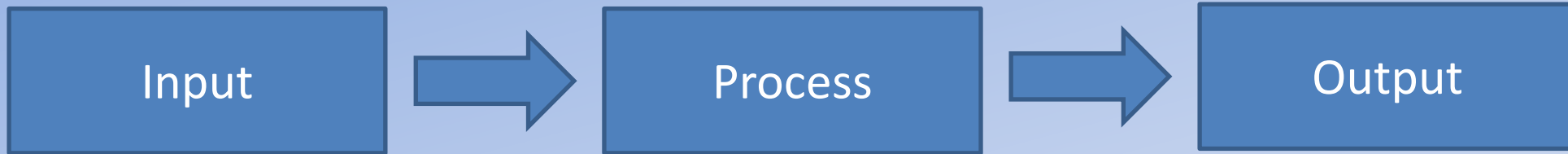
如何用`input()`輸入不同型別的數數

輸出`print()`有兩個參數`sep`和`end`來控制輸出

Python 的算數運算包括：加減乘除(+ - \* /), 次方(\*\*), 商(/)和取餘數(%)。加減乘除(+ - \* /), 次方(\*\*)的計算結果是浮點數。商(/)和取餘數(%)的計算結果是整數。



# IPO Model (3)



`input()`輸入多個變數  
使用`split()`轉換成串列

`for-loop`(迴圈),  
多重指定敘述  
索引和切片  
`range()`函數的使用  
串列(`List`)的初始化

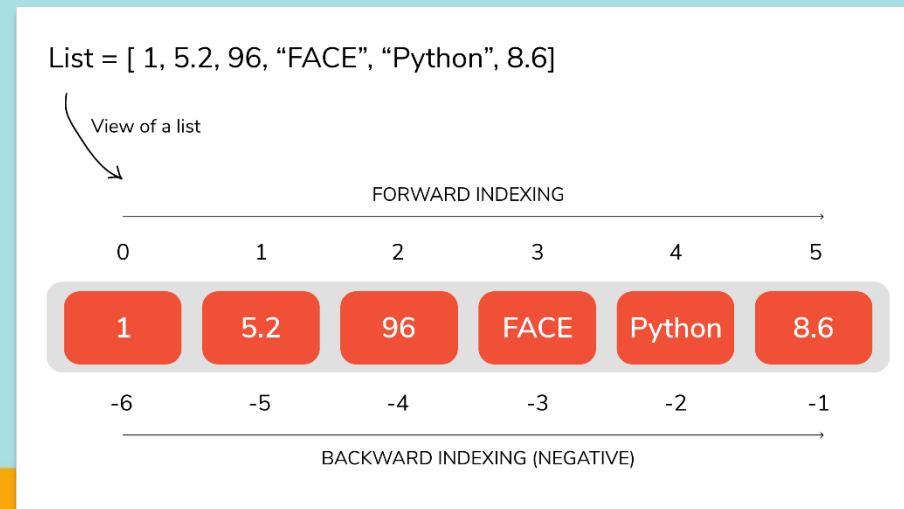
`print()`的格式化輸出  
f-格式化字串

本章基本觀念是同學要知道：  
如何用`input()`輸入多個變數  
用f-格式化字串的格式化輸出  
迴圈的使用。  
什麼是索引(`Indexing`)和切片(`Slicing`)



# Topic 1(主題1)-串列(list)

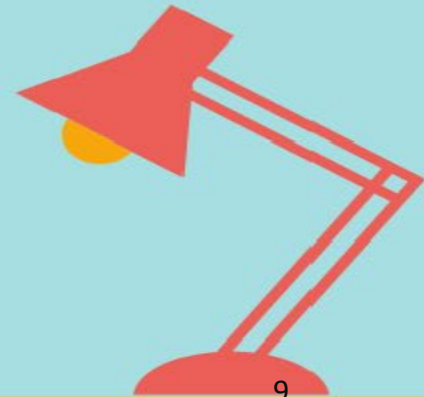
- 串列把一堆變數放入[]中，形成一排資料
- 串列變數是一堆變數的集合
- Step 1: 不使用 list 變數的一堆變數
- Step 2: 使用 list 變數當集合變數, 用索引鍵得到變數/indexing of list elements
- Step 3: 負索引鍵得到變數
- Step 4: list 的長度/Length of a list
- Step 5: 增加串列(list)的元素





# Topic 2(主題2)-字串切割(str.split())

- Step 1: 用空白間隔的子字串
- Step 2: 用split()切開子字串
- Step 3: 把input的字串用split()切開
- Step 4: 把input的字串用split()切開，再轉成整數



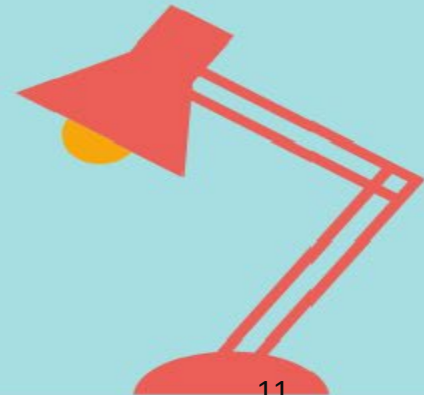
# Topic 3(主題3)-for迴圈/for-loop

- Step 1: 對list變數使用迴圈
- Step 2: 使用迴圈,印出1-9/print out the numbers 1-9



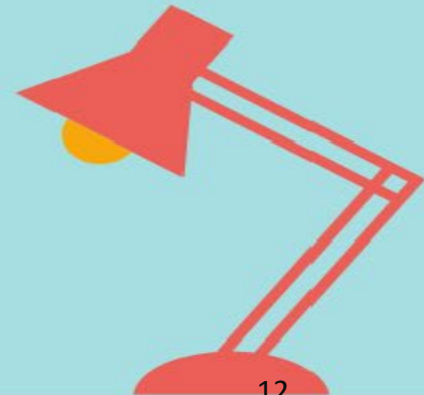
# Topic 4: 多重指定敘述(multiple assignment)

- Step 1: 各別指定
- Step 2: 多重指定(multiple assignment), 又稱為反打包(unpacking)
- Step 3: 部份反打包(unpacking)



# Topic 5: f-格式化字符串和print()的格式化输出

- Step 1: f-字符串
- Step 2: 使用f-字符串的格式化输出
- Step 3: 数字的格式化



# f-string格式化字串

A = 435; B = 59.058



```
print(f"Art:{A:5d}, Price per Unit: {B:8.2f}")
```

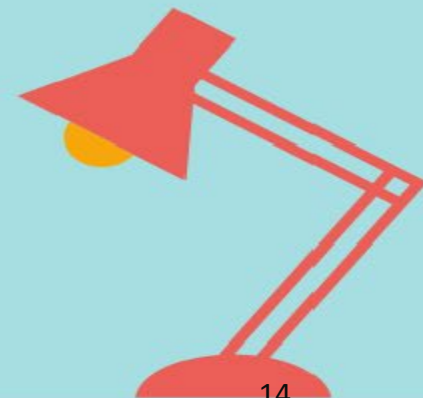
說明：

{A:5d} #將變數A以寬度5格印成整數字串

{B:8.2f} #將變數B以寬度8格印成小數2位的字串

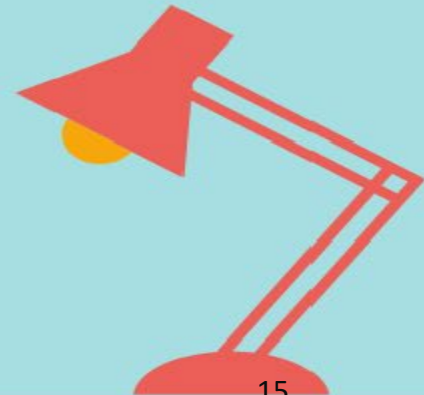
# Topic 6: 印出99乘法表

- Step 1: 使用1層迴圈,印出1-9
- Step 2: 使用2層迴圈,印出99乘法表



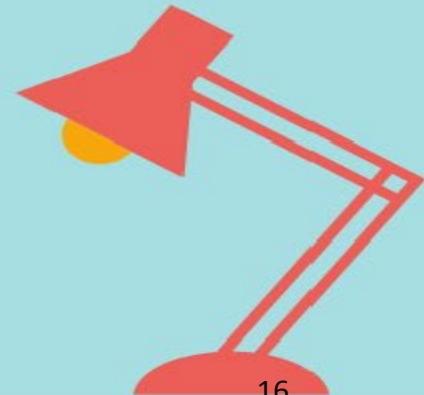
# Topic 7(主題7)-range()函數

- Step 1: parameters of range(): begin/end
- Step 2: parameters of range(): step
- Step 3: 使用range(),印出99乘法表



# Topic 8(主題8)-舊式字串格式化

- Step 1: %-formatting 格式化列印
- Step 2: str-format ( Python 2.6+ ) 格式化列印
- Step 3: f-string ( Python 3.6+ ) 格式化列印





# 格式化列印函數

說明：

**%-formatting** 格式化列印

```
print('Art: %5d, Price per Unit: %8.2f' % (435, 59.058))
```

**str-format ( Python 2.6+ )** 格式化列印

```
print("Art: {0:5d}, Price per Unit: {1:8.2f}".format(435, 59.058))
```

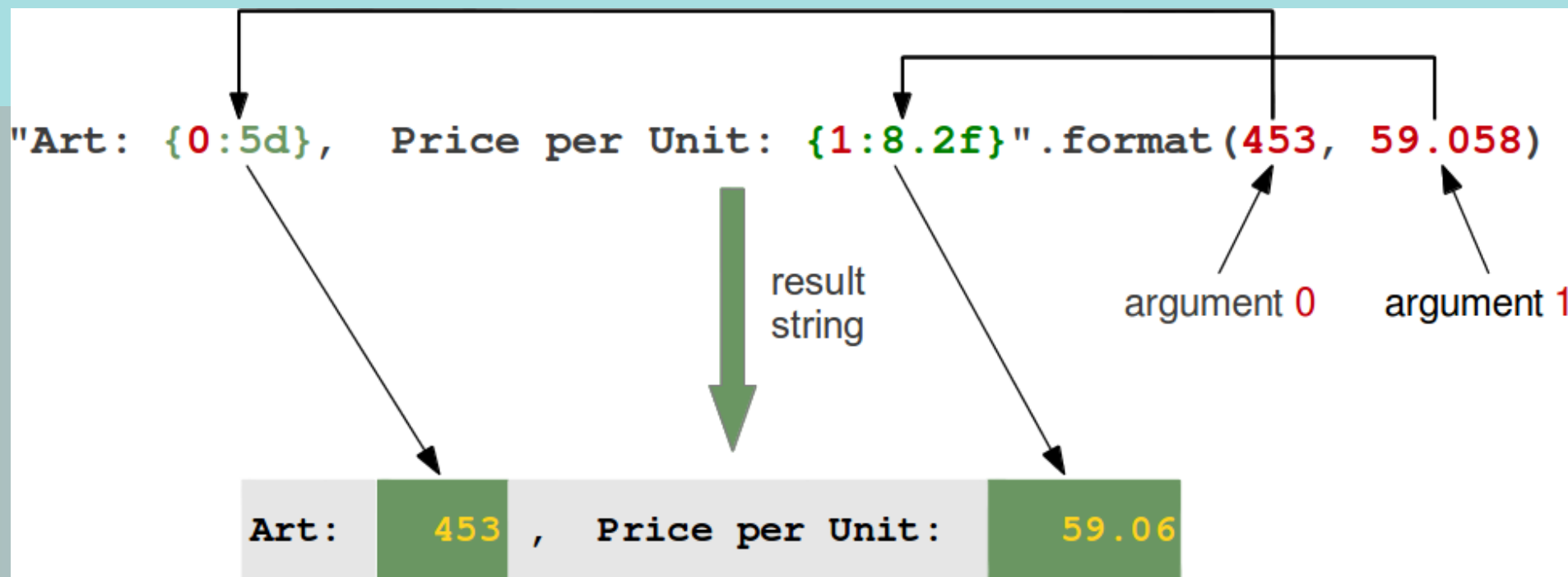
**f-string ( Python 3.6+ )** 格式化列印

A = 435; B = 59.058

```
print(f"Art:{A:5d}, Price per Unit: {B:8.2f}")
```

# str-format格式化字串

Var = "{} {}".format(varA, varB)



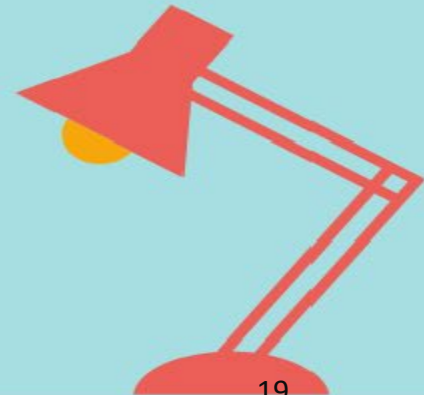
說明：

`{0:5d}` #將後面第0個變數以寬度5格印成整數字串

`{1:8.2f}` #將後面第1個變數以寬度8格印成小數2位的字串

# Topic 9(主題9)-串列(List)的初始化和解析式

- Step 1: 條列初始化
- Step 2: 使用串列解析式(List Comprehension)來初始化





Thanks!

Q&A

