■ 碁峰書號：AEL027900

■ 書名：Hello！Python 程式設計

■ 勘誤內容：

頁碼	更正說明						
P2-9 勘誤日期： 2025/6/13	<p>更正例題 4 與其下的一整段文字</p> <p>例題 4</p> <hr/> <pre>print(f'{1.5:.0f} {2.5:.0f}') #輸出2 2</pre> <hr/> <p>Python 小數的進位並非都採用四捨五入，而是採用四捨六入五成雙，也就是四捨五入到最接近的偶數。上例中，1.5 取小數 0 位，最接近的偶數是 2，2.5 取小數 0 位，也是 2。但有些數值可能會因為精度問題，影響捨入結果。</p>						
P2-23 第 12 行 勘誤日期： 2025/6/13	<p>更正「說明」底下的程式</p> <p>說明</p> <p>可將運算式直接寫在 f 字串內，第 3 ~ 7 行可改寫成</p> <hr/> <pre>print(f'{2*pi*r:.1f} {pi*r**2:.1f} {4/3*pi*r**3:.1f}')</pre> <hr/>						
P3-18 第 12 行 勘誤日期： 2025/6/13	<p>更正「執行結果」</p> <p>執行結果</p> <table style="border-collapse: collapse; width: 100%;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">1 2 3</td> <td style="border-right: 1px solid black; padding: 5px;">3 4 5</td> <td style="padding: 5px;">11 15 19</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">無法構成三角形</td> <td style="border-right: 1px solid black; padding: 5px;">36</td> <td style="padding: 5px;">6792</td> </tr> </table> <hr/>	1 2 3	3 4 5	11 15 19	無法構成三角形	36	6792
1 2 3	3 4 5	11 15 19					
無法構成三角形	36	6792					
P3-33 倒數第 2 行 勘誤日期： 2025/6/13	<p>刪除「應用題」的第五題</p> <p>4. 寫一程式，輸入一字元後，能判斷輸入的字元是小寫字母、大寫字母、或數字。如果都不是，輸出 '特殊字元'。</p> <p>5. 寫一程式，輸入一個整數 n 後，若最多有 n 個糖果要分裝，7 個一數餘 5，11 個一數餘 5，17 個一數餘 5。請輸出所有可能的分裝結果。</p>						

頁碼	更正說明
<p>P5-8</p> <p>勘誤日期： 2025/6/13</p>	<p>更正如下黃標處</p> <p>以下程式碼是計算 bmi 值，並用 f 字串輸出，輸出結果為「James 身高 170 體重 58.5 bmi 值 20.2」。</p> <hr/> <pre>name = 'James' h, w = 170, 58.5 m = h / 100 print(f'{name}身高{h}體重{w} bmi值{w/m**2:.1f}')</pre> <hr/> <p style="text-align: right;">取至小數第 1 位 浮點數</p> <pre>print(f'{name} 身高{h} 體重{w} bmi值{w/m**2 :.1f}')</pre> <p style="text-align: center;"> James 170 58.5 計算出之 bmi 值 </p>
<p>P5-15</p> <p>勘誤日期： 2025/6/13</p>	<p>更正流程圖的「ing 結尾」</p> <pre> graph TD Start([輸入字串 s]) --> Len{長度 len(s) > 2} Len -- False --> Output([輸出字串 s]) Len -- True --> Ing{ing 結尾 s[-3:] == 'ing'} Ing -- True --> Ly[s = s + 'ly'] Ing -- False --> E{字尾是 e s[-1] = 'e'} E -- True --> EIng[s = s[:-1] + 'ing'] E -- False --> AddIng[s = s + 'ing'] Ly --> Output EIng --> Output AddIng --> Output </pre>
<p>P5-23</p> <p>勘誤日期： 2025/6/13</p>	<p>更正如下黃標處</p> <p>3. 解題演算法可設計如下：</p> <p>輸入代表英文句子的字串 s</p> <p>標點符號組成的字串 pun = r'!"()~[]{};:","<>./#@#%^^&* _~"'</p> <p>無標點符號的字串 nopun 初始值設為空字串 "</p>

頁碼	更正說明
P5-24 勘誤日期： 2025/6/13	更正「程式設計」的第 1 行 <div style="background-color: #e0e0e0; padding: 2px; margin: 5px 0;">程式設計</div> <hr/> <pre> 1 pun = r'!'!()-[]{};:'"\, <> . / ? @ # \$ % ^ & * _ ~ ' ' ' #pun是標點符號組成的字串 2 s = input()</pre>
P6-21 勘誤日期： 2025/6/13	將原第一行與第二行的文字，更正如下： <p>如果 f 的長度 = 1，輸出 -1，否則輸出串列 f 的元素，元素間用 * 分隔</p>
P6-24 勘誤日期： 2025/6/13	更正「解題方法」的第 2 項 <div style="background-color: #e0e0e0; padding: 2px; margin: 5px 0;">解題方法</div> <ol style="list-style-type: none"> 1. 讀取輸入的資料，轉成整數，存放到串列 a。 2. 出席同學的座號在串列 a 第 3 個元素以後，所以使用切片 n = a[2:]，將出席座號存放到串列 n。
P6-27 勘誤日期： 2025/6/13	更正「範例 6.3-4」的第 3 行 <div style="background-color: #e0e0e0; padding: 2px; margin: 5px 0;">範例 6.3-4 保齡球遊戲</div> <p>打保齡球時，球道上會有 10 個球瓶，玩家目標是要擊倒所有球瓶。有一類似保齡球的遊戲，球瓶和球的數量會變動，若有 n 個球瓶和 k 顆球，球瓶從 1 到 n 編號，記錄玩家每顆球擊倒球瓶的情形，在所有球都滾完後，輸出每個球瓶的狀況。</p>
P6-43 勘誤日期： 2025/6/13	更正第 11 題題目的程式 <p>11. 執行以下程式，結果為何？</p> <pre> a = [92, 85, 80, 100, 78, 41, 34, 90, 60, 59] count_1 = count_2 = 0 for i in range(10): if a[i] >= 80: count_1 = count_1 + 1 if a[i] < 60: count_2 = count_2 + 1 print((count_1 + count_2) % 4)</pre>

頁碼	更正說明										
P6-44 勘誤日期： 2025/6/13	<p>更正第 15 題题目的程式</p> <pre> for i in range(1, 9): if a[i] >= a[index]: index = i </pre>										
P6-45 勘誤日期： 2025/6/13	<p>更正第 5 題題目</p> <p>5. 寫一程式，輸入一串正整數，找出這些正整數的最大數及其索引，例如：輸入 5 1 9 6 8 3，輸出 9 2。(b002)</p>										
P7-4 勘誤日期： 2025/6/13	<p>更正「第 4 回合」底下說明</p> <p>第 4 回合</p> <p>1 $a[0] < a[1]$</p> <table border="1" style="display: inline-table; text-align: center;"> <tr> <td style="width: 20px;">0</td> <td style="width: 20px;">1</td> <td style="width: 20px;">2</td> <td style="width: 20px;">3</td> <td style="width: 20px;">4</td> </tr> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> </tr> </table> <p style="text-align: center;">不交換</p>	0	1	2	3	4	1	2	3	4	5
0	1	2	3	4							
1	2	3	4	5							
P7-7 勘誤日期： 2025/6/13	<p>更正如下黃標處</p> <p>2. i 由 $1 \sim n-1$ 變成 $0 \sim n-2$，j 也由 $0 \sim n-i-1$ 變成 $0 \sim n-i-2$。</p> <pre> for i in range(n - 1): for j in range(n - 1 - i): if a[j] > a[j + 1]: </pre> <p>3. $j = n-1 \sim i-1$，所以每回合變成由串列尾端開始，往前兩兩比較。</p> <pre> for i in range(n - 1): for j in range(n - 1, i, -1): if a[j - 1] > a[j]: </pre>										

頁碼	更正說明
P7-18 勘誤日期： 2025/6/13	<p>更正如下黃標處</p> <p>5. 解題演算法可設計如下：</p> <p>讀取輸入，轉為整數，存在串列 a</p> <p>輸入要搜尋的鍵值 key</p> <p>比較次數 c 初始值設為 0</p> <p>設定左邊界 L 為 0，右邊界 R 是串列長度 - 1</p> <p>當 $L \leq R$ 時，執行重複結構</p>
P7-23 勘誤日期： 2025/6/13	<p>更正如下黃標處</p> <p>一個 $m \times n$ (m 列 n 行) 的二維串列 a，第 i 列第 j 行的元素是 $a[i - 1][j - 1]$，其中 i 的範圍是 $0 \sim m - 1$，j 是 $0 \sim n - 1$ (圖 7-8)。</p> <div data-bbox="662 929 1161 1209" style="text-align: center;"> </div> <p>圖 7-8 二維串列第 i 列第 j 行的元素是 $a[i - 1][j - 1]$</p>

頁碼	更正說明
P7-40 勘誤日期： 2025/6/13	<p>更正第 5 ~ 9 題題目</p> <p>5. 寫一程式，輸入一串已排序好的整數，再輸入一個搜尋值，使用二分搜尋，輸出 $\leq$ 搜尋值之最大值的索引。</p> <p>例如：輸入 1 3 5 7 9 11 及搜尋值 2 時，≤ 2 的最大值是 1，所以輸出元素 1 的索引 0。</p> <p>6. 承上題，輸出 $\geq$ 搜尋值之最小值的索引。</p> <p>7. 寫一程式，輸入一串已排序好的整數，再輸入一個搜尋值，使用二分搜尋，輸出搜尋值的索引，若有多個相同的整數，輸出最小索引，若搜尋不到，輸出「沒搜尋到」。</p> <p>例如：輸入 1 2 3 3 3 5 7 9 11 及搜尋值 3 時，輸出元素 3 最小的索引 2。</p> <p>8. 一個有 n 個整數的序列，相鄰兩數之差的絕對值序列是 1 到 $n - 1$，稱為 jolly jumper。例如：1 2 4 1 5，相鄰 2 數差的絕對值為 1, 2, 3, 4，所以是 jolly jumper ($n = 5$)。</p> <p>寫一程式，輸入一串整數，第一個正整數為 n ($n < 100$)，代表此整數序列的長度，判斷此整數序列是否為 jolly jumper。(d097)</p> <p>9. 寫一程式，輸入整數 n 後，依序用數字 0, 1, 2, 3, …… $n * n - 1$，創建一個大小為 $n \times n$ 的二維串列，並輸出所建立的串列。</p>