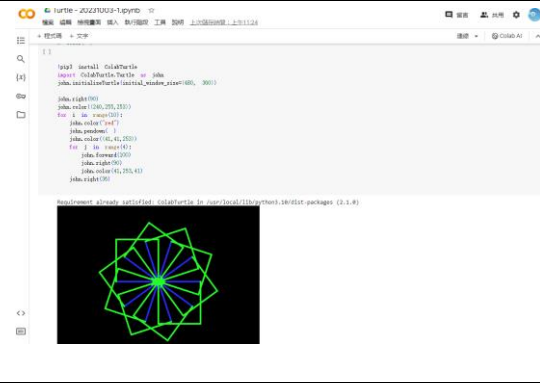
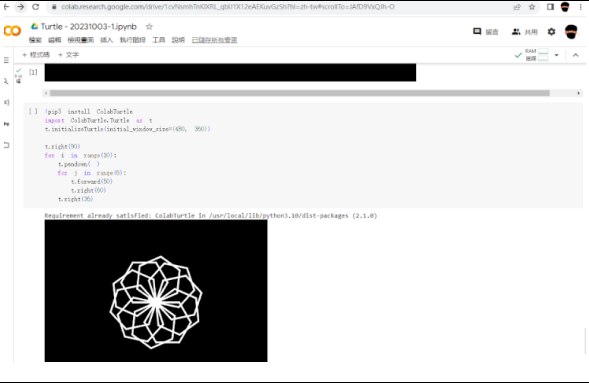

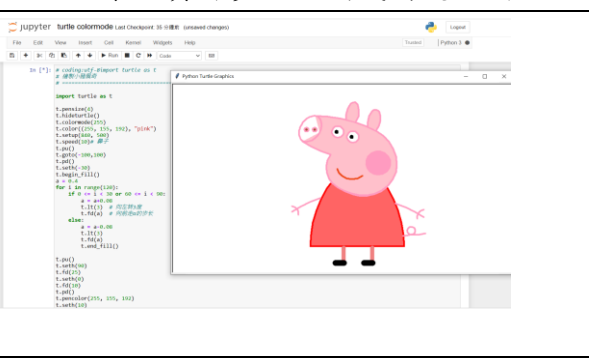


111至114年度新北市數位學習精進計畫

112年度數位學習創新教案設計

| | | | | |
|---|--|--|------|---|
| 服務學校 | 新北市市立丹鳳高中 | | 設計者 | 林秀貞 |
| 領域/科目 | 資訊科技 | | 實施年級 | 三年級 |
| 單元名稱 | Python 程式設計應用—turtle 模組 | | 總節數 | 共 4 節， <u>180</u> 分鐘 |
| 行動載具 作業系統 | ■Android 系統 ■Chrome 系統 □iOS 系統 ■Windows 系統 | | | |
| 設計依據 | | | | |
| 學習重點 | 學習表現 | <ul style="list-style-type: none"> ● 運 t-IV-4 能應用運算思維解析問題。 ● 設 c-IV-1 能運用設計流程，實際設計並製作科技產品以解決問題。 ● 設 s-IV-1 能繪製可正確傳達設計理念的平面或立體設計圖。 ● 設 c-IV-3 能具備與人溝通、協調、合作的能力。 | 核心素養 | <ul style="list-style-type: none"> ● 科-J-B3 了解美感應用於科技的特質，並進行科技創作與分享。 ● 科-J-A3 利用科技資源，擬定與執行科技專題活動。 |
| | 學習內容 | <ul style="list-style-type: none"> ● 資 P-IV-1 程式語言基本概念、功能及應用。 ● 資 P-IV-5 模組化程式設計與問題解決實作。 | | |
| 議題融入 | 實質內涵 | <p>設計理念：</p> <p>讓學生了解與學習如何使用 Python 程式碼製作多邊形多重重複迴圈圖案，同時鼓勵學生進一步探索程式設計和圖形繪製的世界。學生根據自身的學習程度和進度，再調整自主學習內容和難度。最後，利用簡報與影音剪輯技巧將學習過程彙整成自主學習歷程檔案。</p> | | |
| | 所融入之學習重點 | <ul style="list-style-type: none"> ● CHAT GPT 線上詢問程碼技巧學習。 ● 如何將程碼修改為適合自己的創作並將結果呈現。 ● 利用 Canva 簡報與影音剪輯技巧將學習過程彙整成自主學習歷程檔案 | | |
| 與其他領域/科目的連結 | <ul style="list-style-type: none"> ● 數學 ● 藝術 — 美術與色彩學 | | | |
| 教材來源 | <p>教師授課：投影片</p> <p>學生解題任務：學習單、Google Colab、Codingbar</p> <p>課堂討論：投影機、行動載具、課堂報告影片與簡報</p> | | | |
| 教學設備/資源 | Windows OS、 | | | |
| 使用軟體、數位資源或 APP 內容 | <ol style="list-style-type: none"> 1. Ipad 的相機及照片 App，會操作截圖、AirDrop 及拍照影片剪輯技巧 2. 使用【因材網】進行課前預習、課間學習、組內共學及課後複習 3. Google Colab、Codingbar、Google Drive 雲端操作 4. Canva 簡報、影音剪輯軟體 | | | |
| 學習目標 | | | | |
| <ol style="list-style-type: none"> 1. 理解如何使用 Python 程式碼製作多邊形 2. 學會使用 for 迴圈重複繪製多邊形，形成圖案 3. 學會利用 Chat GPT 與 Chat GPT 溝通互動協助例題的程碼撰寫 4. 分析自己與 GPT 撰寫程式的邏輯思考差異點 5. 利用 Canva 簡報與影音剪輯技巧將學習過程彙整成完整的自主學習歷程檔案 | | | | |

| 教學活動設計 | | |
|--|--|--|
| 教學活動內容及實施方式 | 時間 | 使用軟體、數位資源或 APP 內容 |
| 1. 時間45分鐘/1節，總共180分鐘/4節 2. 課堂用 Google Colab、Codingbar 平台操作 3. 教學對象：九年級學生 4. 了解與學習運用 python 程式製作多重重複迴圈的圖案 5. 學會如何與 GhatGPT 溝通，產生參考的程式碼，解決程式碼撰寫問題 6. 程式碼與變化圖形上傳 Google Classroom、因材網作業分享區 7. 分組討論、小組成員分工合作、小組共學 8. 各小組自主學習歷程電子檔製作 | 4節，180分鐘(每節45分鐘) | 1. Ipad 的相機及照片 App，會操作截圖、AirDrop 及拍照技巧。 2. 使用因材網進行課前預習、課間學習、組內共學及課後複習。 3. Google Colab 平台操作。 |
| 教學成果 |  |  |
| | 說明：學生實作多重迴圈幾何變化圖形 | 說明：學生實作多重迴圈幾何變化圖形 |
| |  |  |
| | 說明：學生實作多重迴圈幾何變化圖形 | 說明：學生實作多重迴圈幾何變化圖形 |
| 教學心得與省思 | (含教學調整的脈絡、成效分析、教學省思、修正建議等) 1. 讓學生了解與運用 Python 函式設定與多重迴圈重複繪製圖形的概念 2. 讓學生嘗試繪製多邊形圖案 3. 提供範例供學生設計製作參考 4. 小組分工討論共學 5. 錄製程式碼創作過程與成果產出 6. 創作品線上分享 | |
| 參考資料 | 1. 翰林資訊科技(3上) 2. 高中資訊教師黃建庭的教學網站 https://sites.google.com/view/zsgitit/home/python-cheng-shi-she-ji/python-xuan-ze-jie-gou 3. 輕鬆玩 Python 程式設計，黃建庭，全華圖書，2018 4. Python turtle 模組的進階應用 https://webnas.bhes.ntpc.edu.tw/wordpress/wpcontent/uploads/2017/09/PythonTurtleAdvanced.pdf 5. Python 繪圖專題 https://hackmd.io/@h3BSPiUpQhCVyCHgJAWWHg/BJKeNNBLV 6. Python 迴圈常見用法 https://www.learncodewithmike.com/2019/12/python.html | |
| 附錄 | 學習單或其他相關資料：如附件 | |

教案內容

| | | |
|--|---|---|
| 教師姓名 | 林秀貞 | |
| 四學應用 | 知識獲得：教學影片、WSQ 學習單 知識應用：程式撰寫(Codingbar)、組內分享、Colaboratory 平臺、 解題策略、螢幕錄製、影片剪輯 學生自學：WSQ 學習單、影片 組間互學：合作解難、展示匯報 教師導學：教師提問、問題釐清、總結延伸、指導編輯自主學習歷程電子檔 | |
| 學科領域 | 資訊科技 | |
| 教學目標 | 1. 於培養學生的邏輯思考與解決問題能力。 2. 具備基礎程式實作能力，並可自主學習其他程式。 3. 培養與加強同學們的創造力、資訊能力、表達能力及團隊合作能力。 4. 指導並培養學生編輯自主學習歷程電子檔的能力 | |
| 授課單元/主題 | Python 程式設計應用：turtle 模組 | |
| 本單元能力指標： 資 P-IV-1 程式語言基本概念、功能及應用。 資 P-IV-5 模組化程式設計與問題解決實作。 資 T-IV-2 資訊科技應用專題。 | | |
| 教學方式 | 教師授課、觀看影片、課堂討論、程式實作 | |
| 資源/設備/書籍 | 教師授課：投影片、學習平台 觀看影片：行動載具、Youtube 影片 學生解學任務：學習平台、Colaboratory、Codingbar ... 等 課堂討論：行動載具、學習平台 | |
| 評量方式 | 1. 作業繳交：學習平台 20% 2. 程式撰寫 (含 Colaboratory、Codingbar 各 25%) 50% 3. 上課表現 20% 4. 學生互評 10% | |
| 單元教學時間 | 每週2節，共計 4 節 180 分鐘 | |
| 課堂組織 | 學習任務 | 教學支援 |
| 課前自學 | | |
| 1. 學生自學 | 1. 指派因材網的課程包學習任務，讓學生於課前完成任務。 2. 透過影片學習單完成影片觀看，並回答相關知識問題。 3. Colaboratory 平台中，記錄程式碼撰寫的學習重點。 | 1. 教師觀看學生學習進度，了解學生學習狀況。 2. Colaboratory 記錄程式碼學習重點與討論內容 (表 1、圖 1)。 |
| 單元第二節課 45 分鐘 | | |
| 1. 教師導入 (20分鐘) | 課前複習單向選擇結構 if、雙向選擇結構 if、else、多向選擇結構 if、elif、else 原理與應用 1. 說明本節課學習重點。 (1) 能了解3種選擇結構的差異與用法 (2) 能理解常見的3種選擇結構運用 2. 進入 Codingbar 討論並撰寫程式 | 1. 教師分析因材網中的學習任務 2. 說明程式撰寫的學習問題點 3. 小組工作分配表 (附件一) |
| 2. 進行小組合作學習 (20分鐘) | 1. 各組登入因材網，進入課程包 — 選擇結構例題與程式撰寫 (小組任務) (1) <u>各組組長</u> 主持，進行小組討論 (2) <u>記錄長</u> 紀錄討論後的正確程式碼 (3) <u>設計長</u> 清楚解決問題的邏輯過程 | 1. 教師課間巡視，觀察各小組討論內容是否可執行、適時給予意見指導 2. 獎勵討論用心且認真的小組 3. 記錄學生提示討論時出現的疑問 |

| | | |
|-----------------------|--|--|
| 3. 教師導學 (5分鐘) | 1. 依據學生提出的問題點，進行說明 2. 預告下一節課進行流程。 3. 提醒利用課餘時間複習因材網影片 | 1. 引導學生完成自主學習單—單向、雙向選擇結構(表2、表3、圖2、圖3) 2. 完成 Codingbar 題目 3. 利用因材網影片複習 |
| 單元第三節課45分鐘 | | |
| 1. 教師導入 (10分鐘) | 1. 複習前一節課的學習重點 2. 課前複習 for 迴圈的原理與應用 3. 說明小組作業提繳與組內互評規則 4. 說明個人自評規則評分表細項規則 | 1. 教師巡察各小組討論內容的思考點，適時給予意見指導。 2. 獎勵認真入、用心討論的小組 3. 記錄學生討論時所出現的疑問 |
| 2. 進行小組合作學習 (20分鐘) | 1. 各組登入因材網，進入課程包—for 迴圈例題與程式撰寫(小組任務) (1) 各組組長 主持，進行小組討論 (2) 記錄長 紀錄討論後的正確程式碼 (3) 設計長 清楚解決問題的邏輯過過程 2. 小組內互評：每位組員都必須進行組內自評，組長協助確認組員完成組內互評 3. 統計分數：將各項評分加總，總分紀錄在小組成績 4. 教師依照各小組得分與表現，給予獎勵點數 | 1. 引導學生整理並紀錄資料 2. 聆聽各組成員的報告 3. 記錄報告重點與口頭評量 4. 小組內互評規則評分表(附件二) |
| 3. 隨堂測驗 (10分鐘) | 1. 利用因材網課程內的單元評量功能進行課堂隨堂測驗 2. 測驗題目解說 | 1. 因材網課程包單元評量 2. 總結提問 |
| 4. 教師導學 (5分鐘) | 1. 引導學生完成個人自評規則評分表 2. 提醒學生利用因材網影片複習並 GOOGLE 搜尋更多網路資源協助學習 | 1. 個人自評規則評分表(附件三) 2. 引導學生完成自主學習單—for 迴圈的原理與應用(表4、圖4) 3. 利用因材網影片複習或其他網路資源 |
| 單元第四節課45分鐘 | | |
| 1. 教師導入 (10分鐘) | 1. 複習前一節課的學習重點 2. 鼓勵學生進一步探索程式設計和圖形繪製的世界 3. 示範與解釋如何使用 Python—turtle 模組程式碼繪製多邊形，例如：正方形、五邊形、彩色多邊形... 4. 示範並讓學生了解與學習如何使用 Python 程式碼製作多邊形多重重複迴圈圖案，例如：旋轉正方形 | 1. 引導學生完成自主學習單—turtle 模組程式碼應用(表5、圖5) 2. 教師巡察各小組討論內容的思考點，適時給予意見指導。 3. 獎勵認真入、用心討論的小組 4. 記錄學生討論時所出現的疑問 |
| 2. 進行小組合作學習 (15分鐘) | 1. 各組登入因材網，進入課程包—多邊形多重重複迴圈圖案設計與程式撰寫(小組任務) (1) 各組組長 主持，進行小組討論 (2) 記錄長 紀錄討論後的正確程式碼 (3) 設計長 清楚解決問題的邏輯過過程 2. 小組內互評：每位組員都必須進行組內自評，組長協助確認組員完成組內互評 3. 統計分數：將各項評分加總，總分紀錄在小組成績 | 1. 引導學生整理並紀錄資料 2. 聆聽各組成員的報告 3. 記錄報告重點與口頭評量 4. 小組內互評規則評分表(附件三) |

| | | |
|-------------------|---|---|
| | 4. 教師依照各小組得分與表現，給予獎勵點數 | |
| 3. 任務製作 (15分鐘) | <ol style="list-style-type: none"> 1. 學生分享他們所繪製的圖案成果 2. 討論學生的經驗和觀察，並確保理解如何使用迴圈來重複繪製圖案 3. 學生記錄如何與 Ghat GPT 溝通並藉助 Chat GPT 的協助思考解決程式碼撰寫的問題點 4. 學生分析並記錄 Ghat GPT 撰寫的程式碼與自己所撰寫程式碼的差異性 | <ol style="list-style-type: none"> 1. 因材網課程包單元評量 2. 總結提問 |
| 4. 教師導學 (5分鐘) | <ol style="list-style-type: none"> 1. 引導各小組分享成果 2. 各小組分享創作圖案時，所遇到的程式碼撰寫困難點與心得 3. 各小組利用螢幕錄製與影音剪輯軟體，製作各組的自主學習歷程電子檔 4. 總結本課程的內容 | <ol style="list-style-type: none"> 1. 各小組互評規則評分表(附件四) 2. 利用因材網影片複習或其他網路資源 |

高級中等學校科技輔助自主學習教案設計

新北市立丹鳳高中資訊科技課程—國中部

| | | | |
|-------|-----------------------------|-----|---------------------|
| 日期： | 20231116、20231123 | 時間： | 4 節課180 分鐘 (每節45分鐘) |
| 班級： | 908 | 科目： | 資訊科技 |
| 章節： | Python 程式設計應用— turtle 模組 | 節數： | 第1、2節 (2週) |
| 授課教師： | 林秀貞 | 教室： | 電室203R |

智慧互動黑板

| 第四排 | 第三排 | 第二排 | 第一排 |
|-----|-----|-----|-----|
| 第八組 | 第五組 | 第三組 | 第一組 |
| 第七組 | 第六組 | 第四組 | 第二組 |

附件一

科技輔助自主學習合作分組小組工作分配單

組別：第一組 第二組 第三組 第四組 第五組 第六組

學習重點：

- (1) 能了解 Python 程式中 turtle 模組的功能。
- (2) 能理解數幾何圖形的構造。
- (3) 能應用 turtle 模組建構數學幾何多變化圖形，並結合色彩進行運用。

組內工作分配表：(請各小組成員討論後填入姓名)

| 編號 | 代號 | 小組成員分配任務 | 學生姓名 |
|----|-----|--|------|
| 1 | 組長 | 1. 主持小組成員討論 (引導小組成員討論並分析解題方式) 2. 確認組內小組合作流程並完成組內自評 | |
| 2 | 副組長 | 1. 彙整小組解題記錄與執行成果 2. 進行確認他組表現評分是否正確 (互評標準：逐條確認是否符合互評標準表單) | |
| 3 | 設計長 | 1. 依據小組成員討論並分析解題方式，實際撰寫程式碼，若有撰寫盲點可借助 Ghat GPT 2. 比較並標註自行撰寫的程碼與 Ghat GPT 撰寫的程碼有何差異 | |
| 4 | 記錄長 | 彙整小組解題成果並剪輯成 mp4檔案上傳雲室 | |
| 5 | 檢核長 | 確認解題是否正確並說明解題的邏輯概念 | |

附件二

科技輔助自主學習合作 小組內 互評規則評分表

評分組別：第一組 第二組 第三組 第四組 第五組 第六組

學習重點：

(1) 小組成員分工合作

(2) 能尊重並聆聽同學的邏輯分析看法，並歸納完成程式作業

互評標準：(請評分同學逐條確認)得分為 1-5 顆星

| 順序 | 評 分 標 準 | 1 | 2 | 3 | 4 | 5 |
|----|-------------------------|---|---|---|---|---|
| 1 | 能依照分工任務表，完成任務的內容 | | | | | |
| 2 | 用心參與討論與回答 | | | | | |
| 3 | 分析問題的解法與編寫程式，執行出結果 | | | | | |
| 4 | 訓練與 ChatGPT 溝通得到撰寫程式碼協助 | | | | | |
| 5 | 尊重小組成員的不同意見、分工互助合作 | | | | | |

附件三

科技輔助自主學習 個人 自評規則評分表

自評標準：(請評分同學逐條確認) 得分為 1-5 顆星

| 順序 | 檢查確認 | 評 分 標 準 | 得 分 |
|----|---|----------------------------|-----|
| 1 | <input type="checkbox"/> 有 <input type="checkbox"/> 無 | 能理解數學幾何圖形的邏輯構造 | |
| 2 | <input type="checkbox"/> 有 <input type="checkbox"/> 無 | 能理解 Python — Colab 繪圖模組的用法 | |
| 3 | <input type="checkbox"/> 有 <input type="checkbox"/> 無 | 能理解 Python — turtle 模組的用法 | |
| 4 | <input type="checkbox"/> 有 <input type="checkbox"/> 無 | 能理解單向選擇結構 if 應用 | |
| 5 | <input type="checkbox"/> 有 <input type="checkbox"/> 無 | 能理解多向選擇結構 if、elif、else 應用 | |
| 6 | <input type="checkbox"/> 有 <input type="checkbox"/> 無 | 能理解 for 迴圈應用 | |

附件四

科技輔助自主學習合作 各小組 互評規則評分表

評分組別：第一組 第二組 第三組 第四組 第五組 第六組

學習重點：

(1) 各小組代表發表與分享小組學習成果

(2) 能尊重並聆聽各小組代表同學的邏輯分析看法，並提出正向建議

互評標準：(請評分組別逐條確認)得分為 1-5 顆星

| 順序 | 評 分 標 準 | 1 | 2 | 3 | 4 | 5 |
|----|------------------------------------|---|---|---|---|---|
| 1 | 用心認真分享小組所繪製的圖案成果 | | | | | |
| 2 | 記錄與分享如何與 Ghat GPT 溝通的技巧 | | | | | |
| 3 | 分享透過 Chat GPT 協助思考解決程式碼撰寫的問題點 | | | | | |
| 4 | 記錄並分享 Ghat GPT 撰寫的程式碼與自己所撰寫程式碼的差異性 | | | | | |
| 5 | 尊重各小組代表提出的意見，並回饋正向看法與觀點 | | | | | |

表1. 自主學習單 — 單向選擇結構

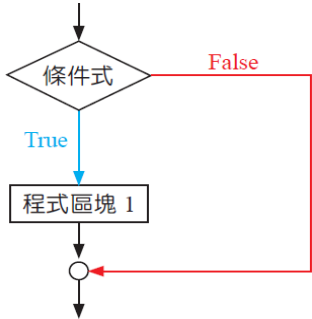
| 主題 | 重點整理 | 程式舉例 (至少 3個) |
|--|---|--|
| <p>單向選擇程式語法</p> <p>if 條件判斷：</p> <p>條件成立的敘述</p> | <p>(1) 只做測試條件為真時，執行條件為真的動作，只有一個方向的選擇</p> | <pre>if score >= 60: print("很好，請繼續保持下去")</pre> |
| | <p>(2) 例如：「若週末天氣好的話，我們就去打球」</p> | <pre>a = 10 b = 8 if (a > b): print("變數 a 比較大")</pre> |
| | <p>(3) 「if 條件判斷：」後面接的條件成立的動作</p> | <pre>c = int(input('請評分餐點 1~5分：')) if (c >= 3): print("滿意")</pre> |
| |  | |

表2. 自主學習單 — 雙向選擇結構

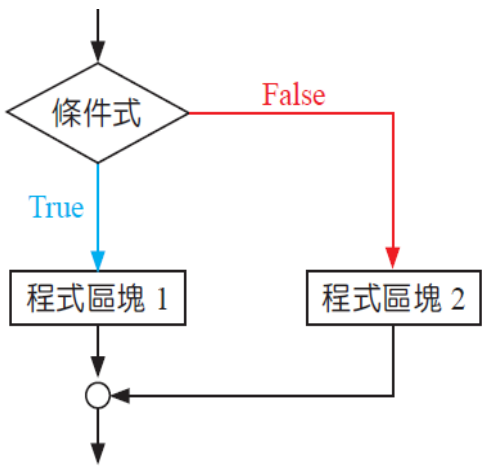
| 主題 | 重點整理 | 程式舉例 (至少 3個) |
|---|---------------------------------------|---|
| 雙向選擇程式語法 if 條件判斷： 條件成立的敘述 | (1) 當測試條件為真時，執行測試條件為真的動作；否則做測試條件為假的動作 | 若 cost 大於等於 2000，則顯示為 cost 值打九折，否則顯示 cost 值 <pre>if cost >= 2000: print(cost*0.9) else: print(cost)</pre> |
| else: 條件不成立的敘述 | (2) 有兩個方向的選擇結構 | 判斷奇、偶數 <pre>d = int(input('請輸入一個數字：')) if (d % 2 == 0): print('偶數') else: print('奇數')</pre> |
| | (3) 例如：「若週末天氣好的話，我們就出去打球，否則去看電影」 | <pre>e = int(input('請評分餐點 1~5 分：')) if (e >= 3): print('滿意') else: print('不滿意')</pre>  |

表3. 自主學習筆記 — 多向選擇結構

| 主題 | 重點整理 | 程式舉例 (至少 3個) |
|--|---|---|
| <p>多向選擇程式語法</p> <p>if 條件判斷1:</p> <p>條件判斷1成立的敘述</p> | <p>(1) 選擇結構中還可以加入選擇結構，讓程式有無限可能執行的路徑與狀態</p> | <pre>f = int(input('請評分餐點 1~5 分：')) if (f == 5): print('非常滿意') elif (f >= 3): print('滿意') else: print('不滿意')</pre> |
| <p>elif 條件判斷2:</p> <p>條件判斷2成立的敘述</p> <p>else:</p> <p>條件判斷2不成立的敘述</p> | <p>(2) 用多個 if-elif-else 達成多項選擇結構</p> | <p>若成績大於等於80分，評語為「非常好」，否則若成績大於等於60分，評語為「不錯喔」，否則評語為「要加油」</p> <pre>score = int(input('請輸入一個成績？')) if score >= 80: print('非常好') elif score >= 60: print('不錯喔') else: print('要加油')</pre> |
| | <p>(3) 例如：某快遞公司以重量為計算郵資的依據，若重量小於等於5公斤，則顯示「50」，否則若重量小於等於10公斤，則顯示「70」，若重量小於等於15公斤，則顯示「90」，若重量小於等於20公斤，則顯示「110」，否則顯示「超過20公斤無法寄送」</p> | <pre>w = float(input('請輸入物品重量？')) if w <= 5: print('所需郵資為50元') elif w <= 10: print('所需郵資為70元') elif w <= 15: print('所需郵資為90元') elif w <= 20: print('所需郵資為110元') else: print('超過20公斤無法寄送')</pre> |

表4. 自主學習筆記 – for 迴圈

| 主題 | 重點整理 | 程式舉例 (至少 3個) |
|------------------------------|---|--|
| for 迴圈變數 in 串列: 程式區塊 | (1) 處理重複步驟，會從一個串列中逐一取出元素，指定給迴圈變數，因為串列中的元素個數，就是重複迴圈中程式碼執行的次數，稱為計次迴圈 | 累加計算 product = 0 n = int(input('請輸入數字 n = ')) for i in range(1, n+1): product = product+i print('1+2+...+', n, '=', product) |
| range(起始值,結束值,遞增(減)值) | (2) 計次迴圈經常用 range()函式，建立整數循序串列 | 累乘計算 product = 1 n = int(input('請輸入數字 n = ')) for i in range(1, n+1): product = product*i print('1*2*...* ', n, '=', product) |
| Break 與 continue 控制迴圈的執行流程指令 | (3) break：直接中斷迴圈，在 break 指令之後的運算皆不會執行 | for number in range(1, 10): if number % 3 == 0: break print(number) |
| | (4) continue：同樣的在 continue 指令之後的運算不會執行，但是不會中斷迴圈，而是會繼續讀取下一個元素。 | for number in range(1, 10): if number % 3 == 0: continue print(number) |
| Python Nested Loops (巢狀迴圈) | (5) 迴圈中又有一層迴圈，外迴圈(Outer loop)及內迴圈 (Inner loop)，程式執行，先從外部迴圈取得第一個元素 M，再執行內部迴圈，直到內部迴圈執行完畢，才回到外部迴圈取得第二個元素 i，再一次執行內部迴圈，依此類推，直到外部迴圈的目標物都讀取完畢，整個巢狀迴圈結束 | for letter in "Mike": for number in range(3): print(letter, number) |

表5. 自主學習筆記 — Python turtle 模組

| 主題 | 重點整理 | 程式舉例 (至少 3個) |
|--------------------------------|--|---|
| <p>利用簡單的指令操作烏龜的行走路徑，繪製幾何圖形</p> | <p>(1) 先產生視窗 (Screen)，設定視窗大小、標題、背景顏色，然後誕生一隻小烏龜，再操作烏龜前進及轉向繪製幾何圖形 — 畫彩色八邊形</p> | <pre>!pip3 install ColabTurtle import ColabTurtle.Turtle as john john.initializeTurtle() john.pencolor('red') john.backward(100) john.left(45) john.pencolor('blue') john.backward(100) john.left(45) john.pencolor('green') john.backward(100) john.left(45) john.pencolor('purple') john.backward(100) john.left(45) john.pencolor('pink') john.backward(100) john.left(45) john.pencolor('silver') john.backward(100) john.left(45) john.pencolor('orange') john.backward(100) john.left(45) john.pencolor('yellow') john.backward(100) john.left(45)</pre> |
| | <p>(2) 利用 for 迴圈畫單色十邊形</p> | <pre>!pip3 install ColabTurtle import ColabTurtle.Turtle as john john.initializeTurtle(initial_window_size=(800, 600)) for i in range(10): john.color(255, 0, 255) john.pendown() john.forward(100) john.left(36)</pre> |
| | <p>(3) 利用巢狀迴圈畫單色多個五邊形</p> | <pre>!pip3 install ColabTurtle import ColabTurtle.Turtle as john john.initializeTurtle(initial_window_size=(800, 600)) for i in range(4): john.color(176, 196, 222) john.pendown() for j in range(5): john.forward(30) john.right(72) john.penup() john.forward(70)</pre> |

圖1. Colaboratory 筆記_實作 Python 單向選擇結構

| | |
|--|---|
| <p><u>程式碼</u></p> <pre>score = int(input('請輸入一個成績?')) if score >= 60: print('很好，請繼續保持下去')</pre> | <p><u>執行結果</u></p> <p>輸入成績「60」，結果顯示在螢幕。</p> <p>請輸入一個成績?60 很好，請繼續保持下去</p> |
| <div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> 9秒 </div> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <pre>score = int(input('請輸入一個成績?')) if score >= 60: print('很好，請繼續保持下去')</pre> </div> </div> <div style="margin-top: 10px; border: 1px solid #ccc; padding: 5px;"> ⊗ 請輸入一個成績? 68 很好，請繼續保持下去 </div> | |
| <p><u>程式碼</u></p> <pre>a = 10 b = 8 if (a > b): print('變數 a 比較大')</pre> | <p><u>執行結果</u></p> <p>輸入兩變數 a、b，判斷變數 a 值與 b 值的大小，若變數 a 大於變數 b，則顯示變數 a 比較大在螢幕。</p> <p>請輸入變數 a 為 10 請輸入變數 b 為 8 判斷變數 a 大於變數 b</p> |
| <p><u>程式碼</u></p> <pre>c = int(input('請評分餐點1至5分:')) if (c >= 3): print('滿意')</pre> | <p><u>執行結果</u></p> <p>輸入變數 c 的評分數值，判斷變數 c 數值是否大於等於3，若大於等於3，則顯示滿意在螢幕。</p> <p>請輸入變數 a 為 10 請輸入變數 b 為 8 判斷變數 a 大於變數 b</p> |
| <div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> 4秒 </div> <div style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <pre>a = 10 b = 8 if (a > b): print('變數 a 比較大') c = int(input('請評分餐點 1 至 5 分 : ')) if (c >= 3): print('滿意')</pre> </div> </div> <div style="margin-top: 10px; border: 1px solid #ccc; padding: 5px;"> ⇒ 變數 a 比較大 請評分餐點 1 至 5 分 : 3 滿意 </div> | |

圖2. Colaboratory 筆記 — 實作 Python 雙向選擇結構

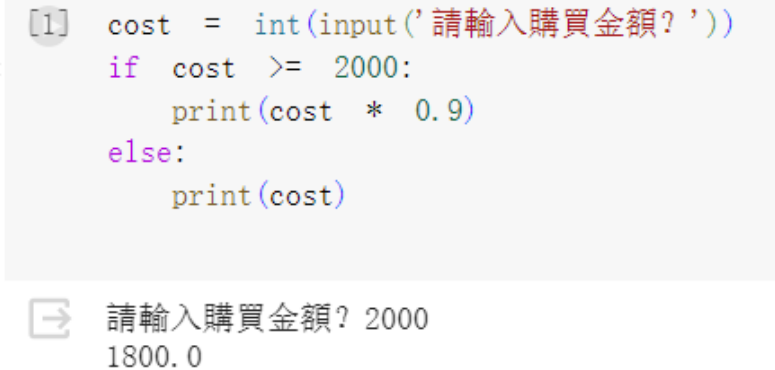
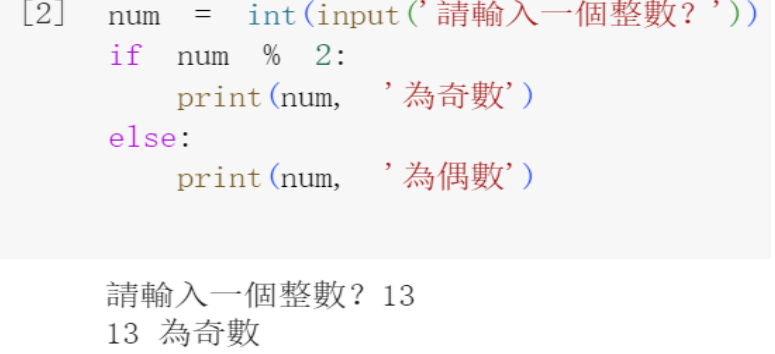
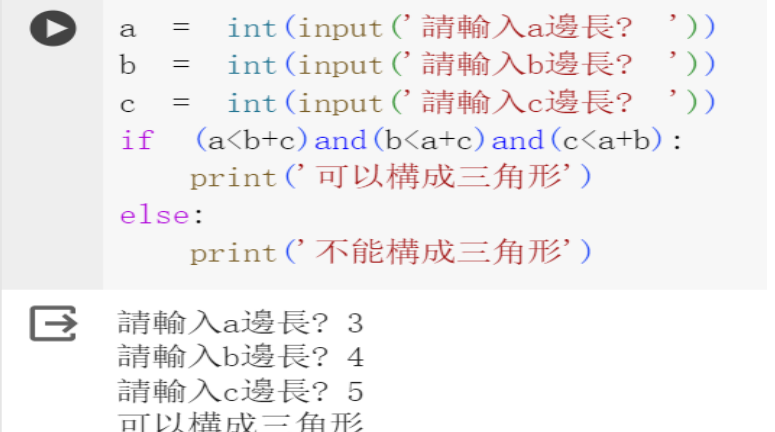
| | |
|---|---|
| <p><u>程式碼</u></p> <pre>cost = int(input('請輸入購買金額?')) if cost >= 2000: print(cost * 0.9) else: print(cost)</pre> | <p><u>執行結果</u></p> <p>輸入購買金額「2000」，結果顯示在螢幕。</p> <p>請輸入購買金額? 2000 1800.0</p> |
|  | |
| <p><u>程式碼</u></p> <pre>num = int(input('請輸入一個整數?')) if num%2: print(num, '為奇數') else: print(num, '為偶數')</pre> | <p><u>執行結果</u></p> <p>輸入一個數字「13」，顯示結果在螢幕上。</p> <p>請輸入一個整數? 13 13為奇數</p> |
|  | |
| <p><u>程式碼</u></p> <pre>a = int(input('請輸入 a 邊長?')) b = int(input('請輸入 b 邊長?')) c = int(input('請輸入 c 邊長?')) if (a<b+c)and(b<a+c)and(c<a+b): print('可以構成三角形') else: print('不能構成三角形')</pre> | <p><u>執行結果</u></p> <p>輸入三角形三邊長，分別為變數 a、b、c，根據三角形中任兩邊相加要大於第三邊，判斷是否為三角形。</p> <p>請輸入 a 邊長? 3 請輸入 b 邊長? 4 請輸入 c 邊長? 5 可以構成三角形</p> |
|  | |

圖3. Colaboratory 筆記 — 實作 Python 多向選擇結構

| | |
|---|--|
| <p><u>程式碼</u></p> <pre>score = int(input('請輸入一個成績?')) if score >= 80: print('非常好') elif score >= 60: print('不錯喔') else: print('要加油')</pre> | <p><u>執行結果</u></p> <p>輸入成績「60」，結果顯示在螢幕。 請輸入一個成績? 60 很好，請繼續保持下去</p> |
|  | |
| <p><u>程式碼</u></p> <pre>w = float(input('請輸入體重(KG)?')) h = float(input('請輸入身高(M)?')) bmi = w/(h*h) print('BMI 為',bmi) if bmi < 18: print('體重過輕') elif bmi < 24: print('體重正常') elif bmi < 27: print('體重過重') else : print('體重肥胖')</pre> | <p><u>執行結果</u></p> <p>請輸入體重(KG)? 65 請輸入身高(M)? 1.65 BMI 為 23.875114784205696 體重正常</p> |
|  | |

圖4. Colaboratory 筆記 — 實作 Python for 迴圈結構

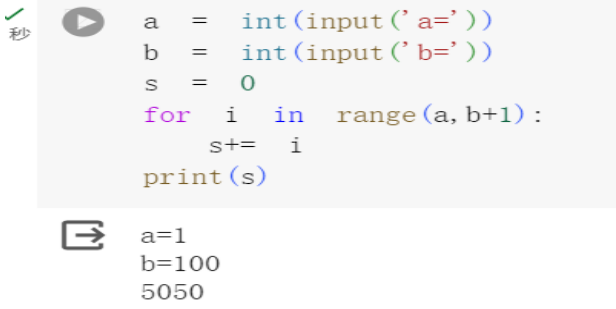
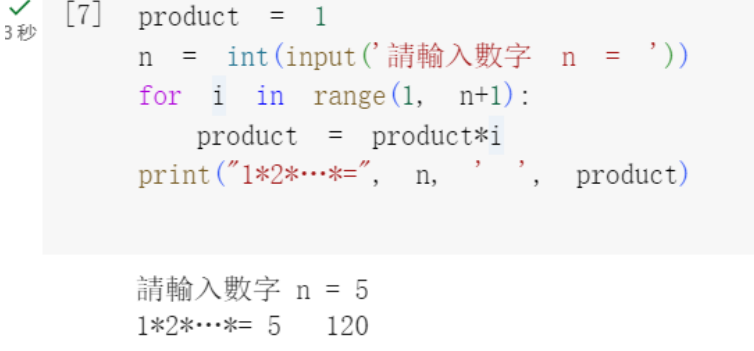
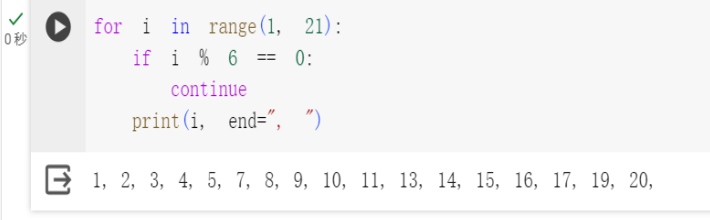
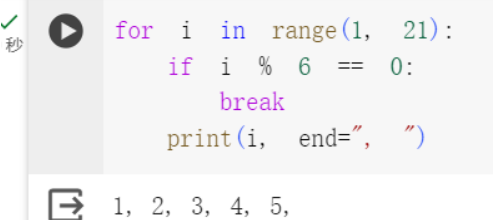
| | |
|--|---|
| <p><u>程式碼</u></p> <pre>a = int(input('a=')) b = int(input('b=')) s = 0 for i in range(a,b+1): s+= i print(s)</pre> | <p><u>執行結果</u></p> <p>輸入兩正整數 a、b，其中 b>a，求 a 加到 b 的總和，利用迴圈直接計算加總，總和為 a+a+1+a+2+a+3+...+b-1+b 並顯示在螢幕上</p> |
|  | |
| <p><u>程式碼</u></p> <pre>product = 1 n = int(input('請輸入數字 n = ')) for i in range(1, n+1): product = product*i print("1*2*...*=", n, ', ', product)</pre> | <p><u>執行結果</u></p> <p>輸入一個正整數 n，利用迴圈計算並輸出 n! 的值</p> <p>利用迴圈直接計算累乘，總和為 a*a+1*a+2*a+3*...*b-1*b 並顯示在螢幕上</p> |
|  | |
| <p><u>程式碼</u></p> <pre>for i in range(1, 21): if i % 6 == 0: continue print(i, end=" ",)</pre> | <p><u>執行結果</u></p> <p>要印出 1~20 中，非 6 的倍數的所有數字，並且將數字間用逗號隔開</p> <p>if i % 6 == 0，判斷變數 i 是否為 6 的倍數，如果是，則 continue，跳過該數，輸出結果：</p> <p>1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 20</p> |
| <pre>for i in range(1, 21): if i % 6 == 0: break print(i, end=" ",)</pre> | <p><u>執行結果</u></p> <p>要印出 1~20 中，非 6 的倍數的所有數字，並且將數字間用逗號隔開，將 continue 改為 break 直接將整個迴圈停止運作，輸出結果：1, 2, 3, 4, 5</p> |
|  |  |

圖5. Colaboratory 筆記 — 實作 Python turtle 模組

程式碼

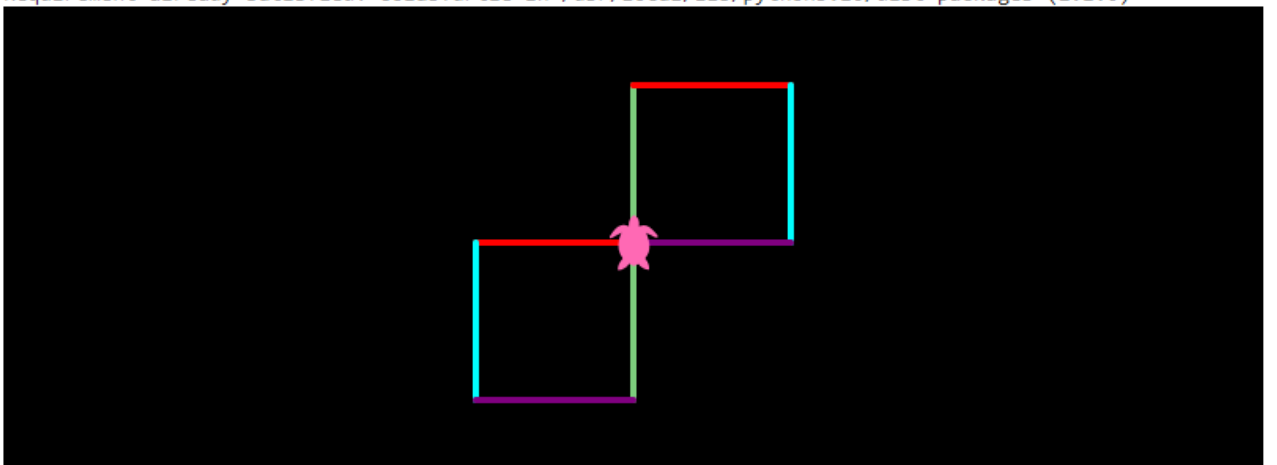
```
!pip3 install ColabTurtle
import ColabTurtle.Turtle as john
john.initializeTurtle( )
john.color(124, 205, 124)
john.forward(100)
john.left(90)
john.color('red')
john.forward(100)
john.left(90)
john.color(0, 255, 255)
john.forward(100)
john.left(90)
john.color('purple')
john.forward(100)
john.left(90)
john.color(255, 105, 180)

john.penup( )
john.forward(100)
john.pendown( )

john.color(124, 205, 124)
john.forward(100)
john.right(90)
john.color('red')
john.forward(100)
john.right(90)
john.color(0, 255, 255)
john.forward(100)
john.right(90)
john.color('purple')
john.forward(100)
john.right(90)
john.color(255, 105, 180)
```

執行結果

Requirement already satisfied: ColabTurtle in /usr/local/lib/python3.10/dist-packages (2.1.0)



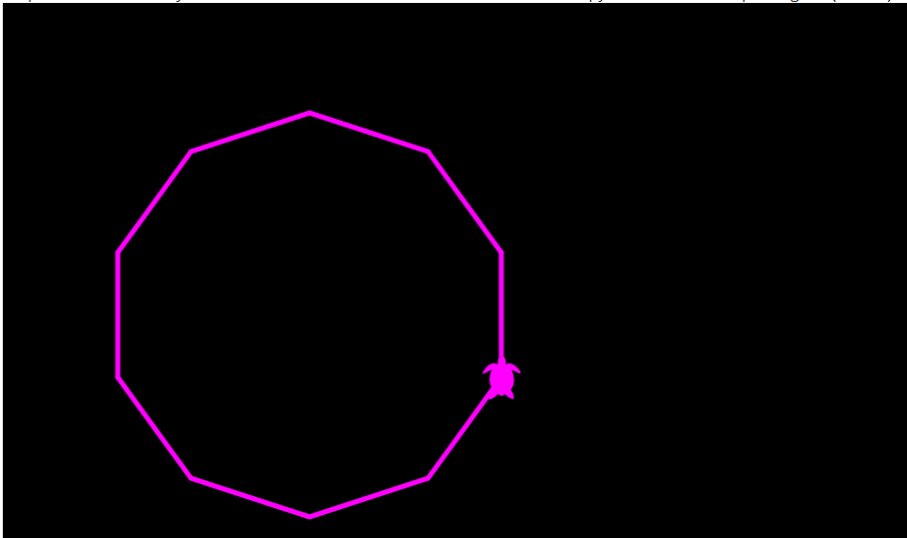
程式碼

```
▶ !pip3 install ColabTurtle
import ColabTurtle.Turtle as john
john.initializeTurtle(initial_window_size=(800, 600))

for i in range(10):
    john.color(255, 0, 255)
    john.pendown( )
    john.forward(100)
    john.left(36)
```

執行結果

▶ Requirement already satisfied: ColabTurtle in /usr/local/lib/python3.10/dist-packages (2.1.0)



程式碼

```
▶ !pip3 install ColabTurtle
import ColabTurtle.Turtle as john
john.initializeTurtle(initial_window_size=(800, 600))

for i in range(4):
    john.color(176, 196, 222)
    john.pendown( )
    for j in range(5):
        john.forward(30)
        john.right(72)
    john.penup( )
    john.forward(70)
```

執行結果

▶ Requirement already satisfied: ColabTurtle in /usr/local/lib/python3.10/dist-packages (2.1.0)

