

決策主題（一）：電腦懂得真與假

by 田弘華 Hung-Hua Tien

1 布林值

我們把真(True)和假(False)稱為布林值(Boolean value)。

- 這是以英國數學家George Boole命名。
- 他在1800年代中期所發明的布林代數是現代數位電腦邏輯的基礎。
- 電燈開與關、投影機螢幕放下與升起、麥克風開與關...等，生活上有許多布林的例子。

1-1 真值與假值

- 布林只有 True 與 False 這兩個值。
- Python對於英文的大小寫是敏感的（case-sensitive）。
 - 只有True 會被識別為布林
 - 但是 TRUE 或者 true 會被視作物件名稱。

```
# 布林只有 True 與 False 這兩個值。
True
False
print(type(True))
print(type(False))

# 大小寫不同
true
false
TRUE
FALSE
print(type(true))
print(type(false))
print(type(TRUE))
print(type(FALSE))
```

1-2 建立布林值與數學運算

(1)建立布林資料

```
# 方法（一）：直接鍵入True或False
print(True)
print(False)

# 方法（二）：用bool()函式建立。
print(bool(1))
print(bool(0))
```

(2)布林值可視為數值資料做數學運算

- 在 Python 中，True 跟數值 1 相等；False 跟數值 0 相等。
- 在數值運算中納入了布林不會產生任何問題。

```
# True 跟數值 1 相等； False 跟數值 0 相等。  
# 數值運算中可以納入了布林。  
print(False + 3) # False為0  
print(True + 3) # True為1
```

2. 關係條件比較

電腦可以決定一個敘述是真(True)還是假(False)。

- 如「24是一個偶數」這個敘述是True。
- 又如「25是一個偶數」這個敘述是False。

我們可以透過條件判斷式得到布林值。當進行條件判斷或者資料篩選時，我們需要仰賴布林值。

2-1 布林條件判斷式

有True或False的任何敘述式，稱為布林條件判斷式。

(1)用兩個等號來比較兩個項目。如果相同結果值為True，反之則是False。

例：變數player_score預設為0，把它和0比較，Python回傳True。

```
player_score = 0 # 這會把0放入變數Player_score  
player_score == 0 # 這會檢查player_scor裡的值是否為0
```

(2)檢查文字字串會區分大小寫。

例：變數name儲存Z大寫的Zoe，並與z小寫的zoe比較，檢查後回傳False。

```
name = "Zoe" # 將變數name設為值Zoe  
name == "zoe"
```

(3)用其他運算元來檢查真假

```
# 數字  
5 > 2 # 檢查5是否大於2  
5 < 2 # 假查5是否小於2  
5 != 2 # 檢查5是否不等於2  
  
# 文字  
chocie = "yes"  
choice == "yes"  
choice != "quit" # 檢查choice是否不等於quit
```

2-2 關係運算子

常見的Python關係運算子共有六個==、!=、>、>=、<、<=。

表 4-3.1 邏輯運算子的相關運算

運算子	功能	例子	說明
1	==	x==y	若 x 等於 y，則結果為真
2	!=	x!=y	若 x 不等於 y，則結果為真
3	<	x<y	若 x 小於 y，則結果為真
4	>	x>y	若 x 大於 y，則結果為真
5	<=	x<=y	若 x 小於等於 y，則結果為真
6	>=	x>=y	若 x 大於等於 y，則結果為真

關係運算子所構成的運算式稱之為**條件運算式**。當用來比較兩者之間的關係，結果有 True（真）或是 False（假）兩種。

```
# 布林（一）：比較關係運算子 = 比較運算子 = 關係運算子
## 比較運算子用於比較兩數的大於、小於和等於關係。

print(8 == 7) # 判斷 8 是否等於 7，請注意不要混淆 = 和 ==。
print(8 != 7) # 判斷 8 是否不等於 7
print(8 > 7) # 判斷 8 是否大於 7
print(8 >= 7) # 判斷 8 是否大於等於 7
print(8 < 7) # 判斷 8 是否小於 7
print(8 <= 7) # 判斷 8 是否小於等於 7
```

```
# 資料篩選
## 電腦評分是否高於8分

movie_rating = 8.8
print(movie_rating > 8.0)
print(type(movie_rating > 8.0))
```

隨堂練習：身分證字號尾數是否為奇數？

```
id_last_digit = input("請輸入您身分證字號的尾數：")
ans = int(id_last_digit) % 2 == 1
print("身分證尾數除以2的餘數是否為1：", ans)
```

3. 邏輯的運算

～我們常將不同的布林值結合為新的條件式來幫助我們做決定。

電腦能結合True或False的布林表示式，來建立新的布林表示式。

3-1 基本觀念

例：帶雨傘 **Take an umbrella.**

(1)且

如果我們知道「在下雨」是**True**，（且）知道「我有雨傘」也是**True**，則我們知道在下雨時出門可以帶雨傘。因此「帶雨傘」只有在「兩個」敘述式都為**True**時才為**True**。

(2)或

如果我們知道「風大」是**True**，（或）知道「很冷」是**True**，則可以決定是否帶外套。如果「風大」或如果「冷」，或如果「同時風大又冷」，我們都會帶外套。因此「帶外套」在這兩個敘述式任一或同時為**True**時為**True**。

(3)非真

如果我們知道「外面溫暖」是**Fasle**，（非真），則可以決定帶外套。因此，「帶外套」在「外面溫暖」為**False**實為**True**，即為「相反」。

3-2 邏輯運算子

～可以使用邏輯運算子連結多個條件。

邏輯運算子有三種運算子，**and**（而且）、**or**（或）、**not**（相反）。

- 若要多個條件須同時為**True**，結果才會為 **True** 時，就使用「**and**」結合這些條件；

1. x and y：

當只有 x、y 都是 **True**，結果才會是 **True**；x 與 y 只要其中一個為 **False**，結果為 **False**。

變數 x	變數 y	結果
True	True	True
True	False	False
False	True	False
False	False	False

X and Y	Y=True	Y=False
X=True	True	False
X=False	False	False

- 若只要其中之一條件為**True**，結果就為 **True**，就使用「**or**」結合這些條件；

2. x or y :

當 x 與 y 其中一個或兩者都是 True，則結果為 True；當 x 與 y 兩者都是 False，則結果為 False。

變數 x	變數 y	結果
True	True	True
True	False	True
False	True	True
False	False	False

X or Y	Y=True	Y=False
X=True	True	True
X=False	True	False

- 假如要相反的結果就使用「not」，「not」必須置於該條件的前面。

3. not x :

若 x 為 True，not x 的結果為 False；若 x 為 False，not x 的結果為 True。

變數 x	結果
True	False
False	True

	not X
X=True	False
X=False	True

例1: 布林：邏輯運算子

```
# 邏輯運算子有三種運算子:且(and)、或(or)、非(not)。  
  
print(True and False) # 判斷 True 與 False 的交集  
print(True and True)  
print(False and False)  
print(False and True)  
  
print(True or False) # 判斷 True 與 False 的聯集  
print(True or True)  
print(False or False)  
print(False or True)  
  
print(not True) # False  
print(not False) # True  
print(not(8 > 7)) # 反轉 8 是否大於 7 的判斷
```

例2: 布林：邏輯運算子

舉例	X 值	結果	說明
((X>60) and (X<80))	70	True	條件 (70>60) 為 True，而 條件 (70<80) 為 True，經由 and(且) 運算結果為 True
((X>60) and (X<80))	60	False	條件 (60>60) 為 False，只要有一個條件 False，經由 and(且) 運算結果就為 False
((X>60) or (X<80))	60	True	條件 (60<80) 為 True，經由 or(或) 運算只要有一個條件為 True，結果就為 True。
not(X>60)	60	True	條件 (60>60) 為 False，取 not(非) 運算，結果變成 True。

3-3 成員與身份算子

(1) `[x in y]` 判斷x是否為y其中一個元素。

```
# 文字
print("H" in "Hello world") # 判斷H是否存在於Hello world 之中

# 數字串列
x = 1
y = [1,2,3]
print(x in y)
print(x not in y)
```

(2) `[x is y]` 判斷x與y的id是否相同（在此不是相等==的判斷）。

```
# 數字
print(8 is 7) # 判斷 8 與 7 是否為同樣的類別或值
print(8 is not 7) # 判斷 8 與 7 是否為相異的類別或值

print(5566 is 5566.0) # False
print(5566 is not "5566") # True
```

範例:變數在電腦記憶體位置是否相同的判斷

```
x = [1,2,3]
y = [1,2,3]
print(id(x),id(y)) #x 與y 的位址不同
print(x is y) #x與y在電腦有不同的記憶空間，所以x是x、y是y。
print(x == y) # x 與y 的值相等，True
```

執行結果說明

- 變數x 參考整數串列「1, 2, 3」時，id(x) 為5192360，
- 變數y 參考到整數串列「1, 2, 3」時，id(y) 為5156576，
- 因x 與y 的id 不同，所以「x is y」的結果為False，因x 與y都是整數串列「1, 2, 3」，所以「x == y」的結果為True。

隨堂練習：請問

(1)若x = "Python", y = "Python"，則(x is y) 和 (x==y)的結果是True還是False?

(2)若x = "Python", y = x，則(x is y) 和 (x==y)的結果是True還是False?

(3) 二者答案為什麼一樣或不一樣？

```
# 觀念補充
## 不可變物件的答案是True，位址相同
x = "Python"
y = "Python"
print(id(x),id(y)) #x 與y 的位址相同
print(x is y)
print(x==y)

x = "Python"
y = x
print(id(x),id(y)) #x 與y 的位址相同
print(x is y)
print(x==y)

x = (1,2,3)
y = (1,2,3)
print(id(x),id(y)) #x 與y 的位址相同
print(x is y)
print(x == y)

## 可變物件的答案是False，位址不同
x = [1,2,3]
y = [1,2,3]
print(id(x),id(y)) #x 與y 的位址不同
print(x is y) #x與y在電腦有不同的記憶空間，所以x是x、y是y。
print(x == y) # x 與y 的值相等，True

x = {1,2,3}
y = {1,2,3}
print(id(x),id(y)) #x 與y 的位址不同
print(x is y) #x與y在電腦有不同的記憶空間，所以x是x、y是y。
print(x == y) # x 與y 的值相等，True
```

Summary

True和False在Python被視為布林值，且區分大小寫。

當進行判斷條件或者資料篩選的時候，我們需要仰賴布林值。除了直接輸入True或False之外，我們可以透過條件判斷式得到布林值。

常用的布林運算符號

I.比較運算子 = 關係運算子

(1)== 、 != ：等於以及不等於

(1)> 、 >= 、 < 、 <= ：大於、大於等於、小於以及小於等於

II. 邏輯運算子

(2)and 、 or ：交集與聯集

(2)not ：非

III. 成員與身份運算子

(3)is 、 is not ：是否為相同的值與類型

(3)in ：是否存在於

課後習題：時尚偵測師

時尚達人Cher 和 Dionne要參加一個華麗的派對。他們想確保他們的穿搭與眾不同，但也有想保留一點共同的地方，比如兩人都喜歡粉紅色。讓我們寫一些程式，確保兩人的穿搭都個性十足！

～決定兩位的穿著

```
cher_dress_color = "pink"
cher_shoe_color = "white"
cher_has_earrings = True
dionne_dress_color = "purple"
dionne_shoe_color = "pink"
dionne_has_earrings = True
```

～時尚偵測:穿搭檢查

1. 至少有一人穿紫色 At least one person is wearing purple? True
2. 兩人穿不同顏色的衣服Cher and Dionne hve different dress colors.
3. 兩人都戴耳環Cher and Dionne are both earing earrings.
4. 至少有一人穿粉紅色 At least one person iw wearing pink.
5. 沒人穿綠色。 No one is wearing green.
6. 兩人穿同顏色的鞋子 Cher and Dionne have the same shoe color.

```
print (f'Both girls have different dress colors? {cher_dress_color != "purple" and dionne_dress_color != "pink"}')
print(f"Both girls are waering earrings? {cher_has_earrings == True and dionne_has_earrings == True}")
print(f'At least one person is wearing pink? {cher_dress_color == "pink" or dionne_dress_color == "pink"}')
print(f'No one is wearing green? {cher_dress_color != "green" and dionne_dress_color != "green"}')
print(f'Both girls have the same shoe colors? {(cher_shoe_color=="pink" and dionne_shoe_color == "pink") or (cher_shoe_color == "white" and dionne_shoe_color == "white")}')

```


補充說明：布林值的三大應用場景 - 條件選擇、條件迴圈與資料篩選

```
# 用途一：if 條件選擇
last_id_digit = input('請輸入身分證字號的尾數:')
last_id_digit = int(last_id_digit)

remainder = last_id_digit % 2
if remainder == 0: # bool
    ans = "偶數"
else:
    ans = "奇數"
print("身分證字號尾數除以2的餘數為:{}, 因此為{}".format(remainder, ans))

# 用途二：條件迴圈：while 迴圈
i = 65
while i <= 90: # bool
    print("Character: {}, Order: {}, Binary: {}".format(chr(i), i,
    bin(i)))
    i += 1

# 用途三：資料篩選
member_ages = [18, 20, 22, 25, 35, 42, 65]
list(filter(lambda x: x <= 25 and x >= 18, member_ages))
```

4. 理論：范恩圖（文氏圖）

4-1. 一個集合的文氏圖

- 字集合 **U (universal set)**：以方塊表示，代表所有需要討論的元素。
- 集合 **A(set)**：以圓圈表示，代表擁有相同特性的元素，如學生。
- 空集合 (**empty set**)：以符號 \emptyset 表示，代表沒有元素的集合。

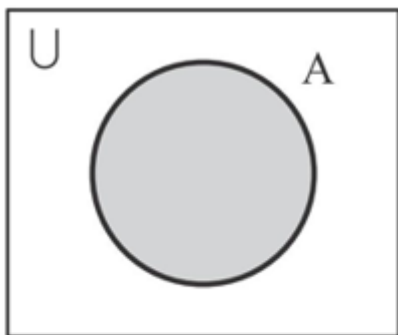


圖 3-1

- 補集合 (**complement set**)：通常以符號 S^c 表示，S的補集合。代表所有不是學生(教師，職員等)的元素。

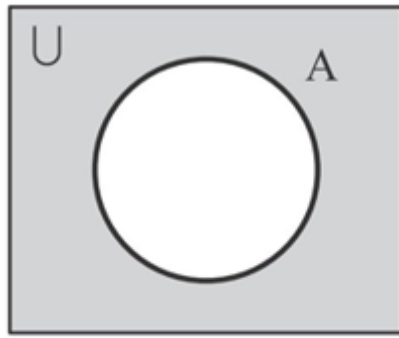


圖 3-2

4-2 兩個集合的文氏圖

(1) 兩集合 (A, B) 的一般化表示如下

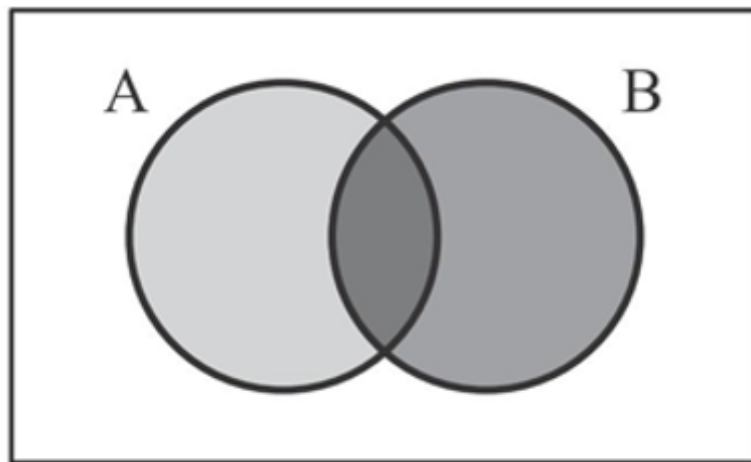


圖 3-3

範例：

- A：演唱家，B：作曲家
- A：金卡客戶，B：本日消費大於5千元的客戶

(2) 兩個集合的文氏圖特例（一）：包含

若兩集合 (類別) 有包含關係 (A 包含 B)，則下左圖中右邊會是空集合 \emptyset 。

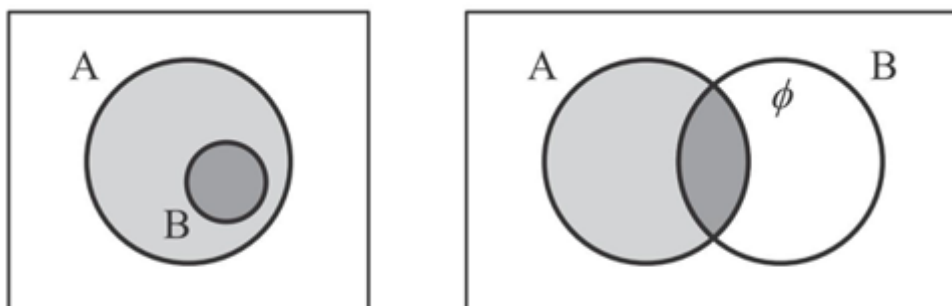


圖 3-4

敘述若為：B 是 A 的形式，就隱含了包含關係。

- 鳳梨 是 水果。

範例：A：汽車，B：雙門跑車

範例：A：客戶，B：金卡客戶

兩個集合的文氏圖特例（二）：互斥

(3)若兩集合 (類別) 若沒有交集 (A,B互斥)，則??會是空集合。

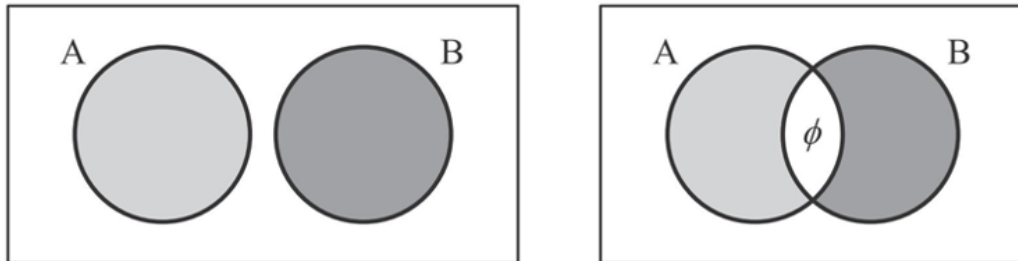


圖 3-5

範例：A：金卡客戶，B：銀卡客戶 (都在客戶集合中)

範例：A：大學生，B：研究生 (都在學生集合中)

4-3 集合的運算

(1)聯集 (union)：

$A \cup B = \{x \in U \mid x \in A \text{ 或 } x \in B\}$ 。

- 集合A表示：喜歡喝咖啡的人
- 集合B表示：喜歡喝茶的人
- $A \cup B$ 表示：喜歡喝咖啡或茶的人
- $(A \cup B)^c$ 表示：不喜歡喝咖啡也不喜歡喝茶的人

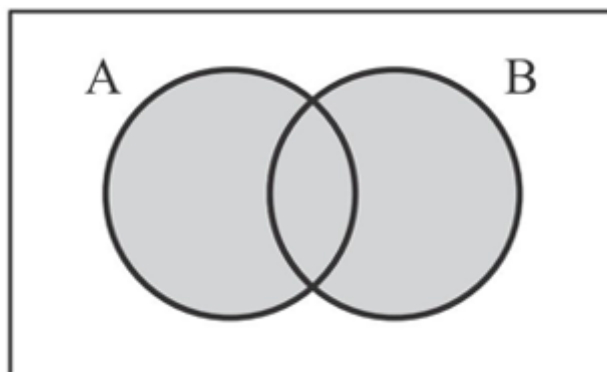


圖 3-7

(2)交集 (intersect)：

$A \cap B = \{x \in U \mid x \in A \text{ 且 } x \in B\}$ 。

- 集合A表示：喜歡喝咖啡的人
- 集合B表示：喜歡喝茶的人
- $A \cap B$ 表示：喜歡喝咖啡且喜歡喝茶的人

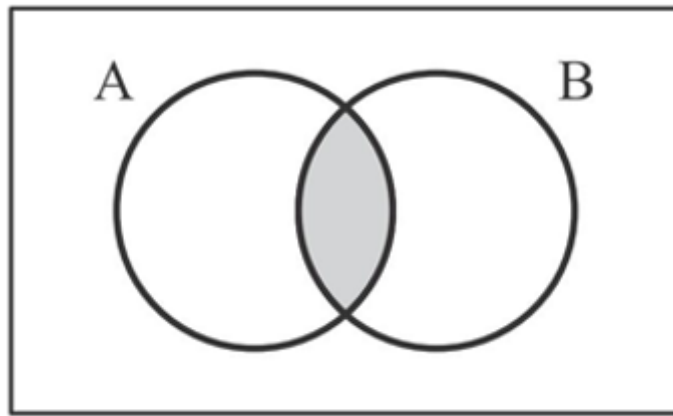


圖 3-8

(3) 差集 (difference) :

$A \setminus B = A - B = \{ x \in U \mid x \in A \text{ 且 } x \notin B \}$ 。

- 集合A表示：喜歡喝咖啡的人
- 集合B表示：喜歡喝茶的人
- $A - B$ 表示：喜歡喝咖啡且不喜歡喝茶的人
- $B - A$ 表示：喜歡喝茶且不喜歡喝咖啡的人

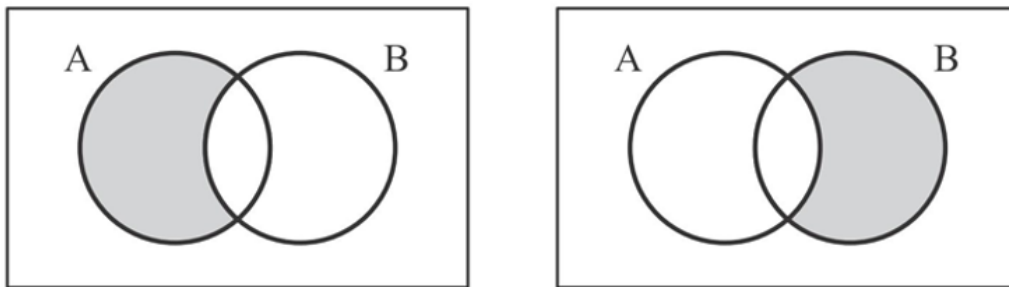


圖 3-9

(4) 對稱差集

- 集合A表示：喜歡喝咖啡的人
- 集合B表示：喜歡喝茶的人
- $A \Delta B$ 表示：喜歡喝咖啡或喜歡喝茶，但是不會兩個都喜歡的人

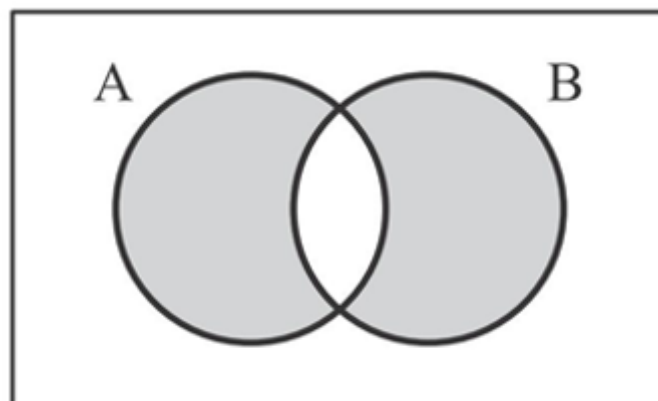


圖 3-10

練習1：甲酒吧老闆嘗試分析客戶

例：A表示：甲酒吧的客戶，B表示：會點外國酒品的酒客。

練習2：民調公司嘗試分析民意

例：A表示：20~29歲的選民，B表示：大專(以上)學歷選民。

(1) $A \cup B$ (2) $(A \cup B)^c$ (3) $A \cap B$ (4) $(A \cap B)^c$ (5) $A - B$ (6) $(A \cup B) - (A \cap B)$

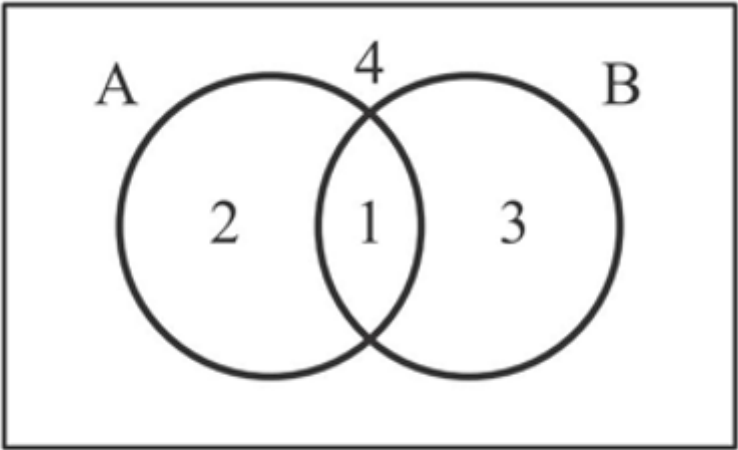


圖 3-11

參考答案

表 3-1

集合運算	區塊
$A \cup B$	1, 2, 3
$(A \cup B)^c$	4
$A \cap B$	1
$(A \cap B)^c$	2, 3, 4
$A - B$	2
$(A \cap B)^c$	1, 4
$(A \cup B) - (A \cap B)$	2, 3

練習3：民調公司嘗試分析民意

例：A表示：20~29歲的選民，B表示：碩士學歷的選民，C表示：住都會區的選民。

(1) $(A \cup B) \cap C$ (2) $(A \cup C)^c$ (3) $A \cap B \cap C$ (4) $A - (B \cap C)$ (5) $(A \cap B) - C$ (6) $(A \cup B) - (B \cap C)$

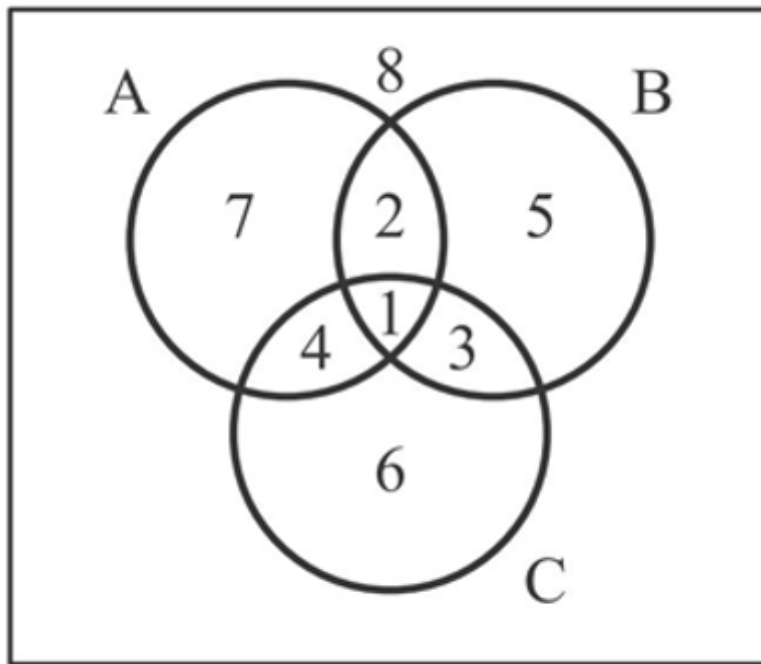


圖 3-12

參考答案：

表 3-2

集合運算	區塊
$(A \cup B) \cap C$	1, 3, 4
$(A \cup C)^c$	5, 8
$A \cap B \cap C$	1
$A \cap (B \cap C)$	2, 3, 4, 7
$(A \cap B) - C$	2
$(A \cup B) - (A \cap C)$	2, 3, 5, 7

4-4. 三個集合的表示法

例：A：單號同學 + B：雙號同學 + C：及格同學

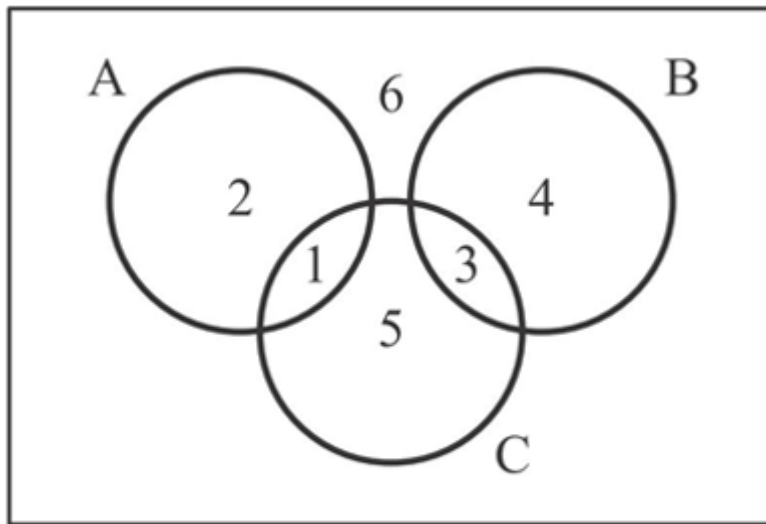


圖 3-14

表 3-4

集合運算	說明	區塊
$A \cap C$	期末考及格的單號同學	1
$A - C$	期末考不及格的單號同學	2
$B \cap C$	期末考及格的雙號同學	3
$B - C$	期末考不及格的雙號同學	4
C^c	期末考不及格的同學	2, 4, 6
$C - A - B$	期末考及格的無學號同學	5
$U - A - B - C$	期末考不及格的無學號同學	6