

Topic03_學習目標

- 迴圈、判斷式及其組合
- 次數有限的重覆動作：迴圈 (for...)
- 次數無限的重覆動作：迴圈 (while...)
- 類似的動作：判斷式 (if...elif...else...)
- 重覆的類似動作：迴圈 + 判斷式！

for ... 迴圈：次數有限的重複動作

- Santa prepares a toy for **every kid** in **this room**.
- Santa 對房間裡的每個孩子都準備了玩具

for everyKid **in** thisRoom:

Santa prepares a toy

縮排 4 個空格

既然有 "in"，那麼就直接把 "thisRoom" 視為是某種「容器」或是「可以裝東西的空間」吧。

- Santa prepares a toy... 重複的動作
- for every kid in this room... 孩子的數量有限

for ... 迴圈 - 範例

- 對一個字串中的每一個符號，都以 print 處理

```
story = "He broke the glass."
```

```
for element in story:
```

```
    print(element)
```

把 story 視為某種內含字串的「容器」

- 都以 print() 處理… (重覆的動作)
- 對字串中的每一個符號… (符號的數量有限)
- for.. 迴圈的意思：
 - 對某個容器中的每個元素，都做以下的動作…

for... 迴圈 - 範例

- 「容器」不只有字串而已，容器也可以是「列表」（或其它…）
wordList = ["he", "borke", "the", "window", "glass", "last", "night."]
for element **in** wordList:
 print(element)
- 都以 print() 處理…（重覆的動作）
- 對列表中的每一個成員…（成員的數量有限）
- **for.. 迴圈的意思：**
 - 對某個容器中的每個元素，都做以下的動作…

for... 迴圈 - 實作

- 對一個列表中的每一個元素，都以 print 處理
 - `wordOneList = ["He", "broke", "the", "glass."]`
- 對一個列表中的每一個元素，都印出它的長度
 - `wordTwoList = ["He", "broke", "the", "glass", "."]`
- 對一個列表中的每一個元素，都印出它的型態
 - `wordThreeList = ["He", "broke", 3, "glass."]`

for ... 迴圈：用 range() 另解

- range(x, y)：是一種小工具；提供從 x ~ y 之間的數字，並包裝成一個列表。
 - numList = range(0, 10)
 - print(numList)
- 應用：把這些數字當做 index 使用！
 - | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|
| H | e | | b | r | o | k | e | | t | h | e | | g | l | a | s | s | . |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
 - story[0]
 - story[1]
 - story[n]

for... 迴圈 - 另解

- 回顧第一個例子：

- 對一個字串中的每一個符號，都以 `print()` 處理

```
story = "He broke the glass."
```

```
endding = len(story)
```

```
for n in range(0, endding):  
    print(story[n])
```

可以被視為...

```
for n in [0, 1, 2, 3]:  
    print(story[n])
```

比較一下

```
for element in story:  
    print(element)
```

兩種 for... 迴圈：語言上的應用

- 兩種 for... 迴圈的適用場景：

wordList = ["He", "broke", "the", "glass"]

- 提供「容器」給 for... 迴圈：

```
for x in someList:
```

```
    print(x)
```

- 提供「index」給 for... 迴圈：

```
for n in range(x, y):
```

```
    print(n-1, n, n+1)
```

```
    print wordList[n-1], wordList[n], wordList[n+1]
```

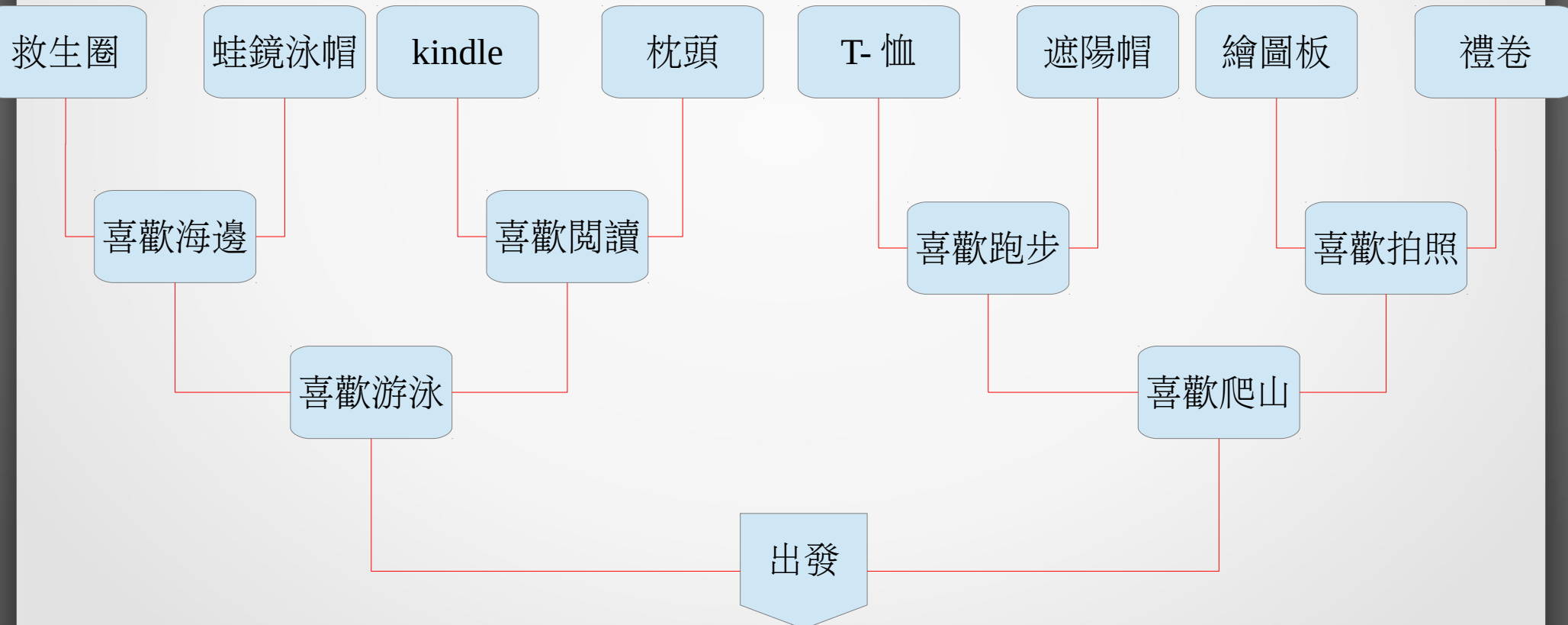

第二個 for... 迴圈：實作

- 對列表中的每個成員，都印出自己及下一個。
 - `wordOneList = ["He", "broke", "the", "glass."]`
- 對列表中的每個成員，都印出自己及下一個的長度。
 - `wordTwoList = ["He", "broke", "the", "glass", "."]`
- 對列表中的每個成員，都印出自己及上一個的型態。
 - `wordThreeList = ["He", "broke", 3, "glass."]`

if...elif...else... 判斷式

- 向左走向右走

朋友的生日禮物怎麼選呢？



if...elif...else... 判斷式

- 牛刀小試

- 請使用者輸入身高。如果身高超過 130，印出「請購票後進入遊樂區」；否則的話，印出「免購票，但請由大人陪同進入遊樂區。」

```
height = raw_input(" 請輸入你的身高： ")
```

```
if height > 130:
```

```
    print(" 請購票後進入遊樂區 ")
```

```
else:
```

```
    print(" 免購票，但請由大人陪同進入遊樂區。 ")
```

if...elif...else... 判斷式

- 牛刀小試之二

- 請使用者輸入一個英文動詞的進行式。如果這個字是進行式的話，印出「這個字的原型應該是…」及該字的原型；否則的話，再試試這個字是否為過去式，若是的話，印出「這個字的原型可能是…」，如果都不是的話，印出「這個字大概不是動詞吧！」

```
Verb = raw_input(u"請輸入一個英文動詞的進行式或過去式… ")
```

```
if Verb.endswith("ing"):
```

```
    print(Verb + u" 的原型應該是… " + Verb[0:-3])
```

```
elif Verb.endswith("ed"):
```

```
    print(Verb + u" 的原型可能是… " + Verb[0:-2])
```

```
else:
```

```
    print(Verb + u" 這個字大概不是動詞吧！ ")
```

if...elif...else... 判斷式

- 實作

- 請開啟 Amis.txt 並用一個 if 判斷式來檢查「Amis」中是否含有 "ka" 或是 "ga" 這兩個音。
- 如果有 "ka" 的話，印出 "Amis 有 ka !"，
如果有 "ga" 的話，印出 "Amis 有 ga !"，
如果都沒有的話，印出 "Amis 沒有這兩個音！"

迴圈加上判斷式

- 概念

- for... 迴圈讓我們可以對一個物件的內容「依次」處理。
- if...elif...else... 判斷式讓我們可以對一個物件「依判斷內容」而有不同的處理。
- 若能結合在一起...

```
for kid in thisRoom:
```

```
    if kid == "good kid":
```

```
        Santa prepares a toy
```

```
    else:
```

```
        Santa ignores him/her.
```

迴圈加上判斷式

- 試作

- 如果身高超過 130 ，印出「請購票後進入遊樂區」；否則的話，印出「免購票，但請由大人陪同進入遊樂區。」
- heightList = [120, 111, 130, 143, 171, 167, 140]
 for h in heightList:
 if h > 130:
 print(h, u" 請購票後進入遊樂區 ")
 else:
 print(h, u" 免購票，但請由大人陪同進入遊樂區。 ")

字串與列表

- 進階操作

- **string** 當主詞的時候，可以做的事情有…
 - string.split(patternString)
 - string.replace(oldString, newString)
 - string.decode(codeNameString)
- **list** 當主詞的時候，可以做的事情有…
 - list.append(item)
 - list.insert(index, item)
 - list.remove(item)
 - list.pop(index)
 - list.sort()

字串與列表

- 進階操作

- **list** 當主詞，在 prompt 裡試試以下動作…
 - nonVerbList = ["He", "the", "glass"]
 - nonVerbList.append("blue")
 - nonVerbList.append(7)
 - nonVerbList.insert(1, "window")
 - nonVerbList.insert(4, "door")
 - nonVerbList.remove("man")
 - nonVerbList.remove(7)
 - nonVerbList.pop()
 - nonVerbList.sort()

while... 迴圈

- 天長地久有時盡，此 while 綿綿無絕期…

- 大部份的問題都能用 for... 迴圈解決。
- while... 迴圈外面一定要先設一個「起始值」！

```
harryPotterFile = open("Harry Potter.txt", "r")
wordList = harryPotterFile.read().split(" ")
HarryCounter = 0 # 起始值
harryVerbList = []
while HarryCounter < 1000: # 終止條件
    if wordList[0] == "Harry":
        harryVerbList.append(wordList[1])
        wordList.pop(0)
        HarryCounter = HarryCounter + 1 # 向「終止條件」靠近一步
    else:
        wordList.pop(0)
```

課後練習 1-1, 1-2

- 請建立一個名為 "homework.py" 的 python 程式。
- 在 homework.py 中加入適當的程式碼，讓程式執行以後可以達成以下要求：
 - 1-1: (完成者，本次作業分數 75 起計)
 - 讀取同目錄下名為 "sample.txt" 的純文字檔案。
 - 將 sample.txt 的內容長度超過 5 的詞彙加入一個名為 sampleWordList 的列表中並顯示出來。
 - 1-2 : (完成者，本次作業分數 80 起計)
 - 做完以上動作以後，再加一個 while... 迴圈，讓使用者輸入任何五個字母以上的英文字，然後告訴使用者剛才輸入的字，是否在 sampleWordList 裡！

本節完成目標

- 迴圈、判斷式及其組合
- 次數有限的重覆動作：迴圈 (for...)
- 次數無限的重覆動作：迴圈 (while...)
- 類似的動作：判斷式 (if...elif...else...)
- 重覆的類似動作：迴圈 + 判斷式！