

四、本教材資訊科技篇的特色如下：

- (一) 章節結構嚴謹，能充分反映國中資訊科技課程綱要的學習重點。
- (二) 重視運算思維歷程，培養學生能面對問題、提出問題、分析問題、並解決問題的能力。
- (三) 圖文並茂讓課本內容生動活潑，能吸引學生注意並引起學習興趣。
- (四) 簡明扼要，以保留資訊科技發展快速之內容的適時性。
- (五) 強調實/操作，除可避免傳統記誦教學外，也可減輕學生學習的負擔，同時可提升學生的學習動機，維持學習的興趣。
- (六) 具有彈性，所有硬體及軟體未明確指定設備或環境，此種考慮也是為了增加適時性，選擇教學當時最適當的教學及學習環境。
- (七) 強調合作與共創學習，以培養學生團隊合作的態度與精神。
- (八) 視學習成果，主要考慮維持學生的學習動機與學習成就。
- (九) 配合學生之認知能力發展，個人的生活及學習經驗，使學生在課堂中能愉快且自然地學習資訊科技之基本能力及技能。

五、本教材期望能夠讓教師在授課時，能夠有豐富的參考資源，依據自己的教學需求，選用適切的活動或內容以進行教學。此外，學生也能夠把本教材當作是一本最棒的工具書，當遇到日常生活中的相關問題時，可以隨時拿出來查詢，以協助他們解決問題。

六、本教材如有未盡周詳之處，敬請各校教師隨時提供意見，以供修訂時之參考，不勝感激。

阿顯



努力、勇於挑戰。

星兒



溫柔、善解人意。

拜特



熱心、大家的資訊站。

波波



穩重、憨厚又樂於助人。

吉嘎



聽話、喜歡跟著星兒。

資訊科技

第一冊

目次

國民中學

科技

壹 生活科技

為了教導各位同學能夠設計與製作一個機構玩具，本書主要請大家一起團隊合作進行闖關活動，希望在經過七年級的生活科技課程學習之後，能夠協助大家設計與製作出一個科技玩具。

想要設計與製作出一個科技玩具，必須具備做、用、想的能力，為了按部就班的引導大家學習以具備這些能力，七年級主要設計六個關卡。

相信大家已經準備好且迫不及待的想開始學習了！請大家一起來**做、用、想**，設計與製作出屬於自己的科技產品。

關卡 介紹生活科技教室的使用規範、創意思考與科技問題解決。

1 生活科技導論

挑戰1 生活科技教室使用規範

闖關任務 生活科技教室安全規範同意書

挑戰2 創意與思考

闖關任務 我是創意大師

挑戰3 科技問題解決

闖關任務 設計與製作氣球車

關卡 介紹什麼是科技、科技的重要性、日常生活中科技產品的選用原則。

2 認識科技

挑戰1 看見科技 I see you

闖關任務 新興科技大探索

挑戰2 建立科技系統的概念

闖關任務 科技系統網路大解密

挑戰3 探索科技的發展與影響

闖關任務 垃圾處理停看聽

挑戰4 聰明的科技產品選用者

闖關任務 選用科技產品小達人

關卡 學習視圖與製圖能力、基本電腦輔助設計，及手工工具的操作與使用方法。

3 設計與製作的基礎

挑戰1 無所不在的視圖與製圖

闖關任務 我是小小工程師

挑戰2 電腦輔助設計與應用

闖關任務 3D 繪圖大師

挑戰3 處處可見的工具

闖關任務 製作微型椅

闖關任務 我是遊戲製作大師—彈珠臺

生活科技相關競賽介紹

 做一做

自己動手操作與觀察，培養「做」的能力。

 用一用

查詢相關資料或報告分享，學習使「用」科技產品。

 想一想

用腦力激盪設計與批判科技，培養「想」的能力。

 補給站

課程內容的延伸補充，增強知識的深度與廣度。

 闖關任務

依條件進行闖關，透過做、用、想解決問題，並分享成品。

使用說明



資訊科技 貳

第 1 章	資訊科技導論	140
1-1	資訊科技與人類生活	142
1-2	資訊科技發展簡史	144
1-3	個人電腦及其周邊設備	148
1-4	資訊科技與問題解決	150
1-5	資訊科技及其相關議題	152
1-6	資訊科技與跨領域整合	158

第 2 章	基礎程式設計(1)	162
2-1	認識演算法與程式語言	164
2-2	Scratch 程式設計 — 基礎篇	172
2-3	Scratch 程式設計 — 計算篇	184
2-4	Scratch 程式設計 — 繪圖篇	201

第 3 章	資料處理與分析	212
3-1	資料的形式與意義	214
3-2	資料搜尋	216
3-3	資料處理與分析工具	222

附錄

生科圖片來源	
資訊圖片來源	249
資訊中英文對照	250
生科 Onshape 繪圖軟體操作手冊	
附件 1 生科三角格紙	
附件 2 生科方格紙	
附件 3~6 生科三視圖	

使用說明

小知識

課程內容的延伸補充，增強知識的深度與廣度。

小提示

針對陌生、易產生迷思的概念，進行補充及說明。

分析

有系統的歸納重要觀念、運算、程式…等。

範例

活用課文中提到的概念去延伸學習。

問題拆解

將範例問題做步驟式提問，引導學生思考。

第 1 章

資訊科技導論

歡迎來到資訊科技的舞臺，這是見證資訊科技神奇發展與變化的時刻。本課程將會學到有關於資訊科技的知識及應用，這些都是身處資訊社會應該要具備的知能。

本章為導覽式章節，首先說明資訊科技與人類生活的關係，並介紹資訊科技發展簡史與個人電腦及其周邊設備。接著扼要說明運算思維與問題解決的關係，並舉例說明。隨後將資訊科技與媒體所衍生的諸多社會相關議題做簡要說明。最後強調 STEAM 學科領域的關係，以及跨領域整合的重要性。

- 1-1 資訊科技與人類生活
- 1-2 資訊科技發展簡史
- 1-3 個人電腦及其周邊設備
- 1-4 資訊科技與問題解決
- 1-5 資訊科技及其相關議題
- 1-6 資訊科技與跨領域整合





1-1 資訊科技與人類生活

資訊科技（Information Technology，簡稱 IT）也稱資訊及通訊科技（Information and Communication Technology，簡稱 ICT），是指用於管理及處理資訊所採用各種技術的總稱。由於它的功能日趨強大，操作親和性逐步提高，已經發展到一般民眾都可以輕易使用的地步。各種產業為了提高運作的效率，也不斷的進行資訊化。資訊化的結果帶給人們在生活及工作上非常多的便利，例如：人際溝通、線上訂票、醫療掛號、網路購物、銀行存提款及轉帳、線上學習（圖 1-1），以及線上直播等，不勝枚舉。



▼ 圖 1-1 在資訊時代，透過線上學習，隨時隨地可獲得不同領域的知識。



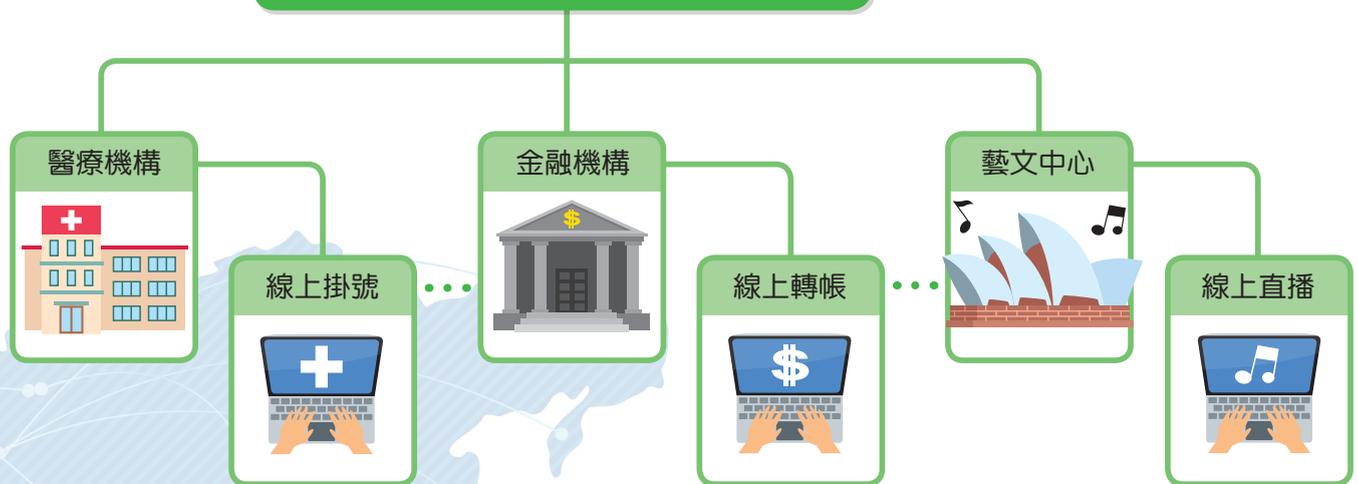
現在我們就要透過本課程，有系統的學習資訊科技的基本原理與技術。同時也要探討因受資訊科技的影響，人們如何去面對、調適及改變既有的觀念，進而適應資訊時代的生活與工作。從以下的示意圖，我們可以明顯的感受到工作與生活方式時時在改變中。

小知識

Apps

Mobile Application 的簡稱，中文稱行動應用程式，是為智慧型手機、平板電腦或其他行動裝置設計的應用程式。

資訊科技影響各種行業的運作方式

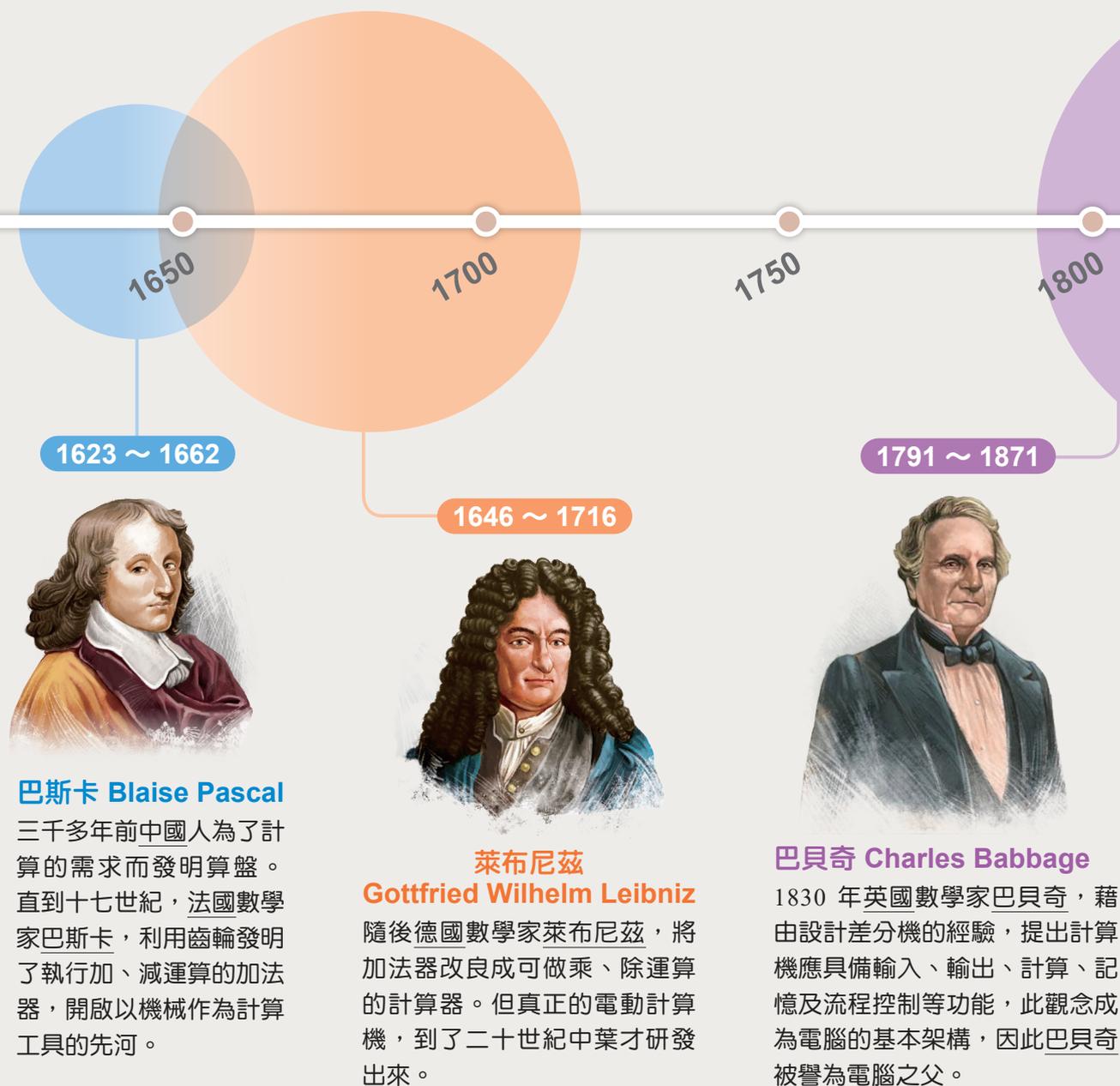


1-2 資訊科技發展簡史

資訊科技對人類社會的影響既深且廣，它的發展主要是經由前人的創意與不斷努力，才有今日的成果。了解資訊科技發展的簡要歷史，相信會帶給我們很多的啟示。

資訊科技的發展，可以分別從計算工具及電子元件的發展來說明，首先來看計算工具的發展簡史。

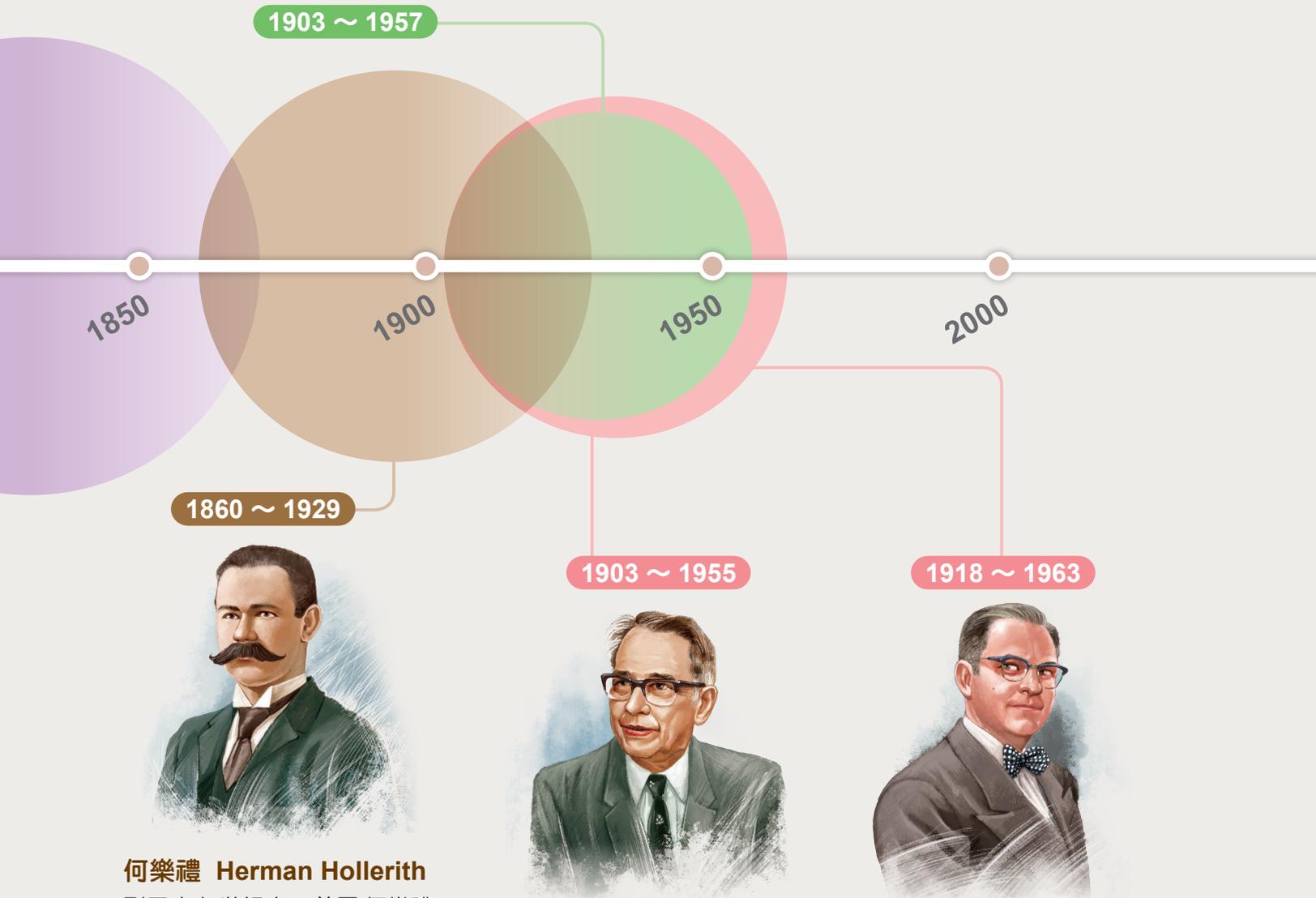
1-2-1 計算工具的發展





馮紐曼 John von Neumann

1945 年，匈牙利裔美國 普林斯頓大學的馮紐曼提出二進制取代十進制及將程式儲存在記憶體內的概念。奠定了日後電腦在工作前，先將程式載入記憶體的理論架構，對現代電腦結構影響甚大，也因此被譽為現代電腦之父。



何樂禮 Herman Hollerith

到了十九世紀末，美國何樂禮為了處理大量人口普查資料，發明打孔卡片，成為最早的自動化資料處理工具。



阿塔納索夫

John Vincent Atanasoff

在 1938 年間，阿塔納索夫和貝理合作研發出全球第一部電腦（Atanasoff-Berry Computer，簡稱 ABC 電腦），為接下來自動化電子數位電腦的發展奠定基礎。



貝理

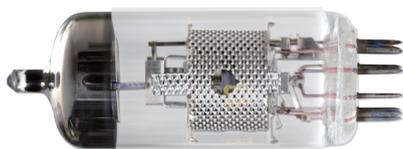
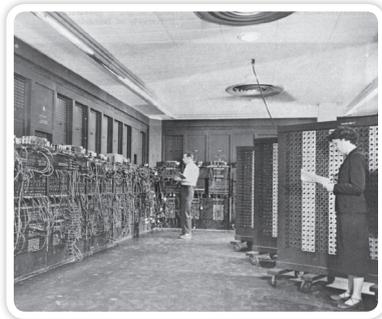
Clifford Edward Berry

1-2-2 電子元件的發展

計算機要靠電子元件發揮功能才能運作。從發明真空管開始，大約經過半世紀，現在已逐步進入人工智慧的時代。從以下圖示，我們可以看到不同世代的進展及功能的演進。

第一代電腦
約 1946~1958

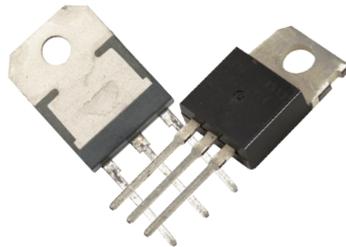
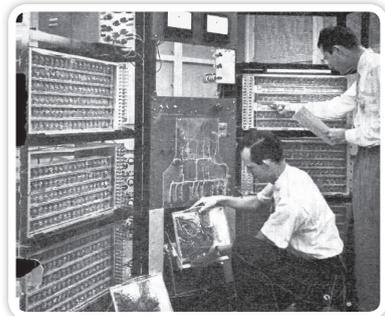
1946 年美國賓州大學完成了以真空管組成的計算機，這是全球第一部使用電子元件組成的電子數位電腦（圖 1-2）。



▲ 圖 1-2 真空管 (Vacuum tube)

第二代電腦
約 1959~1963

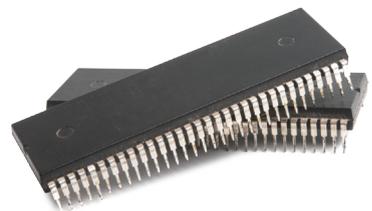
以真空管組成的電腦體積十分龐大，耗電量高，且記憶體容量少，因此又研發第二代電腦，以耗電少、體積小的電晶體來處理訊號（圖 1-3）。



▲ 圖 1-3 電晶體 (Transistor)

第三代電腦
約 1964~1971

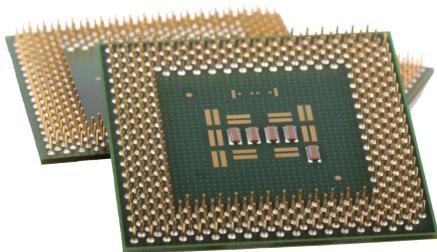
演進到第三代電腦，則使用體積更小的積體電路 (IC)，一小片 IC 的功能就相當於數百個真空管來處理訊號（圖 1-4）。



▲ 圖 1-4 積體電路
(Integrated Circuit, 簡稱 IC)

第四代電腦 約 1972~1989

到了第四代電腦，則利用比 IC 更小，而功能更強的超大型積體電路（VLSI）來處理訊號（圖 1-5）。

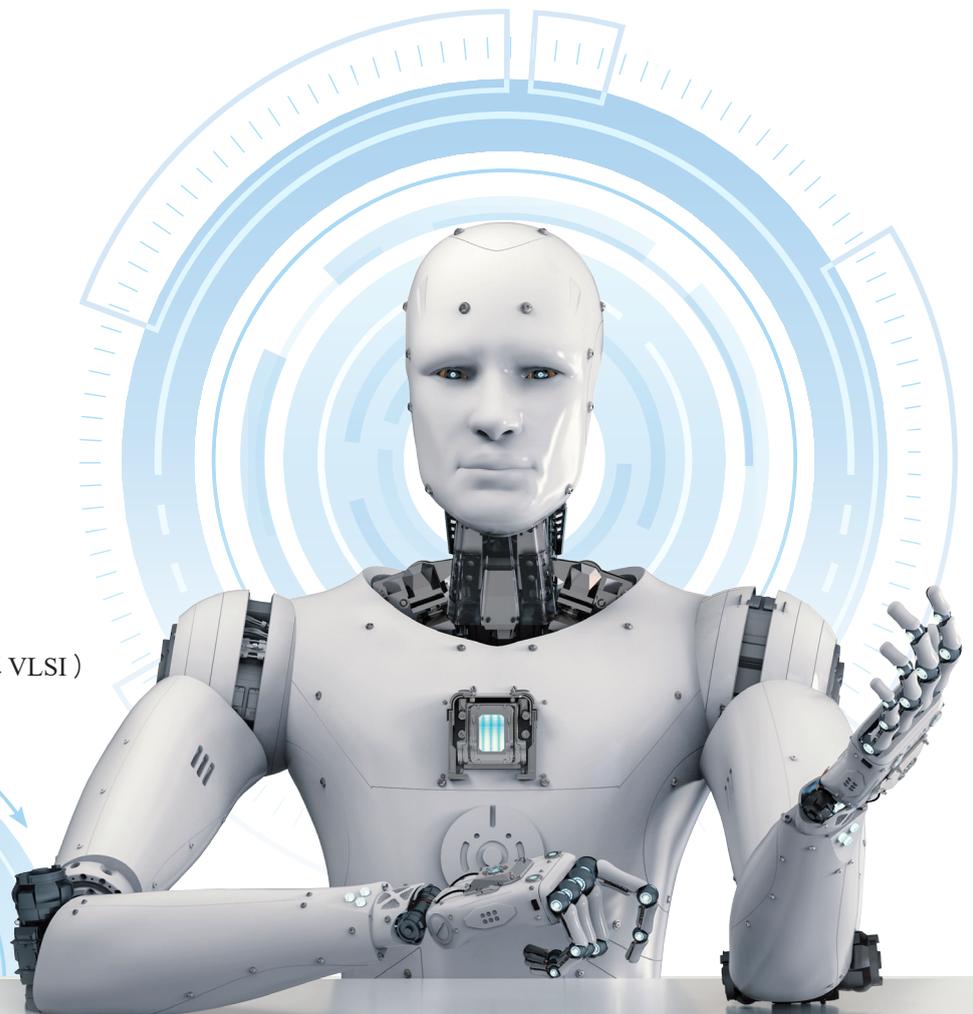


▲圖 1-5 超大型積體電路
（Very-large-scale Integration，簡稱 VLSI）

第五代電腦 約 1990 以後

目前資訊界正在積極研發具有更多功能—處理聲音與影像、記憶且有認知能力的智慧型電腦。資訊科技也邁入人工智慧（AI）時代，並持續發展中（圖 1-6）。

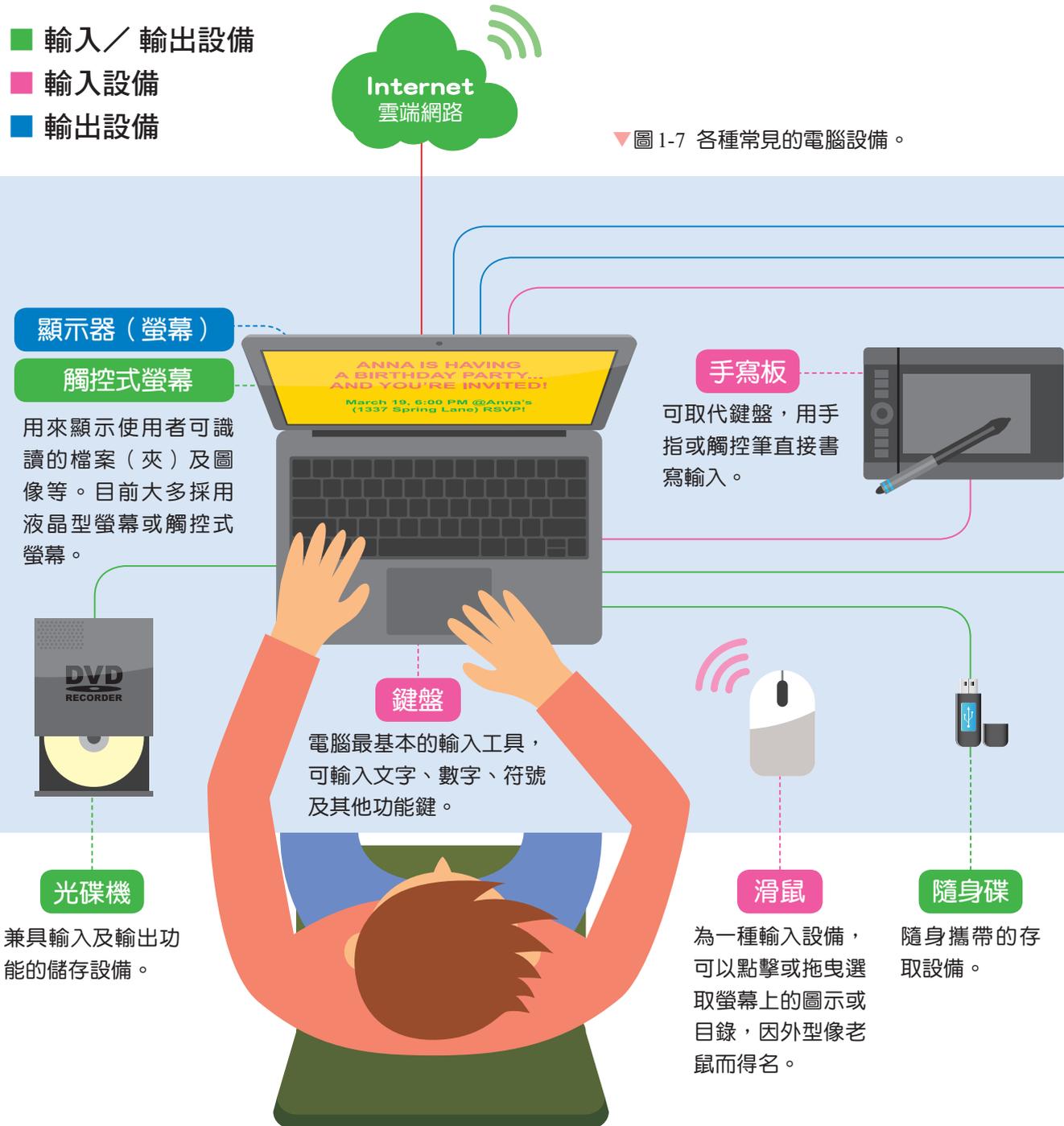
▼圖 1-6 人工智慧示意圖
（Artificial Intelligence，簡稱 AI）



1-3 個人電腦及其周邊設備

身處資訊與網路社會，人們在生活及工作上，經常需要借助電腦及網路來處理很多事情。一般在居家、學校或上班族的工作場所，大概都可以看到如圖 1-7 所示的電腦主機及其周邊設備與網路。由電腦主機連接出來的裝置，可分為輸入設備、輸出設備、以及輸入／輸出設備（合稱為周邊設備）。

- 輸入／輸出設備
- 輸入設備
- 輸出設備



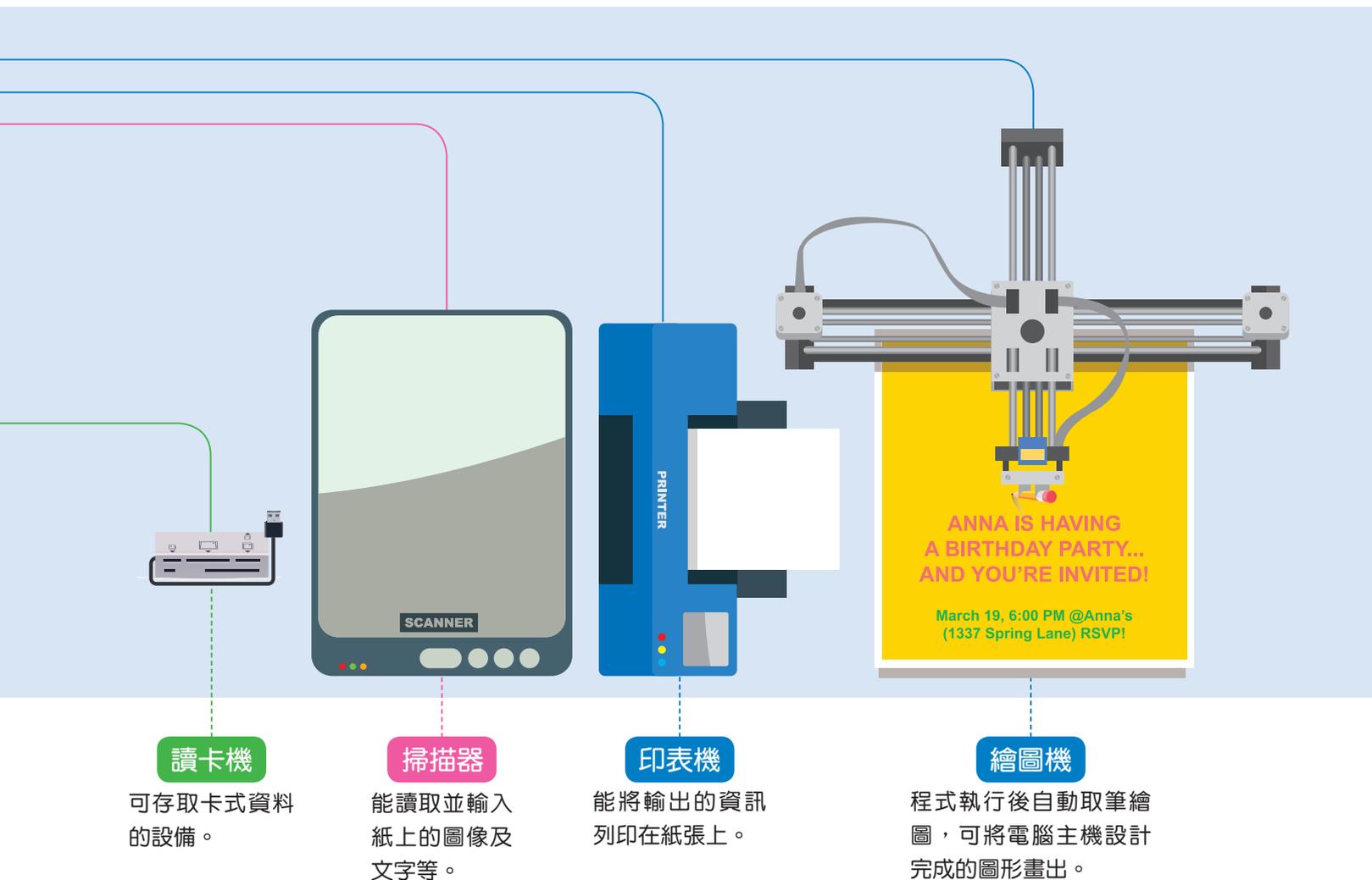
▼圖 1-7 各種常見的電腦設備。

在電腦作業環境裡，主機當然是必要設備，周邊及網路則是選擇性的設備，這麼多種周邊設備視工作需要再決定取捨。至於其他設備，如平板電腦及行動載具等，本課程將會另外安排專章（例如：系統平臺的發展、組成架構、與基本運作原理等）討論。而有關網路（例如：網路技術的概念與網路服務等）同樣有專章討論。

小知識

隨身碟

全名為 USB 隨身碟（USB Flash Drive），是一種存取的設備。一般的隨身碟體積小、重量輕、可重複寫入，問世後迅速普及並取代傳統的軟碟片（floppy disk）及軟碟機（floppy drive）。



1-4 資訊科技與問題解決

學習資訊科技並非只著重於操作或技術，更重要的是要進一步運用學會的技能來解決生活上及學習中遇到的問題，因此動腦思考如何解決問題更為重要！

1-4-1 問題解決的思維模式

人們身處資訊時代，各種事物與現象瞬息萬變，所產生的問題也日趨複雜，依過去的知識已不足以解決問題，因此找出解決問題的方式是我們必須具備的能力。

隨著資訊科技的發展，運算思維（computational thinking）的概念應運而生。運算思維是一種解決問題的思維模式，先是拆解、分析問題，再將解決問題的方法步驟化（圖 1-8），而學習演算法、程式設計與資料處理等課程，就是為了培養問題解決的能力。

在資訊科技領域，演算法是指運算的具體步驟，也就是為了解決問題所設計的一套有限運算規則的集合，而程式設計就是依邏輯順序安排指令以實踐這套運算規則。資料處理則是運用多種運算規則，來分析數據資料，以展示資料意義的方法。

由於演算法具有步驟化的特性，非常適合用來釐清看來毫無頭緒的問題。因此在日常生活的應用上，可以把演算法視為解決問題的方法，並利用流程圖或文字敘述的方式來表示解決問題的步驟。

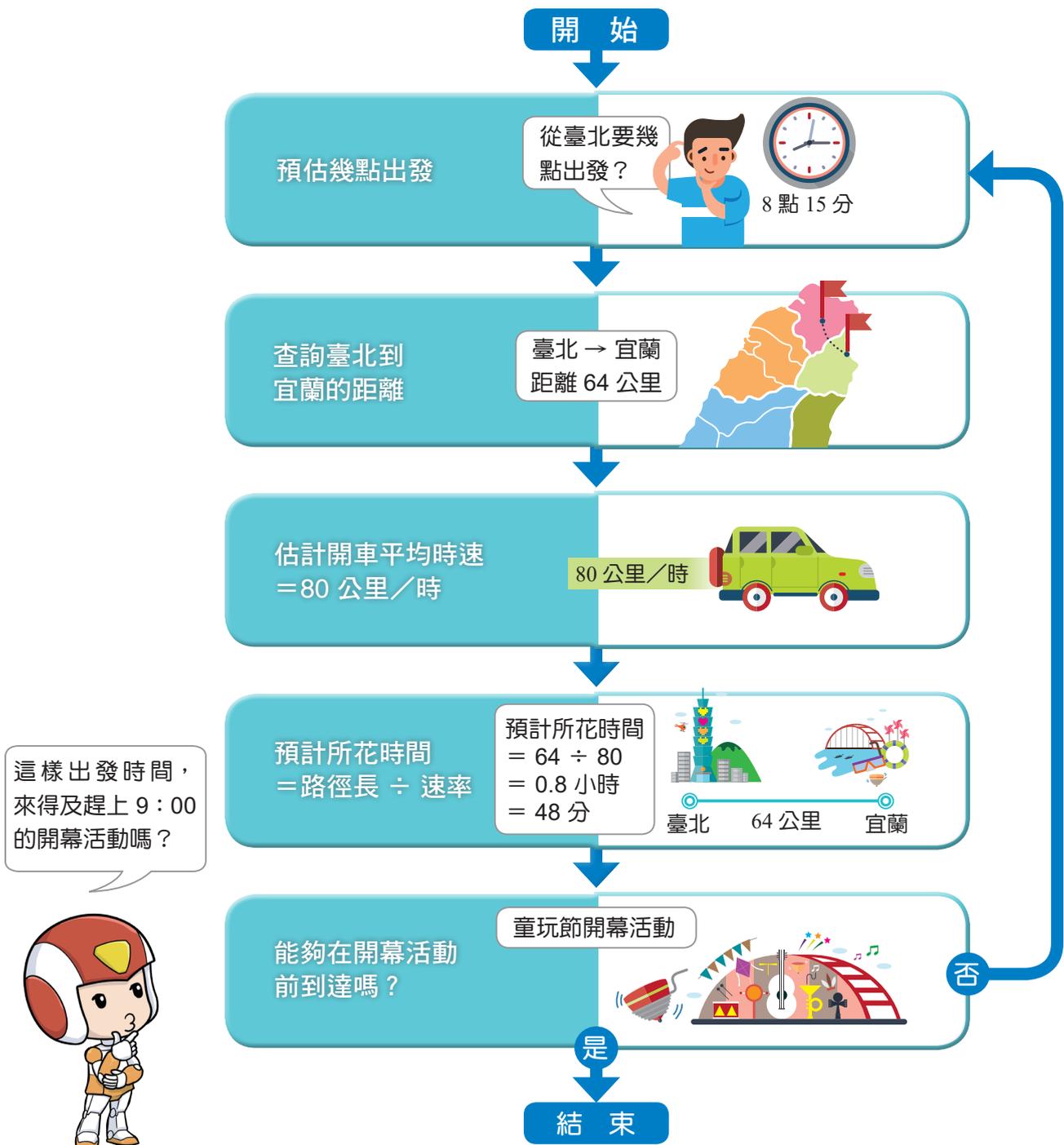
如今在資訊時代裡，日常生活及工作中常需依賴電腦，因此學習與電腦相關的思維模式與解決問題的方法變得很重要，而程式設計課程可以培養我們用電腦來解決問題，藉由程式設計及資料處理，讓大家在實作的過程中，學習解決問題的具體方式。



◀ 圖 1-8 利用運算思維解決生活中的問題。

1-4-2 解決問題示例

小明全家從臺北開車到宜蘭，參加上午 9：00 的童玩節開幕活動，如果希望爸爸以 80 公里／時的速率開車，行駛的距離是 64 公里，若是 8：15 出發，在沒有塞車的情況下，能否在開幕活動前到達？如果不能，你要如何處理？



1-5 資訊科技及其相關議題

資訊科技對人類生活的影響既多元又廣泛，雖然讓我們生活與工作更加便利，但也衍生出許多問題。因此，養成正確的習慣與態度同樣重要。接下來，簡要介紹下列相關議題。

1-5-1 資料保護及資訊安全

資訊安全是與資訊系統及使用者關係最密切的議題（圖 1-9），例如：散布電腦病毒、非法入侵他人網站、竊取個人資料等，都涉及到系統及個人的資訊與資料安全。因此，維護系統安全及保護個人資料極為重要，本課程將於下學期第 4 章加以探討。



▲ 圖 1-9 安全認證系統可以保障大多數使用者的權益。

1-5-2 數位著作合理使用原則

資訊科技的普及，使各種紙本資料及檔案邁向數位化，這些透過資訊科技的技術而產生的創作就是數位著作，加上網路的發達也使取得數位著作變得容易。因此，身處資訊社會，使用數位著作是能夠快速獲得資訊或知識的機會，但只有在不違反法律（如著作權法）規定下，才是合理使用的原則，本課程將於下學期第 6 章加以探討。

1-5-3 資訊倫理

資訊倫理是數位公民態度的展現，例如：尊重隱私權、著作與所有權、培養得體的網路禮儀與遵守網路社群規範等，都是資訊倫理的議題（圖 1-10）。除此之外，資訊近用權（Accessibility）也是資訊倫理的議題，當個人在社經地位或居住環境等有差異，以致近用數位產品的機會不等，就會產生數位落差，本課程將於八年級上學期第 1 章加以探討。



▲圖 1-10 勿濫用方便、快速的科技。

1-5-4 資訊科技與相關法律

資料或資訊因在數位媒體及網路上非常容易進行交換、散布、修改或複製，造成侵犯著作權及隱私權的事件層出不窮，此部分可以用著作權法及個人資料保護法等加以規範（圖 1-11）。此外，因為網路發達，利用電腦或網路犯罪的方式愈來愈廣泛，但兩者在法律的規範上略有不同，本課程將於八年級上學期第 3 章加以探討。



▲圖 1-11 隨便使用或轉貼網路上的圖文，非常容易觸犯相關法律。

1-5-5 媒體與資訊科技相關議題

媒體包括平面媒體（如報紙、雜誌）及電子媒體（如廣播、電視）等大眾媒體（圖 1-12、1-13）。近年來因網路及雲端科技的快速發展，又產生了新興的媒體，如影音分享平臺（YouTube）、直播（live broadcast）等（圖 1-14 ~ 1-16）。此種發展對媒體傳播資訊的方式起了很大的變化，過去大部分的民眾都是被動的接受傳播媒體發布的資訊；而現在人人可以發布資訊，擁有散布資訊的主動權。也因為如此，資訊的產生日趨多元且複雜，媒體素養（media literacy，也稱媒體識讀）的概念也隨之改變。

小知識

YouTube

中文可稱為影音分享平臺，源自於美國的影片分享網站。研發者的原意是為了方便朋友之間分享影音片段，後來逐漸演變成網友的影音作品發布平臺與儲存的場所。使用者可以主動上傳、分享、觀賞、甚至評論，充分發揮了網站的口號：“Broadcast Yourself”（展現你自己）！

直播

直播是指直播主在某現場透過媒體將自己的影音即時傳遞給網際網路上的網民收看。網路直播發揮網際網路的優勢，利用視訊方式進行線上現場直播。直播的特色是即時性，又因為現場播出無法同時剪接及後製，可以呈現最好的真實性。



▲圖 1-12 平面媒體



▲圖 1-13 電子媒體

媒體素養意指能善用傳播媒體的能力。更詳細的說，泛指使用者在使用媒體時，能理解、詮釋傳達的資訊，也能夠透過媒體產生有用的資訊。做為一位資訊社會的數位公民，不論是資訊的接收者或是提供者，具備充分的媒體素養，將能適應數位社會的生活，也是資訊社會不可或缺的能力。

在便捷的數位環境中，生活及工作更方便，社會也更加開放，但卻衍生很多新的議題，例如：假資訊充斥、網路霸凌等，而這些都是值得關注的問題，本課程將於八年級下學期第 5 章加以探討。

▼圖 1-16 社群媒體以多種不同的形式呈現，包括文字、圖像、影片、音樂等。



▲圖 1-14 閱讀部落格文章



▲圖 1-15 觀看 YouTube 影片

1-5-6 常見資訊產業的特性與種類

資訊產業（information industry）也稱 IT 產業（information technology industry），是一種技術、智慧、知識密集型的產業，早期稱為資訊工業或電腦工業，此種產業是指與電腦或資訊有關的軟硬體行業的總稱。在眾多產業中，諸多國家也常將資訊產業列為重點發展項目，可見其受重視的程度。

軟體設計



網路通訊



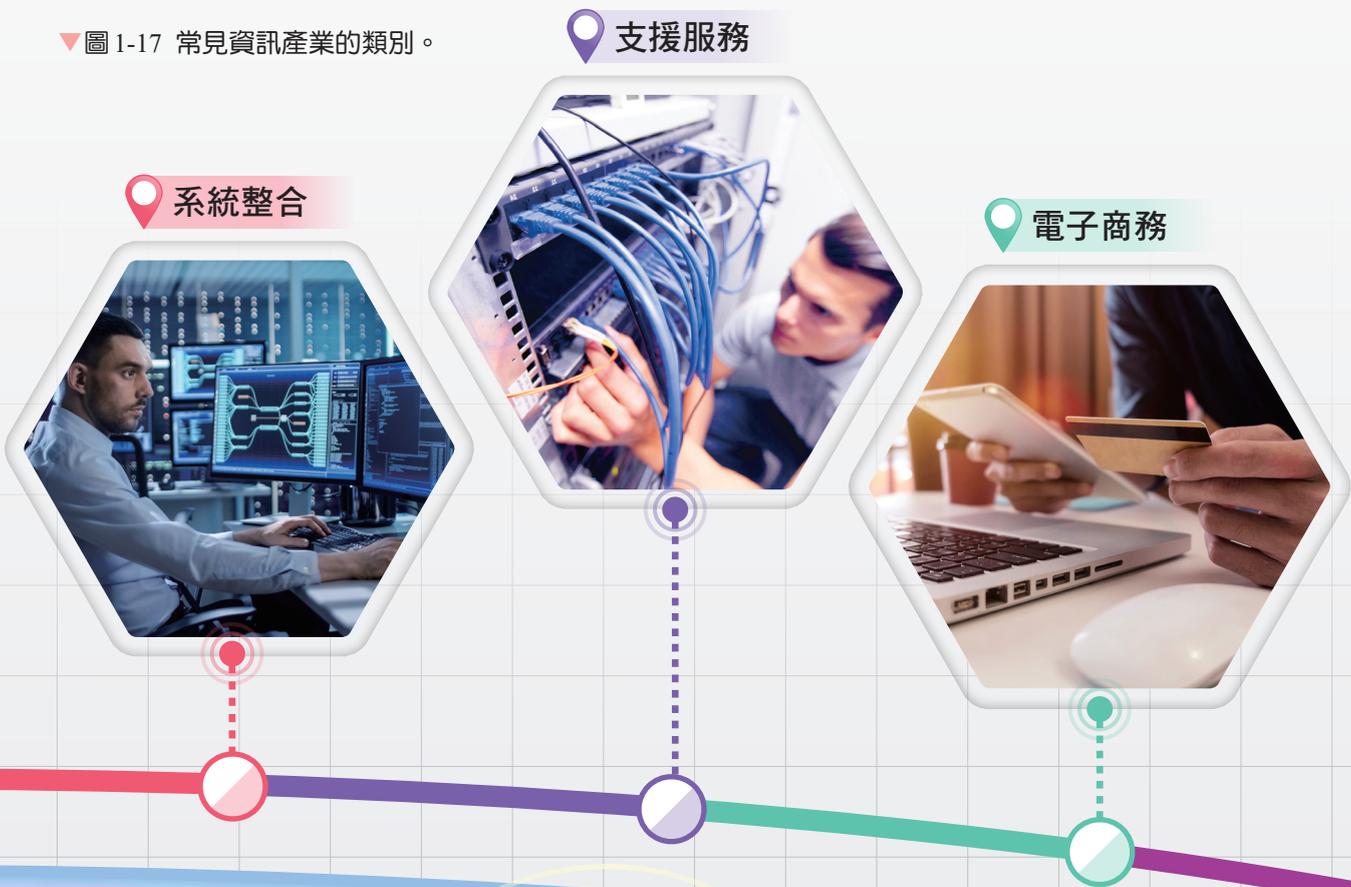
硬體製造



一個國家的 IT 產業發展，往往會帶動其他產業的數位化。各種產業數位化的結果，常會讓產業產生組織及營運模式的改變，促進產業升級，創造異於傳統的商業模式。這些改變及創新，將會使經濟活動更活絡、社會活動更多元、民眾生活方式更多樣化。

IT 產業可以依其特性，大致分為硬體製造、軟體設計、網路通訊、系統整合、支援服務、以及電子商務等類別（圖 1-17）。而相較於其他產業，資訊產業具有對其從業人員素質要求高、產業間競爭劇烈、產品生命週期短、以及產業營運國際化程度高等特徵。本課程將於九年級下學期第 6 章進一步討論。

▼圖 1-17 常見資訊產業的類別。



我想選的科系和未來的資訊發展有什麼關聯呢？



1-6 資訊科技與跨領域整合

1-6-1 資訊科技與 STEM / STEAM

學習資訊科技的知識及技能，培養問題解決的能力，也會同時增進學習相關領域；事實上資訊科技本身就具有科學（S）、科技（T）、工程（E），及數學（M）的性質，也可視為是 STEM 集合體。STEM 旨在整合科學、科技、工程與數學不同學科知識，其重點在於對問題之思考與應用能力，並創造解決問題的方法。後來又加上藝術（A），成為 STEAM（圖 1-18）。在學習過程中，STEM 或 STEAM 其精神在於跨領域整合，透過團隊合作、溝通協調，以引發思考與創意。

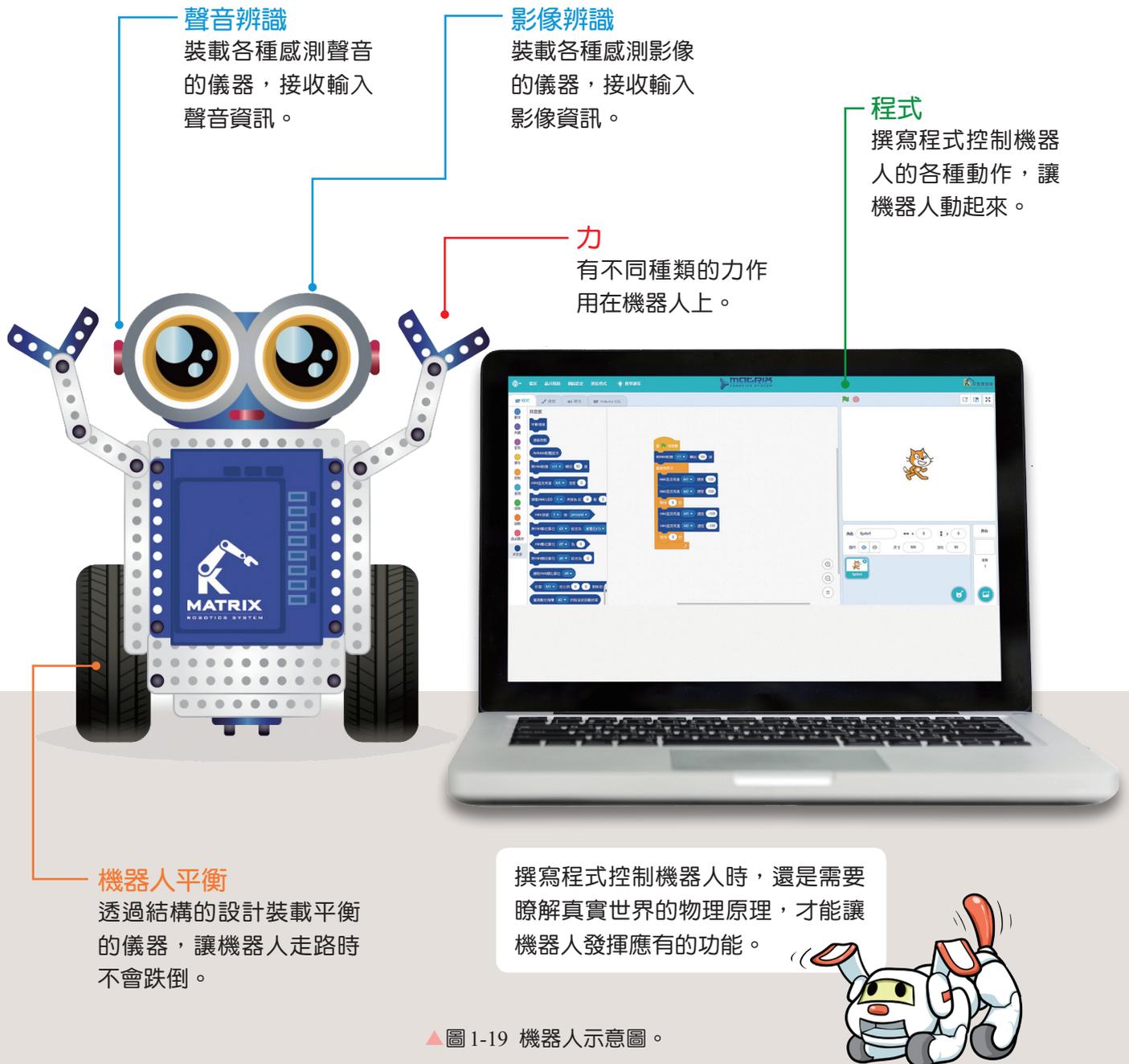
在基礎程式設計模擬篇，就會學到如何利用程式來模擬電子琴演奏，這就是跨領域整合的例子。現代社會的學術研究及各種行業，都非常強調跨領域整合，現在有資訊科技助一臂之力，讓整合變得更加容易。



▲圖 1-18 STEAM 的學習領域

1-6-2 資訊科技與跨領域整合—以機器人為例

以下就用機器人示意圖為例（圖 1-19）來顯示資訊科技與其他領域的整合，由此我們也可以了解跨領域整合，在未來學術或科技發展非常重要。



從上圖的標示，可以看到機器人的製作，需資訊科技跨領域（如聲音、影像、程式語言、物理學及機械工程等）的整合，才能順利完成，這也是新課綱強調合作共創的原因。

重點
回顧

本章名為資訊科技導論，顧名思義，在為所要學習的內容做導引。首先，概括說明資訊科技（IT）是指用於管理與處理資訊所採用各種技術的總稱，也稱為資訊及通訊科技（ICT），IT 與 ICT 兩者常被交換使用。資訊科技的內涵多元且廣泛，發展變化又極快速，不容易對此概念下一個周延的定義。本課程依照課綱引導各位學習資訊科技的主要內涵。透過本課程，逐步學習常用資訊科技的基本原理與技能，也將能逐漸了解其意義及內涵。

回顧資訊科技的發展簡史，特別是計算工具及電子元件的發展，將會讓我們了解資訊科技是智慧及創意的結晶。目前資訊界正在積極研發具有更多功能－處理聲音與影像、記憶且有認知能力的智慧型電腦。資訊科技也邁入人工智慧（AI）時代，並持續發展中。



看完了本章內容，是否對於生活中資訊科技的應用以及歷史有更多的了解呢？

學習資訊科技並非只著重於操作或技術，更重要的是要運用學會的知能來解決問題，因此動腦思考如何解決問題更為重要！也就是遇到問題要思考如何分析問題，並將解決問題的方法步驟化。

資訊科技的應用雖然讓我們生活更加便利，工作更有效率，但也產生許多社會議題。因此，本課程也要同時探討諸如資料保護及資訊安全、數位著作合理使用原則、資訊倫理、資訊科技與相關法律、媒體與資訊科技等相關議題，同時也要了解常見的資訊產業的特性與種類。以上所列，本章都扼要說明其意涵，此可視為往後章節的引文。

本章最後以資訊科技與 STEM 或 STEAM 的關係，以及機器人示例做結尾，目的是要提醒大家，現在不論是學習或是產業發展，都很強調跨領域整合，這也是本課程強調要培養合作共創的能力。

第 2 章就要來學習有趣的程式積木，想想看，你應該要先具備哪些能力呢？



第 2 章

基礎程式設計(1)

身處資訊時代，各種事物與現象瞬息萬變，若還是依循過去的知識已不足以解決現在日趨頻繁且複雜的問題，因此問題解決是我們必須具備的能力。隨著資訊科技發展，運算思維概念應運而生。簡單的說，運算思維是一種解決問題的思考模式，也就是思考如何分析問題，並將解決問題的方法步驟化。

在資訊時代，幾乎人人日常生活及工作中都在使用電腦，因此，學習電腦相關的思考模式與解決問題的方式非常重要，而程式設計正可以培養我們解決問題的能力。

本章將先扼要介紹演算法及程式語言等基本概念，接著設計了一套易學的 Scratch 課程，讓同學在實作的過程中，體驗視覺化程式設計的樂趣，進而在不知不覺中透過運算思維的歷程，解決實際的問題。



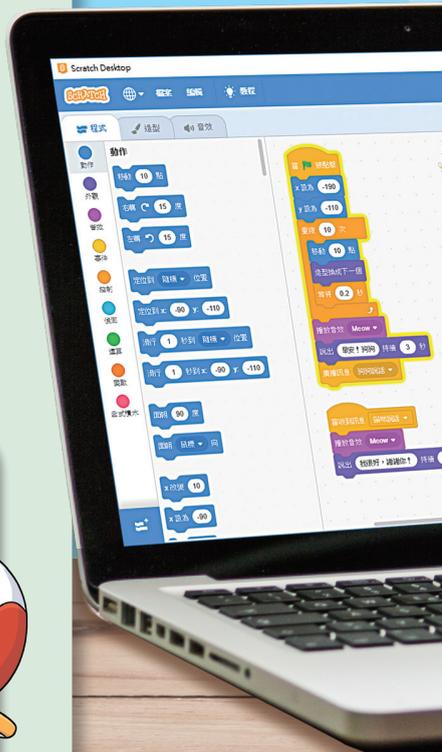
※本教材使用 Scratch 3.0 版本，因 Scratch 會不定期更新，介面可能與本教材略有不同。

2-1 認識演算法與程式語言

2-2 Scratch 程式設計 - 基礎篇

2-3 Scratch 程式設計 - 計算篇

2-4 Scratch 程式設計 - 繪圖篇



問 輸入平時表現成績 並等待

數 平時表現 ▾ 設為 詢問的答案

變數 和 ▾ 設為 和 + 數字

並等待
詢問的答案

並等待
詢問的答案

為 $A + B / 2$

均數是 平均數 持續 2 秒

重複 10 次
移動 10 點
造型換成下一個
等待 0.2 秒

和 + 數字

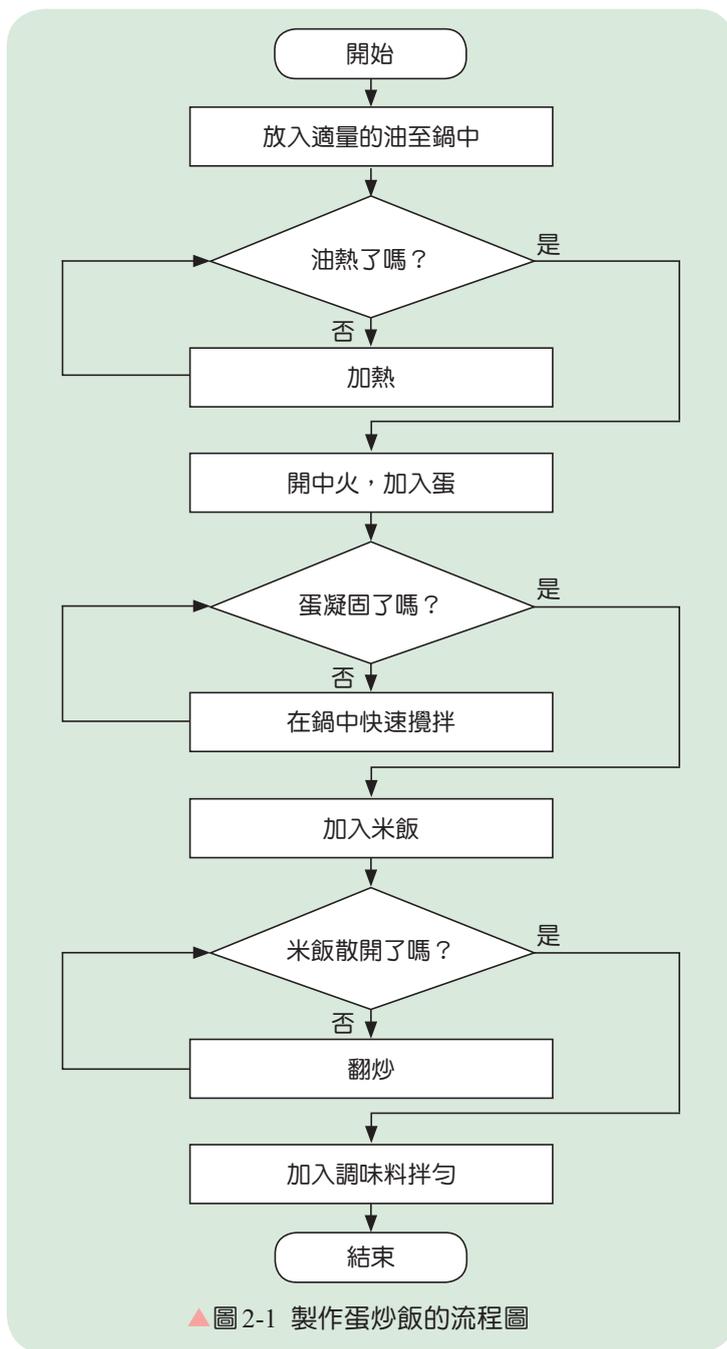
重複 N 次



2-1 認識演算法與程式語言

在資訊科技領域，演算法是一種解決問題的方法，程式語言則是實踐演算法的工具，兩者相輔相成，在此一併討論。

在日常生活中，食譜也類似一種演算法，例如：我們可以把蛋炒飯的步驟畫成流程圖來表示（圖 2-1）。但食譜和電腦的演算法不盡相同，最主要的差別在於食譜不夠精準，會因不同人的解讀而有不同的結果，而電腦的演算法必須表示得很精確，不容許有模糊空間。



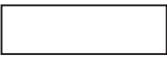
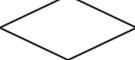
一般來說，食譜雖然把製作步驟描述得很詳細，但是不同人來解讀與實作還是有很大的差異，即使是同一個人在不同時間實作，其結果也會有不相同的品質，這個現象尤其容易出現在新手身上。以上述的蛋炒飯為例，「放入適量的油至鍋中」，而這個適量的敘述就不明確，再者「開中火」，到底是指多大的火候？至於「加入米飯」，使用剛煮好的熱米飯或隔夜冷飯，其結果又會有不同的口感。

2-1-1 演算法的基本概念

我們生活中經常遇到待解決的問題，簡單的問題，憑直覺就可以解決，例如：數字相加，對於熟悉算術的人，根本不是什麼問題；但對沒有學過加法運算的人，就會是一個問題。廣義的說，解決問題的方法就是一種演算法，我們可以把解決這些日常生活問題的方法與過程寫成**步驟**，再依照這個步驟去**執行**。

為了清晰的表示演算法，可把解決問題的步驟整理成符號來表示。美國國家標準學會（American National Standards Institute，簡稱 ANSI）於 1970 年制定標準的流程圖（Flow chart）符號，以方便判讀與相互交流解決問題的步驟，常用的流程圖符號如下表。

流程圖的符號與功能說明

符 號	意 義	說 明
	開始 / 結束	流程圖開始或結束
	處理	處理一項工作
	流程方向	流程進行的方向
	輸入 / 輸出	進行資料輸入或輸出的工作
	決策	依條件比較結果進行不同的處理
	迴圈	迴圈變數初值與終值的描述
	連接	流程的連接點

接下來就以實際的例子，將解決問題的方法與過程，以流程圖來表示。以求任意數的所有因數為問題解決的實例，如果選擇使用窮舉法來解決此問題，就是從 1 開始，針對每一個整數依序進行測試，以找出符合條件的數字。當我們面對一個新問題，又不知道如何下手時，可以考慮先採用窮舉法來求解，問題解決之後，再進一步思考如何改進解決方法。



問題分析

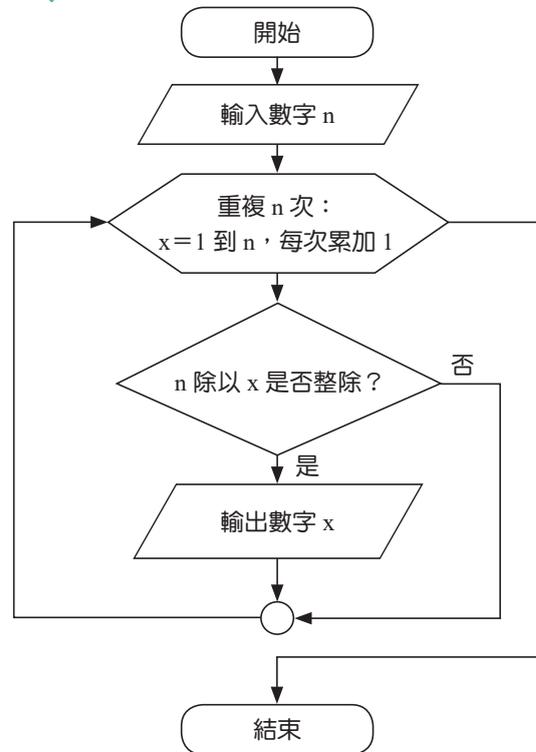
在這個流程圖中，假設使用者輸入的數字 $n=20$ ，接著進入計次式迴圈，迴圈內的敘述會重複執行 20 次。在這個例子中，還有一個迴圈變數 x ：

執行迴圈	處理	判斷	輸出
第一次	$x=1$	20 除以 1 整除	1
第二次	$x=2$	20 除以 2 整除	2
第三次	$x=3$	20 除以 3 沒有整除	沒有執行任何敘述
第四次	$x=4$	20 除以 4 整除	4
⋮			
第二十次	$x=20$	20 除以 20 整除	20

這個例子，很明確的定義了每一個步驟要執行的敘述，不會因為不同人解讀而有不同的結果。



畫流程圖



為了對演算法進行檢驗，必須將演算法轉換成電腦程式，因此最好的途徑是透過程式設計來實作演算法，要學習程式設計則需先選定一種程式語言。由於每個人的思考方式不同，解決問題的辦法也不盡相同，因此設計出來的演算法可能也會不同，但是最重要的，是要考慮演算法的正確性，也就是執行之後是否能產生正確的結果。

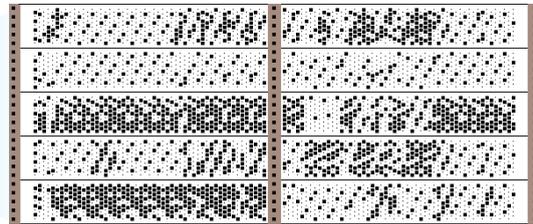
2-1-2 程式語言的基本概念

程式語言發展的歷史遠比電腦來得早，在電腦未發明之前，人們就開始產生編碼的概念。1801年，法國人約瑟夫·瑪麗·雅卡爾（Joseph Marie Jacquard，1752～1834）運用木板打孔的方式，設計出可更改編織圖案的提花織布機（圖 2-2）。他是第一個以程式的概念設計機器的人，此機器展現了兩個程式設計的重要概念：

1. 複雜的設計也可以編譯成機器能了解的程式碼。
2. 依照程式碼指示，機器可不斷重複工作直到完成。



利用設計的孔洞排列，編寫輸入織布的圖案。



▲圖 2-2 提花織布機

而被認為是史上第一位電腦程式設計師是英國人愛達·勒芙蕾絲（Ada Lovelace，1815～1852，圖 2-3），她於 1843 年運用巴貝奇的分析機來計算伯努利數的方法，被認為是史上第一個電腦程式。直到 1940 年代，以電力驅動的現代化計算機出現後，也開啟了各種程式語言的發展。電腦只是一部機器，只要給予指令，它就會照指令執行工作，然後將結果輸出，而這些指令的組合就成為程式。

▶圖 2-3 愛達·勒芙蕾絲



2-1-3 程式語言的演變與發展

最早期的程式是使用 0 與 1 來編寫，稱為機器語言。由於電腦只看得懂 0 與 1，但是人在閱讀與編寫上都很不方便，因此電腦科學家發明了組合語言，使用一些簡單且有意義的文字來撰寫程式，例如：ADD（加）。利用組合語言寫成的程式，必須經由組譯程式的處理，才可在機器上執行。它執行的速度雖然相當快，但仍是相當複雜，必須對電腦硬體結構有相當了解的人才能撰寫，因此被歸類為低階語言（圖 2-4）。

低階語言

低階語言是一種以硬體為導向來描述指令的語言，例如：機器語言與組合語言。雖然一般人比較難看得懂，但是因為它可以直接與硬體的中央處理器溝通，因此執行效能較好。



組合語言

電腦的運算主要是靠中央處理器，而組合語言即是一種與中央處理器硬體結構密切相關的語言，若使用不同指令集的中央處理器，所對應的組合語言的語法也會不一樣。

組譯

機器語言

機器語言是由 0 與 1 兩種符號組成的代碼，它能夠直接被電腦理解並執行，是電腦硬體能夠直接識別的指令集合。但不同電腦硬體架構使用不同機器語言。

轉譯

▲圖 2-4 低階語言經由組譯程式轉換為機器語言。

由於組合語言的撰寫相當費力，而且容易出錯，在 1950 年代科學家又發明了較接近人類思維模式的高階語言（圖 2-5）。經過幾十年的演進，目前已知的程式語言種類很多，既有的程式語言仍有人在使用，但新的程式語言也不斷的出現。

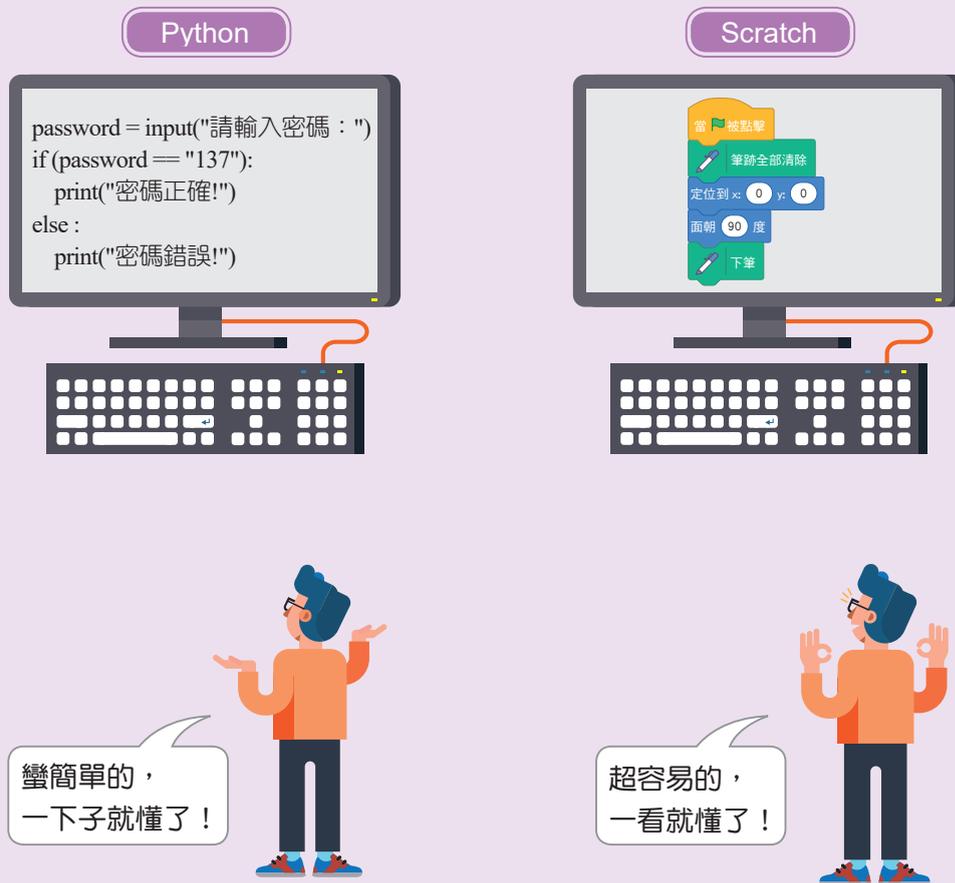
小知識

指令

在低階程式語言（如組合語言），一條要求電腦動作的指示，稱為指令（instruction），如 ADD（加）是一個指令，OR（或）也是一個指令。

高階語言

高階語言是一種語法接近人類語言的程式語言。以高階語言所設計的程式，必須先轉譯成機器語言，才能被電腦硬體執行。

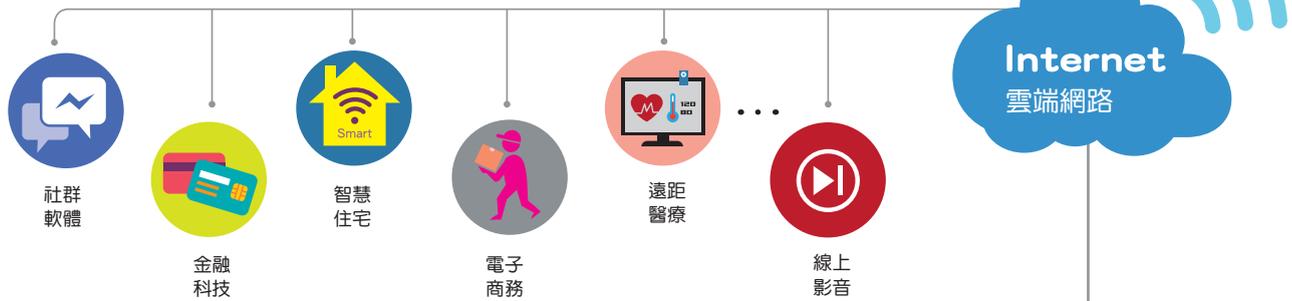


▲圖 2-5 高階語言轉譯為電腦能懂的機器語言。

2-1-4 程式語言的主要功能

綜合前面的介紹，人們為了要指揮電腦完成某項工作，就要寫許多指令，這些指令要按照一定邏輯順序排列，而電腦就依照這種順序接受指令而完成工作。總而言之，這些為完成某項工作而依其邏輯順序寫成的一連串指令，就稱為程式。程式要指揮電腦完成不同的工作，就有各種不同的功能，而主要的功能如下：

- ① 主要的功能就是要啟動（Booting）電腦並分配資源，指揮電腦運作。
- ② 其次是產生人與電腦溝通的介面，讓使用者可以透過介面來操作電腦硬體，因此人與電腦就可以產生互動。



- ③ 把相關的電腦串連起來，特別在網路或是雲端時代，以各種硬體所建構起來的環境，需要靠各種程式發揮功能，將其串連在一起後，讓眾多使用者可以同時在線上互動與溝通。

程式設計師 / 撰寫程式軟體



上傳程式軟體

下載程式軟體

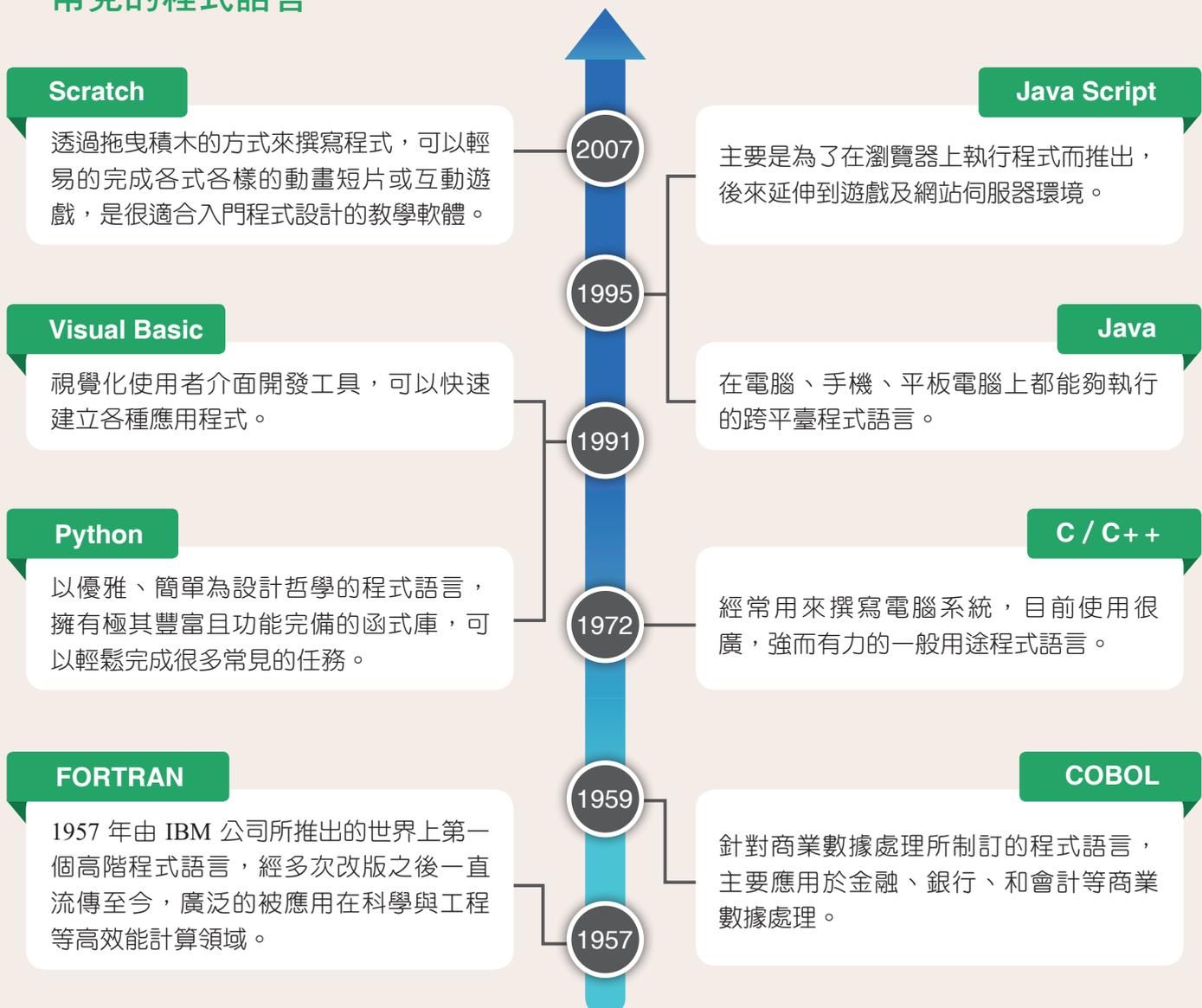


一般使用大眾 / 使用程式軟體

2-1-5 程式語言的應用

從有程式語言至今，目前廣被使用的程式語言種類非常多，但因用途不同，功能也不一樣。大致上，可以分為一般用途及特殊用途。雖然不同程式語言的語法不一樣，但是基本邏輯則類似。一開始可以從一般用途的程式語言入門，建立了程式設計的基本概念後，再因應不同的專業需求，選用特殊用途的程式語言。在本課程中，我們將選用教學用途的 Scratch 語言來學程式設計。

常見的程式語言



2-2 Scratch 程式設計 – 基礎篇

搭配習作附件 7

Scratch 是由美國麻省理工學院媒體實驗室（MIT Media Lab）的終身幼稚園團隊（Lifelong Kindergarten Group）於 2007 年所發表的一套教學軟體。它提供了視覺化的圖形操作介面，學習者只要會用滑鼠拖曳積木，就能輕易的撰寫程式。

Scratch 是一套免費的自由軟體，目前已經被 150 多個國家翻譯成 40 多種語言，所以你可以使用中文來寫程式。Scratch 從 2013 年更新 2.0 版之後開始提供線上版本，只要連上網際網路，就可以用瀏覽器進程式撰寫，不需要任何安裝程序，非常方便；2019 年，Scratch 改版 3.0，在平板電腦上也可以使用。而透過 Scratch 官網的雲端平臺，你可以將創意作品與全球分享。在本課程我們先了解 Scratch 的基本功能後，接著就要以簡單、有趣及實用的計算、繪圖、遊戲及模擬來學習 Scratch 程式設計。

2-2-1 Scratch 的官方網站

① Scratch 3.0 線上版

在瀏覽器的網址列輸入
<https://scratch.mit.edu>。

創造

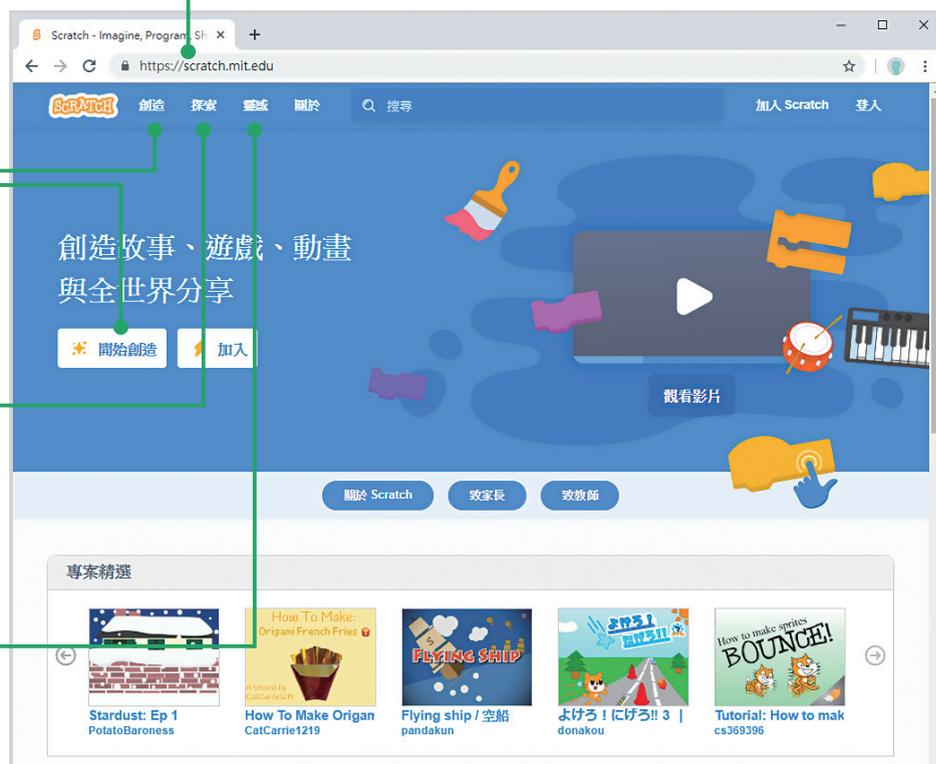
開啟撰寫程式介面。

探索

專案及工作坊的開放平臺，使每個人皆能上傳、分享，並學習他人程式。

靈感

提供教程，並依不同主題分類，步驟化說明影片學習更容易。

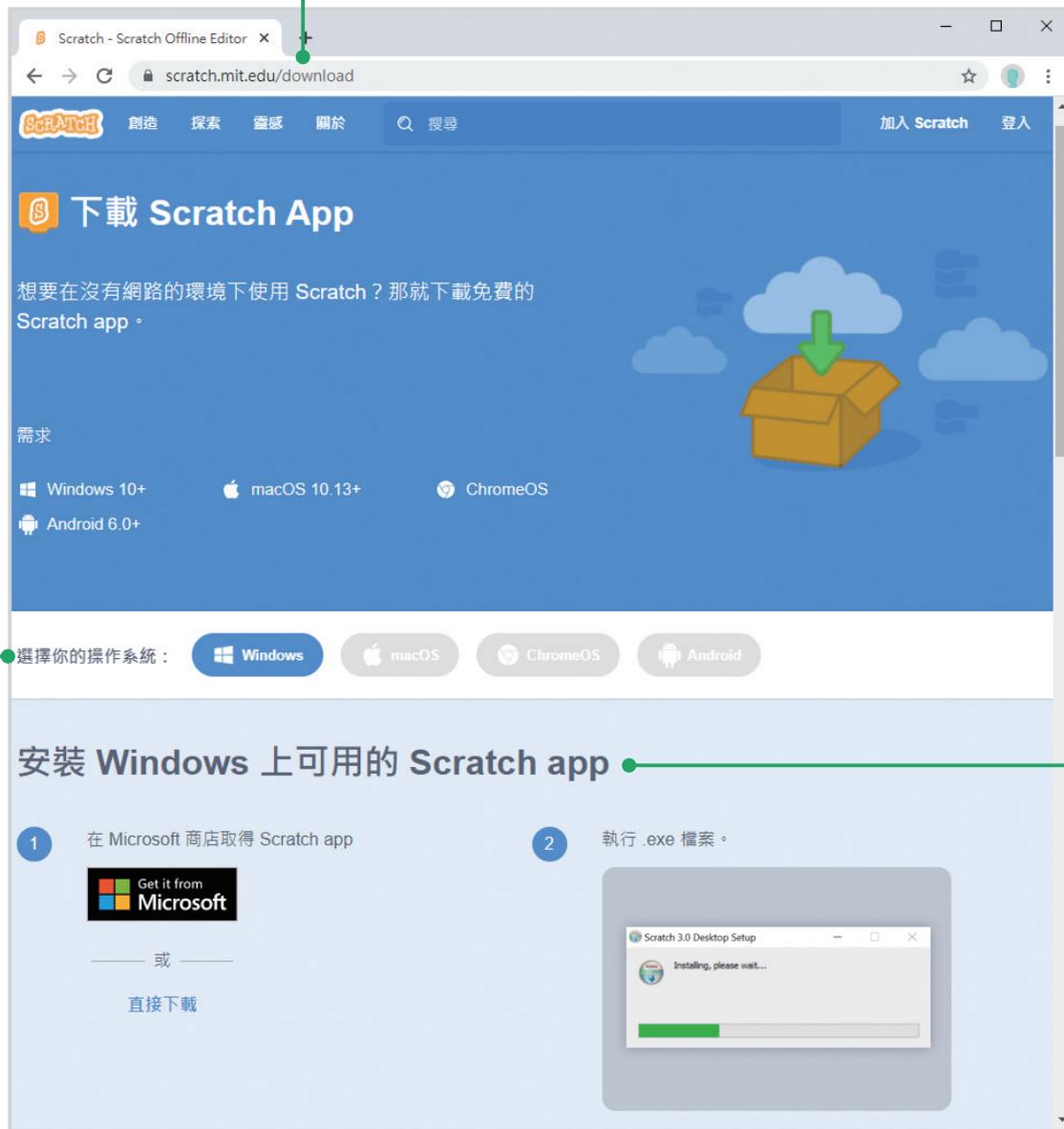


當然，我們也可以選擇從官網上直接下載並安裝離線版，完成後不必連上網路，就可以直接在自己的電腦上進行編輯。

② Scratch 3.0 離線版

在瀏覽器的網址列輸入

<https://scratch.mit.edu/download>。



依電腦操作系統，點選 **Windows** 或 **macOS** 鍵。

依照步驟，安裝 Scratch 離線版。

2-2-2 Scratch 的操作介面介紹

啟動 Scratch 後，會看到如下圖的畫面，這個起始畫面劃分為三個區。

腳本區

此區可以定義角色的造型及聲音，並且組合各式積木以達成你想要的功能。

程式面板

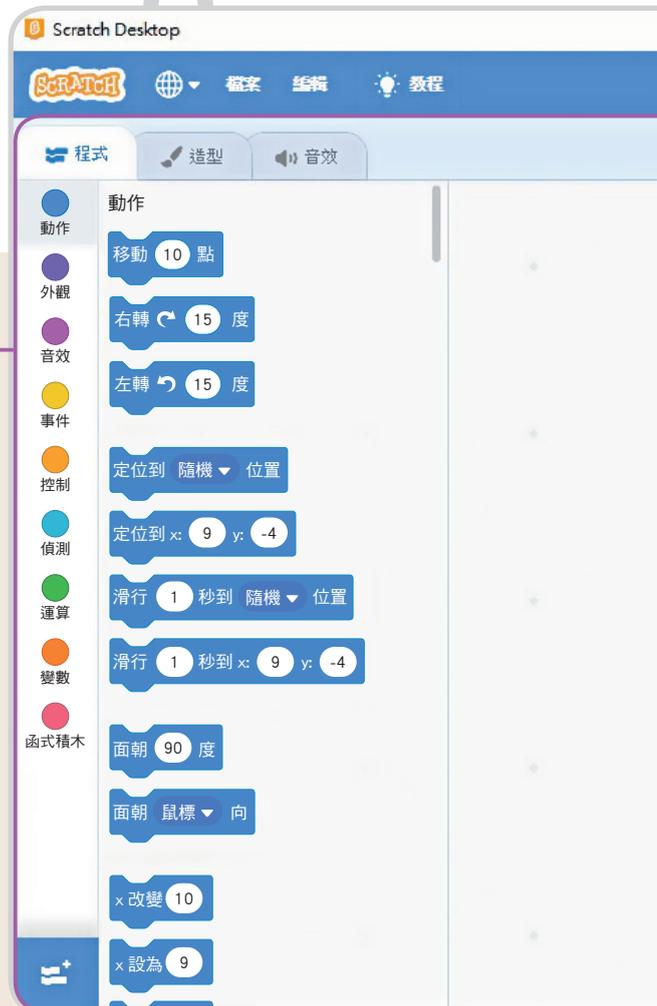
按  鍵顯示此面板，它提供 9 類共 100 多個積木讓你組裝，用來定義角色的行為，不同類別的積木使用不同顏色來區分。

造型面板

按  鍵顯示此面板，可定義該角色所有會出現的造型。

音效面板

按  鍵顯示此面板，可設定聲音檔，讓角色發出聲音的效果。



程式面板的功能



動作

設定角色的位置、方向、旋轉及移動。

外觀

設定角色的造型、特效、大小、及文字顯示。

音效

設定聲音的播放、音量，也可設定樂器與彈奏音符。

事件

設定當某個情況發生時需要執行的程式。

函式積木

自訂積木或擴展積木。

控制

設定程式的流程控制，如判斷、重複等。

偵測

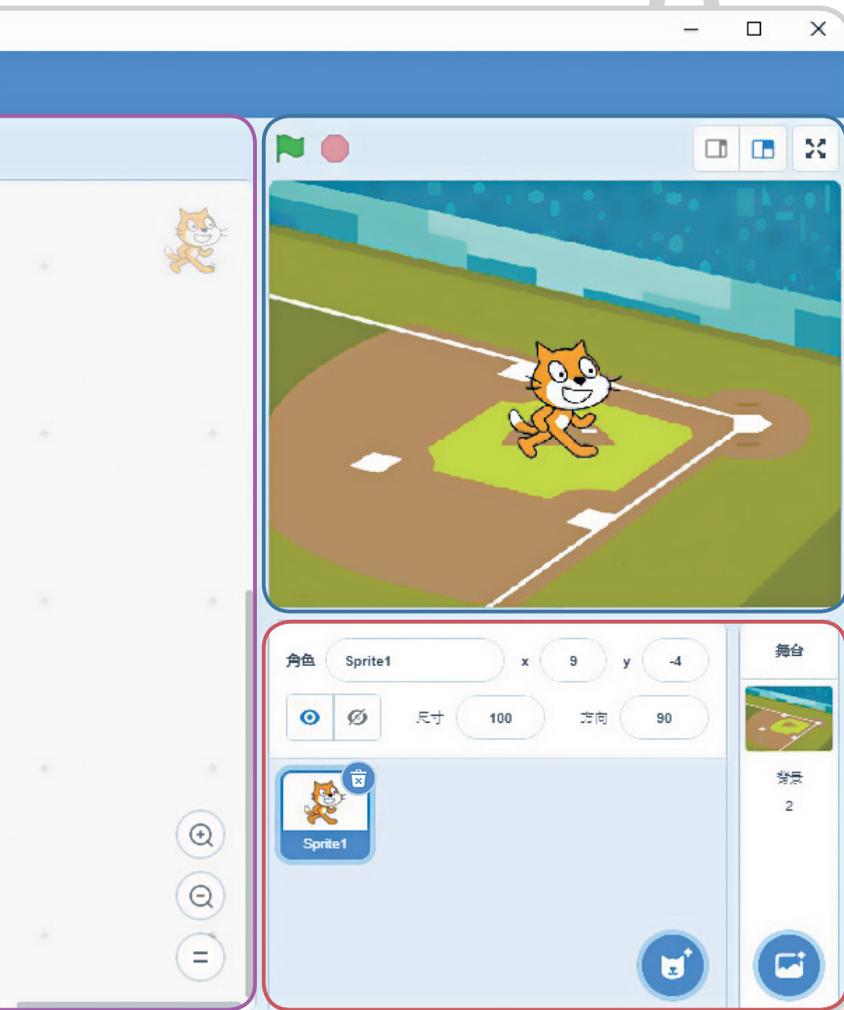
用於取得鍵盤或滑鼠的資訊，以判斷角色是否發生特定狀況。

運算

用於算術運算、邏輯運算、字串運算或產生亂數。

變數

設定變數及清單以儲存資料。



舞臺區

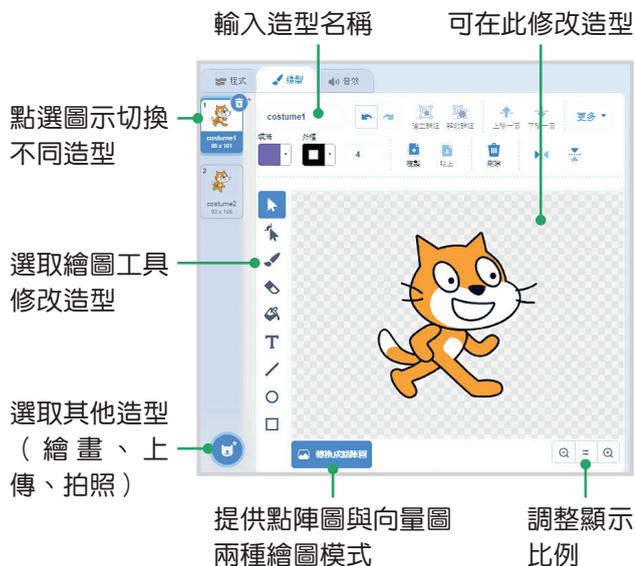
Scratch 提供的繪圖環境，大小是寬 480 點，高 360 點，此區會呈現程式執行結果。



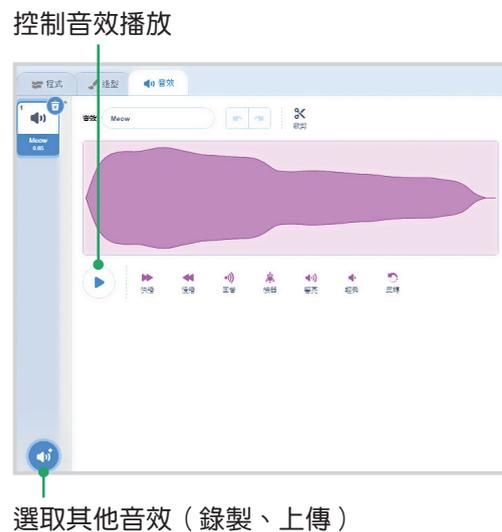
角色區

此區會列出所有用到的角色縮圖。按下角色縮圖，即可在角色資訊區，重新給角色取一個有意義的名稱。在 Scratch 中，雖然只有一個舞臺，但你可以為這個舞臺設定不同的背景。

造型面板的功能



音效面板的功能



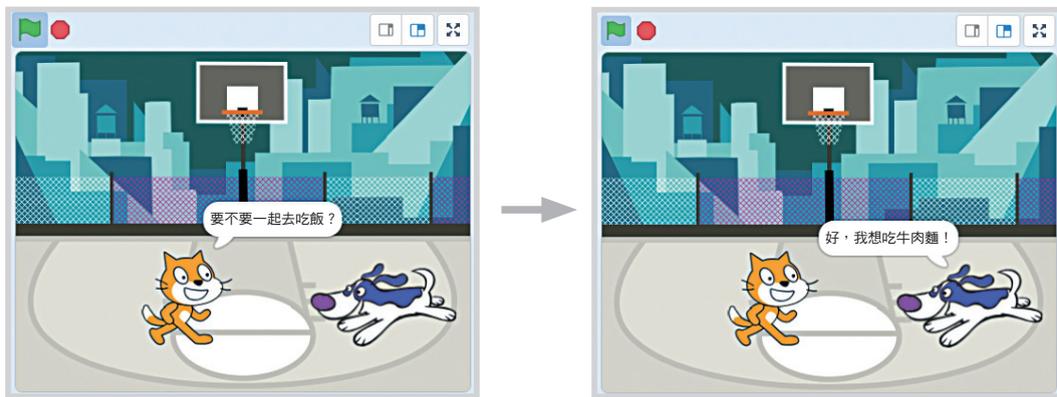
2-2-3 簡易的 Scratch 動畫實作

認識了 Scratch 的操作介面後，接著來實作一個簡易的動畫，以了解這些介面交錯運用所能達到的功能。

範例

對應習作第 43 頁

設定場景在籃球場，小貓和小狗碰面，進行簡單的對話後再相約去吃飯。以下為範例完成時的舞臺區畫面。



準備工作

回到 Scratch 的操作介面，開始設定舞臺與背景。

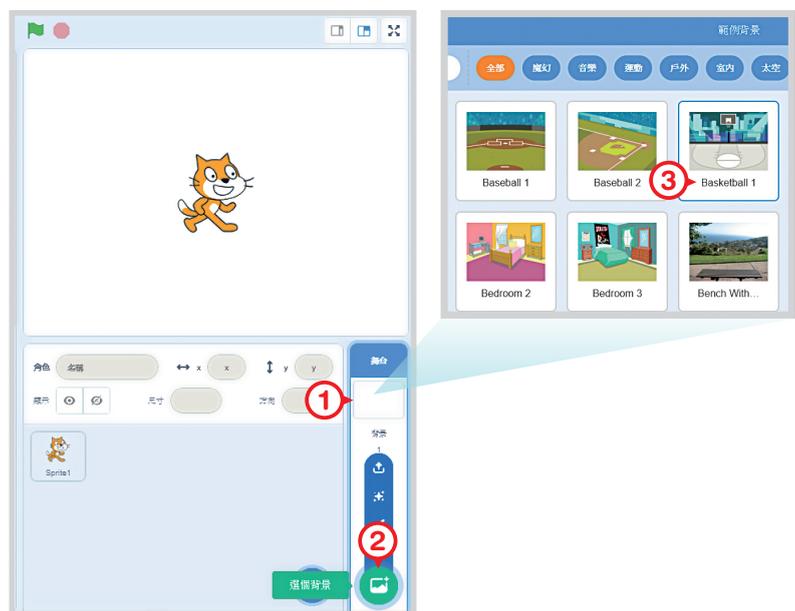
舞臺設計

步驟 1 開新檔案，匯入舞臺背景。

① 在角色區中用滑鼠點選右下方的舞臺。

② 點選下方選個背景按鍵列的 。

③ 從範例背景中，選擇 **Basketball 1** 圖片。





實作動畫

在開啟一個新的專案時，Scratch 會預設一隻小貓角色，而且有兩種造型、一種聲音。我們的動畫剛好有小貓角色，因此可以直接用這隻預設的小貓。除了小貓之外，還需要一隻小狗角色，可以從 Scratch 內建的範例庫中做挑選。



角色安排

步驟
2

新增小狗角色，並轉換小狗方向，讓小貓與小狗互看。

4

在角色區中，下方選個角色按鍵列中，按下 ，準備開啟小狗角色。

5

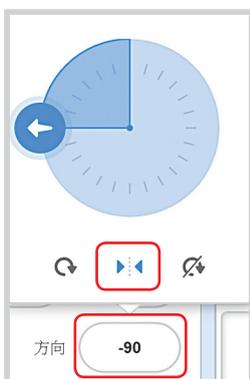
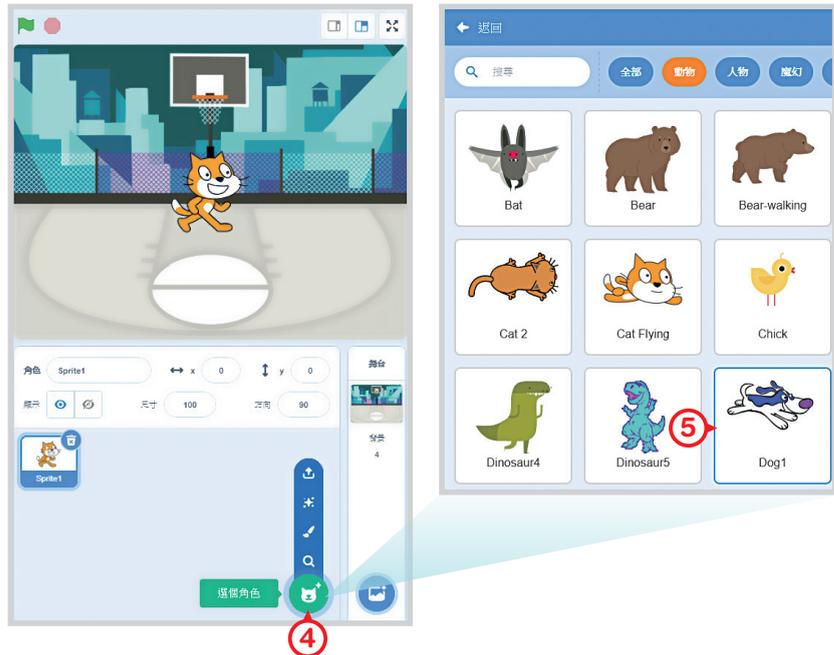
從範例角色中，選擇 Dog1。

6

用滑鼠將小貓與小狗拖曳至舞臺區的左下方與右下方。

7

點選小狗角色縮圖，在角色資訊區調整角色的方向，將預設的方向 90 改為 -90，旋轉方式修改為 。



撰寫程式

步驟 3 小貓準備開始走路。

⑧ 點選**事件**類別。

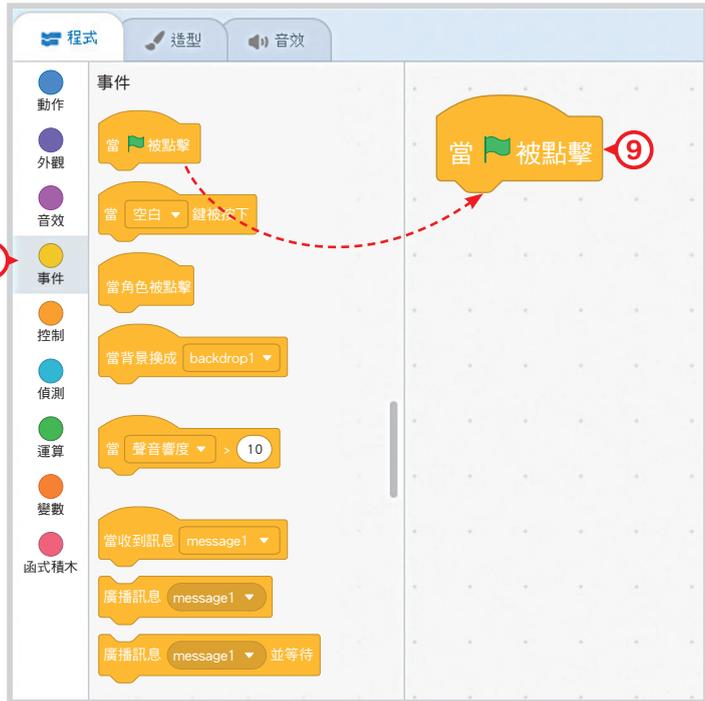
⑨ 將 **當 被點擊** 積木拖曳至腳本區中。

小提示

如何使用此積木？
通常放在程式的起頭，
用來啟動程式。



↑
下方有突起，可以拼接
上方有凹槽的積木。



步驟 4 讓小貓的腳走動 10 次。

⑩ 點選**控制**類別。

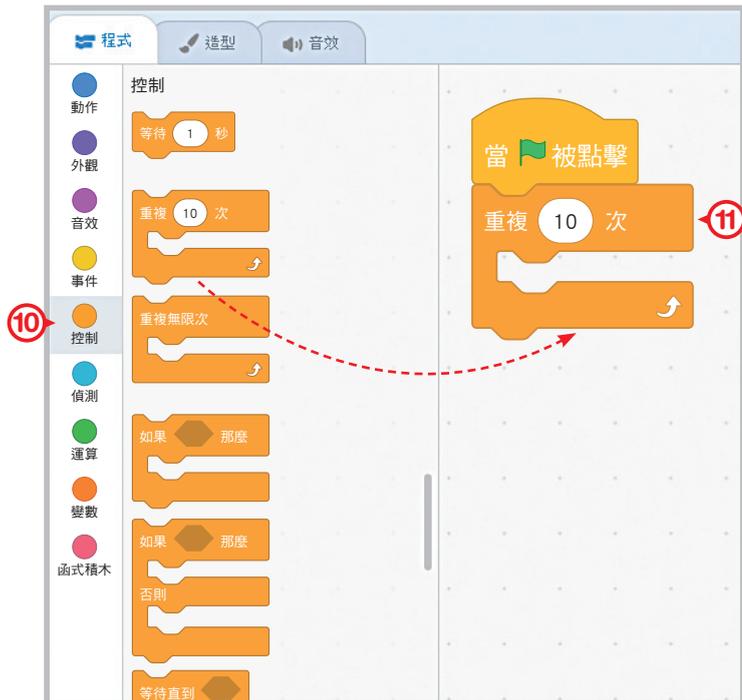
⑪ 拖曳 **重複 10 次** 積木，
接在前一個積木下方。

小提示

如何使用此積木？
在此輸入重複執行次
數，預設是 10 次。



可以嵌入需要重複
執行的積木。



步驟
5

設定讓小貓向前移動 10 點。

12 點選**動作**類別。

13 拖曳 **移動 10 點** 積木，嵌入前一個積木下方。

小提示

如何使用此積木？

設定角色往前移動距離，移動幾點代表移動幾個像素。



12



步驟
6

設定叫出小貓左右腳交替造型。

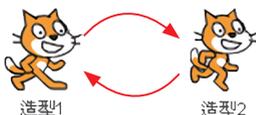
14 點選**外觀**類別。

15 拖曳 **造型換成下一個** 積木，嵌入前一個積木下方。

小提示

如何使用此積木？

設定角色換至下一種造型，或由最後的造型換至第一種造型。



小貓有兩種造型，重複執行讓造型不斷切換，做出走路動作。

14



因電腦的程式執行速度很快，重複 10 次一下子就結束，所以要想辦法讓過程慢下來。

步驟 7

設定小貓每走一步就停頓一下。

16 點選**控制**類別。

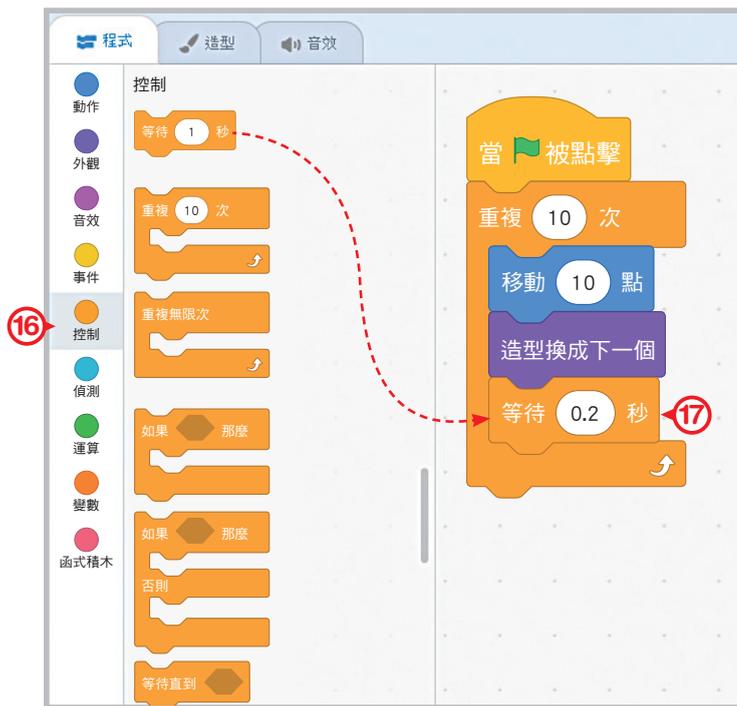
17 拖曳 **等待 1 秒** 積木，嵌入前一個積木下方，並將秒數改為 **等待 0.2 秒**。

小提示

如何使用此積木？

設定角色移動速度。等待時間愈長，移動速度愈慢。

等待 0.2 秒 輸入等待時間



按下舞臺區的綠旗時，小貓就會往前走。如果我們希望下一次再按下綠旗時，小貓會先回到原來的位置再往前走，就要設定此角色的起始位置。

步驟 8

設定小貓從何處開始走路。

18 點選**動作**類別。

19 拖曳 **定位到 x: -159 y: -80** 積木，嵌入 **當 被點擊** 積木下方。

小提示

設定起始位置

用此積木，輸入角色一開始出現的位置坐標。

定位到 x: -159 y: -80



在這個範例中，當小貓走到小狗面前時，就要開始簡單的對話，小貓先問：「要不要一起去吃飯？」，小狗回答：「好，我想吃牛肉麵！」由於小狗的回答是接續著小貓說完才開始講的，需要使用廣播與接收的觀念，讓不同角色間的積木可以串接起來。



步驟
9

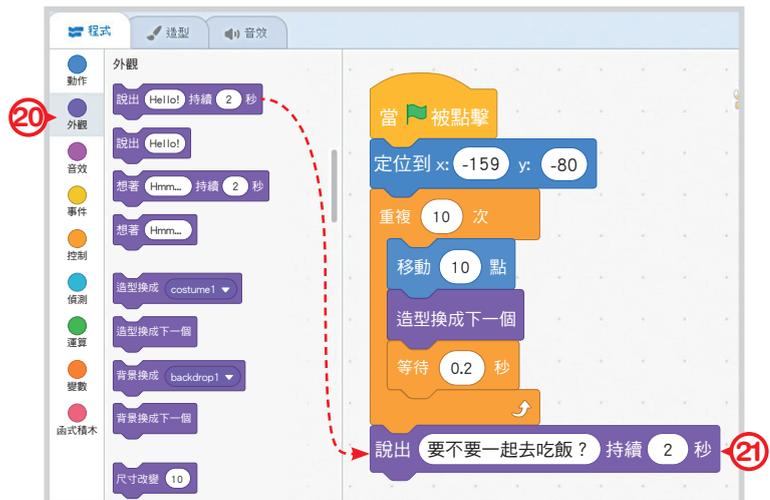
開始撰寫貓狗對話程式，設定讓小貓先說話。

20

點選外觀類別。

21

將「說出 Hello! 持續 2 秒」積木，拼接在之前完成的程式積木下方，改為「說出 要不要一起去吃飯? 持續 2 秒」。



小提示

如何使用此積木？

用來顯示角色表現在畫面的訊息，例如：角色要說的話。



在這裡輸入要說的話

輸入說話持續時間

步驟 10 設定小貓發訊息，啟動小狗程式。

22 點選事件類別。

23 將 廣播訊息 message1 積木，
 拼接在前一個積木下方，改為 廣播訊息 小狗說話。



小提示

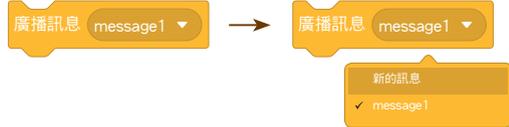
如何使用此積木？

主要使用在此角色對其他角色發送訊號，開始啟動或控制其他角色的程式。

先用滑鼠按下 ▼

出現選單後，點選「新的訊息」。

將跳出「新的訊息」視窗，並填入「小狗說話」，表示用來啟動在小狗角色上的說話程式。



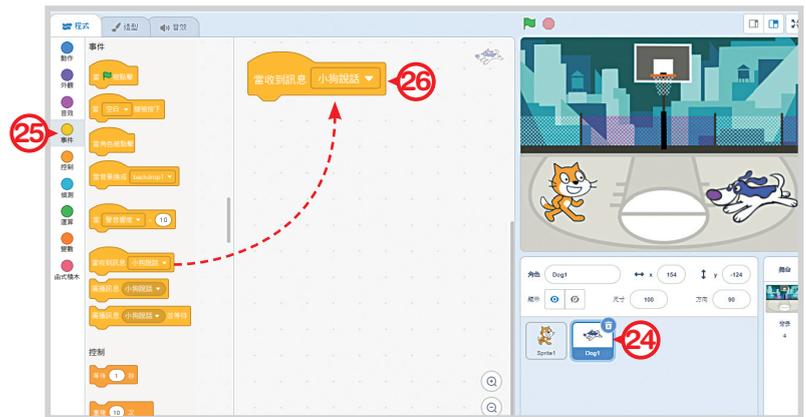
廣播訊息 小狗說話 完成！

步驟 11 小狗接收到訊息後，啟動程式。

24 回到角色區點選小狗。

25 點選事件類別。

26 將 當收到訊息 小狗說話 積木拖曳至腳本區中。



步驟 12 讓小狗說話。

27 點選外觀類別。

28 將 說出 Hello! 持續 2 秒 積木，
 拼接在之前完成的程式積木下方，改為

說出 好，我想吃牛肉麵！ 持續 2 秒



小提示

廣播積木與接收積木的互動方式

- 1 當程式執行到小貓的廣播積木。
- 2 小貓的廣播積木發出小狗說話訊息。
- 3 小狗的接收積木收到訊息後，啟動執行。



按下  測試你的程式。完成了這個 Scratch 動畫創作，你是不是覺得寫程式一點都不難，而且很有趣呢！關於積木排列方式，都會影響程式執行的結果。不妨試試習作附件 7 的小遊戲，練習排列的先後順序。

在此單元中，我學到的有：

1. 明白了完成動畫所需的一系列步驟。
2. 為了讓角色走路，我使用了迴圈執行同一序列動作許多次。
3. 為了讓角色對話，我使用了廣播功能來形成另一事件發生的某事件。
4. 確認程式碼可以運作，包含發現並解決問題。
5. 了解並連結整部動畫與部分程式碼間的關係。
6. 了解運算也是一種創造工具，而我可以進行創造！



以上介紹了 Scratch 的操作介面及範例的實作，你務必要了解且熟練腳本區、舞臺區和角色區所扮演的角色及功能。熟練之後，就可以開始用 Scratch 計算與繪圖了！

2-3 Scratch 程式設計 – 計算篇



▲圖 2-6 結帳時應用的資訊科技。

電腦剛發明時，被稱為電子計算機。它的功能，主要用於幫助人們處理大量的計算問題（圖 2-6），本節將運用流程圖與 Scratch 程式設計來解決一些數學上的計算問題，讓你體驗一下電腦程式的魅力。

在進行算術運算時，必須指定適當的運算符號來表達運算式，如下表所示。

算術運算的類型、符號、Scratch 運算積木對照表

運算類型	運算符號	Scratch 運算積木
加	+	
減	-	
乘	×	
除	÷	

接下來，我們將從簡單的範例開始，學習如何分析問題，之後形成流程圖，接著再依照流程圖寫出 Scratch 程式碼解決問題。

而在解決問題的過程中，可以使用後面介紹的循序結構、選擇結構和重複結構三種基本流程結構，來安排解題步驟的順序與過程。

準備工作

設定積木

由於接下來需要頻繁使用變數積木，因此先熟悉一下基礎的操作過程。

步驟 1 開始設定新變數。

- 到程式面板，點選**變數**類別。
- 按下**建立一個變數**鍵，來新增變數。

步驟 2 選擇名稱與適用範圍。

- 跳出**新的變數**視窗。
- 這裡輸入變數的名稱 **A**。
- 接著點選**適用於所有角色**。
- 按下**確定**鍵。

步驟 3 產生變數積木群組。

- 產生了 5 種變數積木。

小知識

變數

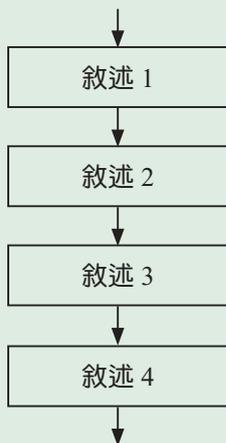
在程式設計過程中，若需要記錄某些文字或數字資料，可以用一個名稱記錄起來，且暫存在電腦中。因資料內容是可變動的，因此，這個名稱就是變數。



2-3-1 循序結構

程式若是由上而下依序執行，就稱為循序結構。

循序結構－流程圖



例如：Scratch 程式碼由上而下依序執行



範例

求 A 與 B 兩數的平均數。

對應習作第 45 頁



問題分析

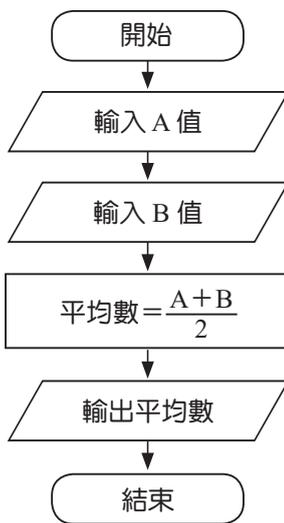
閱讀並分析題目，將問題拆解為輸入、處理、輸出三個階段，請寫出各階段相對應的內容。

階段	內容
輸入	A 與 B 的值
處理	平均數 = $\frac{A+B}{2}$
輸出	平均數



畫流程圖

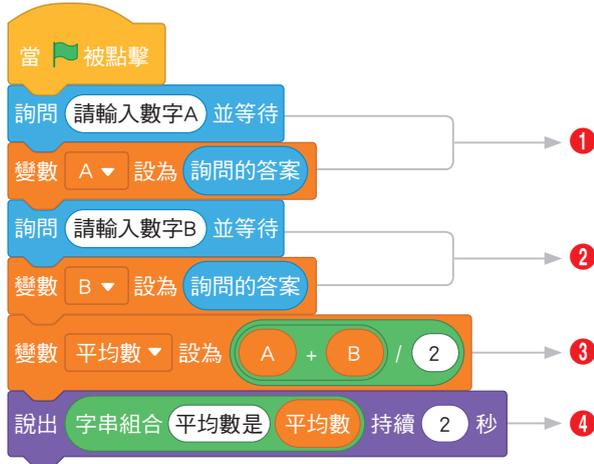
將問題分析的內容，依序畫成流程圖。





撰寫程式

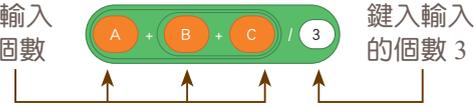
依照流程圖的各步驟，寫出 Scratch 程式碼。



分析

如何撰寫計算兩個數以上平均數的積木？

若輸入 3 個數



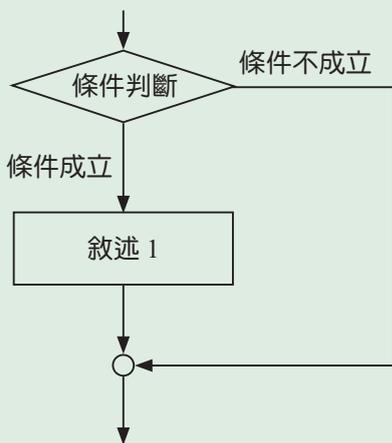
問題解析	問題實作
(A)如何設定輸入 A 值？	<p>1 用此積木，設定變數 A 值為 詢問的答案。</p> <p>鍵入小貓說話文字</p> <p>詢問 請輸入數字A 並等待</p> <p>變數 A 設為 詢問的答案</p> <p>放入 詢問的答案</p>
(B)如何設定輸入 B 值？	<p>2 用此積木，設定變數 B 值為 詢問的答案。</p> <p>鍵入小貓說話文字</p> <p>詢問 請輸入數字B 並等待</p> <p>變數 B 設為 詢問的答案</p> <p>放入 詢問的答案</p>
(C)如何計算 A 與 B 的平均數？	<p>3 用此積木，設定變數平均數，數值是 $\frac{A+B}{2}$。</p> <p>選擇變數為平均數</p> <p>變數 平均數 設為 $(A + B) / 2$</p> <p>放入 $(A + B) / 2$</p>
(D)如何輸出平均數？	<p>4 設定小貓說出平均數的數值，持續 2 秒。</p> <p>鍵入小貓說話文字</p> <p>說出 字串組合 平均數是 平均數 持續 2 秒</p> <p>放入平均數的變數</p>

2-3-2 選擇結構

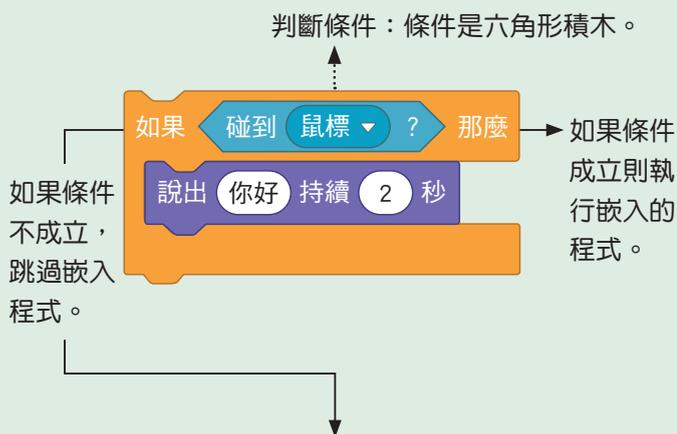
依條件判斷的結果，決定要執行哪一段程式碼敘述。條件判斷的結果，有條件成立（true）及條件不成立（false）兩種。

在單向選擇結構中，只定義了條件成立時要執行的敘述；在雙向選擇結構中，分別定義了條件成立與條件不成立時要執行的敘述。

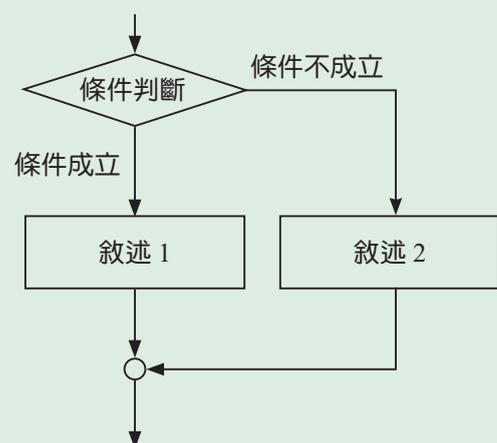
單向選擇結構—流程圖



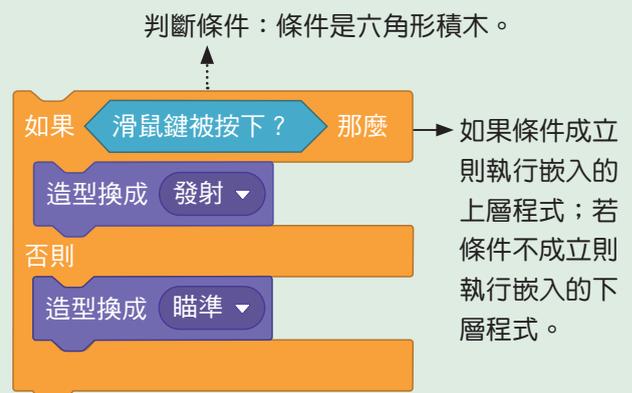
例如：使用單向選擇結構的 Scratch 程式碼



雙向選擇結構—流程圖



例如：使用雙向選擇結構的 Scratch 程式碼



對應習作第 47 頁

範例

將輸入的各項成績換算為學期成績，並判斷學期成績是否及格？

（作業成績占 40%、測驗成績占 40%、平時表現占 20%，學期成績 60 分為及格分數。）

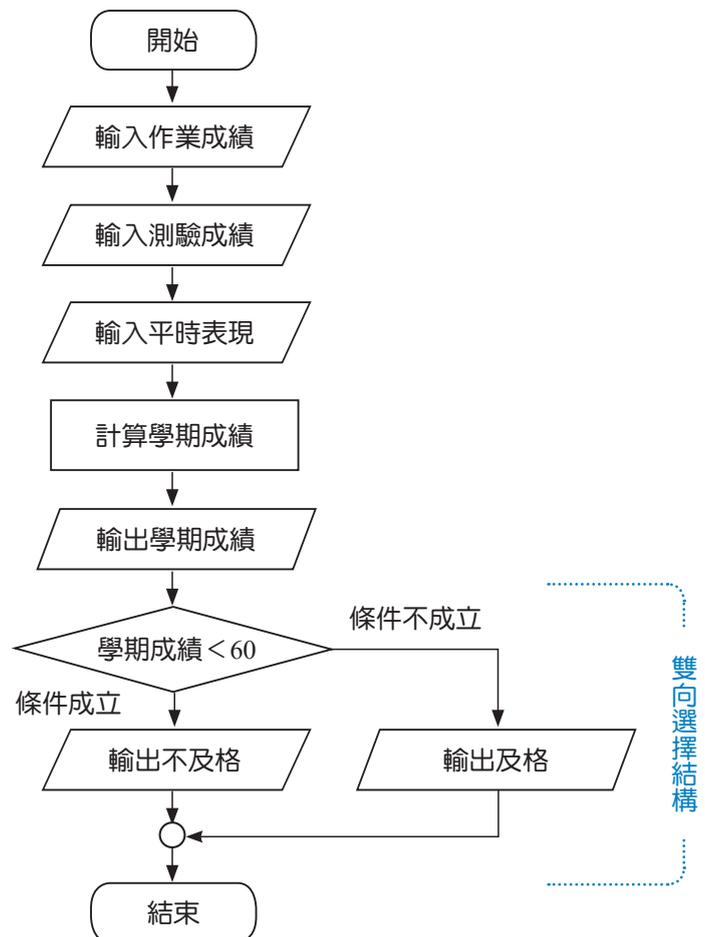
問題分析

閱讀並分析題目，將問題拆解為輸入、處理、輸出、判斷、輸出五個階段，請寫出各階段相對應的內容。

階段	內容
輸入	
處理	
輸出	
判斷	
輸出	

畫流程圖

將問題分析的內容，依序畫成流程圖。





撰寫程式

依照流程圖的各步驟，寫出 Scratch 程式碼。



學期成績 < 60 是否還有其他表示的方法呢？



可以利用第 199 頁小知識的  或 ，就能呈現另一種方法了。



問題解析	問題實作
(A)如何設定輸入作業成績、測驗成績、平時表現？	<p>❶ 設定輸入作業成績。</p> 
	<p>❷ 接著設定輸入測驗成績。</p> 
	<p>❸ 最後設定輸入平時表現。</p> 
(B)如何計算學期成績？	<p>❹ 先寫出算式，再設定學期成績。</p> 
(C)如何輸出學期成績？	<p>❺ 用此積木，說出學期成績是。</p> 
(D)判斷學期成績是否不及格？	<p>❻ 設定此程式為條件，判斷學期成績是否低於60分？</p> 
(E)如何依照條件判斷的結果，控制輸出及格或不及格？	<p>❼ 若條件成立（學期成績 < 60），執行嵌入積木上層程式，也就是❸。 若條件不成立（學期成績 ≥ 60），執行嵌入積木下層程式，也就是❹。</p> 
(F)如何設定輸出學期成績是否及格？	<p>❽ 用此積木，說出不及格。</p> 
	<p>❾ 用此積木，說出及格。</p> 

2-3-3 重複結構

需要重複被執行的程式碼可以使用重複結構，根據被重複執行的情況，可以分為計次式迴圈與條件式迴圈兩種。

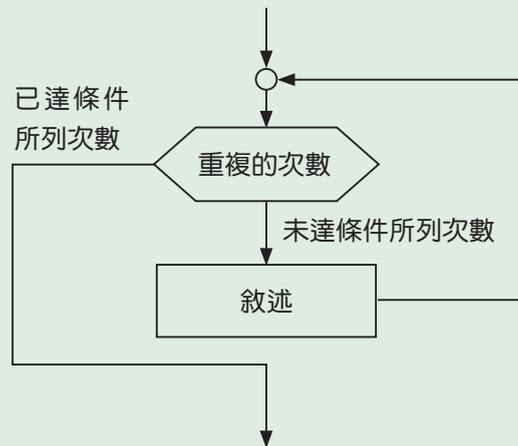
當我們明確知道某段敘述要被重複執行的次數時，就會使用計次式迴圈；當我們不知道某段敘述要被重複執行的次數，而是知道終止情況時，就會使用條件式迴圈。

小知識

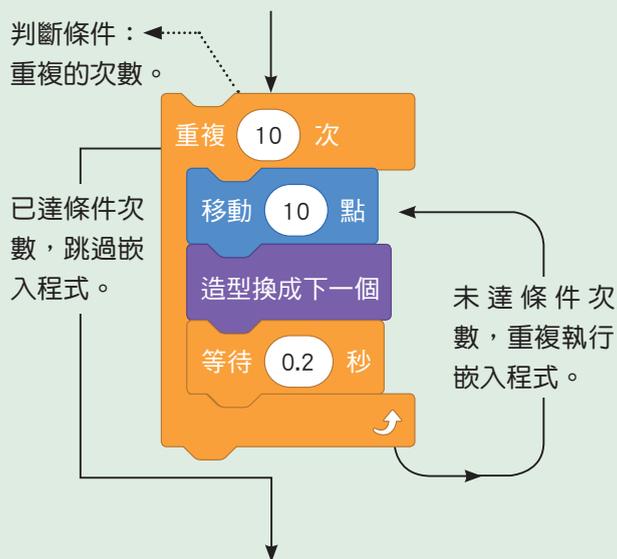
迴圈

在一個程式語言內，可以讓程式碼反覆執行的敘述。

計次式迴圈－流程圖



例如：使用重複結構的 Scratch 程式碼





範例

試計算 $1 + 2 + 3 + 4$ 的值。



問題分析

閱讀並分析題目，將問題拆解為處理、輸出兩個階段，請寫出各階段相對應的內容。

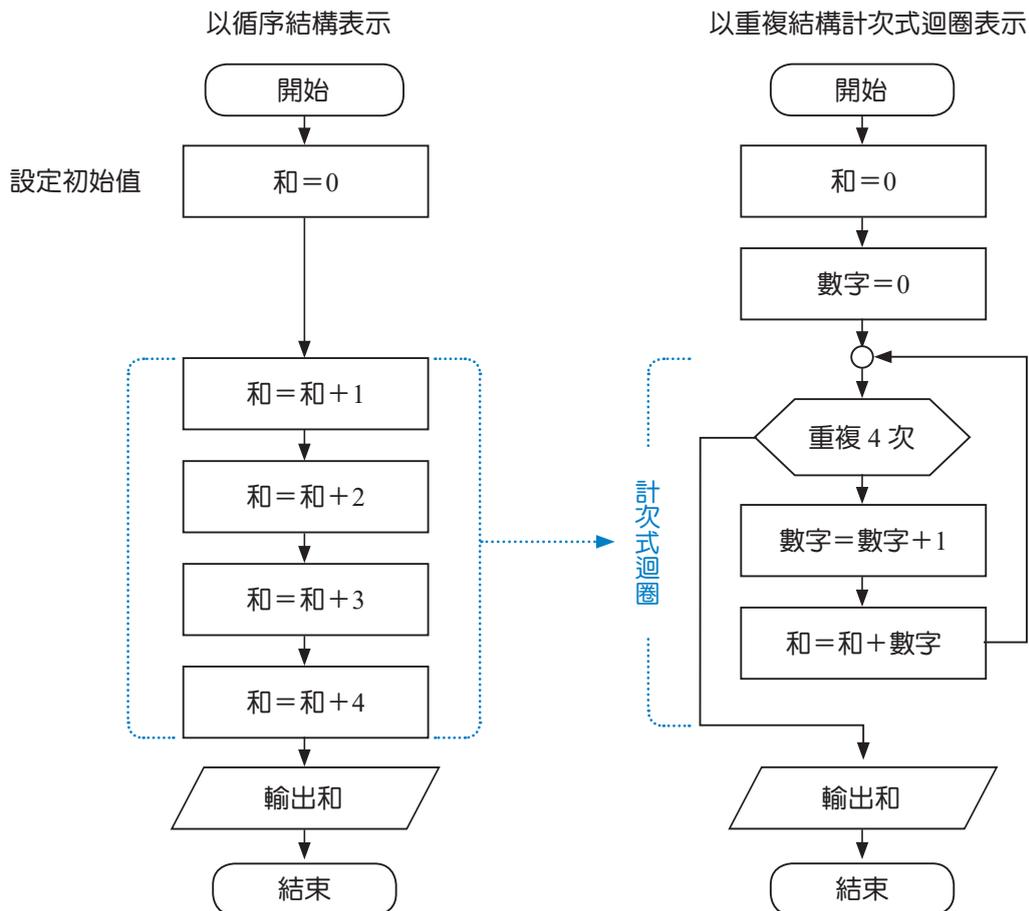
設定初始值：和設為 0。

階段	內容
處理	
輸出	



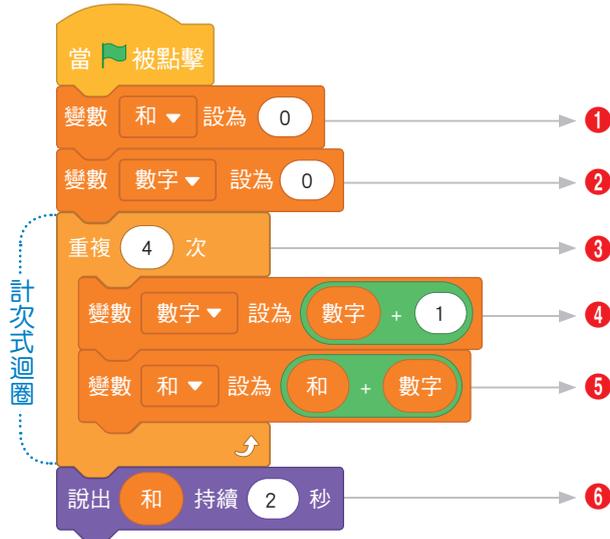
畫流程圖

將問題分析的內容，依序畫成流程圖。



撰寫程式

依照流程圖的各步驟，寫出 Scratch 程式碼。



分析

1. 程式的起頭，通常要先設定每個變數一開始的數值，這種程序稱為設定初始值。
2. 程式設計，除了條件判斷之外，等號的意義並不是左邊 = 右邊，而是指定將等號右邊的數值儲存到等號左邊的變數，這種運算方式稱為指定運算 (assignment operation)。
3. 流程圖中的運算式：



問題解析	問題實作
(A)如何將開始時的和設為 0？	<p>❶ 用此積木，設定和的初始值為 0。</p> ← 鍵入和的初始值 0
(B)如何將開始時的數字設為 0？	<p>❷ 用此積木，設定數字的初始值為 0。</p> ← 鍵入數字的初始值 0
(C)如何重複計算加法 4 次？	<p>❸ 用此積木，讓嵌入的程式，也就是 ❹、❺，重複執行 4 次。</p> ← 鍵入重複執行的次數 4 次
(D)每次重複計算加法時，如何讓數字增加 1？	<p>❹ 執行時，讓數字的數值每次增加 1。</p> ← 放入 數字 + 1
(E)每次重複計算加法時，如何讓和加上數字？	<p>❺ 執行時，讓和每次都加上數字的數值。</p> ← 放入 和 + 數字
(F)如何輸出和的數值？	<p>❻ 用此積木，設定說出和的數值。</p> ← 放入和的變數

用重複結構計次式迴圈來計算 $1 + 2 + 3 + 4$ 編寫程式的速度，好像沒有比使用循序結構快，但是它給了我們很大的擴充彈性，使用計次式迴圈很容易推展到任意數字。畫流程圖時，我們可以設計讓使用者輸入一個數字 N ，再計算從 1 累加到 N 的值。



範例

試計算 $1 + 2 + \dots + N$ 的值。



問題分析

閱讀並分析題目，將問題拆解為輸入、處理、輸出三個階段，請寫出各階段相對應的內容。

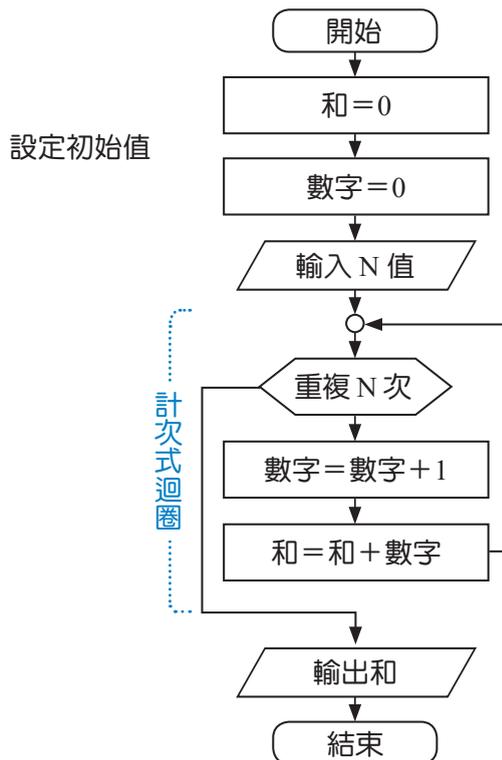
設定初始值：和設為 0，數字設為 0。

階段	內容
輸入	
處理	
輸出	



畫流程圖

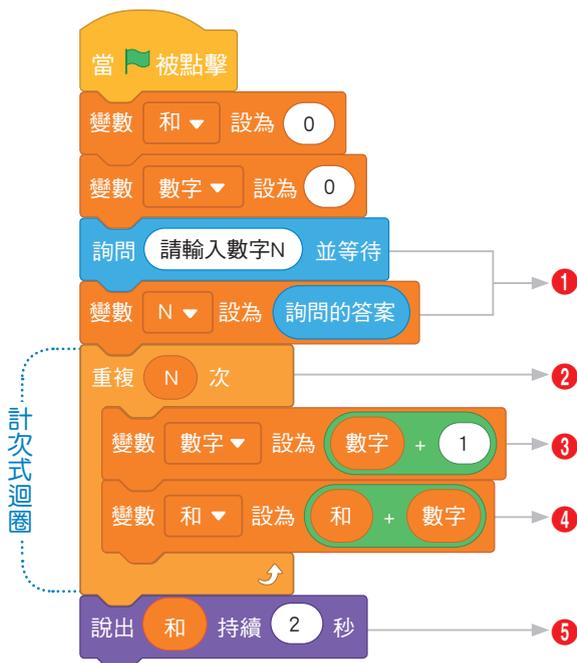
將問題分析的內容，依序畫成流程圖。





撰寫程式

依照流程圖的各步驟，寫出 Scratch 程式碼。

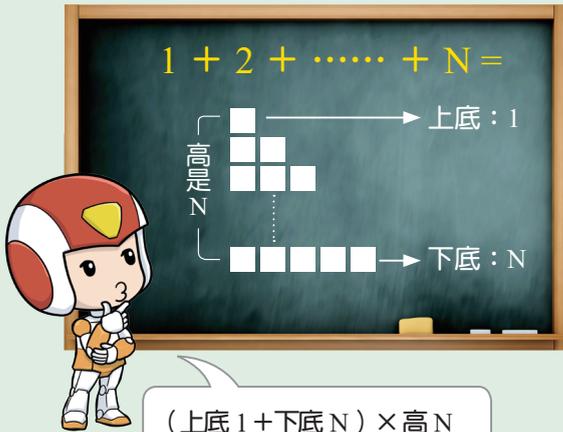


想想看，如果題目改為 $5+6+\dots+N$ ，那麼和、數字的變數，又要如何設定初始值呢？



問題解析	問題實作
(A)如何設定輸入 N 的值？	<p>①用此積木，設定輸入 N 的值。</p>
(B)如何重複計算加法 N 次？	<p>②用此積木，讓嵌入的程式，也就是 ③、④，重複執行 N 次。</p> <p>放入變數 N，重複執行 N 次</p>
(C)每次重複計算加法時，如何讓數字增加 1？	<p>③執行時，讓數字的數值每次增加 1。</p> <p>放入 數字 + 1</p>
(D)每次重複計算加法時，如何讓和加上數字？	<p>④執行時，讓和每次加上數字的數值。</p> <p>放入 和 + 數字</p>
(E)如何輸出和的數值？	<p>⑤用此積木，設定說出和的數值。</p> <p>放入和的變數</p>

你可能會覺得，要計算 $1 + 2 + \dots + N$ 的總和，不是用公式一下子就算出來了嗎？為什麼要那麼麻煩寫一堆程式碼來計算呢？



$$\frac{(\text{上底 } 1 + \text{下底 } N) \times \text{高 } N}{2}$$
，
用此公式就可計算！

其實，雖然累加的確有公式可以速算。但是如果需要計算連乘時怎麼辦呢？試著藉由如右範例，模仿累加的程式碼，用計次式迴圈寫看看連乘吧！



未來數學課程的排列組合會使用到此算式，要怎麼算？



範例

試計算 $1 \times 2 \times \dots \times N$ 的值。



問題分析

閱讀並分析題目，將問題拆解為輸入、處理、輸出三個階段，請寫出各階段相對應的內容。

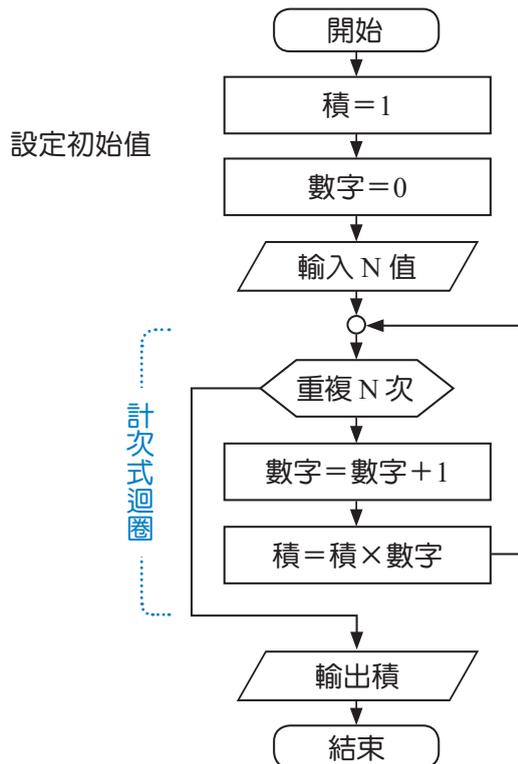
設定初始值：積設為 1，數字設為 0。

階段	內容
輸入	
處理	
輸出	



畫流程圖

將問題分析的內容，依序畫成流程圖。





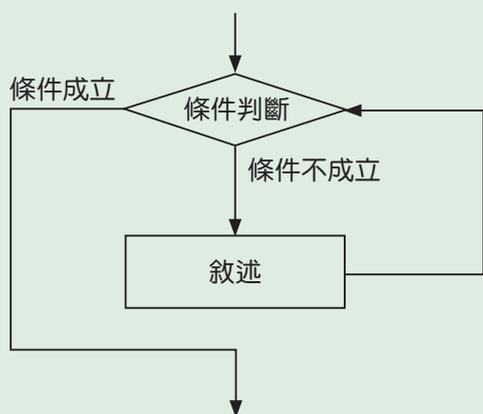
撰寫程式

依照流程圖的各步驟，寫出 Scratch 程式碼。

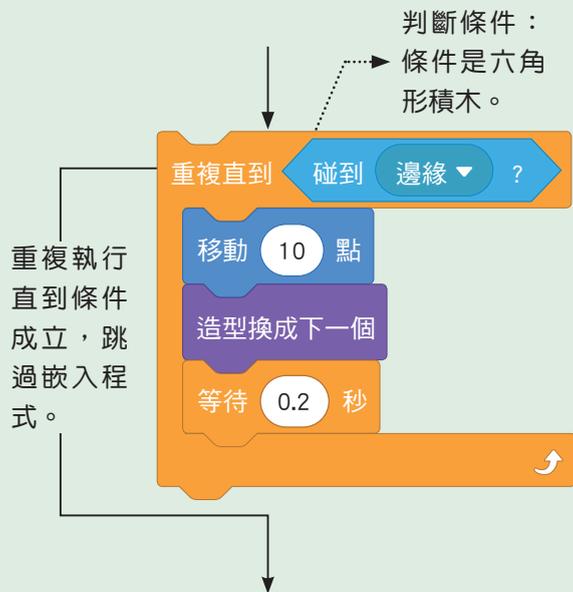


問題解析	問題實作
(A)如何將開始時的積設為 1 ？	<p>❶用此積木，設定積的初始值為 1。</p>  <p>鍵入積的初始值 1</p>
(B)如何將開始時的數字設為 0 ？	<p>❷用此積木，設定數字的初始值為 0。</p>  <p>鍵入數字的初始值 0</p>
(C)如何設定輸入 N 的值？	<p>❸用此積木，設定輸入 N 的值。</p> 
(D)如何重複計算乘法 N 次？	<p>❹用此積木，讓嵌入的程式，也就是 ❺、❻，重複執行 N 次。</p>  <p>放入變數 N，重複執行 N 次。</p>
(E)每次重複計算乘法時，如何讓數字增加 1 ？	<p>❺執行時，讓數字的數值每次增加 1。</p>  <p>放入 數字 + 1</p>
(F)每次重複計算乘法時，如何讓積乘以數字？	<p>❻執行時，讓積每次都乘以數字的數值。</p>  <p>放入 積 * 數字</p>
(G)如何輸出積的數值？	<p>❼用此積木，設定說出積的數值。</p>  <p>放入積的變數</p>

條件式迴圈－流程圖



例如：使用條件式迴圈的 Scratch 程式碼



小知識

且 左右兩邊條件式要同時成立，運算結果才會成立。

或 左右兩邊條件式有一個成立，運算結果就會成立。

不成立 反向運算，當條件成立，運算結果為不成立；當條件不成立，運算結果為成立。

範例

設計一個電腦系統的密碼驗證機制，條件如下。

- (1) 若第一次輸入密碼錯誤後，可再重複嘗試輸入兩次。
- (2) 若輸入三次密碼都錯誤，跳出使用者帳號被鎖定的訊息。

問題分析

閱讀並分析題目，將問題拆解為輸入、判斷、處理、判斷、輸出五個階段，請寫出各階段相對應的內容。

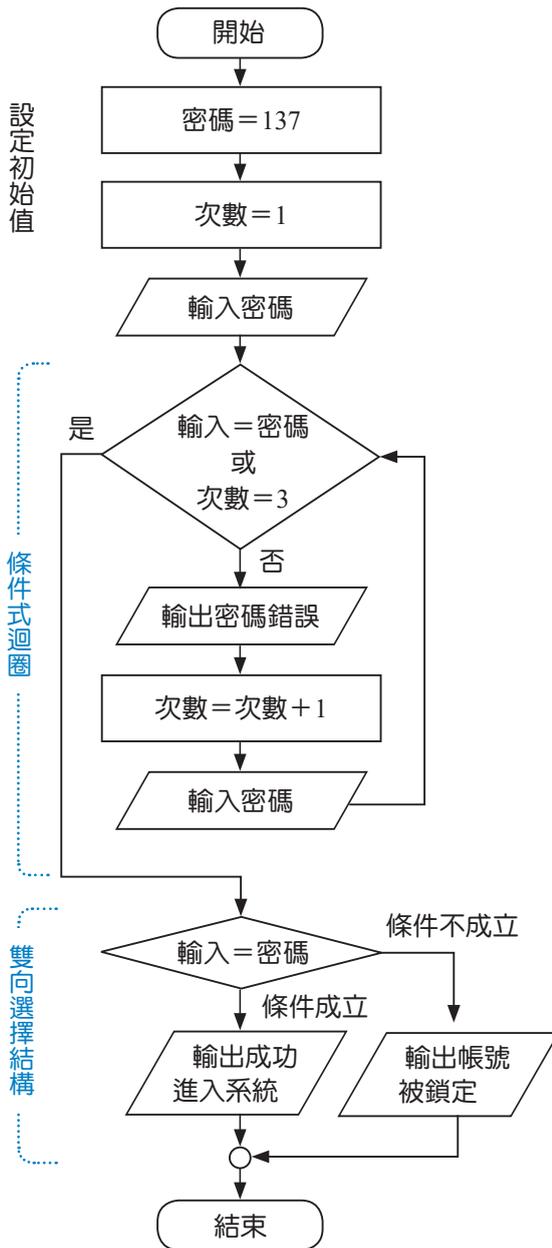
設定初始值：密碼設為 137，次數設為 1。

階段	內容
輸入	
判斷	
處理	
判斷	
輸出	



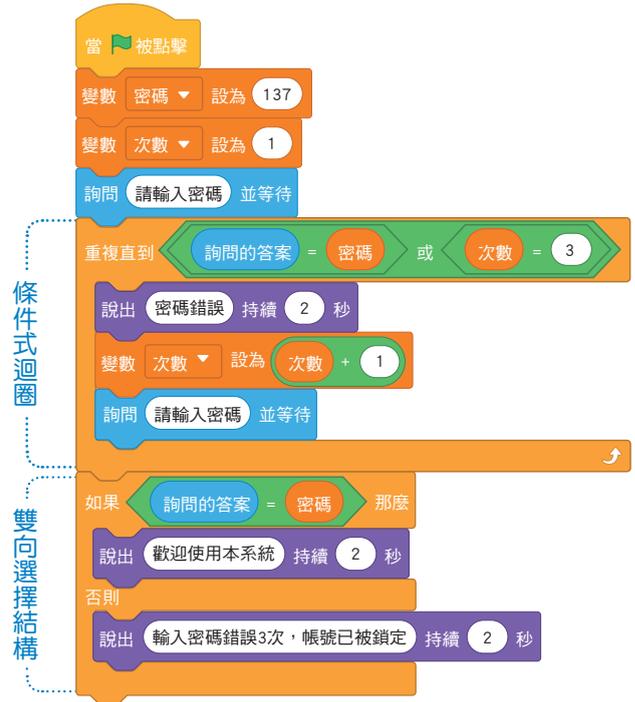
畫流程圖

將問題分析的內容，依序畫成流程圖。



撰寫程式

依照流程圖的各步驟，寫出 Scratch 程式碼。



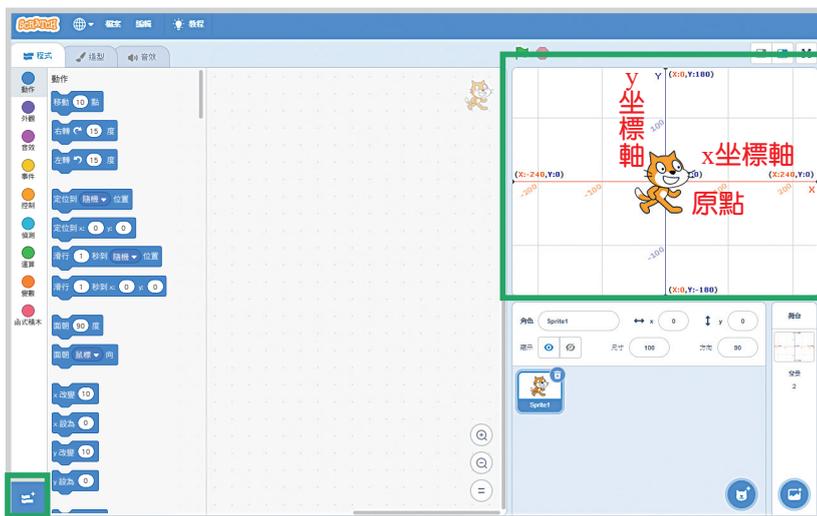
在此單元中，我學到的有：

1. 明白程式碼是由上而下循序執行的序列。
2. 明白程式碼有計次式與條件式兩種不同的迴圈。
3. 明白可以依據條件做決定而執行不同的程式碼。
4. 明白用於數學與邏輯的運算式。
5. 可以使用變數來儲存與更新的資料。

2-4 Scratch 程式設計 – 繪圖篇

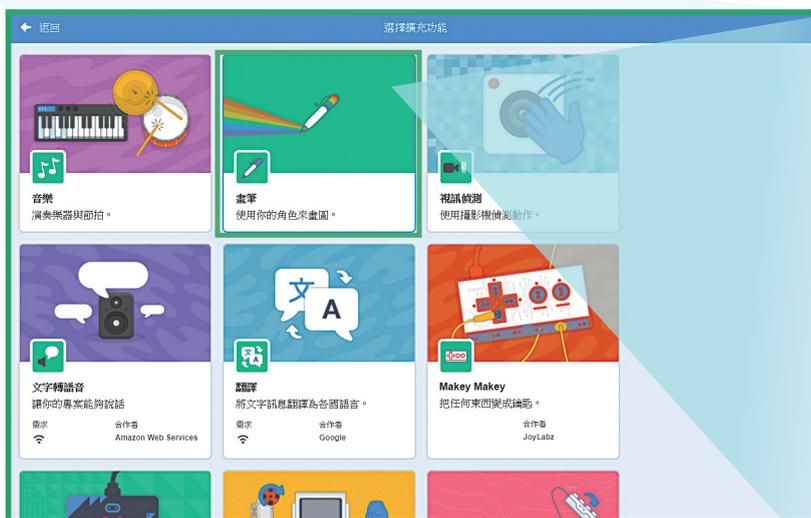
Scratch 提供一個很方便且易學的繪圖環境，如下圖，它的舞臺畫面寬 480 點，高 360 點。在畫面的正中央是坐標系統的原點 (0, 0)。原點往右的 x 軸坐標是正數，往左是負數；原點往上的 y 軸坐標是正數，往下是負數。

透過 Scratch 的畫筆，可以繪製出很多種有趣的幾何圖案。所以，接下來就要實際體驗，如何運用 Scratch 程式設計來繪圖。在使用畫筆前，需要先從擴展功能中新增相關積木。



舞臺區標示出坐標軸與原點

程式面板中展示所有新增的畫筆積木。



腳本區下方可以添加擴展許多功能，本節使用畫筆功能。





準備工作

從下一頁起的三個範例中，需要觀察小貓移動的坐標位置變化，因此先設計舞臺，選擇坐標圖形 xy-grid 作為背景。



舞臺設計

步驟 1

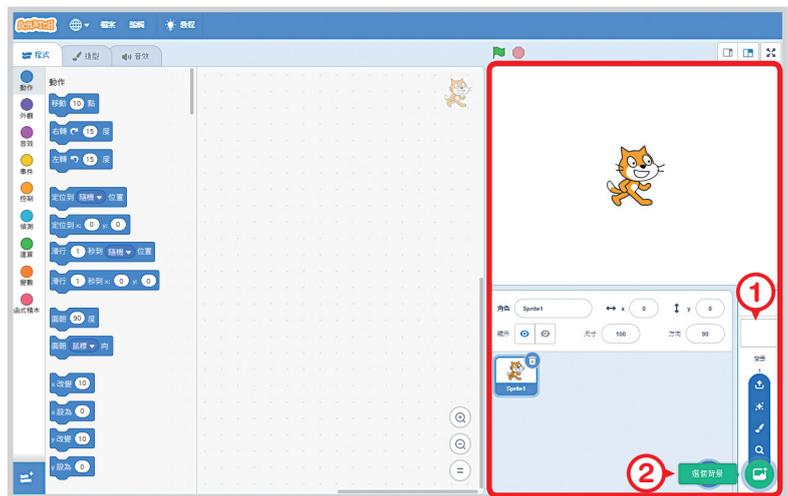
開新檔案，匯入舞臺背景。

1

在角色區中用滑鼠點選右下方的舞臺。

2

點選下方選個背景按鍵列的後，從範例庫中選擇背景。

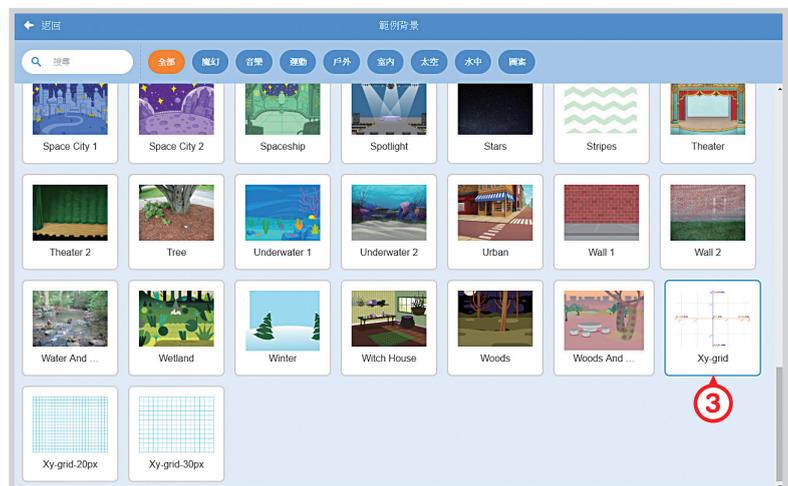


步驟 2

選擇舞臺樣式。

3

從範例背景庫中，選擇 xy-grid 圖片。



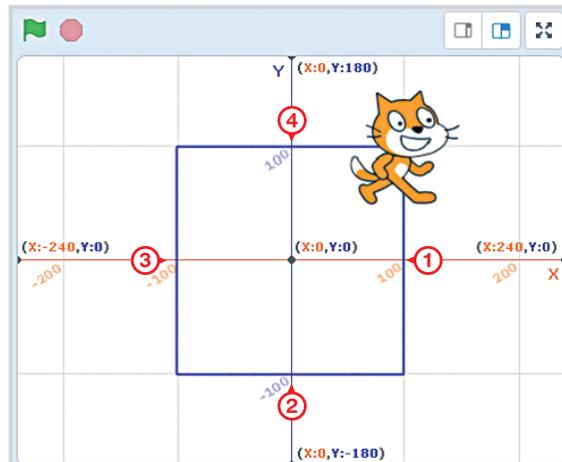
2-4-1 畫正方形

接下來，讓我們利用不同的方法來畫正方形。你可以說出這三種方法的優缺點嗎？

範例

對應習作第 49 頁

利用坐標積木讓小貓畫出一個正方形。



撰寫程式

步驟 1

設定繪圖位置的坐標。

當 旗幟 被點擊

定位到 x: 100 y: 100

下筆

1 滑行 1 秒到 x: 100 y: -100

2 滑行 1 秒到 x: -100 y: -100

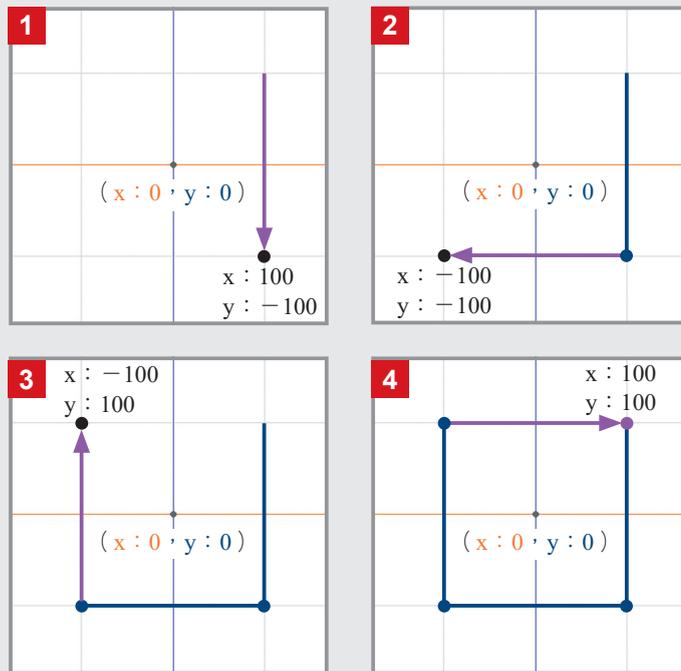
3 滑行 1 秒到 x: -100 y: 100

4 滑行 1 秒到 x: 100 y: 100

停筆

想一想！如果將圖中的點與坐標刪除，利用方向是否也可畫圖呢？

舞臺區繪圖步驟拆解



問題解析	問題實作
(A)如何設定角色的初始位置？	用此積木，輸入角色一開始出現的位置坐標。
(B)如何控制角色滑行至指定位置？	用此積木，輸入滑行時間與終點位置的坐標。

範例

利用**方向**積木讓小貓畫出一個正方形。

撰寫程式

步驟 1

設定前進方向與移動距離。

當 被點擊

筆跡全部清除

定位到 x: 100 y: 100

面朝 90 度

下筆

右轉 90 度

移動 200 點

出現第 1 次。

等待 1 秒

右轉 90 度

移動 200 點

出現第 2 次。

等待 1 秒

右轉 90 度

移動 200 點

出現第 3 次。

等待 1 秒

右轉 90 度

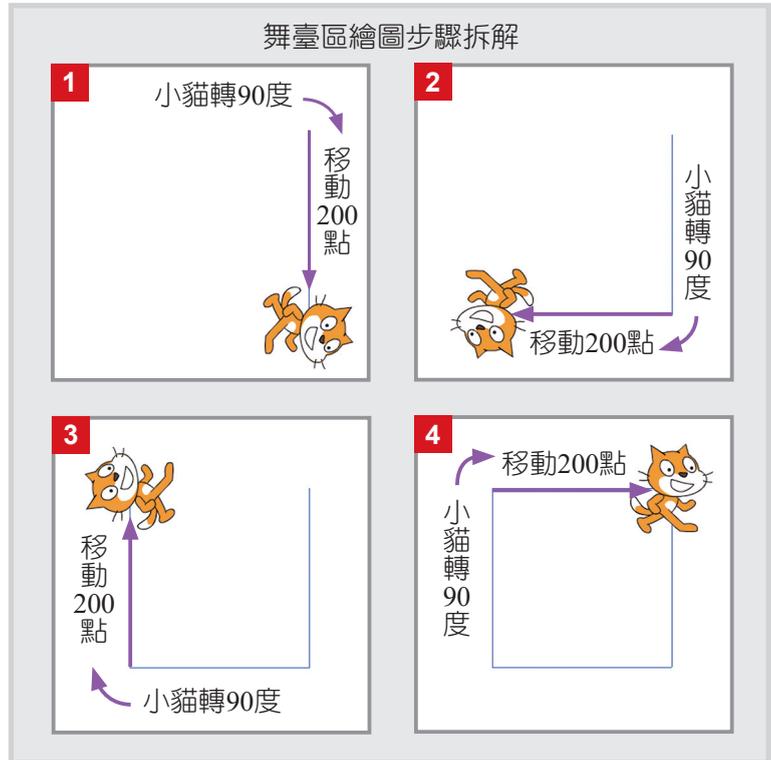
移動 200 點

出現第 4 次。

等待 1 秒

停筆

舞臺區繪圖步驟拆解



想一想！這個程式和上一個直接指定坐標的程式有什麼差別呢？你有沒有發現，當我們使用方向來寫程式時，程式碼會有重複出現的部分呢？對於重複出現的程式碼，我們可以用程式設計的**迴圈**概念來處理，這樣就可以精簡程式碼。

問題解析	問題實作
(A)如何設定角色的初始方位？	用此積木，輸入角色一開始所朝向的方位。  
(B)如何控制角色的轉向？	用此積木，輸入旋轉角度，向不同方向旋轉。    
(C)如何控制角色移動的距離？	用此積木，輸入要往前移動幾個點，代表移動幾個像素。 



範例

對應習作第 50 頁

利用計次式迴圈讓小貓畫出一個正方形。



撰寫程式

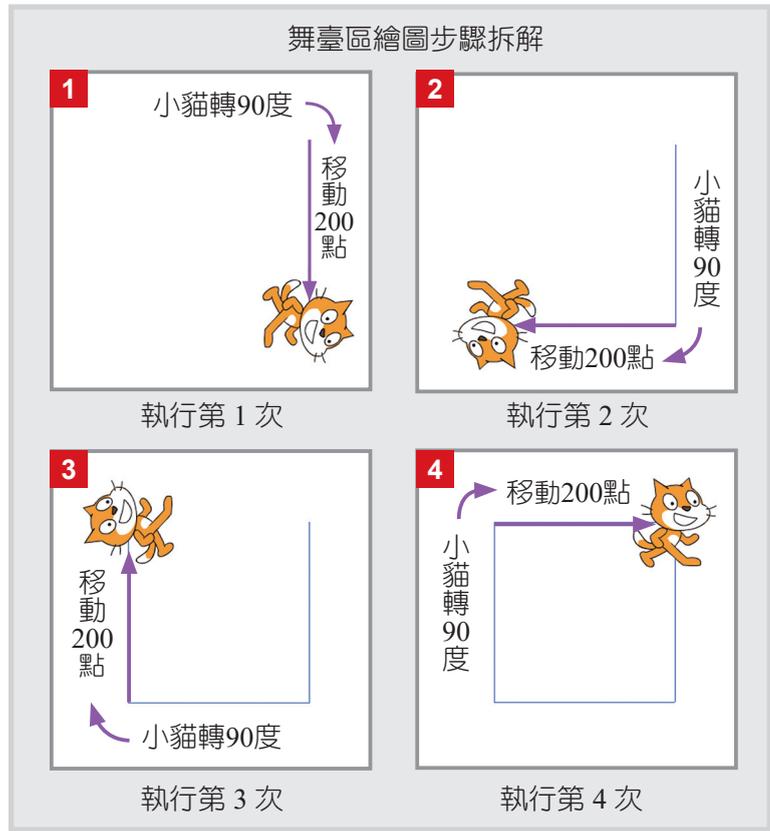
步驟 1

設定計次式迴圈，取代前一頁範例中重複的部分。

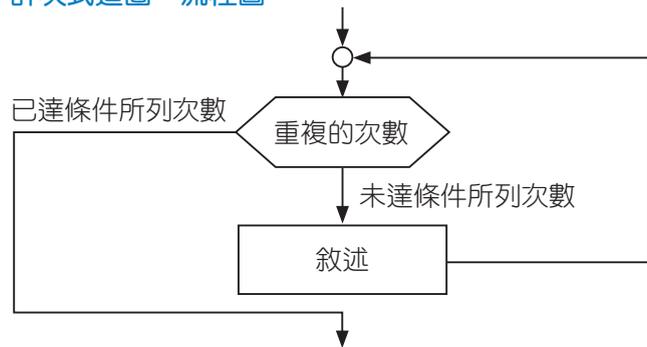
```

    當 被點擊
    筆跡全部清除
    定位到 x: 100 y: 100
    面朝 90 度
    下筆
    重複 4 次
    右轉 90 度
    移動 200 點
    等待 1 秒
    停筆
    
```

重複執行 4 次，對照右上圖的 1 ~ 4。



計次式迴圈－流程圖



問題解析	問題實作
(A)如何設定計次式迴圈？	用此積木取代重複執行程式碼。
(B)如何控制角色的轉向？	用此積木，輸入旋轉角度，向不同方向旋轉。
(C)如何控制角色移動的距離？	用此積木，輸入要往前移動幾個點，代表移動幾個像素。

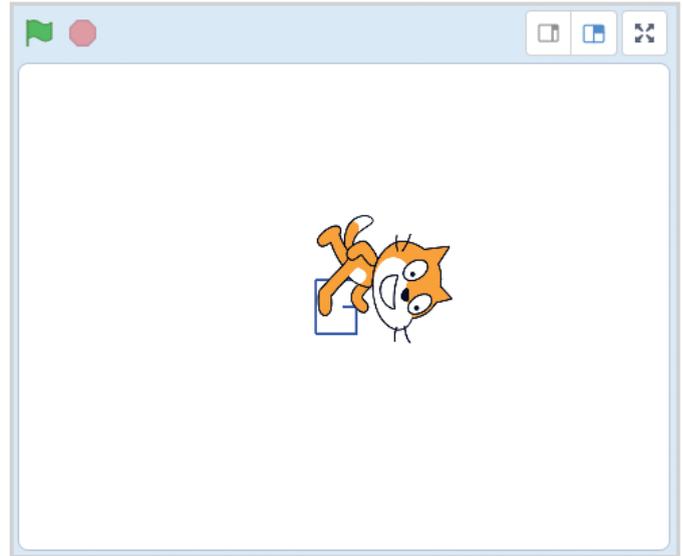
2-4-2 畫擴散方形

現在我們要畫一個擴散的方形，每次移動的距離都增加，並轉 90 度。想一想，有沒有比較快的方法？

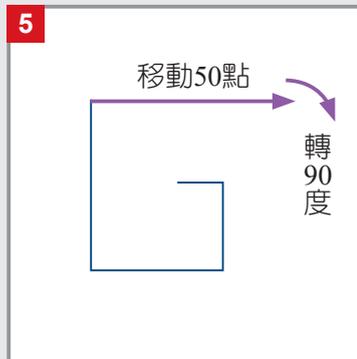
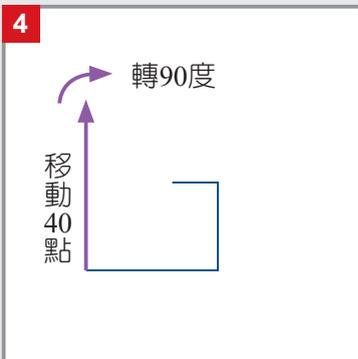
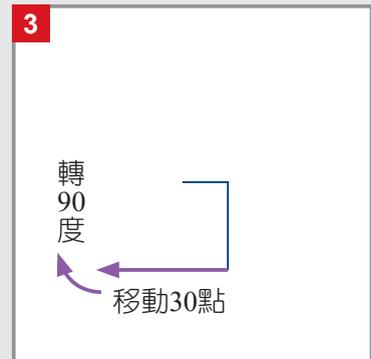
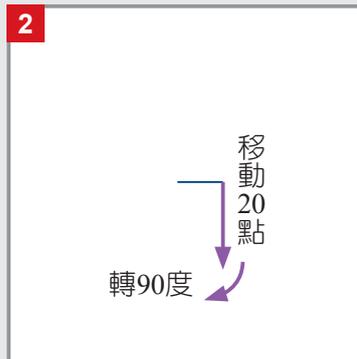


範例

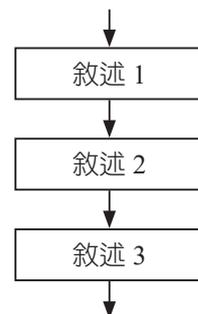
利用**循序結構**畫出一個擴散的方形。



舞臺區繪圖步驟拆解



循序結構－流程圖





撰寫程式

步驟
1

設定方向，但逐漸增加
每次移動的距離。

當被點擊

筆跡全部清除

定位到 x: 0 y: 0

面朝 90 度

下筆

移動 10 點

右轉 90 度

等待 1 秒

移動 20 點

右轉 90 度

等待 1 秒

移動 30 點

右轉 90 度

等待 1 秒

移動 40 點

右轉 90 度

等待 1 秒

移動 50 點

右轉 90 度

等待 1 秒

停筆

出現第 1 次
移動 10 點。

出現第 2 次
移動 20 點。

出現第 3 次
移動 30 點。

出現第 4 次
移動 40 點。

出現第 5 次
移動 50 點。

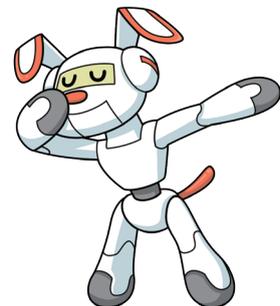
問題解析	問題實作
(A)如何控制角色移動的距離？	用此積木，輸入要往前移動幾個點，並逐漸增加移動的距離。
(B)如何控制角色的轉向？	用此積木，輸入旋轉角度，向不同方向旋轉。

觀察左方程式碼，有沒有重複出現的地方呢？小貓每次移動的點數是有規則的數列：**10、20、30、40、50**，所以我們可以配合**變數**的概念來產生這組數列，以達到精簡程式碼的目的。

這麼多重複的積木怎麼利用變數簡化呢？



讓我們繼續看下個範例吧！





範例

利用計次式迴圈與變數
畫出一個擴散的方形。



撰寫程式

步驟 1

開始設定新變數。

①

到程式面板，點選**變數**類別。

②

按下**建立一個變數**鍵，來新增變數。

步驟 2

選擇名稱與適用範圍。

③

跳出**新的變數**視窗。

④

在這裡輸入變數的名稱**長度**。

⑤

接著點選**適用於所有角色**。

⑥

按下**確定**鍵。

步驟 3

產生變數積木群組。

⑦

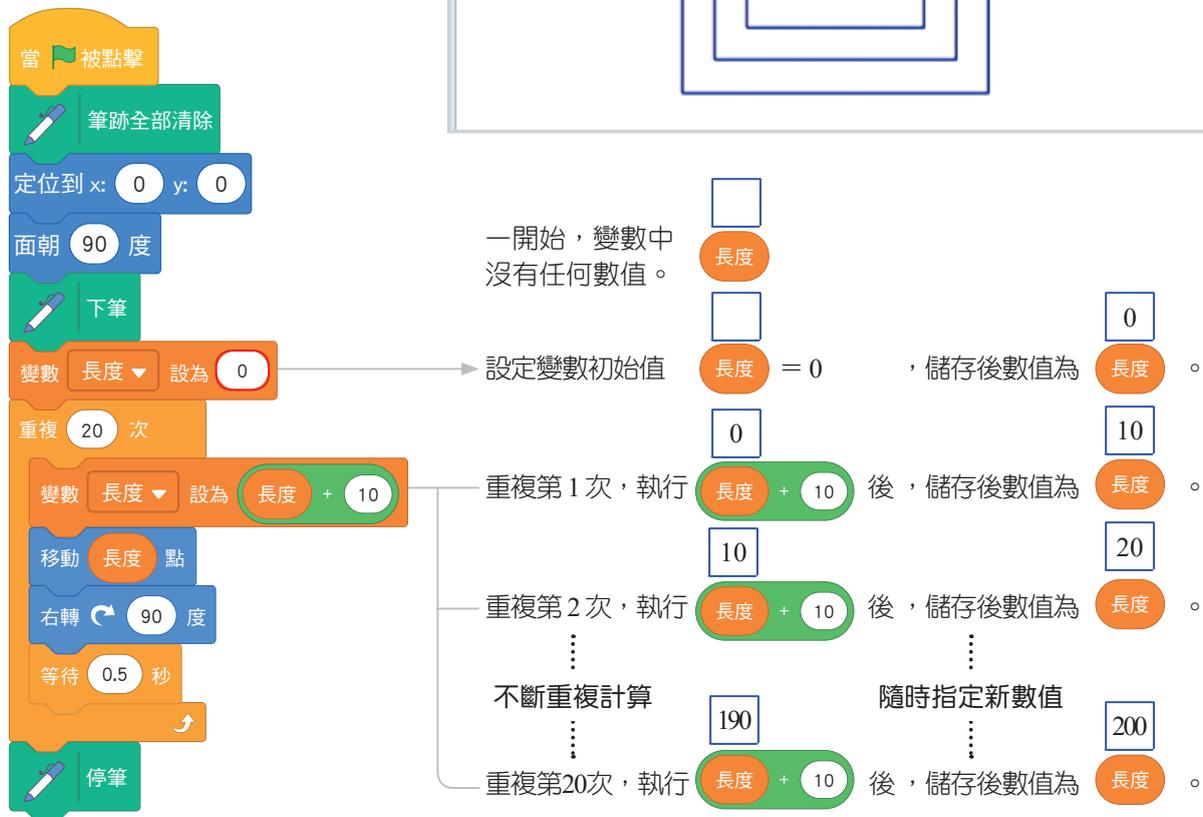
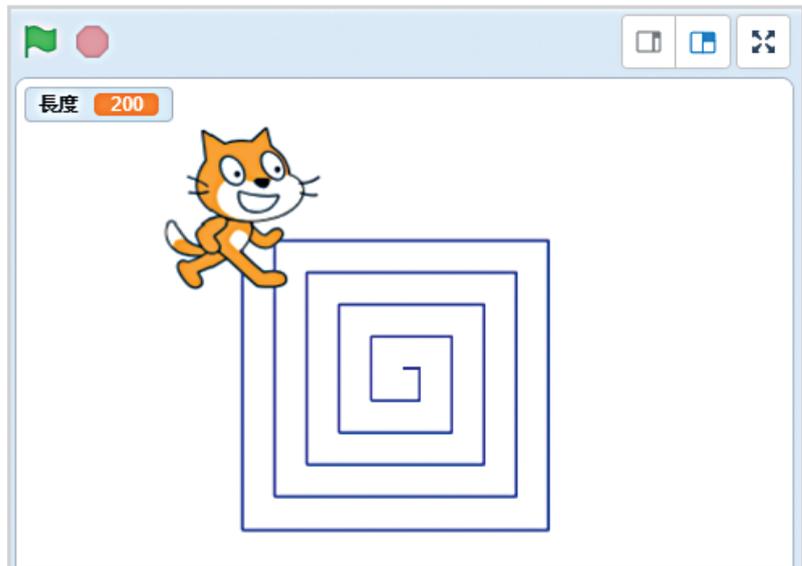
產生了 5 種變數積木。



步驟 4

利用迴圈積木撰寫程式

接著開始撰寫程式，加入變數積木，再利用迴圈積木取代上一個範例中重複的部分後，按綠旗執行。



問題解析	問題實作
(A)如何設定變數的初始值？	用此積木，輸入變數一開始的數值。 ← 鍵入初始值
(B)如何改變變數的數值？	根據指定運算的原則，輸入每次執行到此積木時，要重新儲存到變數的數值。 ← 放入指定變數數值的運算式
(C)如何改變每次移動的距離？	用變數取代固定數值，改變每次移動的距離。 ← 放入長度的變數

2-4-3 畫旋轉正方形

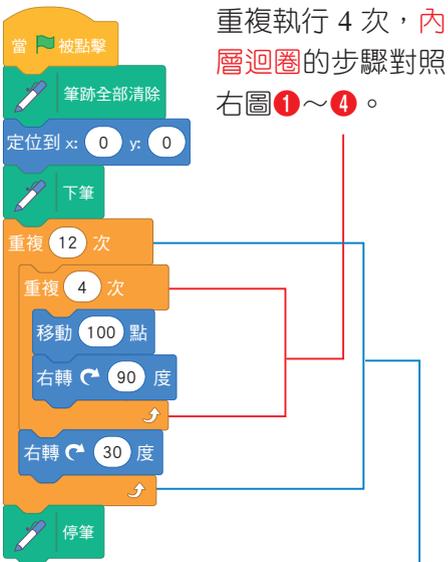
如果每次都先畫一個正方形後，再旋轉 30 度，要怎麼使用迴圈進行繪圖呢？

範例

對應習作第 51、52 頁

利用**巢狀結構**畫出 12 個旋轉的正方形。

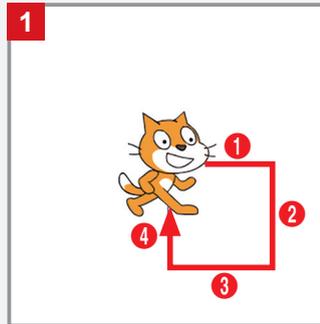
撰寫程式



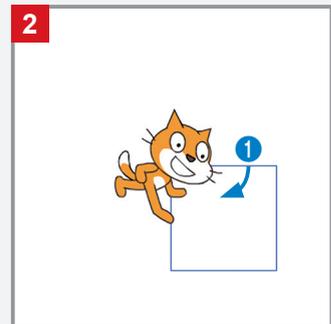
重複執行 12 次，外層迴圈每重複 1 次，內層迴圈就會重複 4 次，對照右圖的 1、2 ……。

巢狀結構指的是結構裡面還有結構。在範例的程式碼中，重複 12 次的迴圈結構中，又包含了一個重複 4 次的迴圈結構，所以是一個巢狀結構的程式。

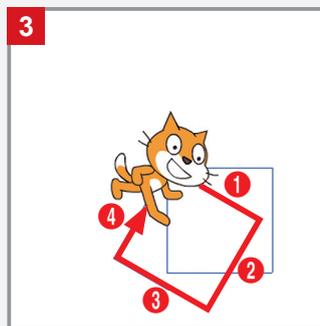
舞臺區繪圖步驟拆解



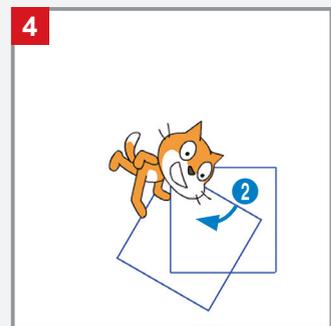
畫邊長 100 點正方形



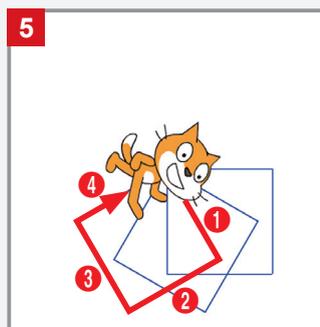
小貓轉 30 度



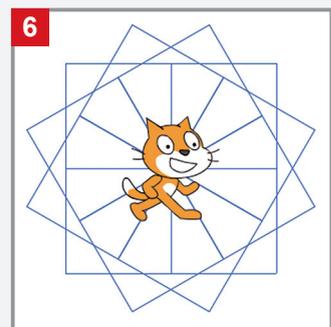
畫邊長 100 點正方形



小貓轉 30 度



畫邊長 100 點正方形 ……



最後完成圖形

在此單元中，我學到的有：

1. 能夠觀察事物特徵，將重複出現的部分，使用迴圈來簡化程式碼。
2. 能夠使用巢狀迴圈來簡化程式碼。
3. 能夠使用不同的方法達到相同的目標，並且了解其差別。

重點 回顧



演算法是一種解決問題的方法，程式語言則是實踐演算法的工具。解決問題的方法與過程可以寫成具體的步驟，再依照步驟去執行。為了表示演算法，可把具體的步驟整理成流程圖，以方便判讀與相互交流。而為了對演算法進行檢驗，必須將演算法轉換成電腦程式。因此，透過程式設計來實作演算法，要學習程式設計則需先選定一種程式語言。由於每人的思考模式不同，解決問題的辦法也不同，設計出來的演算法也會不同，但最重要的就是執行能產生正確的結果。

電腦只是一部機器，為了要指揮電腦完成某項工作，就要配合演算法，編寫許多指令或敘述，這些為完成某項工作而依其邏輯順序寫成的一連串指令，就是程式。只要給予指令，電腦就照指令執行，然後輸出結果。

程式語言發展的歷史遠比電腦來得早。從有程式語言至今，廣被使用的程式語言種類非常多，但因用途不同，功能也不一樣。雖然不同程式語言的語法不一樣，但基本邏輯則類似。建議初學者可以從一般用途的程式語言入門，建立了程式設計的基本概念後，再因應不同的專業需求，選用特殊用途的程式語言。

本課程選用易學有趣的 Scratch 3.0 來學程式設計。Scratch 是免費的自由軟體，它提供了視覺化的圖形操作介面，只要會用滑鼠拖曳積木，就能輕易的撰寫程式。在基礎篇以簡易的動畫實作，介紹操作介面及程式碼。接著在計算篇中，學習程式設計的基本邏輯（循序、選擇及重複）結構，在繪圖篇中，則加強迴圈概念來精簡程式碼。

同學們有想過龐大的統計資料要怎麼整理成有用的資訊嗎？下一章就要來告訴你們！



第 3 章

資料處理與分析

我們隨時隨地都會接觸到各式各樣的資料 (data)，資料也隨時隨地在產生，它與我們的關係極為密切，因此，了解其意義及產生的過程，重要性自不待言。任何形式的資料，有些需要經過適當處理或分析，才可以使用或是展現出具有意義的資訊 (information)。



在上圖中，處理/分析需要適當的程序與有效的工具，處理的工具可用人力或機器，目前電腦已被普遍的應用於資料處理，也被視為是最有效的資料處理工具。

- 3-1 資料的形式與意義
- 3-2 資料搜尋
- 3-3 資料處理與分析工具



10,000
9,000
8,000
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0

Jan



3-1 資料的形式與意義

資料隨時隨地都在產生，它與人的各種紀錄、意見或思想表達，關係極為密切，通常以文字、數字、圖形、影音的形式，或是這些形式的各種排列組合來表示（圖 3-1）。

以文字呈現的資料最為常見，大至政府的各種文件，小至個人的日記，通常以文字組成句子，再以句子構成段落，最後由段落形成有意義的文件資料，例如：作文、公文、報告或論文等（圖 3-2），都是典型的**文字資料**。

另外透過科學方法，把觀察或測量結果用數字記錄下來的資料，通常以數字表示，形成**數值資料**（圖 3-3）。而在數位時代，醫療行為、交通流量、消費市場活動、生產線上的紀錄、社交網路／搜尋網路使用者行為、航太載具蒐集／釋放的訊號等資料，隨時都在產生，也會留下各式各樣的紀錄。



資料隨時隨地在產生，它與人的各種紀錄、意見或思想表達關係極為密切。資料通常以文字、數字、圖形、影音的形式，或是這些形式的多種排列與組合來表示。

▲圖 3-2 思想表達的文字資料。



未經處理的資料稱為原始資料（raw data），必須透過適當的程序處理才能顯現其所代表的意義，這也是資料處理與分析的主要目的。

資料處理與分析常相提並論，資料處理通常要透過整理、分類、編碼及建立檔案等程序；而資料分析則要運用工具（如試算表軟體、統計分析軟體等）對已完成建檔的數位資料，進行計算、比較、排序等工作。經過處理與分析所得到的結果，有些還要加以詮釋，才能使資料成為一般人可以理解的有用資訊。

資料可分為數值資料及非數值資料兩大類，前者可以用算術四則運算加以處理，後者則以分類或排序的方式來處理。

資料的呈現方式通常以文字為主，其他如圖、表、符號等非數值資料，可透過適當的處理程序，顯現所代表的意義（圖 3-4）。當資料經過處理分析後，常會先將資料儲存於記憶體或雲端，等到使用者需要的時候，即可透過網路找到它們，並下載來使用；而數值資料則可以運用現有的軟體工具來處理與分析。

本章著重在搜尋非數值資料（文字、數字、圖形、影音）及試算數值資料的應用。這些內容與日常生活、課業學習，甚至未來就業的關係十分密切，大家要盡量熟練這些軟體工具的操作與使用。



▲圖 3-4 各種形式的條形圖。

3-2 資料搜尋

資料搜尋即是要在眾多的資料中，找到滿足某些條件的資料。對於有網路使用經驗的人來說，大致上都已經有搜尋資料的經驗。但是，若能再進一步了解一些實用性高的操作技巧，就可以更快速的獲取所需要的資料。接下來，以 Google 搜尋引擎為例，介紹資料搜尋技巧。

搜尋技巧 1

如果想找出同時滿足幾個關鍵字的網頁，只要在關鍵字間使用**空格**就可以，不需要使用符號。例如：搜尋所有包含了**果實**和**種子**的網頁，只要輸入**果實 種子**即可。

- ① 在搜尋處輸入**果實 種子**。
- ② 按搜尋鍵。
- ③ 同時出現**果實**和**種子**兩個語詞。

…**果實形態和種子的傳播**…



小提示

Google 的以圖找圖功能

除了以文字來搜尋圖片外，Google 圖片也提供以圖片搜尋圖片功能。

- 1 進入圖片搜尋畫面，點選 ，以圖搜尋。
- 2 填入圖片網址或上傳圖片。



搜尋技巧 2

如果想找出包含個別關鍵字的網頁，使用大寫的 **OR** 將關鍵字隔開，例如：搜尋包含**紫外線指數**或**雲量**的網頁，只要輸入**紫外線指數 OR 雲量**即可。

- 1 在搜尋處輸入**紫外線指數 OR 雲量**。
- 2 按搜尋鍵。
- 3 將只有**紫外線指數**或**雲量**語詞的資料一起列出來。

…**紫外線指數**分級…

…**雲量**是指視野所及…



搜尋技巧 3

如果想要在搜尋的資料內，排除某個關鍵字，在關鍵字前面加上**減號** – 即可。例如：搜尋所有包含**資訊展**，但排除**台北**的網頁，只要輸入**資訊展 - 台北**即可。

- ① 在搜尋處輸入**資訊展**。
- ② 按搜尋鍵。
- ③ 由於**台北**資訊展的相關資料較多，所以出現都是**台北**的**資訊展**。

… **台北資訊展** …



- ④ 為了要過濾掉台北的資訊展，尋找其他地區的資訊展相關資料，因此要在**資訊展**後面加上 **-台北**。
- ⑤ 按搜尋鍵。
- ⑥ 在**資訊展**中有關**台北**的超連結都被排除，剩下其他地區資訊展的超連結。

… 中區**資訊展** …

小提示

輸入減號時，要注意什麼？

注意，輸入減號時，記得在減號前面要輸入一個空格，但是減號後面要與關鍵字語詞連在一起，按**搜尋**鍵時才有作用。



資訊展-台北 - Google 搜尋

https://www.google.com.tw

Google 資訊展-台北 ④ ⑤

約有 15,600,000 項結果 (搜尋時間: 0.38 秒)

2017年中區資訊展8月11日起舉行6天-生活-自由時報電子報
 news.ltn.com.tw/news/life/breakingnews/2156732

2017年8月8日 - 暑期最大3C展, 「2017年中區**資訊展**」將於8月11日起至15日每日上午10點至下午6點, 一連5天, 在大台中國際會展中心盛大登場。主辦單位表示, 本次展覽雖因北、高兩地停辦暑期展覽而略受影響, 但仍有100家廠商、300個攤位共襄盛舉。

展場平面圖 - 台中電腦資訊展覽網
 https://tccashow.org.tw/info/floorplan/

106**資訊展**(台中區) - 參展廠商: ASUS. 主辦單位: 台中市電腦商業同業公會 40652台中市北屯區昌平路一段95-8號9F 洽詢電話: 04-22421717轉展覽組. 台中市電腦商業同業公會版權所有© 2012 All Rights

電腦公會展覽資訊網- 天啊... 中區資訊展實在太多免費入場方式了 ...
 https://www.facebook.com/...中區資訊展實在太多免費入場方 /1100223296672098/
 不是LINE粉絲嗎?沒關係沒關係... 趕緊上活動網站, 索取免費票券!! 數量有限, 領完為止哦... 中區**資訊展** 8/7-11 大台中國際會展中心活動網址: http://tccashow.org.tw/

2017年中區資訊展8月11日起舉行5天-生活-自由時報電子報
 news.ltn.com.tw/news/life/breakingnews/2156732

2017年8月8日 - 暑期最大3C展, 「2017年中區**資訊展**」將於8月11日起至15日每日上午10點至下午6點, 一連5天, 在大台中國際會展中心盛大登場。主辦單位表示, 本 ...

資訊展
 www.phsh.tyc.edu.tw/school/public/AlbumViewer/index.php?aid=714
 資訊展. 資訊科科遊. loading icon. / 資訊展. file icon IMG_1753.JPG · file icon IMG_1754.JPG · file icon IMG_1755.JPG · file icon IMG_1756.JPG · file icon ...

搜尋技巧 4

對於輸入的關鍵字，Google 有時候會將它拆解成幾個關鍵字來搜尋，若是不想被拆開，而是要找出符合某個詞組的網頁，可以在關鍵字前後加上**英文引號**“。例如：搜尋所有包含出現 **alphago 人工智慧**詞組的網頁，只要輸入“**alphago 人工智慧**”即可。

- ① 在搜尋處輸入“**alphago 人工智慧**”。
- ② 按搜尋鍵。
- ③ 單一超連結資料，**alphago 人工智慧**兩語詞接續出現。

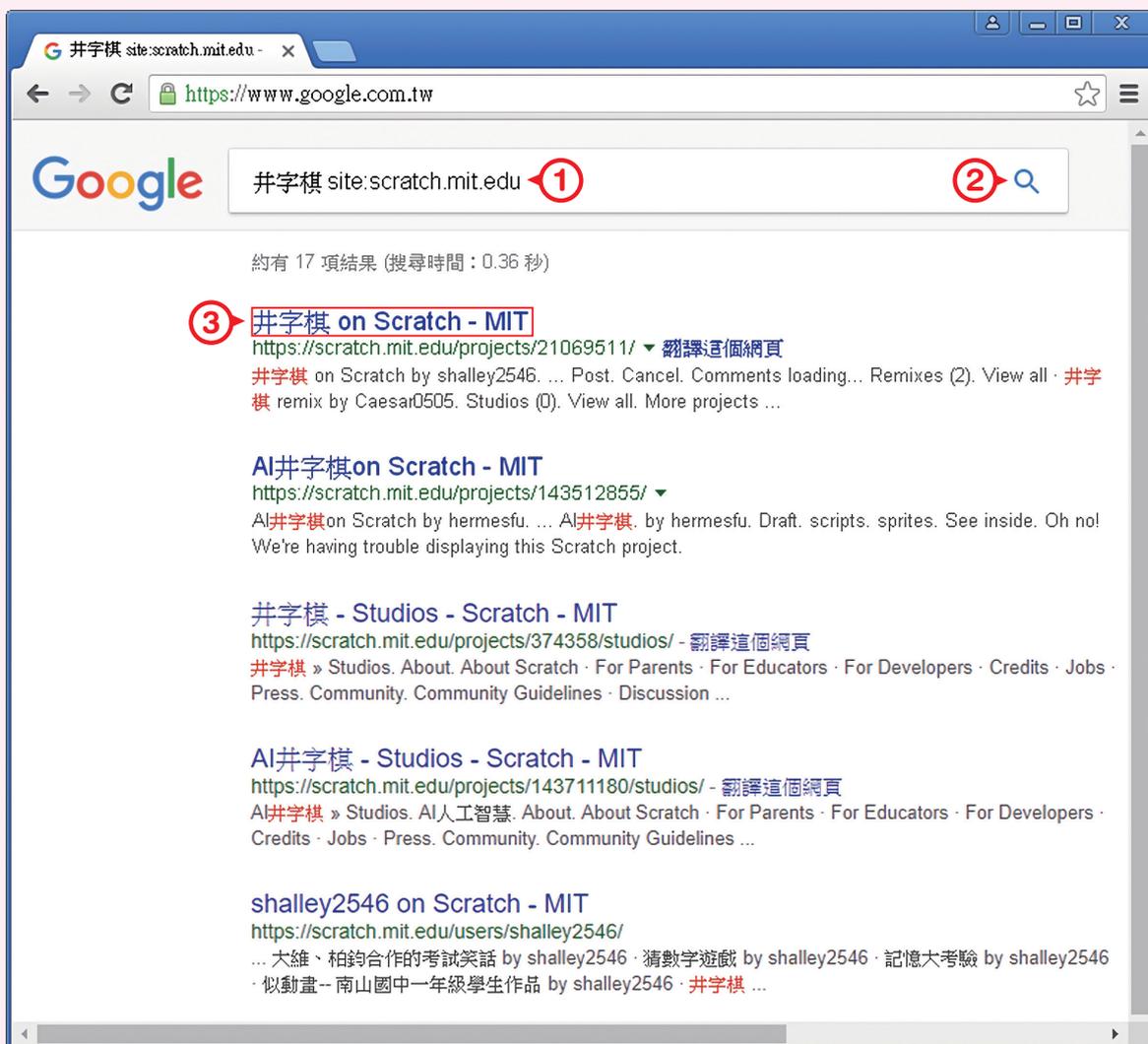
... AlphaGo 人工智慧系統 ...



搜尋技巧 5

如果要將搜尋限制在某個網站，只要在網站名稱前方加上 **site:**。例如：要搜尋 **Scratch** 官網中的井字棋網頁，只要輸入 **井字棋 site:scratch.mit.edu** 即可。

- ① 在搜尋處輸入 **井字棋 site:scratch.mit.edu**。
- ② 按搜尋鍵。
- ③ 出現 **Scratch 官網**中的**井字棋網頁**。



3-3 資料處理與分析工具

日常生活中，人們習慣將數據整理成表格來分析資料。在商業的應用上，通常也會使用表格來記錄、追蹤和預測公司的營運狀況，而這種表通稱為試算表（spreadsheet）。製作出試算表後，藉由改變表格中的某些數值，就可以觀察出將會產生的變化。然而，傳統試算表的計算不但需親自動手做，如果重新計算的內容擴增很多時，就會相當耗費人力。

過去有一位哈佛大學商學院學生丹·布李克林（Dan Bricklin）看到教授在黑板上用人工計算的方式，呈現財務報表的表格，既費時又容易出錯，於是他想出可以將這個過程，使用電腦呈現試算表背後的運算。他將此想法告知就讀麻省理工學院的鮑伯·法蘭克斯頓（Bob Frankston）後，兩人於 1979 年合作設計，推出史上第一個電子試算表軟體 VisiCalc，並將 VisiCalc 在蘋果二號（Apple II）電腦上銷售，頓時成為頗受歡迎的應用軟體。

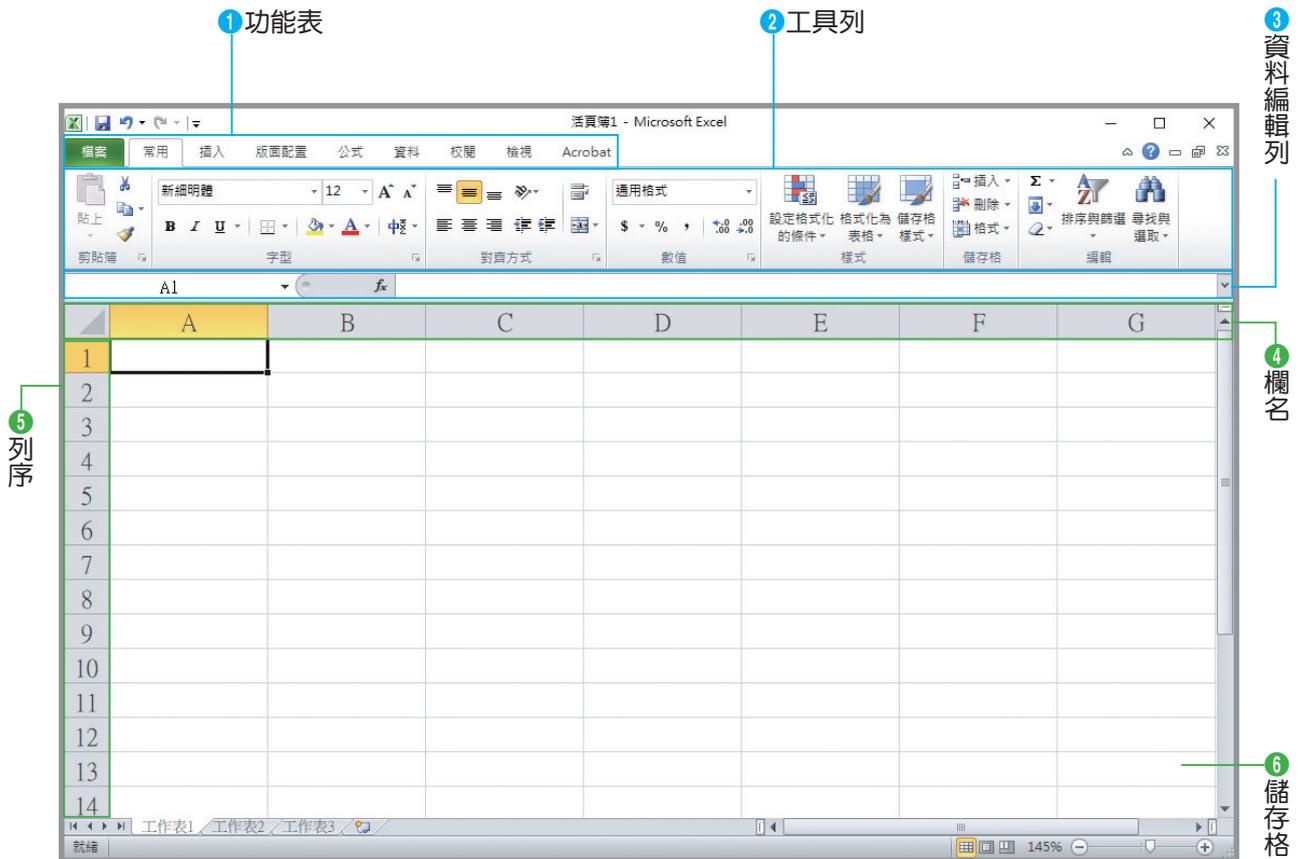
電子試算表的概念是使用者可依照個人需求，在各資料間設定數字關係，一旦變更某個數值，程式會重新自動計算所有相關數值。目前常見的軟體有商業軟體 Microsoft Excel（圖 3-5）、自由軟體 LibreOffice Calc 等。本單元將選用前者介紹試算表的介面及使用方式。



▲圖 3-5 Microsoft Office 中常見的軟體。

3-3-1 Excel 的操作介面介紹

進入 Microsoft Excel 後，你會看到許多細格子，我們稱這些格子為儲存格，每個儲存格都有一個名稱，儲存格上方的英文字母是它的欄名，左邊的數字是它的列序，欄和列對應起來就可分辨出每個儲存格的位置。



❶ 功能表：將常用的功能分類，用滑鼠按下分類選項會跳出子功能表。

❷ 工具列：顯示常用的功能鍵，或設定資料的格式，例如：字體、字的大小等。

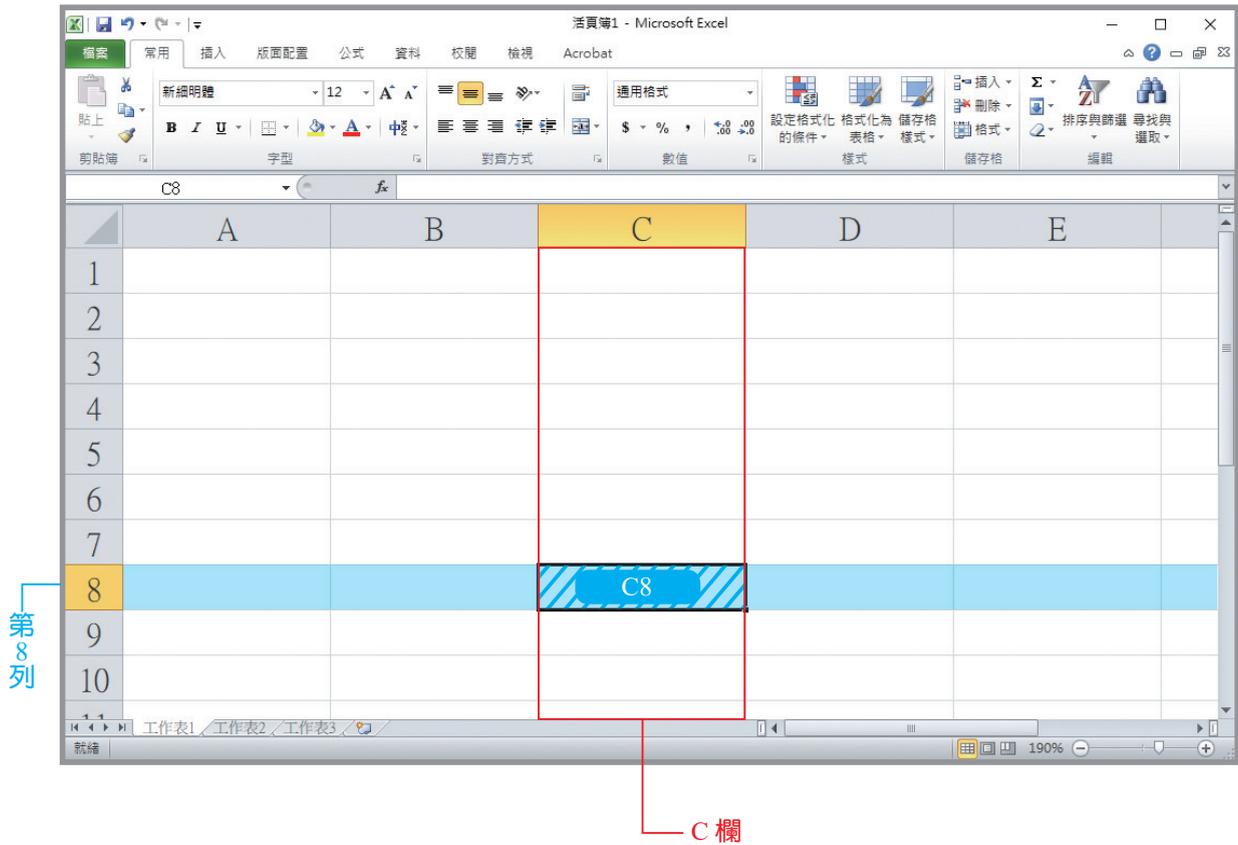
❸ 資料編輯列：顯示選取儲存格的資料內容，可在此進行編輯。

❹ 欄名：標示儲存格所在的欄，由 A 至 XFD，共 16384 欄。

❺ 列序：標示儲存格所在的列，由 1 至 1048576 列。

❻ 儲存格：放置資料的格子，若用滑鼠直接在格子上點兩下，就可在此編輯資料。

3-3-2 Excel 的儲存格欄名與列序



小提示

在這麼多的儲存格中，要如何識別不同的位置呢？

儲存格位置的命名規則：

例如：直行對應 C 欄，橫列對應第 8 列。

儲存格的位置就是 C8

3-3-3 Excel 實作 – 計算一天的花費



輸入資料

現在，我們要學習如何在儲存格中輸入資料。若輸入數字（數值資料），在試算表中預設是靠右對齊；若輸入文字（非數值資料），預設是靠左對齊。因此，直接觀察就可以分辨出在儲存格的資料中，哪些是數字，哪些是文字。

步驟 1

輸入文字。

- ① 在 A1 輸入項目。
- ② 在 B1 輸入金額(元)。
- ③ 在 A2 輸入早餐。
- ④ 在 A3 輸入公車。
- ⑤ 在 A4 輸入午餐。

	A	B	C
1	項目 ①	金額(元) ②	
2	早餐 ③		
3	公車 ④		
4	午餐 ⑤		
5			
6			
7			
8			
9			
10			

步驟 2

輸入數字。

- ⑥ 在 B2 輸入 39。
- ⑦ 在 B3 輸入 12。
- ⑧ 在 B4 輸入 60。

	A	B	C
1	項目	金額(元)	
2	早餐	39 ⑥	
3	公車	12 ⑦	
4	午餐	60 ⑧	
5			
6			
7			
8			
9			
10			



輸入公式

接下來，輸入簡單的公式 $B2 + B3 + B4$ ，讓試算表計算結果。記得要在公式前加上等號 $=$ ，告訴電腦現在輸入的公式要計算出結果。

步驟
3

輸入公式。

9 在 C4 輸入 $=B2+B3+B4$ 。

	A	B	C	D
1	項目	金額(元)		
2	早餐	39		
3	公車	12		
4	午餐	60	$=B2+B3+B4$	
5				
6				
7				
8				
9				

10 看看儲存格中所顯示的計算結果是不是 111？

	A	B	C	D
1	項目	金額(元)		
2	早餐	39		
3	公車	12		
4	午餐	60	111	
5				
6				
7				
8				
9				

小提示

為什麼要使用公式？

輸入公式的意義，就是可以直接讀取儲存格中的資料進行運算。

例如：

輸入公式 $=B2+B3+B4$ 。

B2：讀取 B2 儲存格數值 39。

B3：讀取 B3 儲存格數值 12。

B4：讀取 B4 儲存格數值 60。

因此計算結果就是 $B2+B3+B4$
 $=39+12+60=111$ 。

	A	B	C	D
1	項目	金額(元)		
2	早餐	39		
3	公車	12		
4	午餐	60	$=B2+B3+B4$	
5				
6				
7				
8				
9				

繼續在 A5、B5、A6、B6 輸入資料後，目前共有五筆記錄。如果用公式來計算總金額，必須輸入 $=B2+B3+B4+B5+B6$ 。

步驟 4

資料項目增加時，直接輸入公式麻煩且容易出錯。

① 在 A5 輸入文字**原子筆**，B5 輸入數值 **20**。A6 輸入文字**生日卡片**，B6 輸入數值 **30**。

② 在 C6 輸入公式 $=B2+B3+B4+B5+B6$ 。（打字愈來愈多！）

③ 看看儲存格中所顯示的計算結果是不是 **161**？

	A	B	C	D
1	項目	金額(元)		
2	早餐	39		
3	公車	12		
4	午餐	60	111	
5	原子筆	20		
6	生日卡片	30	$=B2+B3+B4+B5+B6$	
7				
8				
9				

	A	B	C	D
1	項目	金額(元)		
2	早餐	39		
3	公車	12		
4	午餐	60	111	
5	原子筆	20		
6	生日卡片	30	161	
7				
8				
9				

小提示

如何檢查公式資料來源是否有錯？

當公式愈來愈長，輸入資料時容易出錯。因此試算表中設計了一項功能，可顯示公式中引用資料的儲存格位置，以便檢查公式是否有錯。

將游標移至內儲公式儲存格，連擊滑鼠左鍵兩下後，公式資料來源，就會以不同顏色顯示，方便檢查資料來源的儲存格位置。

	A	B	C	D
1	項目	金額(元)		
2	早餐	39		
3	公車	12		
4	午餐	60	111	
5	原子筆	20		
6	生日卡片	30	$=B2+B3+B4+B5+B6$	
7				
8				
9				



輸入函數

隨著項目增加，使用公式時每個儲存格都要輸入一次，很麻煩而且容易出錯，對於這種連續範圍的儲存格，我們可以使用函數來簡化輸入。除此之外，試算表內建各種函數，方便使用者快速計算出所需要的結果。

小知識

函數 (function)

一般大型的程式有主副之分，副程式本身不會主動執行。在程式中，有些重複的工作（例如：累加等），為避免重複撰寫程式，通常會先把這些重複的工作寫成副程式，當要用到時再由主程式呼叫執行。一般套裝軟體裡，都會內建許多副程式，稱為函數。

步驟 5

利用函數來計算。

- 14 在 D6 輸入函數 `=SUM (B2 : B6)` 是不是比較快？
（可參閱 p234）

	A	B	C	D
1	項目	金額(元)		
2	早餐	39		
3	公車	12		
4	午餐	60	111	
5	原子筆	20		
6	生日卡片	30	161	<code>=SUM(B2:B6)</code>
7				14
8				
9				

- 15 看看儲存格中所顯示的計算結果是不是 161？

5	原子筆	20		
6	生日卡片	30	161	15 161

小提示

操作 SUM 函數

用此函數計算總和，方法是鍵入 `=SUM (起始儲存格 : 結束儲存格)`。

例如：

鍵入 `=SUM (B2 : B6)`。

B2：起始儲存格。

B6：結束儲存格。

從起始累加至結束儲存格，就是 $39 + 12 + 60 + 20 + 30 = 161$ 。

	A	B	C	D
1	項目	金額(元)		
2	早餐	起始儲存格 39		
3	公車	12		
4	午餐	60	111	
5	原子筆	20		
6	生日卡片	結束儲存格 30	161	<code>=SUM(B2:B6)</code>
7				
8				
9				



自動重算

自動重算是電子試算表中非常重要的功能，主要是設計用來取代傳統試算表中的人力計算。

試著將 B3 儲存格的數值改為 24，看看其他儲存格的資料是否會重新計算？

步驟
6

自動重算前，只買一段公車票。

	A	B	C	D
1	項目	金額(元)		
2	早餐	39		
3	公車	12		
4	午餐	60	111	
5	原子筆	20		
6	生日卡片	30	161	161
7				
8				
9				
10				

16

B3 的數值是 12。

17

C4 中是公式 $=B2+B3+B4$ ，結果是 111。

步驟
7

改成買兩段公車票後，自動重算。

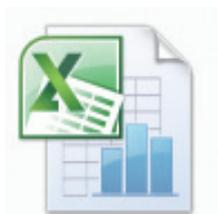
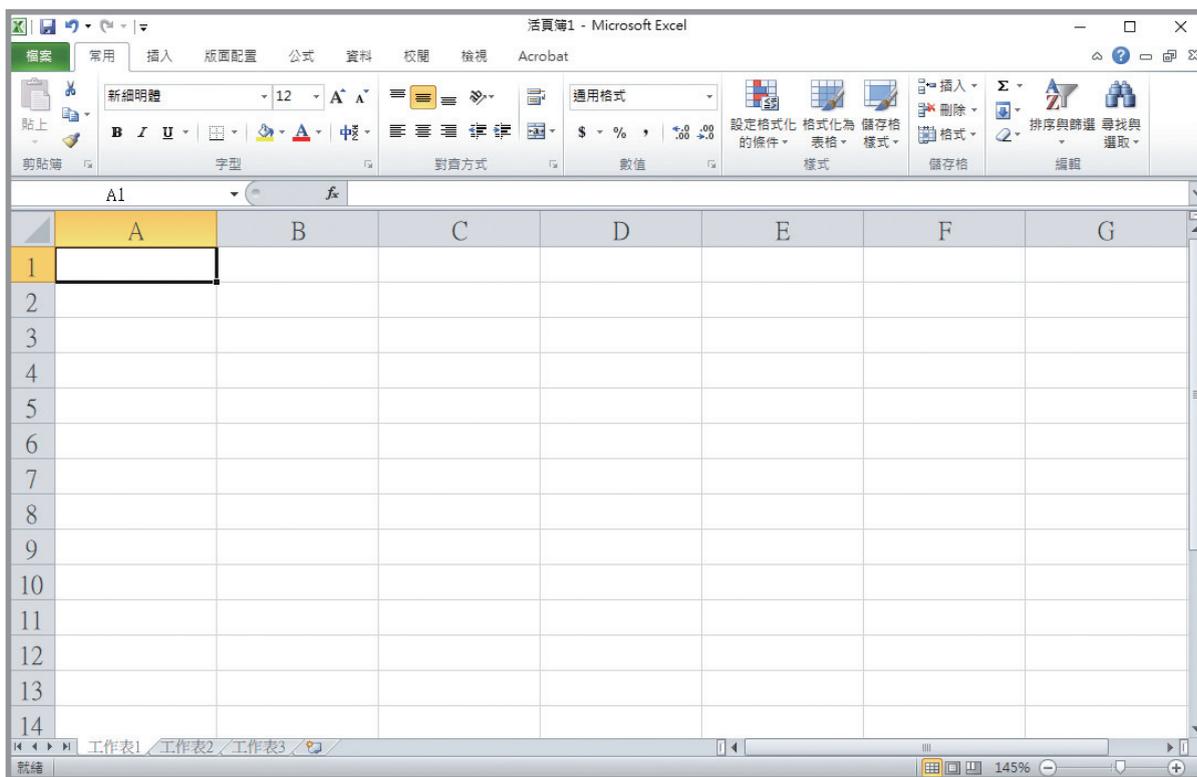
	A	B	C	D
1	項目	金額(元)		
2	早餐	39		
3	公車	24		
4	午餐	60	123	
5	原子筆	20		
6	生日卡片	30	173	173
7				
8				
9				
10				

18

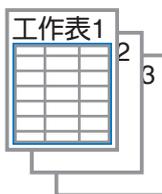
B3 修改為 24 以後，讀取 B3 作為資料來源的公式都會自動重算並更新數值。

資料整理

除了剛才使用過的計算功能之外，當需要處理的資料數量愈來愈多時，就得學習如何有效率的將不同性質的資料，進行分類與管理。現在，我們將要學習如何利用工作表的功能，進行分類並管理資料。



① 每一個儲存的檔案稱為活頁簿。



② 每一個活頁簿中可包含許多張工作表。

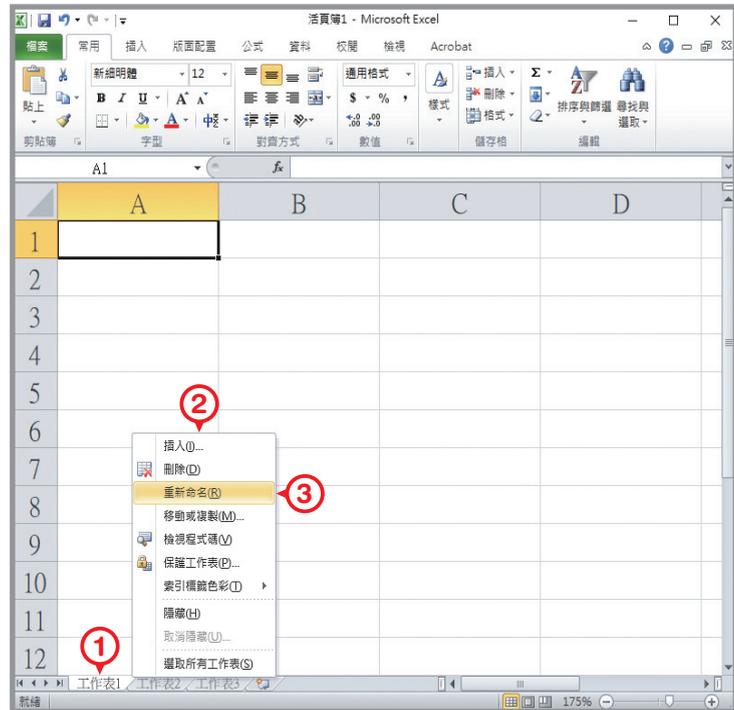


③ 點選視窗下方的工作表，可在不同的工作表間切換。

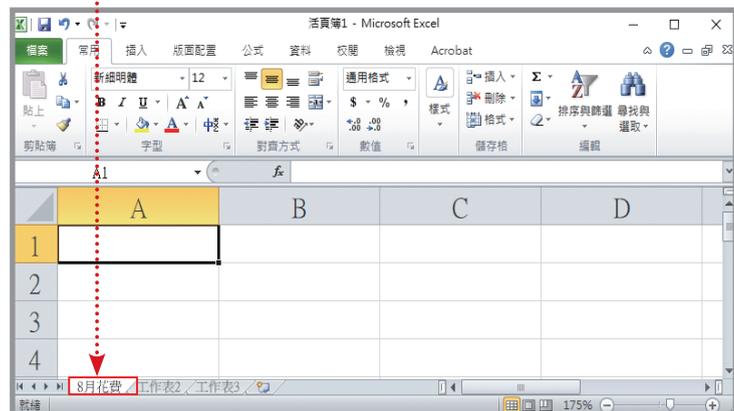
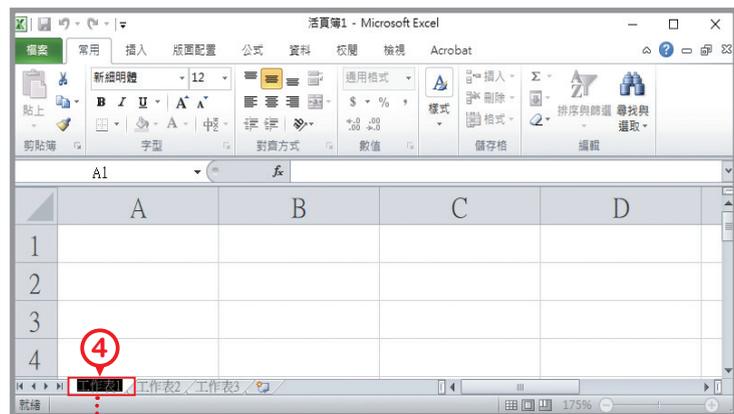
步驟 1

為現有的工作表命名。

- ① 先選取**工作表 1**。
- ② 按滑鼠**右鍵**。
- ③ 在選單中點選**重新命名**。

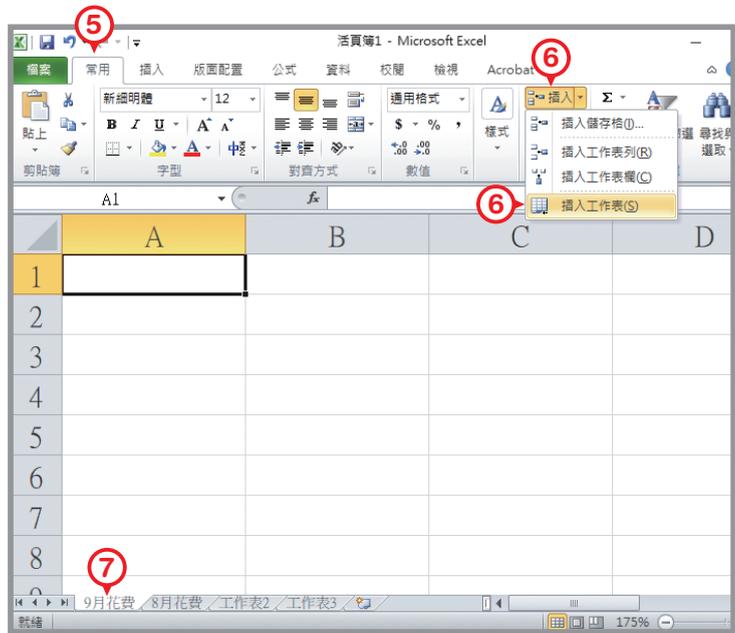


- ④ 將工作表命名為**8月花費**。



步驟 2 插入新的工作表。

- ⑤ 在功能表中按**常用**鍵。
- ⑥ 在工具列中點選**插入**，接著點選**插入工作表**。
- ⑦ 新工作的表單名稱為工作表 4，再依**步驟 1**的方式重新命名為 9 月花費。



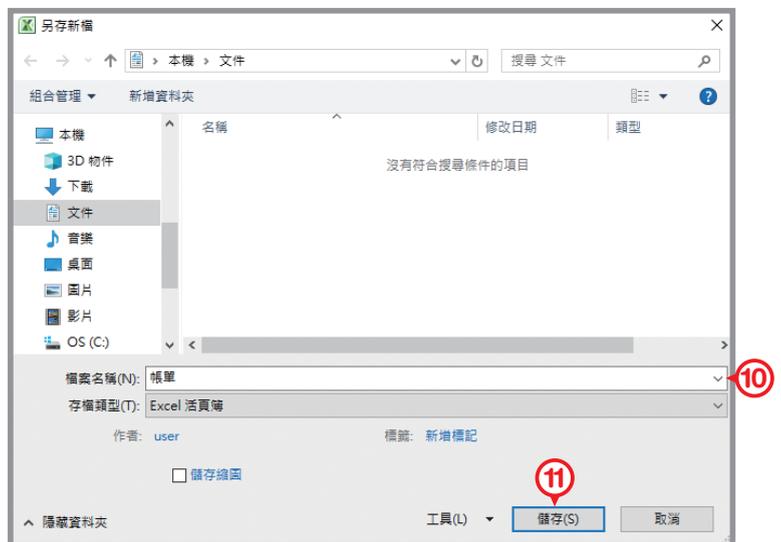
步驟 3 儲存活頁簿。

- ⑧ 在功能表中按**檔案**鍵。
- ⑨ 在選單中點選**另存新檔**。



- ⑩ 在跳出的視窗中，將活頁簿儲存為**帳單**。

- ⑪ 按**儲存**鍵。



3-3-4 Excel 實作 – 製作銷售統計

現在我們要來做一個實際的例子，假設某便利商店在各地設有分店，我們要統計某一週所銷售的礦泉水、飯糰、三明治、巧克力等四項商品，要計算各地區的銷售總金額，並畫圖分析各種商品的總銷售金額。

開啟範例檔－銷售統計.xlsx 後，可以看到目前已有資料包含縣市、品項、銷售金額等資料，如下圖所示。

		銷售金額				
		礦泉水	飯糰	三明治	巧克力	地區小計
3	單價(元)	20	39	35	15	
4	1 基隆市	1480	3315	3570	1665	
5	2 臺北市	36040	54639	56070	38025	
6	3 新北市	23940	23322	27930	23940	
7	4 桃園市	13240	17199	7735	3315	
8	5 新竹市	1780	2574	1540	1995	
9	6 新竹縣	3340	2145	1960	840	
10	7 苗栗縣	4400	3198	7700	1650	
11	8 臺中市	22360	5460	19565	4200	
12	9 彰化縣	10240	4992	13440	7680	
13	10 南投縣	4000	3900	3500	3000	
14	11 雲林縣	1380	2691	7245	2070	
15	12 嘉義市	1620	2106	1890	810	
16	13 嘉義縣	1020	2964	3570	2295	
17	14 臺南市	7540	14703	6580	8475	
18	15 高雄市	11100	21606	29120	4155	
19	16 屏東縣	4960	6435	2905	2490	
20	17 臺東縣	440	858	3080	990	
21	18 花蓮縣	2620	2535	3465	1965	
22	19 宜蘭縣	1820	1755	1610	690	
23	20 澎湖縣	840	819	1470	630	
24	21 金門縣	280	1053	1925	825	
25	22 連江縣	80	78	105	45	
26		礦泉水	飯糰	三明治	巧克力	
27		銷售總金額				

小知識

常用的函數

函數名稱	功能	函數名稱	功能
AVERAGE	計算數值資料的平均值。	MAX	找出選擇範圍資料的最大值。
MIN	找出選擇範圍資料的最小值。	SUM	計算數值資料的總和。

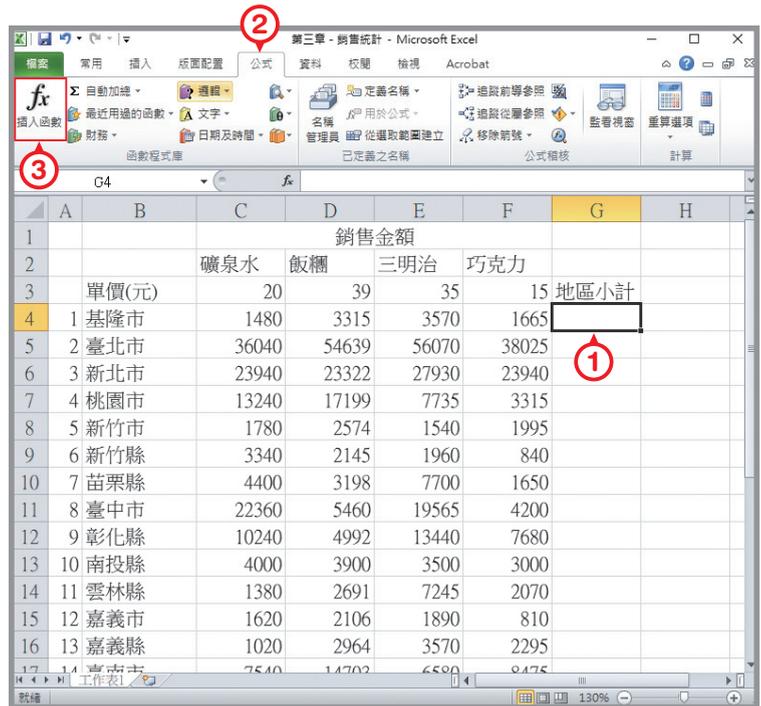


函數應用

步驟
1

現在開始利用函數計算各縣市銷售金額。

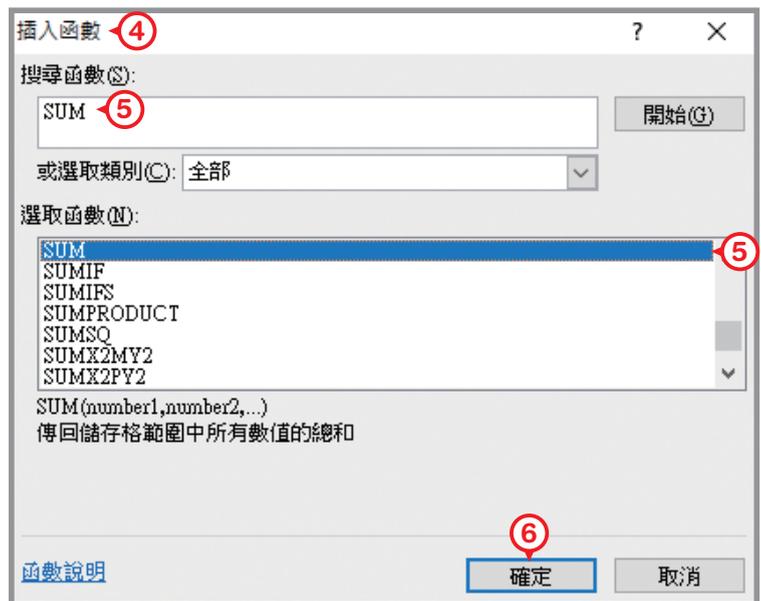
- ① 選取 G4 儲存格。
- ② 在功能表中按**公式**鍵。
- ③ 在工具列中點選**插入函數**。



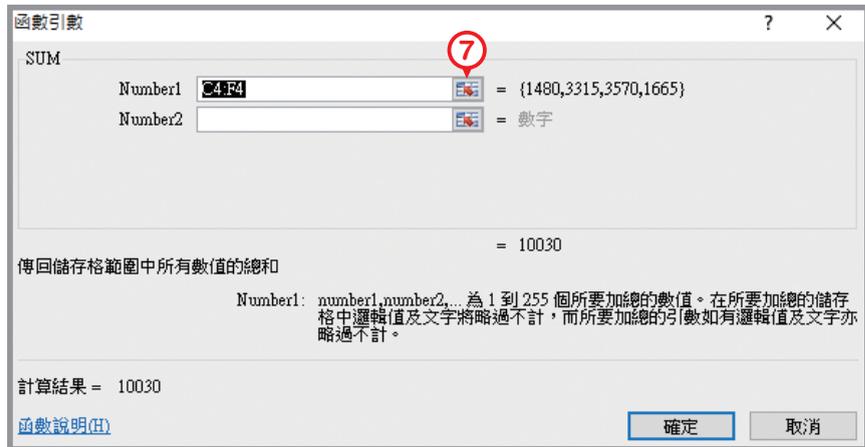
步驟
2

在跳出的視窗選取SUM函數。

- ④ 在試算表視窗上層會跳出插入函數視窗。
- ⑤ 可搜尋或點選 **SUM** 函數，用來加總每一個地區的銷售金額。
- ⑥ 按**確定**鍵。



⑦ 按**選取鍵**。



小提示

1. SUM 函數的意義

SUM 函數用來計算總和，使用方法是鍵入 = SUM (起始儲存格：結束儲存格)。而計算總和的資料範圍，右列三種形式皆可。

① 資料在同一欄／行

起始儲存格



結束儲存格

② 資料在同一列

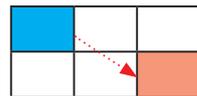
起始儲存格



結束儲存格

③ 資料在長方形區域內

起始儲存格



結束儲存格

2. SUM 函數的快速鍵

用滑鼠按下工具列的 Σ 自動加總鍵，即可直接產生 SUM 函數的表示式 = SUM ()，並自動預測加總的範圍。

3. fx 鍵的用途

按下 fx 鍵可以插入各種函數。



步驟
3

在儲存格選取加總資料範圍，
會跳出視窗完成函數設定。

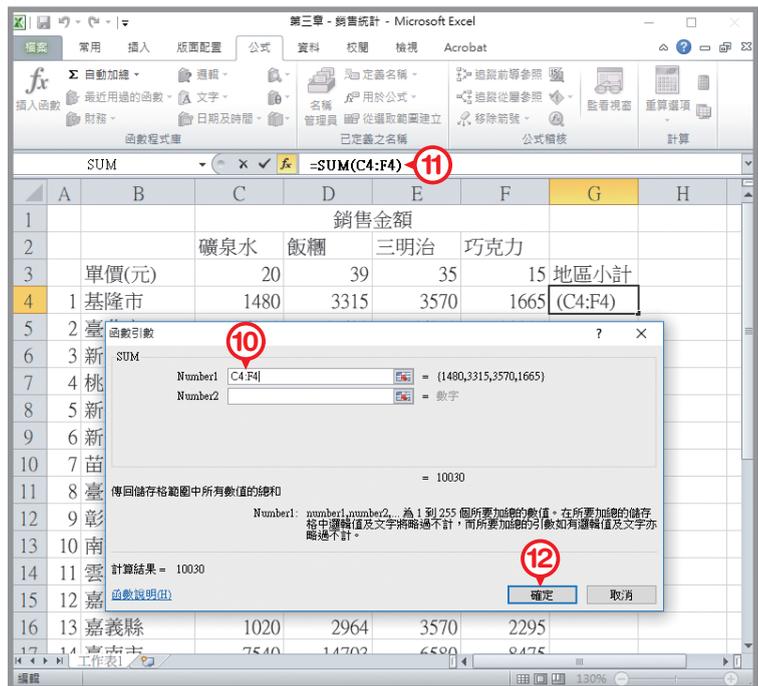
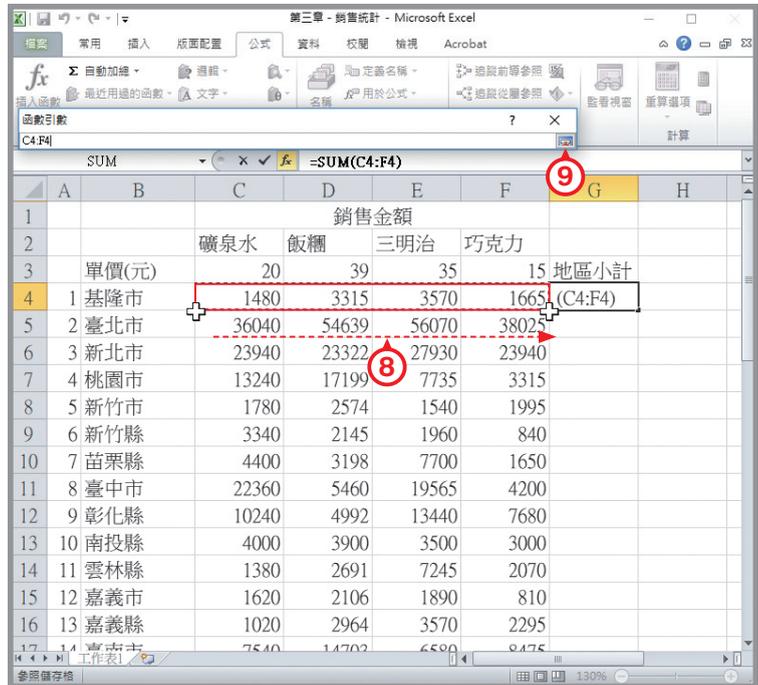
8 點選 C4 儲存格，接著按滑鼠左
鍵不要放開，向右拖曳至 F4 儲存
格，選取要加總的範圍。

9 此時函數引數視窗，已顯示儲存
格範圍，接著按**擴展**鍵。

10 試算表已自動在 Number1 的格子
中填入 C4:F4 文字。

11 最後得到公式 =SUM(C4:
F4)。

12 按**確定**鍵後，完成 SUM 的函數設
定。

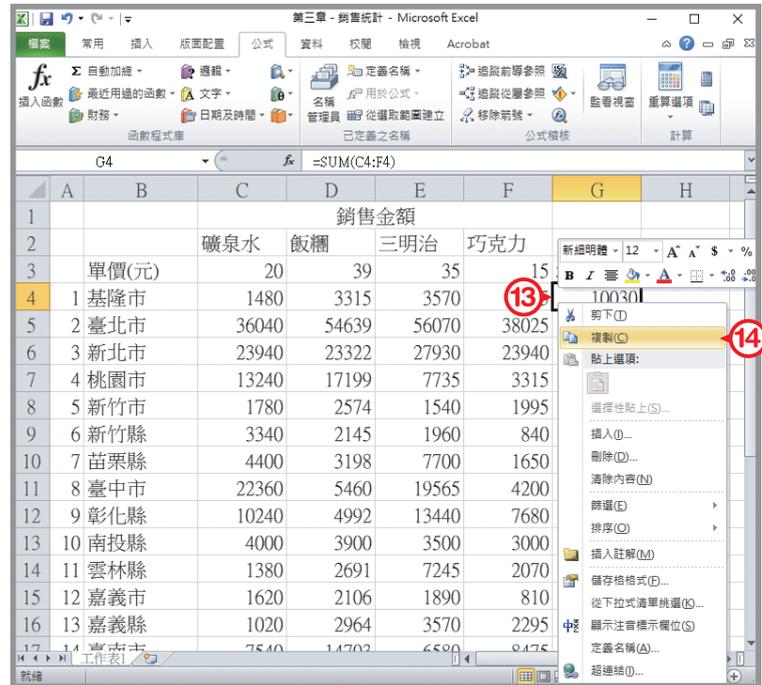


步驟 4

複製地區小計公式後，貼在其他的地區小計儲存格上。

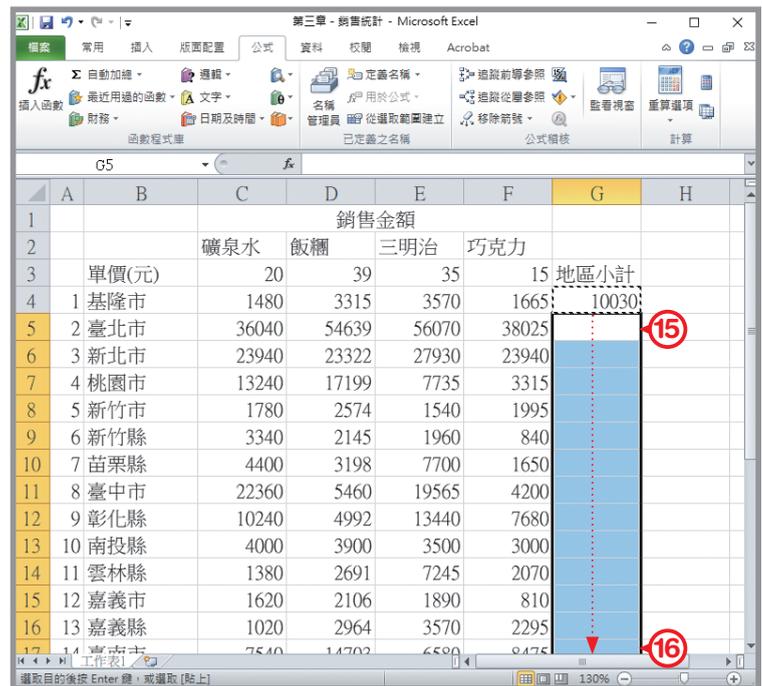
13 在 G4 儲存格上按滑鼠右鍵。

14 在彈出的快選功能表中，點選複製，複製第一個地區的銷售金額小計公式。



15 點選 G5 儲存格，接著按滑鼠左鍵不要放開。

16 從第二個地區開始向下拖曳選取要貼上公式的儲存格，拖曳至最後一個地區後停止。



步驟 6

用 SUM 函數計算品項小計。

20

利用 SUM 函數，分別計算各項商品的銷售總金額，也就是品項小計。

小提示

品項小計的公式寫法

礦泉水：=SUM (C4 : C25)

飯糰：=SUM (D4 : D25)

三明治：=SUM (E4 : E25)

巧克力：=SUM (F4 : F25)

	A	B	C	D	E	F	G	H	I	
1			銷售金額							
2			礦泉水	飯糰	三明治	巧克力				
3		單價(元)	20	39	35	15	地區小計			
4	1	基隆市	1480	3315	3570	1665	10030			
5	2	臺北市	36040	54639	56070	38025	184774			
6	3	新北市	23940	23322	27930	23940	99132			
7	4	桃園市	13240	17199	7735	3315	41489			
8	5	新竹市	1780	2574	1540	1995	7889			
9	6	新竹縣	3340	2145	1960	840	8285			
10	7	苗栗縣	4400	3198	7700	1650	16948			
11	8	臺中市	22360	5460	19565	4200	51585			
12	9	彰化縣	10240	4992	13440	7680	36352			
13	10	南投縣	4000	3900	3500	3000	14400			
14	11	雲林縣	1380	2691	7245	2070	13386			
15	12	嘉義市	1620	2106	1890	810	6426			
16	13	嘉義縣	1020	2964	3570	2295	9849			
17	14	臺南市	7540	14703	6580	8475	37298			
18	15	高雄市	11100	21606	29120	4155	65981			
19	16	屏東縣	4960	6435	2905	2490	16790			
20	17	臺東縣	440	858	3080	990	5368			
21	18	花蓮縣	2620	2535	3465	1965	10585			
22	19	宜蘭縣	1820	1755	1610	690	5875			
23	20	澎湖縣	840	819	1470	630	3759			
24	21	金門縣	280	1053	1925	825	4083			
25	22	連江縣	80	78	105	45	308			
26			礦泉水	飯糰	三明治	巧克力				
27		銷售總金額	154520	178347	205975	111750				

統計圖表

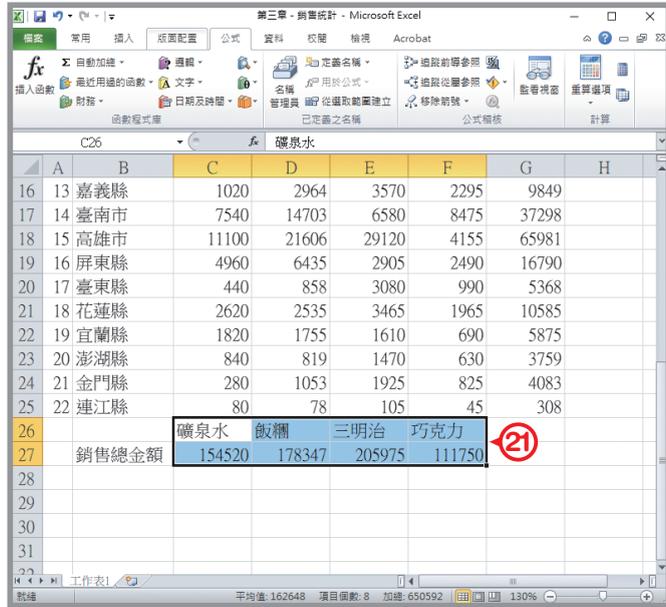
對於大部分的人來說，如果直接閱讀龐大的數值資料，非常容易迷失在數字叢林之中，而無法正確解讀資料數值之間的關係，以及背後所代表的意義。因此，我們可以将這些資料轉換成統計表或統計圖，利用視覺圖像來凸顯數值資料之間的關係與意義，變得更清晰易讀，而且容易理解。

接下來，我們要學習如何將前例中的各項商品銷售總金額，製作成統計表或統計圖。在功能表上選取插入項目中的圖表，以 C26 : F27 的資料，製作簡單的條形圖。

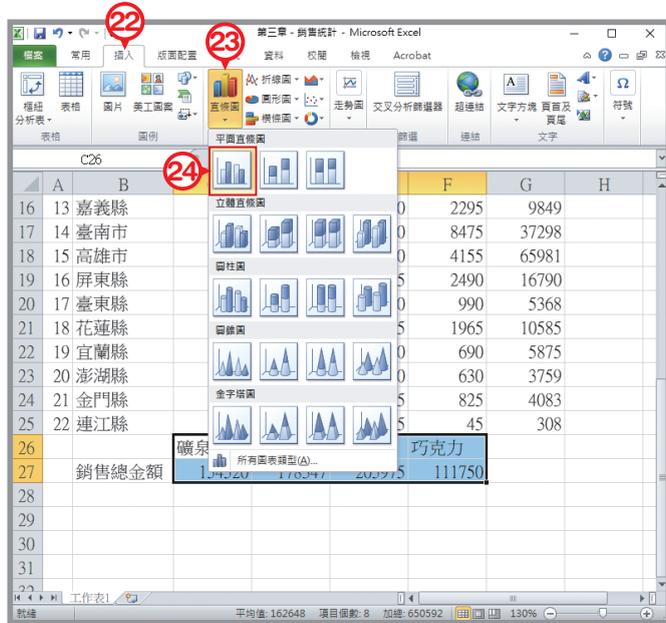
步驟 7

製作條形圖。

21 點選 C26 儲存格，接著按滑鼠左鍵不要放開，向右下拖曳至 F27 儲存格，選取各項商品名稱與銷售總金額。

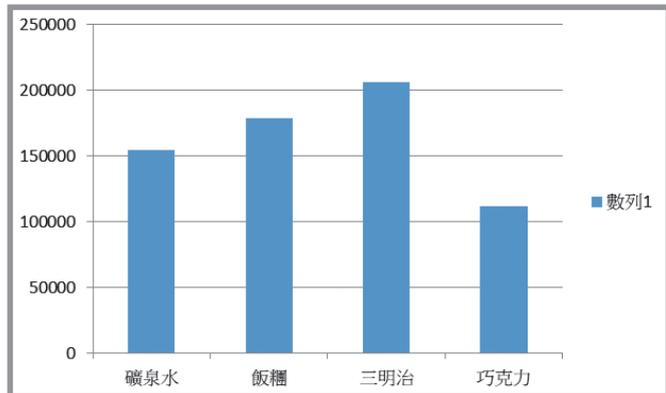


22 在功能表中按插入鍵。



23 在工具列中點選直條圖。

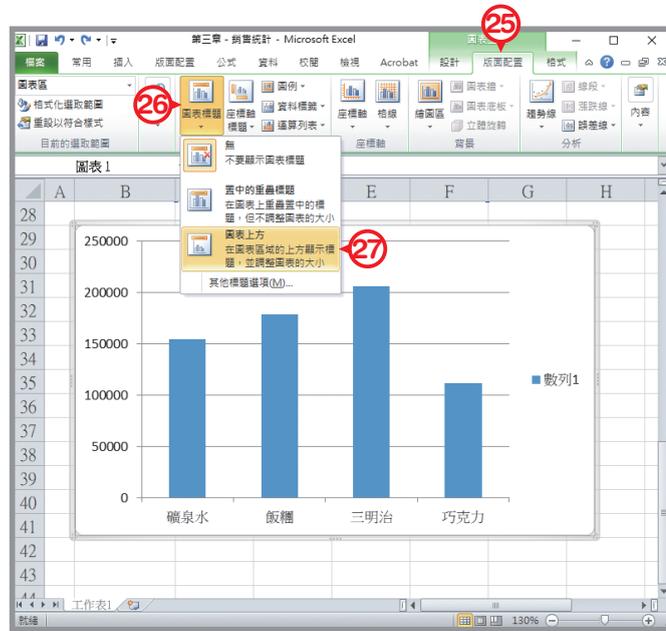
24 在平面直條圖中，點選群組直條圖。



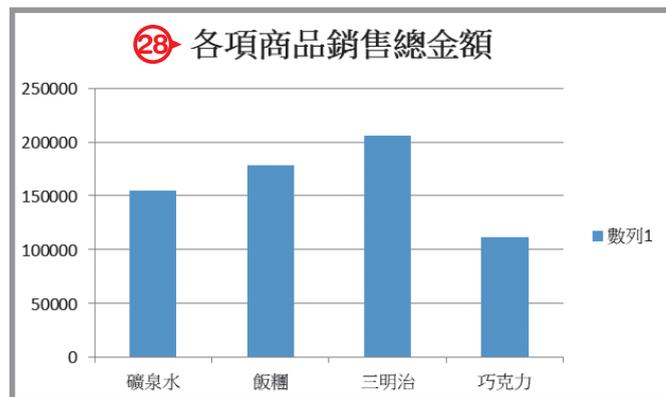
25 在圖表工具的功能表中按**版面配置**鍵。

26 在工具列中點選**圖表標題**。

27 點選**圖表上方**。



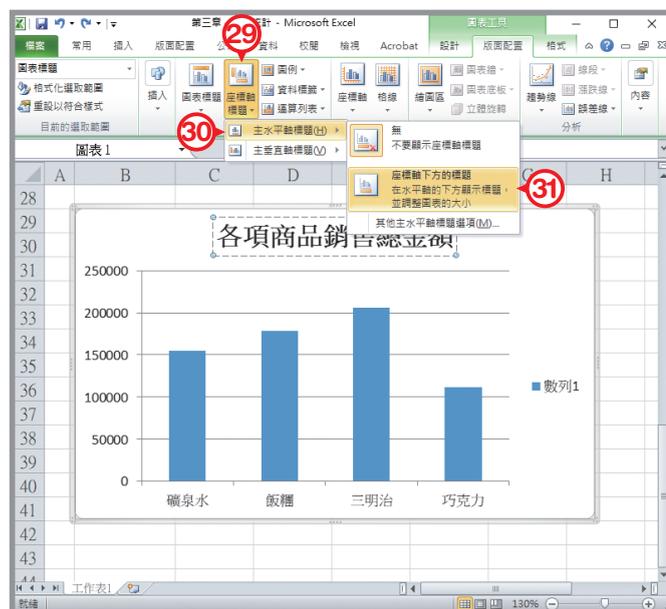
28 將圖表標題命名為**各項商品銷售總金額**。



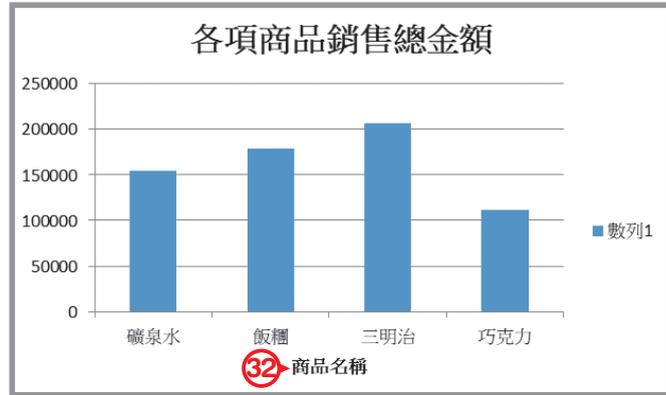
29 在工具列中點選**座標軸標題**。

30 點選**主水平軸標題**。

31 點選**座標軸下方的標題**。



32 將主水平軸標題命名為商品名稱。



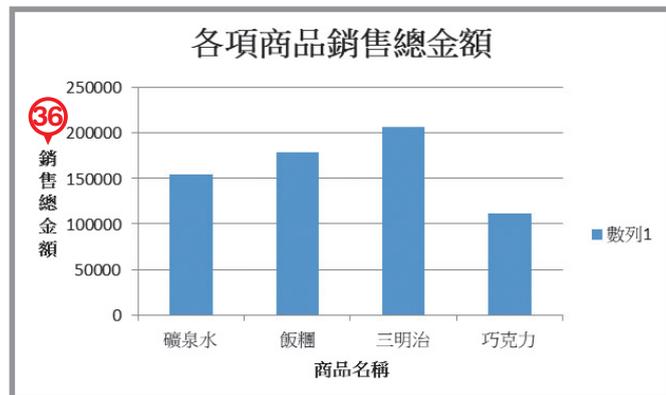
33 在工具列中點選座標軸標題。

34 點選主垂直軸標題。

35 點選垂直標題。



36 將主垂直軸標題命名為銷售總金額。



為了區分事、物或現象，數值資料的產生涉及到分派數值給事、物或現象的歷程；這些數值有四種不同特性，也就是用於區別（名義尺度，如學號）、表示相對關係（次序尺度，如名次）、表示相同單位的相互關係（等距尺度，如溫度），以及表示比率關係（比率尺度，如重量），以上所述之性質即稱為測量尺度。

根據資料的不同測量尺度，可以選用不同的視覺化圖形來呈現，就間斷資料（如學校類別、按讚數）來說，通常我們會選用條形圖，而連續資料（如身高、體重）則是選用折線圖或直方圖來呈現。直方圖用來表示不同數據的多寡，折線圖用來表示發展或變化的趨勢。

小知識

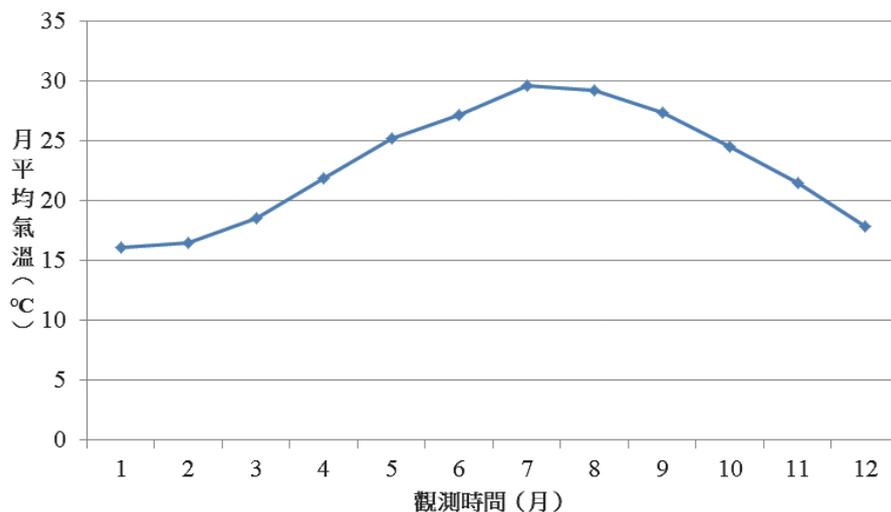
間斷資料 vs. 連續資料

名義及次序尺度是間斷變數，測量的結果是精確值，用整數表示；而等距及比率尺度是連續變數，測量的結果是近似值，也即一段距離，雖然一般也以整數的形式表示，但其實是實數。

連續資料舉例：

臺北市整年的月平均氣溫（中央氣象局統計 1981 ~ 2010 年間資料）

月份	1月	2月	3月	4月	5月	6月	7月	8月	9月	10月	11月	12月
平均氣溫(°C)	16.1	16.5	18.5	21.9	25.2	27.2	29.6	29.2	27.4	24.5	21.5	17.9



臺北市整年平均氣溫的折線圖

排序

步驟 8 將資料依照各地銷售金額由大至小向下排列。

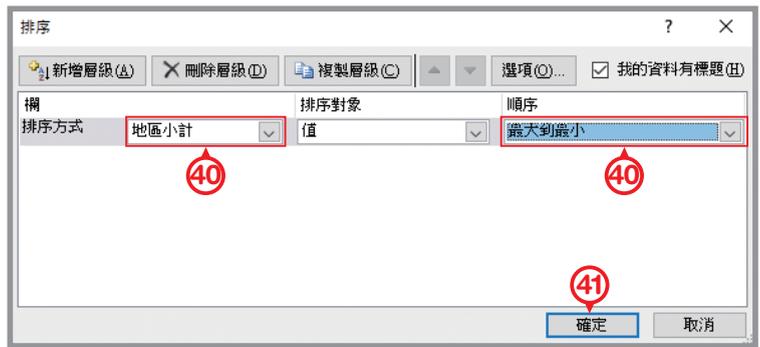
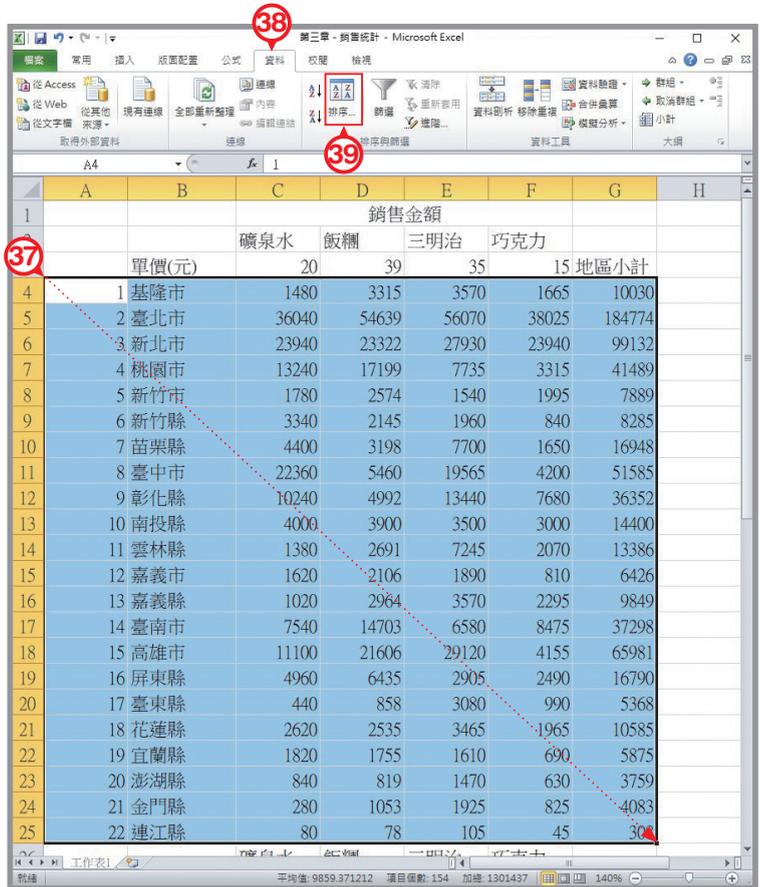
37 從儲存格 A4 起，按下滑鼠左鍵，向右下方拖曳，選取全部的資料範圍。

38 在功能表按資料鍵。

39 在工具列中點選排序。

40 在彈出的視窗中，欄排序方式選取地區小計，順序選取最大到最小。

41 按確定鍵後即完成。



重點 回顧



資料隨時隨地在產生，來源相當多元。例如：機構的各種文件、個人的思想表達等，都是已形成有意義的文件資料。科學或研究則透過觀察或測量的方式，將結果以數字記錄下來而形成**數值資料**。而在數位化時代，例如：醫療行為、交通流量、消費市場活動、生產線上的紀錄、社交網站使用者行為、航太載具蒐集或釋放的訊號等，都會留下大量各式各樣的資料。這些資料通常以文字、數字、符號、圖形、影音的形式，或是這些形式的多種排列與組合來表示。

未經處理的資料可泛稱為原始資料，必須透過適當的程序加以處理與分析才能顯現其所代表的意義，這也是資料處理與分析的主要目的。處理要透過整理、分類、編碼、及建檔等程序；而分析則要運用工具（如電腦）對已完成建檔的數位資料，進行計算、比較、排序、檢定等工作。經過上列處理與分析所得到的結果，有些還要加以詮釋，才能使資料顯示其意義，成為一般人可以理解的有用資訊。

本章著重在搜尋非數值資料（文字、數字、圖形、影音）及試算數值資料的應用；前後兩種資料分別以搜尋引擎及試算表軟體為工具，進行處理與分析。這些工具的運用與日常生活、學習，甚至未來就業與工作的關係十分密切，大家要盡量熟練這些軟體工具的操作與使用。

整理過後的資料，成為了有用的資訊及參考的依據！



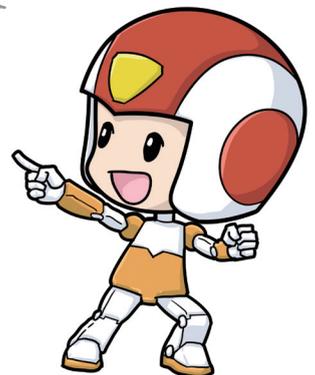
資訊圖片來源

第 1 章 資訊科技導論		
頁碼	圖號內容	提供來源
140-141	章首頁示意圖	depositphotos 提供
142	網路串聯行業別示意圖	編輯部製作
142	圖 1-1 線上學習示意圖	shutterstock 提供
143	資訊科技影響行業別示意圖	編輯部製作
143	課堂學習	depositphotos 提供
144-145	計算工具的發展人物	編輯部製作
146	圖 1-2 真空管	depositphotos 提供
146	圖 1-3 電晶體	shutterstock 提供
146	圖 1-4 積體電路	depositphotos 提供
147	圖 1-5 超大型積體電路	shutterstock 提供
147	圖 1-6 人工智慧示意圖	depositphotos 提供
148-149	圖 1-7 各種常見的電腦設備	depositphotos 提供
150	圖 1-8 利用運算思維解決生活中的問題	depositphotos 提供 編輯部製作
151	解決問題示意圖	編輯部製作
152	圖 1-9 安全認證系統示意圖	編輯部製作
153	圖 1-10 資訊倫理示意圖	編輯部製作
153	圖 1-11 資訊科技相關法律示意圖	編輯部製作
154	圖 1-12 平面媒體示意圖	depositphotos 提供
154	圖 1-13 電子媒體示意圖	shutterstock 提供
155	圖 1-14 閱讀部落格文章示意圖	depositphotos 提供
155	圖 1-15 觀看 YouTube 影片示意圖	shutterstock 提供
155	圖 1-16 社群媒體示意圖	depositphotos 提供
156	圖 1-17 硬體製造示意圖	depositphotos 提供
156	圖 1-17 軟體設計示意圖	shutterstock 提供
156	圖 1-17 網路通訊示意圖	shutterstock 提供
156-157	圖 1-17 地球圖	depositphotos 提供
157	圖 1-17 系統整合示意圖	depositphotos 提供
157	圖 1-17 支援服務示意圖	depositphotos 提供
157	圖 1-17 電子商務示意圖	shutterstock 提供
158	圖 1-18 STEAM 的學習領域	depositphotos 提供
159	圖 1-19 機器人示意圖	貝登堡智能提供

第 2 章 基礎程式設計 (1)		
頁碼	圖號內容	提供來源
162-163	章首頁示意圖	depositphotos 提供
164	廚師	depositphotos 提供
164	圖 2-1 製作蛋炒飯的流程圖	編輯部製作
167	圖 2-2 提花織布機	編輯部製作
167	圖 2-3 愛達·勒芙蕾絲	編輯部製作
168	圖 2-4 低階語言示意圖	編輯部製作
169	圖 2-5 高階語言示意圖	編輯部製作
170	程式語言主要功能示意圖	編輯部製作
171	常見的程式語言	編輯部製作
184	圖 2-6 結帳應用資訊科技示意圖	dreamstime 提供

第 3 章 資料處理與分析		
頁碼	圖號內容	提供來源
212-213	章首頁示意圖	depositphotos 提供
214	圖 3-1 具有意義的符號資料	編輯部製作
214	圖 3-2 思想表達的文字資料	編輯部製作
214	圖 3-3 身高和體重數值示意圖	編輯部製作
215	圖 3-4 各種形式的條形圖	編輯部製作
222	圖 3-5 Microsoft Office 中常見的軟體	編輯部製作

哇！對照圖片讓我更了解課本內容呢！



資訊中英文對照

第 1 章 資訊科技導論

頁碼	中文	英文
142	資訊科技	IT (Information Technology)
142	資訊及通訊科技	ICT (Information and Communication Technology)
143	行動應用程式	Apps (Mobile Application)
144	巴斯卡	Blaise Pascal
144	萊布尼茲	Gottfried Wilhelm Leibniz
144	巴貝奇	Charles Babbage
145	何樂禮	Herman Hollerith
145	阿塔納索夫	John Vincent Atanasoff
145	貝理	Clifford Edward Berry
145	ABC 電腦	Atanasoff-Berry Computer
145	馮紐曼	John von Neumann
146	真空管	Vacuum tube
146	電晶體	Transistor
146	積體電路	IC (Integrated Circuit)
147	超大型積體電路	VLSI (Very-large-scale Integration)
147	人工智慧	AI (Artificial Intelligence)
149	隨身碟	USB Flash Drive
149	軟碟片	floppy disk
149	軟碟機	floppy drive
150	運算思維	computational thinking
153	資訊近用權	Accessibility
154	影音分享平臺	YouTube
154	直播	live broadcast
154	媒體素養 / 媒體識讀	media literacy
156	資訊產業	information industry
156	IT 產業	information technology industry
158	科學	Science
158	科技	Technology
158	工程	Engineering
158	藝術	Arts
158	數學	Mathematics

第 2 章 基礎程式設計 (1)

頁碼	中文	英文
165	美國國家標準學會	ANSI (American National Standards Institute)
165	流程圖	Flow chart
167	約瑟夫·瑪麗·雅卡爾	Joseph Marie Jacquard
167	愛達·勒芙蕾絲	Ada Lovelace
168	加	ADD
169	指令	instruction
169	或	OR
170	啟動	Booting
172	麻省理工學院媒體實驗室	MIT Media Lab
172	終身幼稚園團隊	Lifelong Kindergarten Group
194	指定運算	assignment operation

第 3 章 資料處理與分析

頁碼	中文	英文
212	資料	data
212	資訊	information
215	原始資料	raw data
216	谷歌	Google
222	試算表	spreadsheet
222	丹·布李克林	Dan Bricklin
222	鮑伯·法蘭克斯頓	Bob Frankston
222	蘋果二號	Apple II
228	函數	function

利用中英對照查詢
名詞，會有更多意
想不到的收獲唷！

