國中資訊科技這樣教

林育慈
國立臺灣師範大學 資訊教育研究所

大綱

- 全球資訊教育趨勢
- 資訊科技教育的目的
- 教學設計
- 資訊科技教學策略
  - 運算思維導向教學設計
  - 不插電的電腦科學
  - 動手實作
  - 視覺化
  - STEM/STEAM教學

全球資訊教育趨勢

美國

全球資訊教育趨勢-2017 CSTA K-12 Computer Science Standards

Framework

Concepts

1. Computing Systems
2. Networks and the Internet
3. Data and Analysis
4. Algorithms and Programming
5. Impacts of Computing

Practices

1. Fostering an Inclusive Computing Culture
2. Collaborating Around Computing
3. Recognizing and Defining Computational Problems
4. Developing and Using Abstractions
5. Creating Computational Artifacts
6. Testing and Refining Computational Artifacts
7. Communicating About Computing

全球資訊教育趨勢-2017 CSTA K-12 Computer Science Standards

Level 1A: Grades K-2 (Ages 5-7)

Computing Systems

Identifier	Standard *and Descriptive Statement*	Subconcept	Practice
1A-CS-01	Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use. *People use computing devices to perform a variety of tasks accurately and quickly. Students should be able to select the appropriate app/program to use for tasks they are required to complete. For example, if students are asked to draw a picture, they should be able to open and use a drawing app/program to complete this task, or if they are asked to create a presentation, they should be able to open and use presentation software. In addition, with teacher guidance, students should compare and discuss preferences for software with the same primary functionality. Students could compare different web browsers or word processing, presentation, or drawing programs.*	Devices	1.1
1A-CS-02	Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware). *A computing system is composed of hardware and software.Hardware consists of physical components.Students should be able to identify and describe the function of external hardware, such as desktop computers, laptop computers, tablet devices, monitors, keyboards, mice, and printers.*	Hardware & Software	7.2
1A-CS-03	Describe basic hardware and software problems using accurate terminology. *Problems with computing systems have different causes. Students at this level do not need to understand those causes, but they should be able to communicate a problem with accurate terminology (e.g., when an app or program is not working as expected, a device will not turn on, the sound does not work, etc.). Ideally, students would be able to use simple troubleshooting strategies, including turning a device off and on to reboot it, closing and reopening an app, turning on speakers, or plugging in headphones. These are, however, not specified in the standard, because these problems may not occur.*	Troubleshooting	6.2, 7.2

全球資訊教育趨勢-2017 CSTA K-12 Computer Science Standards

Data and Analysis

1A-DA-05	Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data. *All information stored and processed by a computing device is referred to as data. Data can be images, text documents, audio files, software programs or apps, video files, etc. As students use software to complete tasks on a computing device, they will be manipulating data.*	Storage	4.2
1A-DA-06	Collect and present the same data in various visual formats. *The collection and use of data about the world around them is a routine part of life and influences how people live. Students could collect data on the weather, such as sunny days versus rainy days, the temperature at the beginning of the school day and end of the school day, or the inches of rain over the course of a storm. Students could count the number of pieces of each color of candy in a bag of candy, such as Skittles or M&Ms. Students could create surveys of things that interest them, such as favorite foods, pets, or TV shows, and collect answers to their surveys from their peers and others. The data collected could then be organized into two or more visualizations, such as a bar graph, pie chart, or pictograph.*	Collection Visualization & Transformation	7.1, 4.4
1A-DA-07	Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions. *Data can be used to make inferences or predictions about the world. Students could analyze a graph or pie chart of the colors in a bag of candy or the averages for colors in multiple bags of candy, identify the patterns for which colors are most and least represented, and then make a prediction as to which colors will have most and least in a new bag of candy. Students could analyze graphs of temperatures taken at the beginning of the school day and end of the school day, identify the patterns of when temperatures rise and fall, and predict if they think the temperature will rise or fall at a particular time of the day, based on the pattern observed.*	Inference & Models	4.1

全球資訊教育趨勢-2017 CSTA K-12 Computer Science Standards

Algorithms and Programming

1A-AP-08	Model daily processes by creating and following algorithms (sets of step-by instructions) to complete tasks. *Composition is the combination of smaller tasks into more complex tasks. Students could c algorithms for making simple foods, brushing their teeth, getting ready for school, particip time.*		
1A-AP-09	Model the way programs store and manipulate data by using numbers or other symbols to represent information. *Information in the real world can be represented in computer programs. Students could use thumbs up/down as representations of yes/no, use arrows when writing algorithms to represent direction, or encode and decode words using numbers, pictographs, or other symbols to represent letters or words.*	Variables	4.4
1A-AP-10	Develop programs with sequences and simple loops, to express ideas or address a problem. *Programming is used as a tool to create products that reflect a wide range of interests. Control structures specify the order in which instructions are executed within a program.* *Sequences are the order of instructions in a program. For example, if dialogue is not sequenced correctly when programming a simple animated story, the story will not make sense. If the commands to program a robot are not in the correct order, the robot will not complete the task desired.* *Loops allow for the repetition of a sequence of code multiple times. For example, in a program to show the life cycle of a butterfly, a loop could be combined with move commands to allow continual but controlled movement of the character.*	Control	5.2
1A-AP-11	Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. *Decomposition is the act of breaking down tasks into simpler tasks. Students could break down the steps needed to make a peanut butter and jelly sandwich, to brush their teeth, to draw a shape, to move a character across the screen, or to solve a level of a coding app.*	Modularity	3.2

不明確強調運算思維，但將運算思維融入於各項學習重點

英格蘭

全球資訊教育趨勢-2013 National Curriculum in England

理念

- 電腦科學是在理解運算及探討運算的建模，以及其於電腦系統發展的應用。
- 電腦科學的主要精神是運算思維，一種凌駕於電腦軟硬體之上的思維模式，並能提供理解系統與問題的理論架構。
- 運算思維奠基於許多理論與實務知識，以及能分析、建模、與解決問題的技巧。

全球資訊教育趨勢-2013 National Curriculum in England

目標

- 理解與應用基本的電腦科學原則與概念，包含：抽象化、邏輯、演算法、與資料表示
- 利用運算方法分析問題並能設計程式以解決問題
- 有效評估並應用資訊科技以解決問題
- 對於資訊與通訊科技能是個負責、有能力、自信、且有創意的使用者

全球資訊教育趨勢-2013 National Curriculum in England

主要概念

- Languages, machines, and computation
- Data and representation
- Communication and coordination
- Abstraction and design
- The wider context of computing

全球資訊教育趨勢-2013 National Curriculum in England

主要程序

- Computational thinking
  - Abstraction
  - Modelling
  - Decomposing
  - Generalizing and classifying
- Programming
  - Designing and writing programs
  - Abstraction mechanisms
  - Debugging, testing, and reasoning about programs

全球資訊教育趨勢-2013 National Curriculum in England

各階段能力指標

- Key stage 1 (ages 5-7)
  - 了解什麼是演算法，以及如何利用程式在數位設備上實作演算法，且了解程式將以確切的步驟執行
  - 寫簡單的程式且進行除錯
  - 利用邏輯思維預測簡單程式的行為
  - 利用科技產生、組織、儲存、操作、與擷取數位內容
  - 了解資訊科技的常見用途
  - 安全而尊重地使用科技、保持資訊隱私、了解使用網際網路或其他線上科技遇到問題時，該如何尋求協助

全球資訊教育趨勢-2013 National Curriculum in England

各階段能力指標

- Key stage 2 (ages 7-11)
  - 設計、撰寫、與除錯程式以達到特定目標，包含控制與模擬實體系統、利用解析成更小的問題來解題
  - 利用循序、選擇、與重複結構撰寫程式，能處理變數與不同型態的輸入輸出
  - 利用邏輯思維解釋簡單程式的運作，且能找出並修正演算法與程式的錯誤
  - 了解電腦網路(包含：網際網路)如何提供多元服務，例如全球資訊網，以及電腦網路提如何促進溝通與合作
  - 有效利用搜尋科技以鑑別資料的選擇與排序，並能評估數位內容
  - 為完成特定目的選擇、利用、並組合各種數位設備上不同軟體以設計與創作各種程式、系統與內容(包含：蒐集、分析、評估與展示資料與資訊)
  - 安全、尊重、且負責地使用科技，並能辨識可接受與不可接受之行為，知道如何利用不同方法處理使用科技時相關的問題

全球資訊教育趨勢-2013 National Curriculum in England

各階段能力指標

- Key stage 3 (ages 11-14)
  - 設計、使用與評估運算抽象化以模型化實體問題與系統的狀態與行為
  - 了解能反映運算思維之重要演算法(例如排序與搜尋)，並利用邏輯思維比較同一個問題之不同演算法
  - 利用兩種以上的程式語言，且其中至少有一種為文字型的語言，來解決不同的運算問題，並能適切使用資料結構(例如list、tables、或array)，且能利用程序或函數設計與發展模組化程式
  - 了解簡單的布林邏輯(例如AND、OR、與NOT)及其在電路與程式之使用，並了解如何以二元方法表示數字，且能處理簡單的二元運算(例如二元加法及二進制與十進制的換算)
  - 了解構成電腦系統的軟硬體元件，及其如何彼此溝通或與其他系統溝通
  - 了解指令如何於電腦系統儲存與執行
  - 了解如何以二元方式表示與處理不同形式的數位資料(包含：文字、聲音、與圖片)
  - 能進行創意式的專題製作，並能於製作中選擇、使用、與組合多種設備之應用軟體，以達成具挑戰性之目標(包含：蒐集與分析資料，並設法符合使用者需求)
  - 為特定使用者產生、再利用、與修改數位作品，並考量其可靠性、設計與可用性
  - 了解能安全、尊重、負責、且安全地使用科技的各種方法(包含：保護線上身分與隱私)，並能適切地辨識並報導相關議題

全球資訊教育趨勢-2013 National Curriculum in England

各階段能力指標

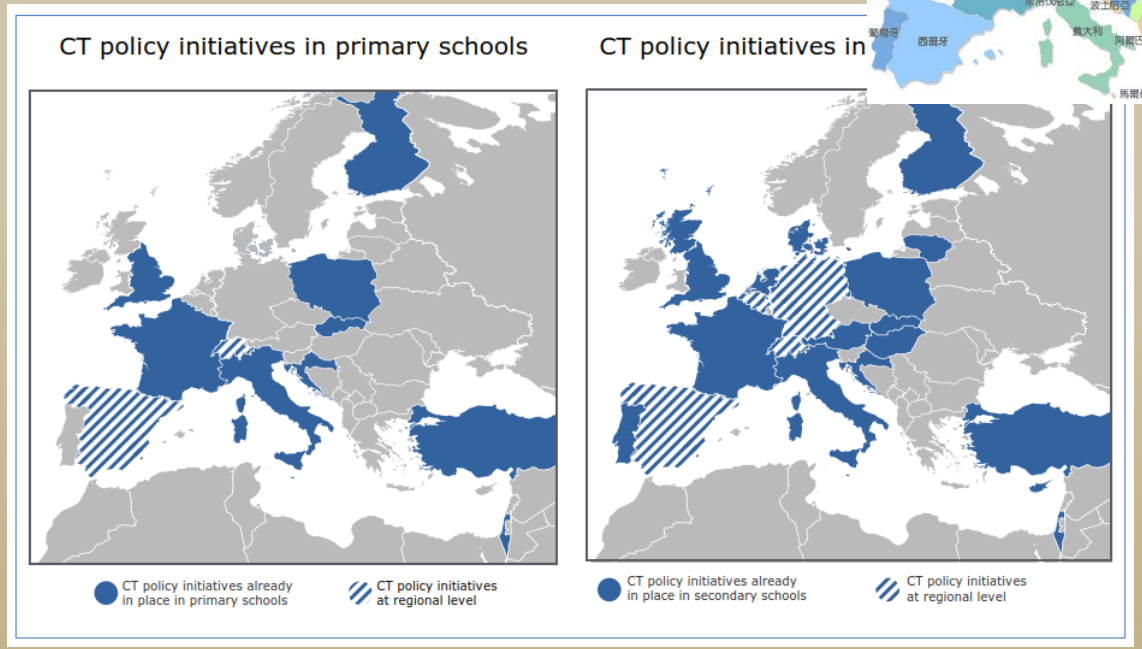
- Key stage 4 (ages 14-16)
  - 提供學生學習足夠深度的資訊科技與電腦科學，以使為進階教育或職業做準備
  - 能發展電腦科學、數位媒體、與資訊科技之能力、創意、與知識
  - 發展並應用解析、問題解決、設計、與運算的思維與技巧
  - 了解科技的改變如何影響安全性(包含：保護線上隱私與身分的方法)，及如何辨識與報導相關問題

將運算思維納入必修

全球資訊教育趨勢

重視運算思維

- 歐洲(以及土耳其、以色列)於小學、中
維教學



資料來源: Developing Computational Thinking in Compulsory Education, JRC（歐盟聯合研究中心）

全球資訊教育趨勢

重視運算思維

- 歐洲國家將運算思維納入必修課的兩大理念
  - 培養幼童與年輕人的運算思維，使其能以不同的方式思考、能透過不同的媒體表達、能解決實體世界的問題、能由不同的角度分析各種議題
  - 培養運算思維以促進國家經濟發展、填補資訊產業相關職缺、預備未來的就業力

資料來源: Developing Computational Thinking in Compulsory Education, JRC（歐盟聯合研究中心）

將程式納入必修

全球資訊教育趨勢

重視程式設計

- 多國將程式設計納入課綱
  - 愛沙尼亞2012年將程式設計納入中小學課綱，是第一個讓程式設計進入小學的國家
  - 全球多數先進國家皆將程式設計納入課綱
  - 芬蘭、法國、立陶宛、波蘭、瑞士、葡萄牙、土耳其等國更在將程式設計能力的培養列為資訊科技課綱的重要目的
  - 美國總統歐巴馬、英國首相卡麥隆、新加坡總理李顯龍等領導人皆親自強調學習程式設計的重要性

全球資訊教育趨勢

重視程式設計

天下雜誌

- **不只有Skype，愛沙尼亞如何成為科技大國？**（2013-08-06）
  - 身為波羅的海三小國的她，全國人口比台北還少，曾經不到一半的人擁有電話線，但卻孕育出Skype等高科技企業，並成為全球第一個讓小學生學寫程式的國家。
  - 當愛沙尼亞1991年剛從蘇聯獨立，只有不到一半的人口擁有電話線，但據《經濟學人》（The Economist）最新報導，20年後，這個過去曾遭瑞典、德國、帝俄殖民，在政治上渺小的愛沙尼亞，已翻身成科技資訊大國。
  - 根據世界銀行（World Bank）統計，2011年愛沙尼亞有14,000家新註冊公司，和2008年同期相比，新設公司大幅成長40%。而現在，這些高科技企業的產值，更吃下愛沙尼亞GDP的15%。
  - 「1980年代的高中男孩都想成為搖滾巨星，」Skype第一號員工、目前也自行創業的塔維特·辛理庫斯（Taavet Hinrikus）對《經濟學人》說，「但現在，這些男孩都想成為創業家。」

國小階段多為必修

全球資訊教育趨勢-各國資訊科技課綱比較

各國課程-多數在小學階段列為必修

Country	What?	How?	Primary	Secondary
Australia	Digital Technologies	Own subject and integrated	Compulsory	Compulsory
England	Computing	Replace existing subject	Compulsory	-
Estonia	Programming (Technology & innovation)	Integrated	Compulsory	Compulsory
Finland	Programming (Digital competence)	Integrated	Compulsory	-
New Zealand	Programming and Computer Science	Own subject	-	Elective
Norway	Programming	Own subject	-	Elective
Sweden	Programming and Digital Competence	Integrated	Compulsory	Elective
South Korea	Informatics	Own subject	Compulsory	Elective
Poland	Computer Science	Own subject	Compulsory	Compulsory
USA	Computer Science	Own subject	-	Elective

Heintz, F., Mannila, L., & Färnqvist, T. (2016, October). A review of models for introducing computational thinking, comput er science and computing in K-12 education. In *2016 IEEE Frontiers in Education conference (FIE)* (pp. 1-9). IEEE

全球資訊教育趨勢-各國資訊科技課綱比較

課程學習年段

台灣	美國 （CSTA, 2017）	英格蘭（DOEE, 2013）	澳洲（ACARA, 2015）
國中七-九年級(13-15歲) 高中一年級(16歲)	Level 1A: Grades K-2 (Age 5-7) Level 1B: Grades 3-5 (Age 8-11) Level 2: Grades 6-8 (Age 11-14) Level 3A: Grades 9-10 (Age 14-16) Level 3B: Grades 11-12 (Age 16-18)	Key Stage 1 (Age 5-7) Key Stage 2 (Age 7-11) Key Stage 3 (Age 11-14) Key Stage 4 (Age 14-16)	Foundation-Year 2 (Age 4-7) Year 3-4 (Age 7-9) Year 5-6 (Age 9-11) Year 7-8 (Age 11-13) Year 9-10 (Age 13-15)

全球資訊教育趨勢-各國資訊科技課綱比較

演算法與程式設計最低學習年段與學習內容

台灣	美國 （CSTA, 2017）	英格蘭（DOEE, 2013）	澳洲（ACARA, 2015）
七年級 (13歲) 演算法基本概念 - 問題解析 - 流程控制 程式語言基本概念、功能及應用 結構化程式設計 - 循序與選擇結構 - 重複結構	Level 1A: Grades K-2 (Age 5-7) • Model daily processes by creating and following algorithms (sets of step-by-step • instructions) to complete tasks. • Model the way programs store and manipulate data by using numbers or other symbols to represent information. • Develop programs with sequences and simple loops, to express ideas or address a problem. Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. • Develop plans that describe a program's sequence of events, goals, and expected outcomes. • Give attribution when using the ideas and creations of others while developing programs. • Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops. • Using correct terminology, describe steps	Key Stage 1 (Age 5-7) • Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions • Create and debug simple programs Use logical reasoning to predict the behaviour of simple programs	Foundation-Year 2 (Age 4-7) • Follow, describe and represent a sequence of steps and decisions (algorithms) needed to solve simple problems Year 3-4 (Age 7-9) • Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them • Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input

演算法教學

Precise and unambiguous instructions?

- 演算法是一套解決問題的程序和方法，不但使用在日常生活，也使用在每天的學習活動

製作蛋糕的食譜

演算法教學

Precise and unambiguous instructions?

演算法教學

Precise and unambiguous instructions?

Must	Write a set of instructions to program the robot to make a jam sandwich
Should	Learn from your own and others mistakes (bugs) and correct them
Could	Write a fully working set of instructions with no bugs

Right Hand	spread	butter	fast
Left Hand	scoop	tub	repeat
Pick up	packet	bread	hard
Press down	knife	slice	soft
cut	blade	plate	forward
Put down	handle	turn	back
hold	jam	top	put
unscrew	jar	bottom	Table
remove	lid	slow	Surface

https://code-it.co.uk/unplugged/jamsandwich

演算法教學

Precise and unambiguous instructions?

Right Hand	put down	bread	handle	tub	bottom	hard
Left Hand	hold	slice	jam	plate	slow	soft
pick up	unscrew	knife	jar	packet	fast	forward
press down	remove	blade	lid	put	repeat	back
cut	spread	scoop	butter	turn	top	

Right Hand	put down	bread	handle	tub	bottom	hard
Left Hand	hold	slice	jam	plate	slow	soft
pick up	unscrew	knife	jar	packet	fast	forward
press down	remove	blade	lid	put	repeat	back
cut	spread	scoop	butter	turn	top	

Right Hand	put down	bread	handle	tub	bottom	hard
Left Hand	hold	slice	jam	plate	slow	soft
pick up	unscrew	knife	jar	packet	fast	forward
press down	remove	blade	lid	put	repeat	back
cut	spread	scoop	butter	turn	top	

Right Hand	put down	bread	handle	tub	bottom	hard
Left Hand	hold	slice	jam	plate	slow	soft
pick up	unscrew	knife	jar	packet	fast	forward
press down	remove	blade	lid	put	repeat	back
cut	spread	scoop	butter	turn	top	

Right Hand	put down	bread	handle	tub	bottom	hard
Left Hand	hold	slice	jam	plate	slow	soft
pick up	unscrew	knife	jar	packet	fast	forward
press down	remove	blade	lid	put	repeat	back
cut	spread	scoop	butter	turn	top	

Right Hand	put down	bread	handle	tub	bottom	hard
Left Hand	hold	slice	jam	plate	slow	soft
pick up	unscrew	knife	jar	packet	fast	forward
press down	remove	blade	lid	put	repeat	back
cut	spread	scoop	butter	turn	top	

Right Hand	put down	bread	handle	tub	bottom	hard
Left Hand	hold	slice	jam	plate	slow	soft
pick up	unscrew	knife	jar	packet	fast	forward
press down	remove	blade	lid	put	repeat	back
cut	spread	scoop	butter	turn	top	

Always start from either right or left hand

Join the words you want to use with a line

https://code-it.co.uk/unplugged/jamsandwich

程式設計教學

Create and debug simple programs Use logical reasoning to predict the behaviour of simple programs?

```
1*1=1   1*2=2   1*3=3   1*4=4   1*5=5   1*6=6   1*7=7   1*8=8   1*9=9
2*1=2   2*2=4   2*3=6   2*4=8   2*5=10  2*6=12  2*7=14  2*8=16  2*9=18
3*1=3   3*2=6   3*3=9   3*4=12  3*5=15  3*6=18  3*7=21  3*8=24  3*9=27

4*1=4   4*2=8   4*3=12  4*4=16  4*5=20  4*6=24  4*7=28  4*8=32  4*9=36
5*1=5   5*2=10  5*3=15  5*4=20  5*5=25  5*6=30  5*7=35  5*8=40  5*9=45
6*1=6   6*2=12  6*3=18  6*4=24  6*5=30  6*6=36  6*7=42  6*8=48  6*9=54

7*1=7   7*2=14  7*3=21  7*4=28  7*5=35  7*6=42  7*7=49  7*8=56  7*9=63
8*1=8   8*2=16  8*3=24  8*4=32  8*5=40  8*6=48  8*7=56  8*8=64  8*9=72
9*1=9   9*2=18  9*3=27  9*4=36  9*5=45  9*6=54  9*7=63  9*8=72  9*9=81
```

```
int main (void){
    int i,j,k;
    for(i=_____)
    {
        for(j=_____)
        {
            for(k=_____)
            {
                cout << _____;
            }
        }
    cout << endl;
    }
}
```

程式設計教學

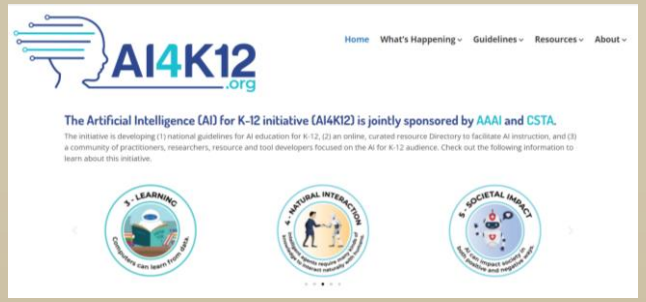
全球資訊教育趨勢

人工智慧的教學

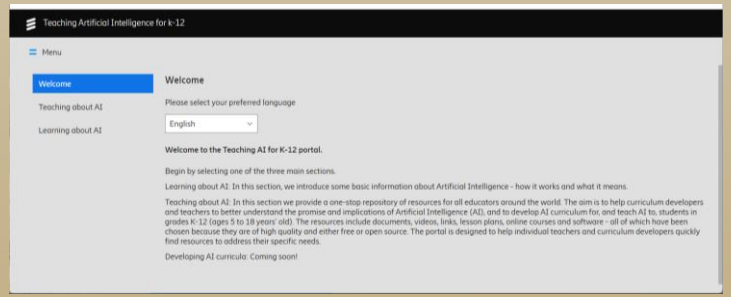
台師大與港大共同發展的AI教材

教育部人工智慧教材 -和AI做朋友

AI4K12
https://ai4k12.org/



Teaching Artificial Intelligence for k12
http://teachingaifork12.org/



全球資訊教育趨勢

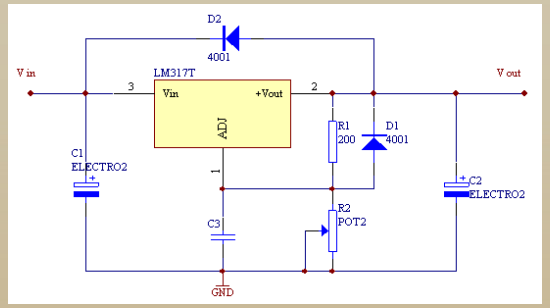
提早學習資訊科學

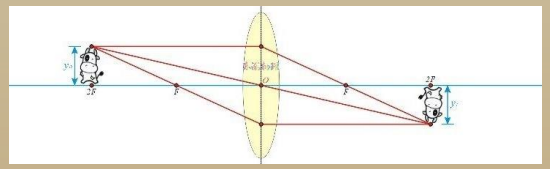
- 幾何學在幾百年前是哲學家或數學家才需要探究的內容，因著時代的進步，現今小學就要開始學幾何
- 不當物理學家的孩子也要學電學、光學，不當生物學家的孩子也要學光合作用
- 隨著科技的進步，資訊科學的學習年段也應向下移動，不當資訊科學家、軟體工程師的孩子也要學資訊科學
  →但需根據學生的認知層次，調整授課內容與教學策略

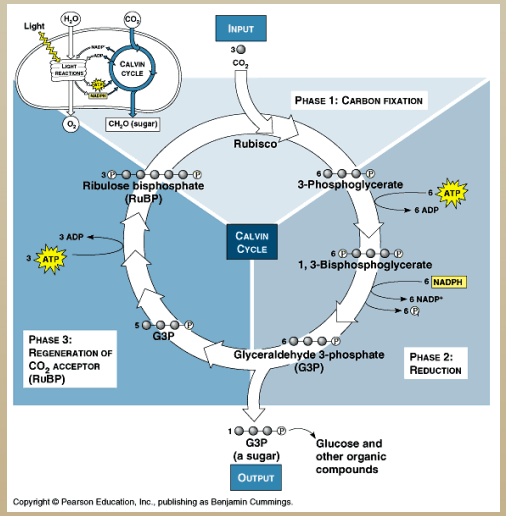
全球資訊教育趨勢

提早學習資訊科學

- 學物理時，只有玩玩電路的組裝嗎?
- 學光學時，只有玩玩光學儀器嗎?
- 學光合作用時，只有種種植物嗎?





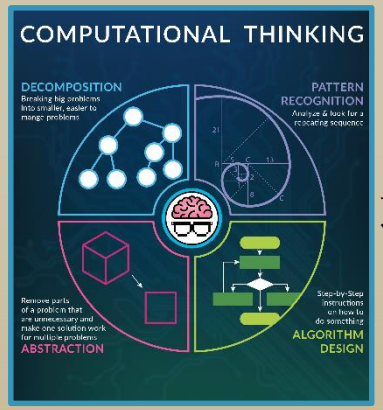


全球資訊教育趨勢

提早學習資訊科學

- 真正的學習，並不是只有體驗，而應真正理解概念
- 若僅有玩玩機器人硬體組裝、操作使用軟體，並非真正的學習

全球資訊教育趨勢



以**運算思維**為主軸

以**程式設計**為工具

以**電腦科學**為基礎

以**高階能力**為目標

這樣的教材，你們學生了解演算法的概念嗎?

資訊科技教育的目的

資訊科技教育的目的

- 學習電腦科學不是只有讓學生使用軟體、建置手機App等，而是可幫助學生發展凌駕於各種學科之外的創造力、創業思維
- 可幫助學生發展系統性思考、問題解決、邏輯演繹、更精確表達自我、批判性思考等高階能力
- 數位時代所需的高階能力與過去有何差異?

資訊科技教育的目的

■ 黑人自拍不再黑麻麻！這支中國手機躋身「非洲之王」

14:432017/06/14 中時新聞網 吳美觀

大陸品牌手機在全球市場迅速崛起，華為在歐洲大有斬獲，聯想和小米則在印度開闢市場，那麼在非洲大陸哪支中國手機賣得最火呢？絕非大家耳熟能詳的大品牌，而是廣東深圳一家名不見經傳的手機品牌「傳音」，默默地吞下非洲40%的手機市場，躋身中國品牌手機的「非洲之王」。

據瞭解，傳音手機席捲非洲最大的利器是，解決當地人自拍的難題。或許你會問，自拍有什麼難解決的，現在的手機不都能自拍嗎？原來是手機拍攝時，會先進行臉部識別，但膚色較深的人種，很難做到準確識別，尤其是在光線不佳的情況下，拍出來的畫面就是一團漆黑。

而傳音為了改良黑人臉部辨識困難問題，還找上台廠聯發科幫忙，開發手機美顏拍照模式，首先大量蒐集當地人的照片，進行臉部輪廓、曝光補償、成像效果的分析調整，再透通眼睛和牙齒來定位，並在此基礎上加強曝光，幫助非洲消費者拍出更加滿意的照片，後來這項在地化功能，果然大受歡迎。

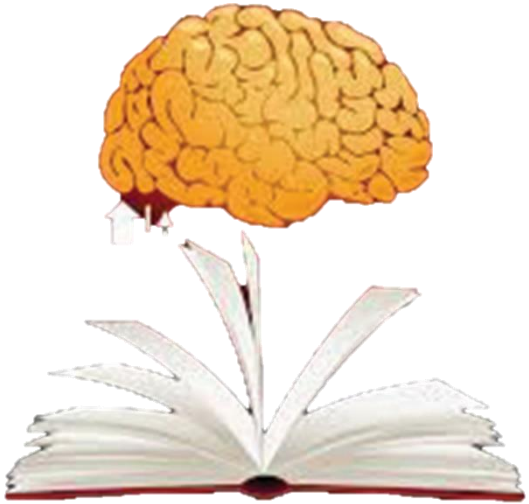
https://www.chinatimes.com/realtimenews/20170614003756-260410?chdtv

資訊科技教育的目的

- 什麼是**批判性思考**?是批評、責難的思考嗎?
  - 客觀分析以分辨有用與無用的資訊，以進一步作合理的判斷或解決問題
  - 須能評估資料、事實、現象、研究結果等

資訊科技教育的目的

- 高階能力皆奠基於紮實的基礎知識



資訊科技教育的目的

- 電腦科學是科學?
- 電腦科學有工程、數學、科學的本質
  - 電腦科學理論有許多數學
  - 電腦科學應用的設計與發展需要工程技術
  - 透過實驗驗證與形成理論與自然科學雷同

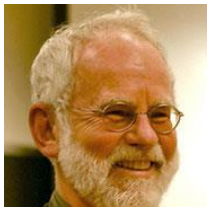
資訊科技涉及紮實的基礎知識，
不是只有動手玩玩的技能!

如何教
運算思維

什麼是運算思維-定義

- “運算思維是利用電腦科學的基本概念進行問題解決、系統設計與人類行為理解的思維模式” (Wing, 2006)

- “運算思維讓我們能擁有電腦科學家面對問題時所持有一種的思維模式” (Grover & Pea, 2013)

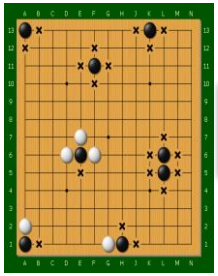


資料來源：
Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.
Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. Educational Researc her, 42(1), 38–43.

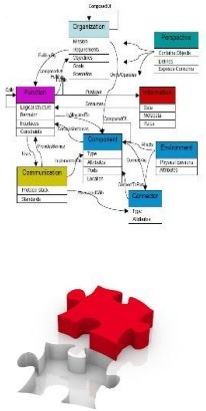
運算思維

- 運算思維在運算歷程中處處可見

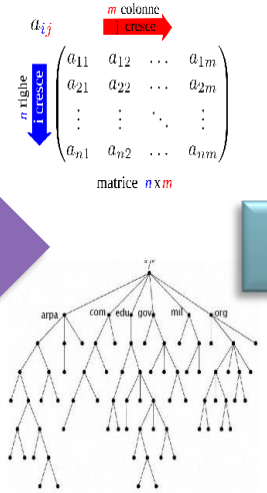
有一定的順序嗎?



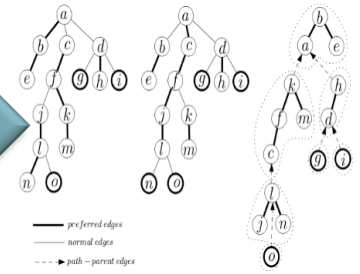
問題解析



資料表示



演算法思維



你是電腦科學家，有固定的解題順序嗎?

討論

- 如何教出運算思維?

如何教運算思維

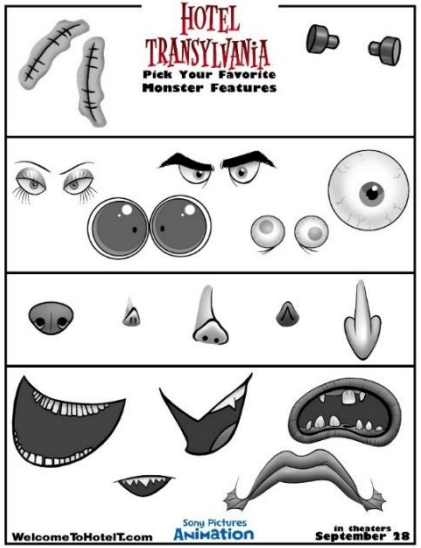
- 運算思維教學宜回歸資訊科技課的本質，重視資訊科學的概念理解與應用

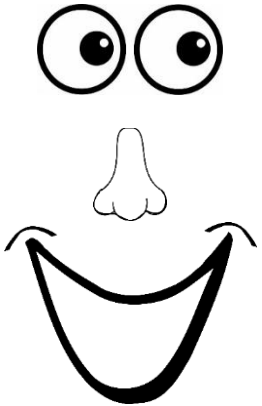
如何教運算思維

- 不宜與資訊科學分離而只去討論運算思維的各元素

如何教運算思維

- 抽象化(Abstraction)
  - 除去細節以簡化並聚焦於重點
  - 辨識與描述普遍化的性質(一般化)





這只是要跟老師們說明什麼是抽象化，但並不是要用這樣的方法教給學生

如何教運算思維

- 藝術的抽象化

國王的哀愁

「畫中他重現了幾個重大的主題-音樂、舞蹈、生活、死亡。畫面中豔麗的色彩和無所不在的花瓣，似乎是對黑色死亡的抗爭。向左倒下的三個人物令人想起舞蹈動作與音樂節奏。在此隱喻中，作者向世人展現對生命快樂的思考，超越裝飾繪畫的貧乏。」(Edina Bemard，現代藝術，閣林國際圖書有限公司)

如何教運算思維

- 藝術的抽象化

Katsushika Hokusai, “South Wind, Clear Sky (Gaif˚ kaisei) or “Red Fuji,” a color woodblock print, Japan,

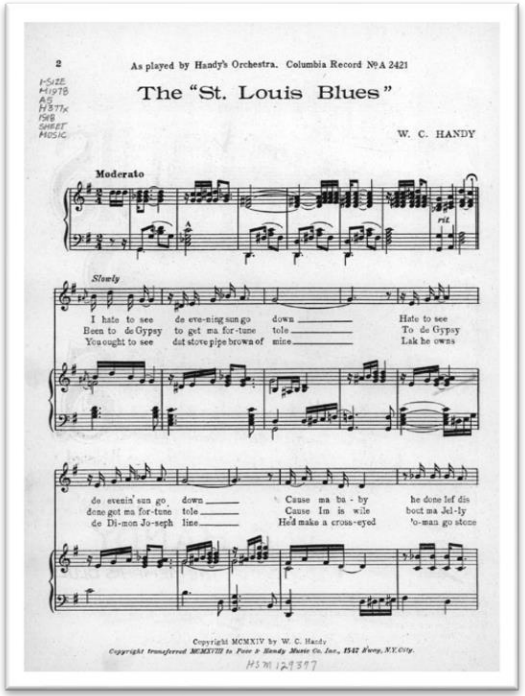
He uses a “perfect balance of colour and composition rendering an abstract form of the mountain to capture its essence.”

如何教運算思維

- 藝術的抽象化

會畫抽象畫是運算思維嗎?
當然不是，那是藝術思維

Icarus的墜落



2

As played by Handy's Orchestra. Columbia Record №A 2421

The "St. Louis Blues"

W. C. HANDY

Moderato

Slowly

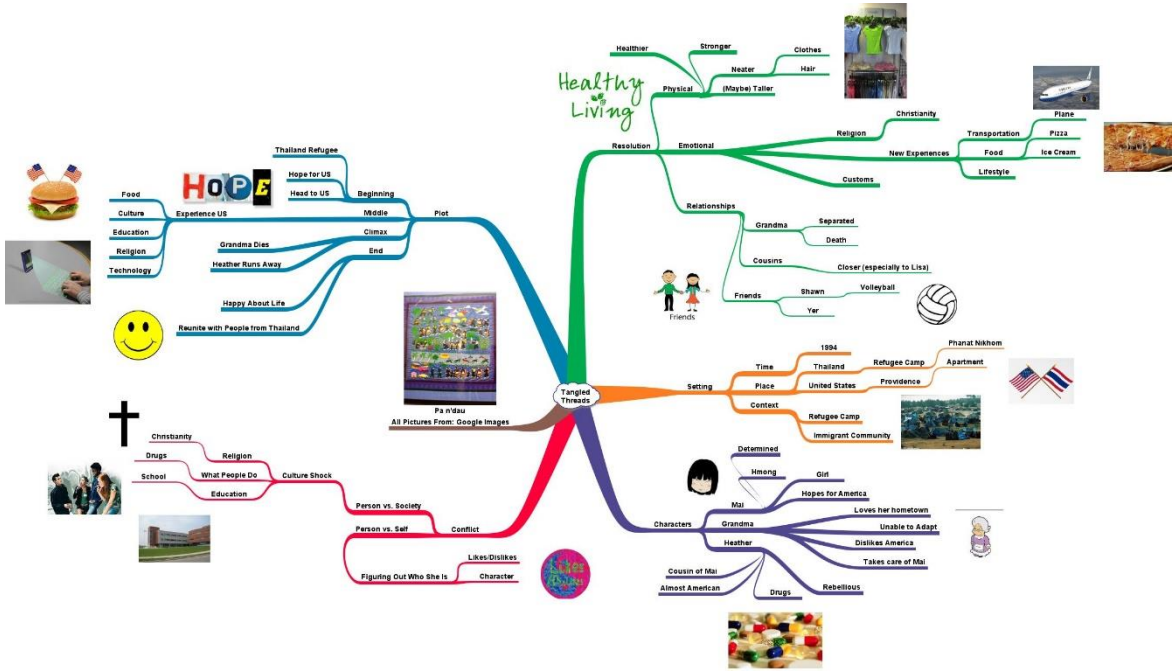
I hate to see de eve-ning sun go down Hate to see
Been to de Gypsy to get ma for-tune tole To de Gypsy
You ought to see dat stove pipe brown of mine Lak he owns

de evenin' sun go down Cause ma ba - by he done lef dis
done got ma for-tune tole Cause Im is wile bout ma Jel-ly
de Di-mon Jo-seph line Hed make a cross-eyed 'o-man go stone

Copyright MCMXIV by W. C. Handy

如何教運算思維

- 語文的抽象化



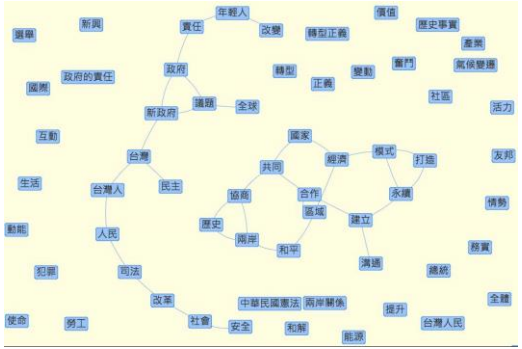
如何教運算思維

- 語文的抽象化

Fault: Clara not only **wants money** but also **fame**.

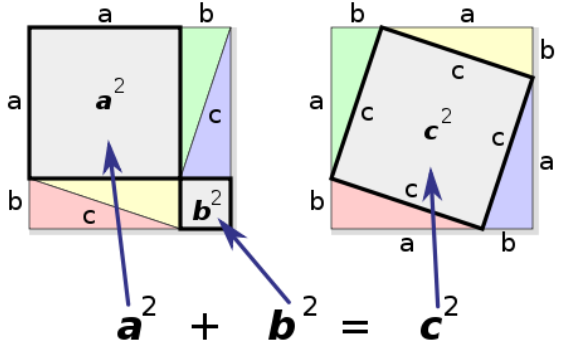
Correct: Clara wants not only **money** but also **fame**.

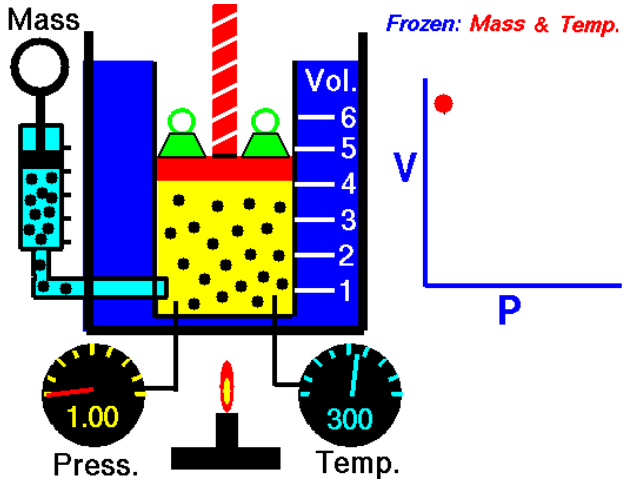
Correct: Clara not only **wants money** but also **wants fame**.



如何教運算思維

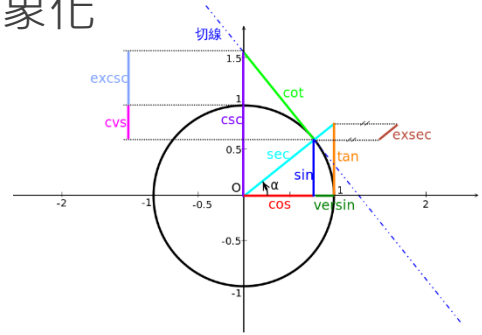
- 數學與科學的抽象化



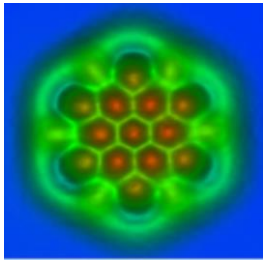


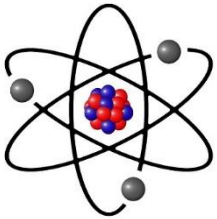
如何教運算思維

- 數學與科學的抽象化



$\sin x = \cos (90^\circ - x)$
$\cos x = \sin (90^\circ - x)$
$\tan x = \sin x / \cos x$

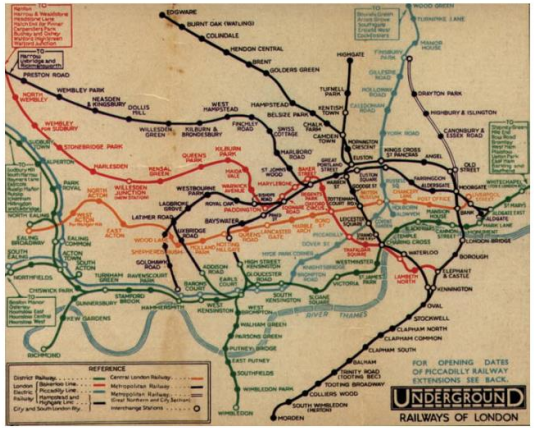




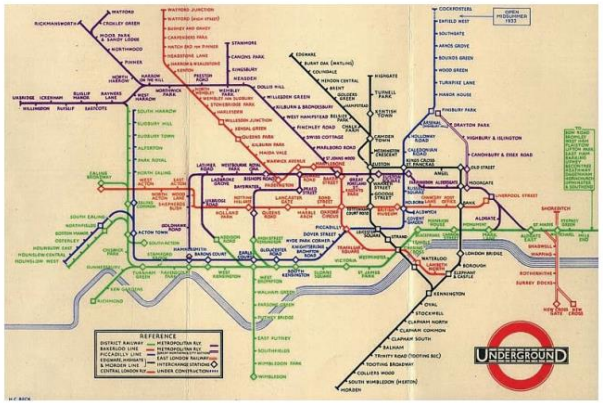
圖片來源：
http://www.businessinsider.com/scientists-can-now-see-individual-atoms-2012-9

如何教運算思維

- 日常生活中的抽象化



(a)



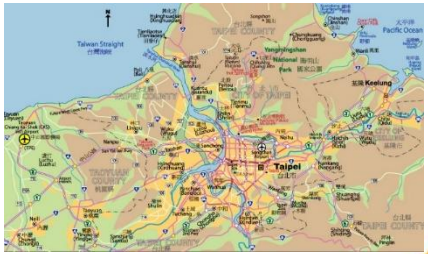
(b)

The London Underground Map (a) the 1928 map and (b) the 1933 map by Harry Beck.

In 1928, the map was essentially an overlay of the underground system onto a conventional geographical map of London. It showed the curves of the train lines and of the River Thames, and the relative distances between the stations. In 1931, Beck produced the first abstract, schematic representation, simplifying the curves to just horizontal, vertical and diagonal lines and where the distances between stations were no longer proportional to the geographical distances.

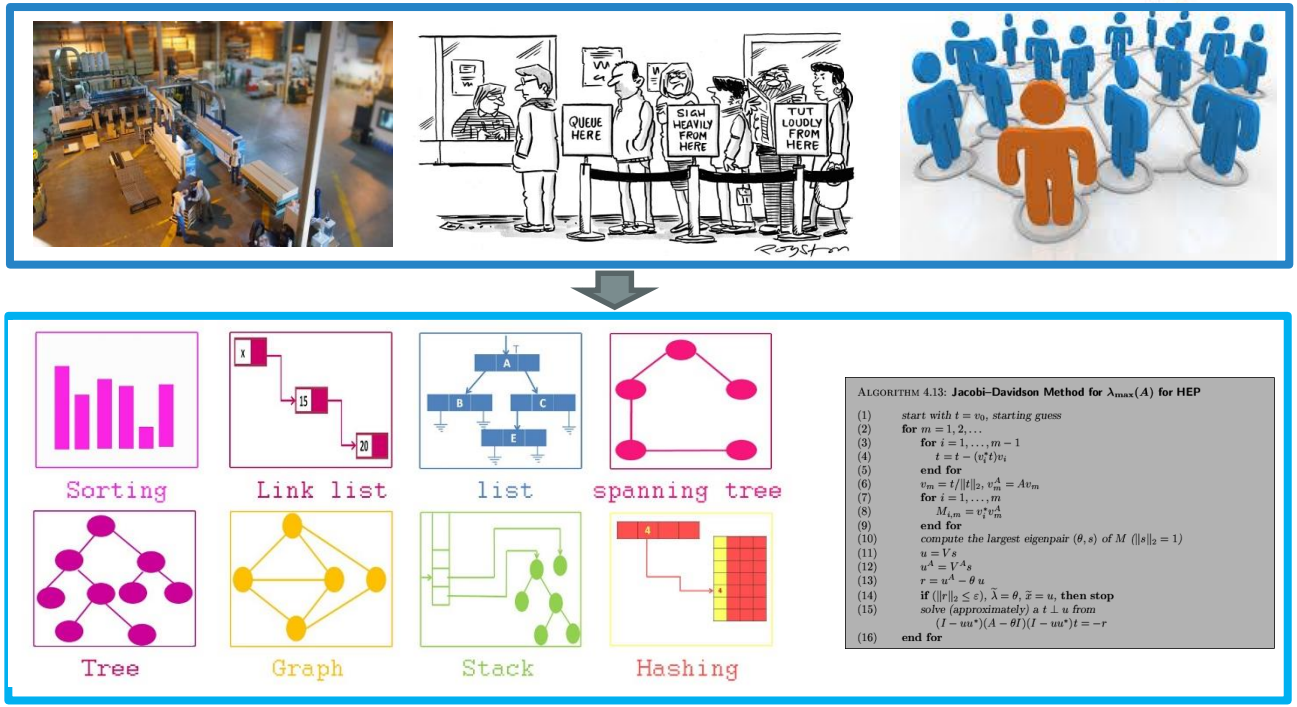
如何教運算思維

- 日常生活中的抽象化



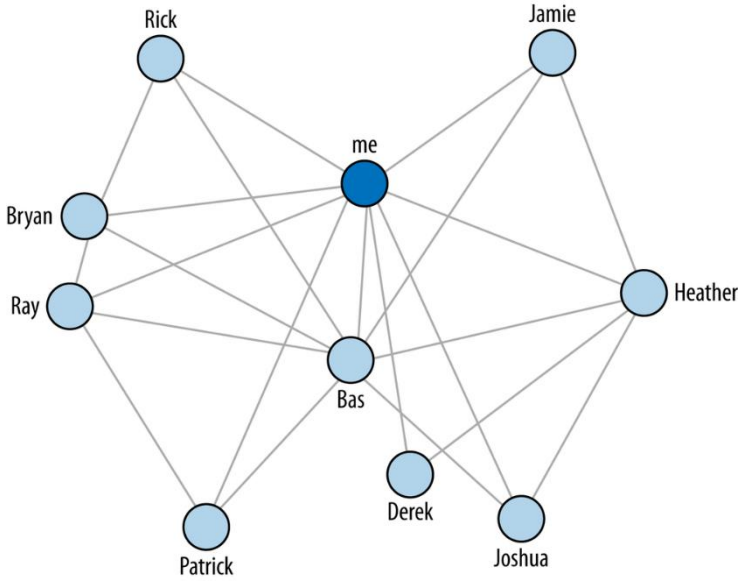
如何教運算思維

- 運算的抽象化



如何教運算思維

- 運算的抽象化

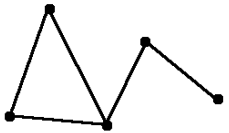


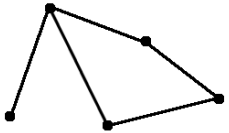
如何教運算思維

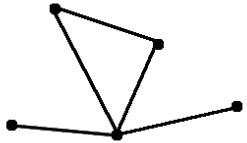
我們知道：
約翰、彼德和湯姆是麥克的朋友。
麥克和安是約翰的朋友。
約翰是安的朋友。
麥克和湯姆是彼德的朋友。
麥克和彼德是湯姆的朋友。

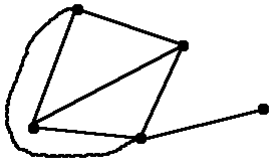
請問: 湯姆跟安是朋友嗎?

有好的表徵方式(抽象化)，
才能有效解題

A 

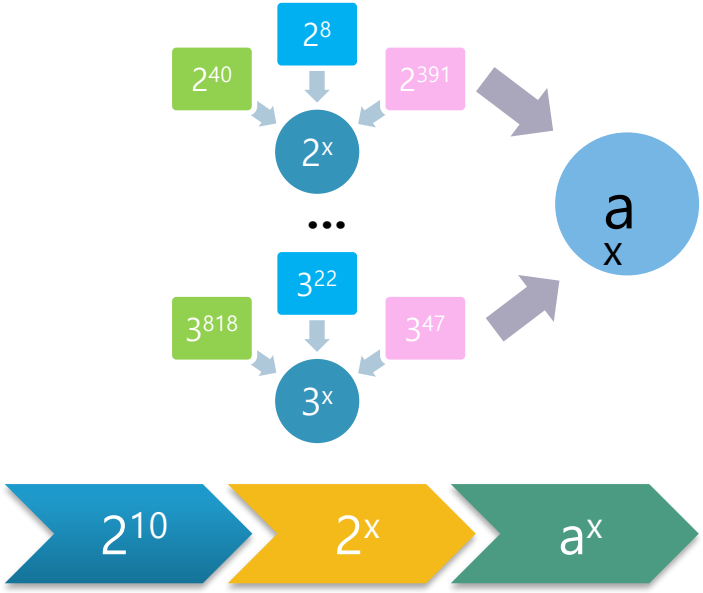
B 

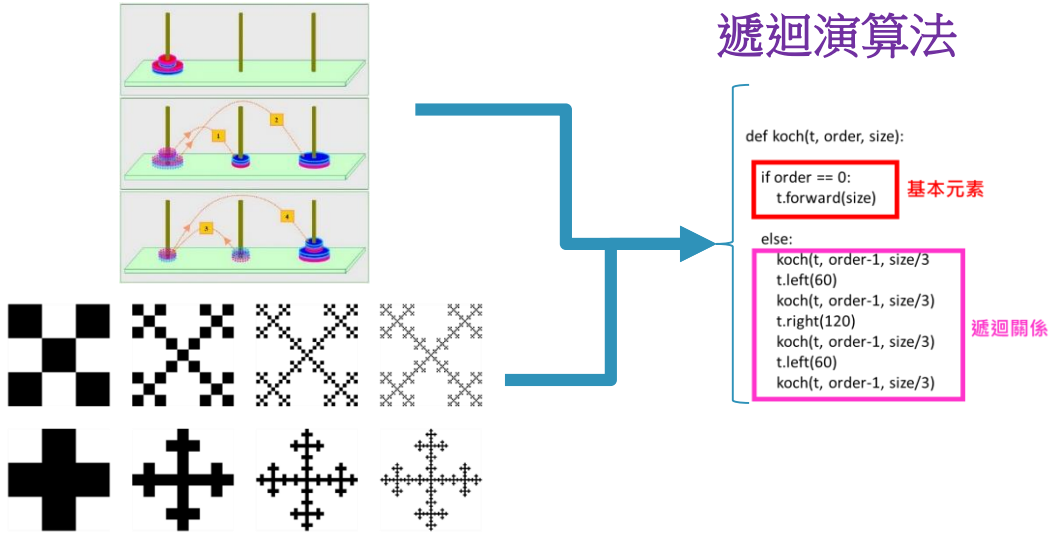
C 

D 

如何教運算思維

- 樣式一般化(Pattern generalization)
  - 產出共通的模式、規則、原則或理論





如何教運算思維

- 我們在撰寫迴圈程式時，亦需要先找出重覆指令的共通特質，例如若有一連串指令：

```
x=x+2;
x=x+4;
x=x+6;
x=x+8;
x=x+10;
```

我們可以找到指令的共同形式是$x=x+2*i$，那麼便可以重整這些指令成為迴圈：

```
for（i=1;i<=5;i++）
   x=x+2*i;
```

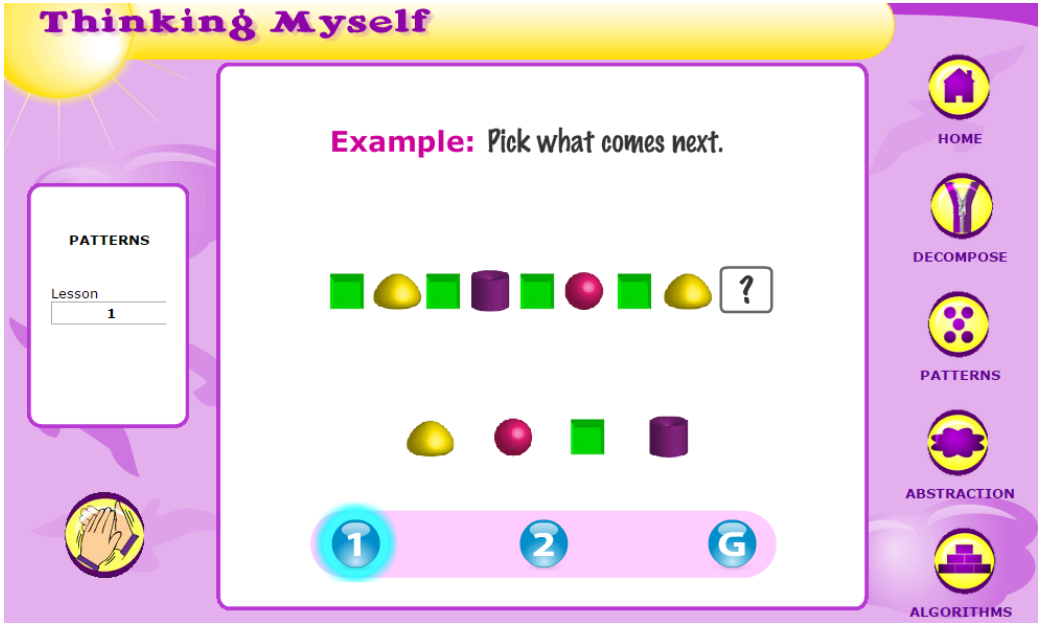
如何教運算思維

- 樣式辨識(Pattern recognition)
  - 在資料中觀察樣式、趨勢或規則
  - http://www.nickjr.com/dora-the-explorer/games/

這只是要跟老師們說明什麼是樣式辨識，但並不是要用這樣的方法教給學生

如何教運算思維

- 樣式辨識(Pattern recognition)
  - http://games.thinkingmyself.com/

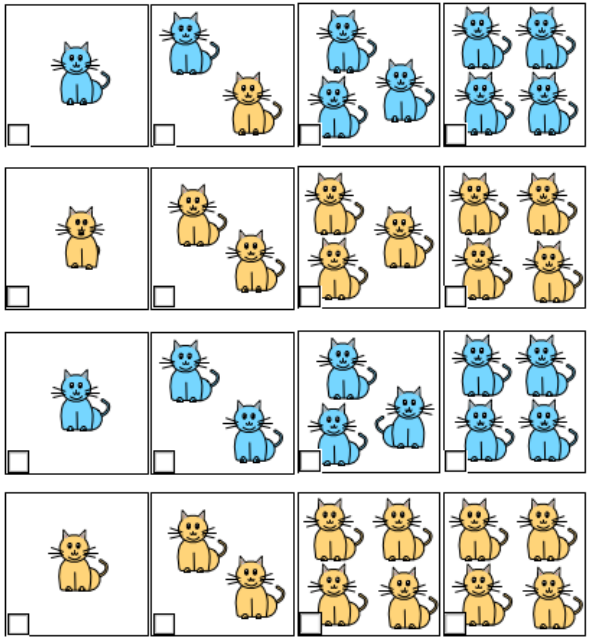


如何教運算思維

- 樣式辨識(Pattern recognition)

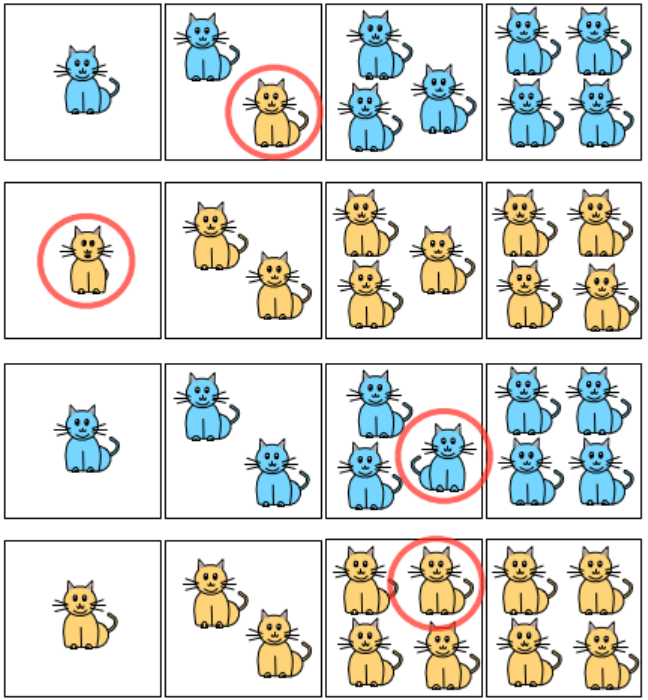
Odd cat out

Put a cross ☒ in the square with the cat or cats that do NOT fit the pattern in ea row of cats



Odd cat out: Solution

The circled cats do NOT fit the pattern in each row of cats.



如何教運算思維

- 樣式辨識(Pattern recognition)

Caterpillar patterns

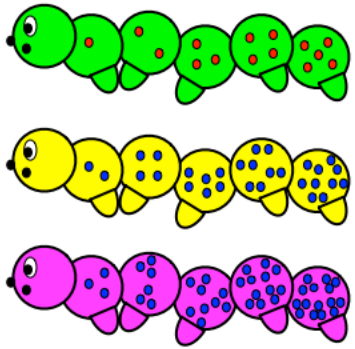
Select the colour of the white part of the rainbow caterpillar with the colour that matches the pattern from the drop down list.

Select the right number of dots to complete the pattern on these spotty caterpillars from the drop down list.

Caterpillar patterns: Solution

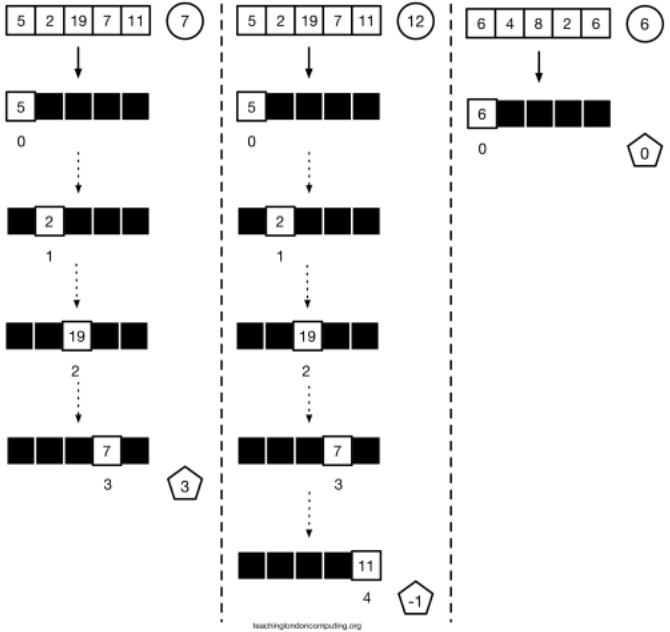
The end segment should be a purple colour to match the rainbow pattern. Violet comes next. Every wondered why we split the rainbow in to 7 colours? You can split it in to any number (we used 6). It was Isaac Newton who chose 7: because the Ancient Greeks thought 7 was a 'magic' number found in nature. Newton was interested in magic and thought the Greeks were right so he gave the rainbow 7 colours, splitting purple in two.

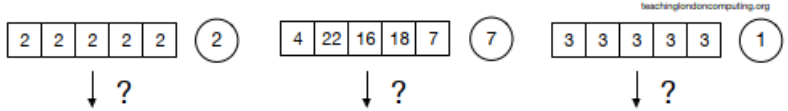
These caterpillar's spots have patterns of counting in 1s, counting in 2's and counting in 3s.



如何教運算思維

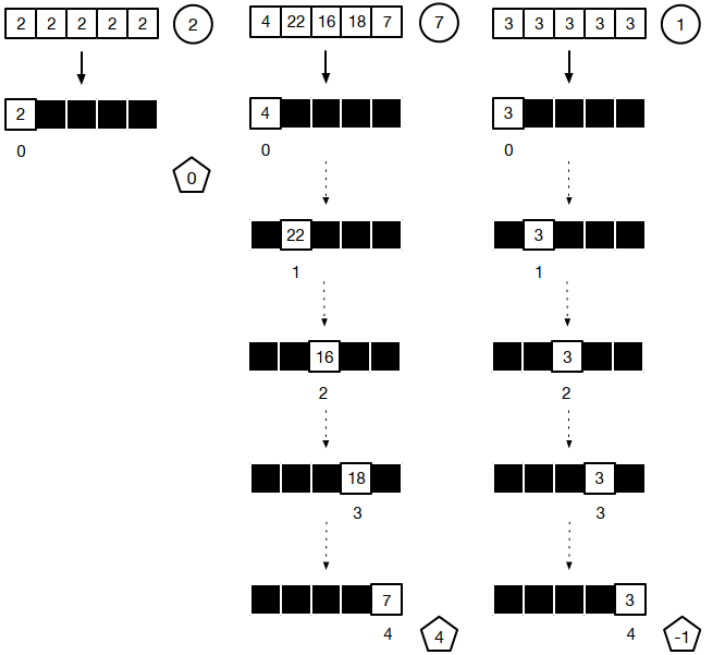
- 樣式辨識(Pattern recognition)





如何教運算思維

- 樣式辨識(Pattern recognition)



如何教運算思維

- 樣式辨識(Pattern recognition)

Pseudocode Pattern Puzzle 1

Work out the pattern and complete the pseudocode by replacing ?, ?? and ??? with different expressions (numbers, variables or calculations).

For example, for the sequence

6 8 10 12…16 ? is 6 ?? is 2 ??? is 16

```
n = ?
i = 0
while (n <= ??)
  begin
    print(n + " ")
    n = n + ???
    i = i + 1
  end
```

Create a dry run / trace table and then write a program fragment in your favourite language to check your answer before looking at the solutions.

1) 2 4 6 8 10 … 20
   ? = _____ ?? = _____ ??? = _____
2) 3 6 9 12 … 33
   ? = _____ ?? = _____ ??? = _____
3) 1 3 5 7 9 … 99
   ? = _____ ?? = _____ ??? = _____
4) 21 28 35 … 63
   ? = _____ ?? = _____ ??? = _____
5) 5 6 7 8 9 … 10
   ? = _____ ?? = _____ ??? = _____
6) 0 0 1 3 6 10 … 21
   ? = _____ ?? = _____ ??? = _____
7) 5 6 8 11 15 … 26
   ? = _____ ?? = _____ ??? = _____
8) 3 3 5 9 15 23 … 33
   ? = _____ ?? = _____ ??? = _____
9) 1 2 4 8 16 … 64
   ? = _____ ?? = _____ ??? = _____
10) 8 16 32 … 128
   ? = _____ ?? = _____ ??? = _____

如何教運算思維

- 樣式辨識(Pattern recognition)

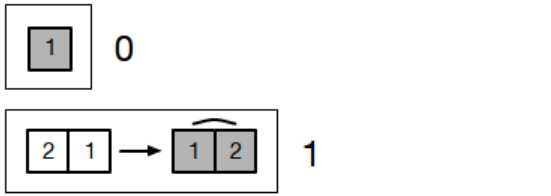
Pseudocode Pattern Puzzle 1: Solutions

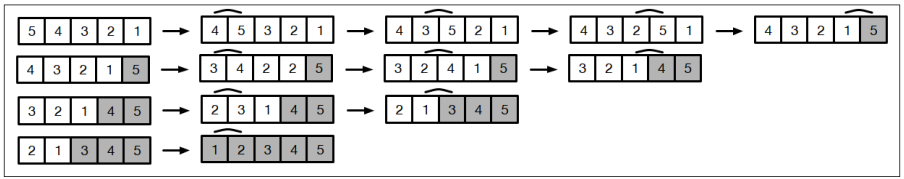
```
n = ?
i = 0
while (n <= ??)
  begin
    print(n + " ")
    n = n + ???
    i = i + 1
  end
```

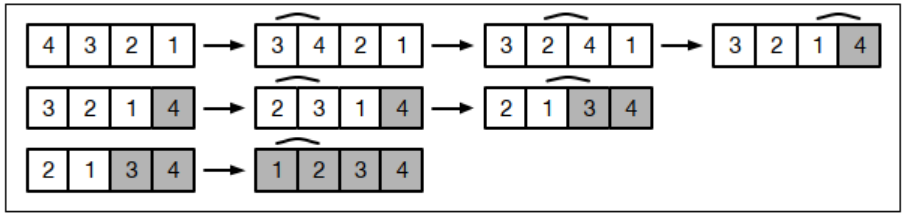
1) 2 4 6 8 10 … 20	? =	2	?? =	2	??? =	20
2) 3 6 9 12 … 33	? =	3	?? =	3	??? =	33
3) 1 3 5 7 9 … 99	? =	1	?? =	2	??? =	99
4) 21 28 35 … 63	? =	21	?? =	7	??? =	63
5) 5 6 7 8 9 … 10	? =	5	?? =	1	??? =	10
6) 0 0 1 3 6 10 … 21	? =	0	?? =	i	??? =	21
7) 5 6 8 11 15 … 26	? =	5	?? =	(i + 1)	??? =	26
8) 3 3 5 9 15 23 … 33	? =	3	?? =	(2 * i)	??? =	33
9) 1 2 4 8 16 … 64	? =	1	?? =	n	??? =	64
10) 8 16 32 … 128	? =	8	?? =	n	??? =	128

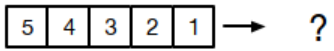
如何教運算思維

- 樣式辨識(Pattern recognition)



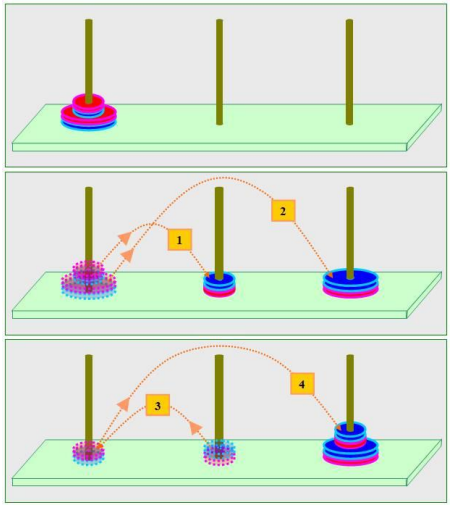




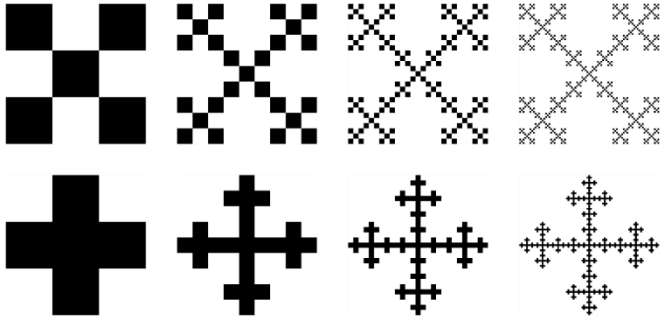


如何教運算思維

- 樣式辨識與一般化讓運算發揮其能力



河內之塔：透過解題規則的找尋，讓運算幫我們處理人腦難以運作的盤子數量

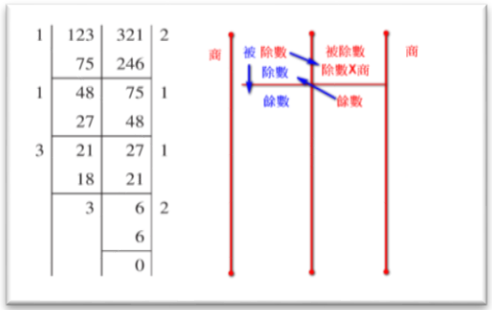


碎形：透過繪製規則的找尋，讓運算幫我們繪製人類難以處理的複雜圖形。

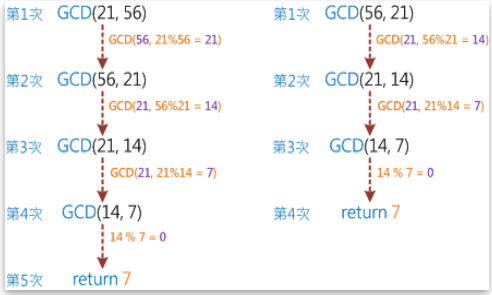
如何教運算思維

- 運算的抽象化

問題: 找出兩個正整數的最大公因數。



數學計算程序



辨識出規則(pattern recognition)

正整數→ A, B
函數GCD

- **Base Case**：
  if (A mod B) == 0 ⇒ return B;
- **General Case**：
  otherwise ⇒ return GCD(B, A mod B)

將規則一般化(pattern generalization)

```
int GCD(int A, int B)
{
    if( (A % B) == 0)
        return B;
    else
        return GCD(B, A%B); }
```

將解題邏輯轉為程式指令

討論

- 你曾經如何透過資訊科技的知識統整解題中需要的通則?
- 如何將這種能力教給學生?

討論

- 若讓學生製作一個程式設計專題，能辨識各種水果的種類，試問樣式辨識的運算思維可能發生在何處?
  - 辨識水果的形狀、顏色等，是此程式的功能，**並非**學生會經歷的運算思維
  - 學生需思考透過什麼樣的規則才能達成辨識水果的功能，因此去整理條件判斷的規則，例如:

    如果水果的外觀完好

    則如果水果的顏色鮮艷

    則判定為好水果

    否則判定為顏色不佳的水果

    否則判定為破損水果

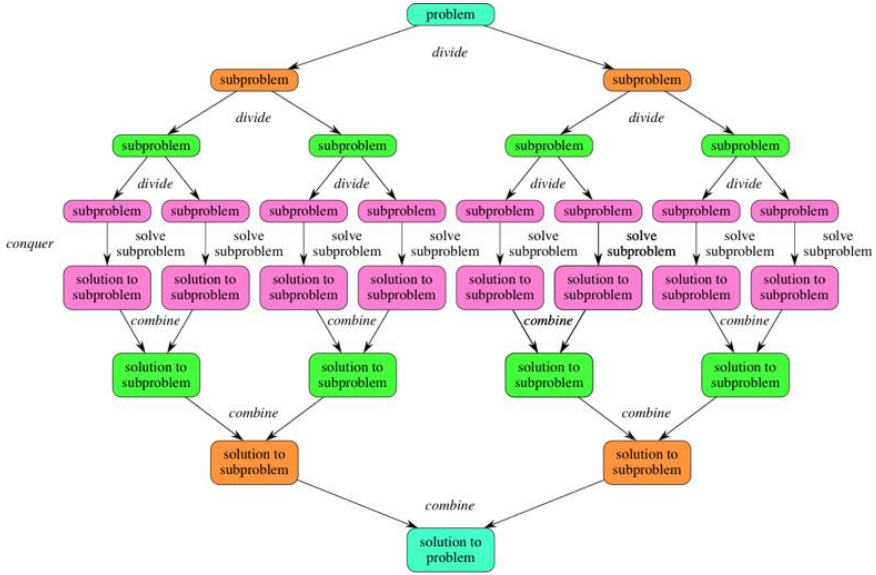
  以作為判定水果種類的通則，這便是一種樣式辨識的歷程。
- 若讓學生製作一個臉部辨識的程式設計專題，樣式辨識又在哪呢?

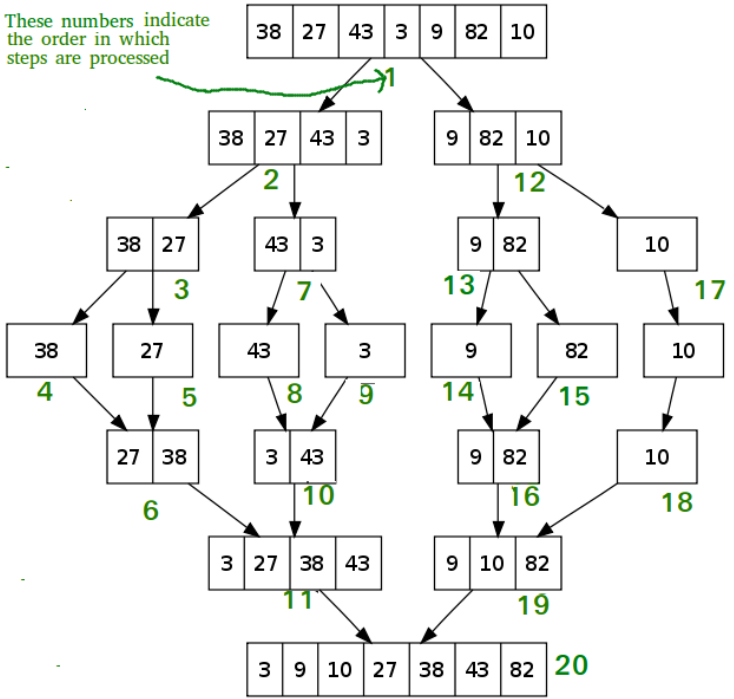
如何教運算思維

- 問題解析(Decomposition)
  - 將資料、程序、問題拆解成較小、較容易處理的部分
  - http://games.thinkingmyself.com/

如何教運算思維

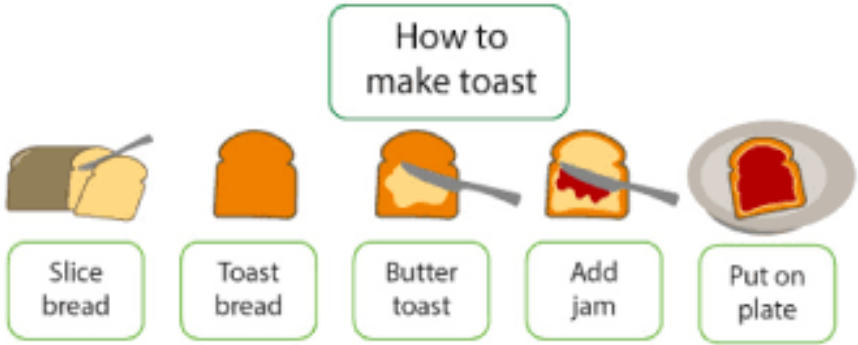
- 問題解析(Decomposition)
  - Divide-and-conquer algorithm

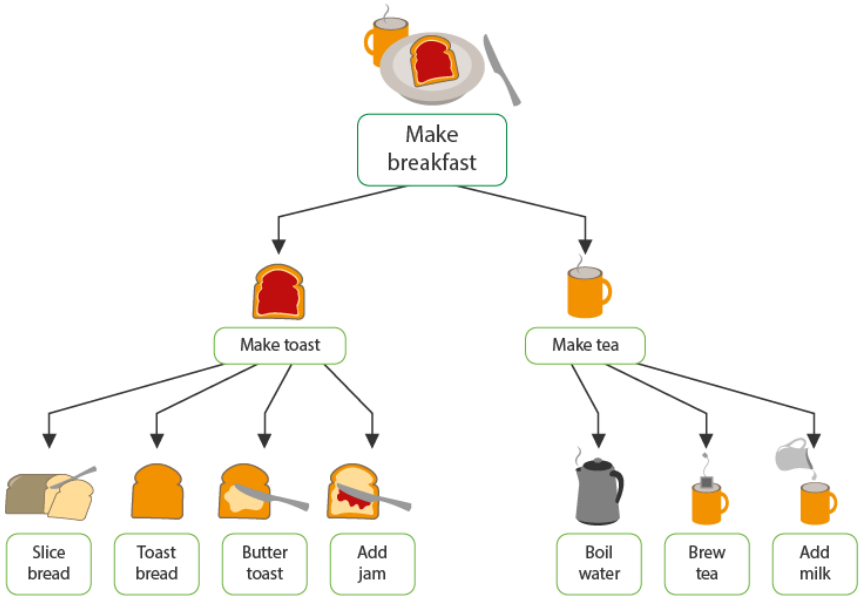




如何教運算思維

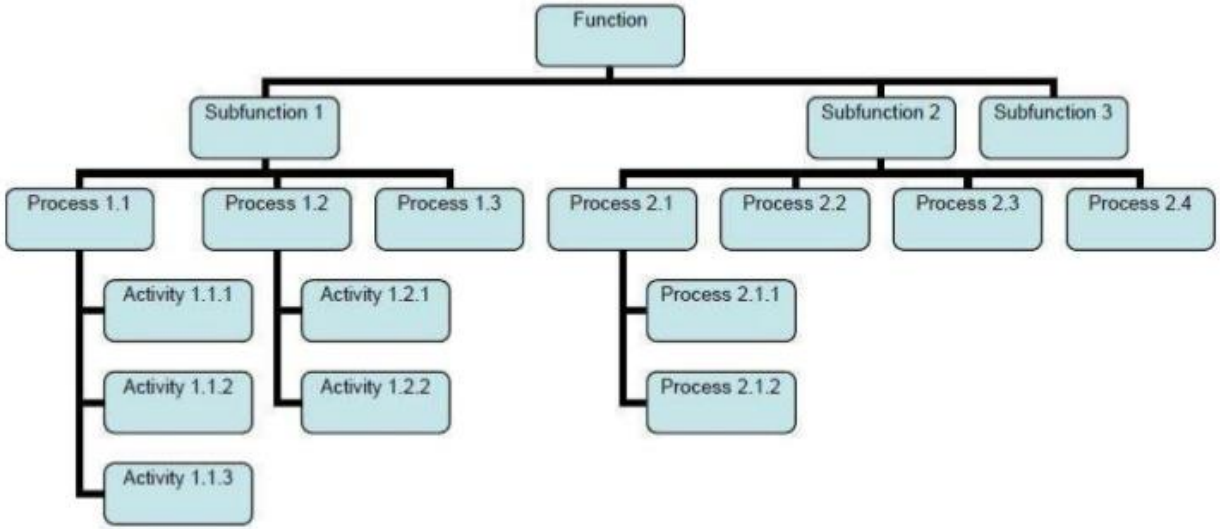
- 問題拆解-非遞迴式的程序拆解
  - 子問題非同類型

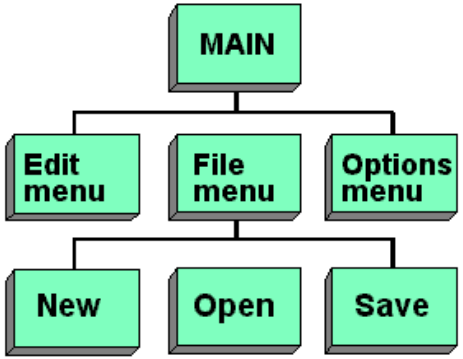




如何教運算思維

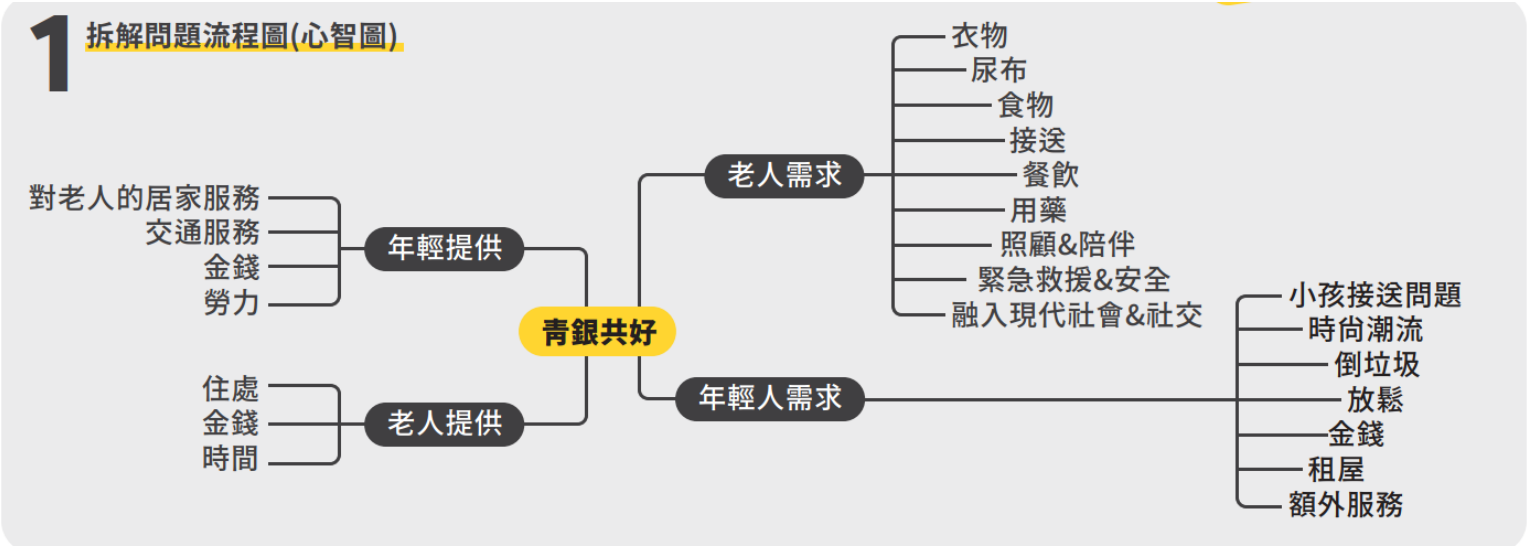
- 問題拆解-功能性的拆解





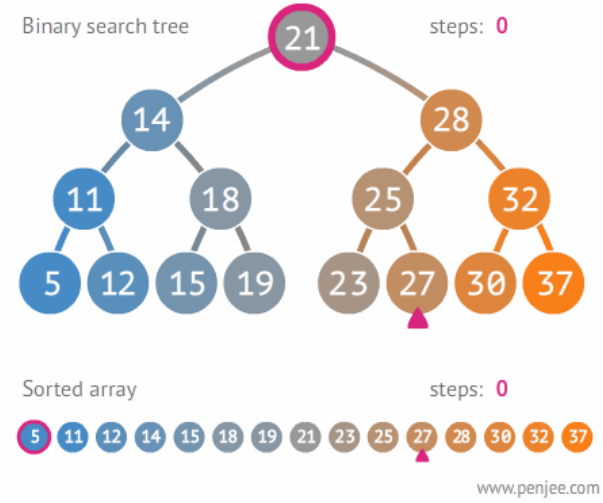
討論

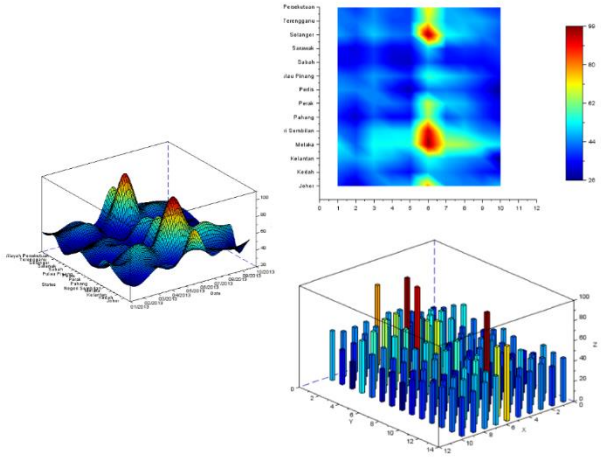
- 問題拆解-以下是好的問題拆解嗎?



如何教運算思維

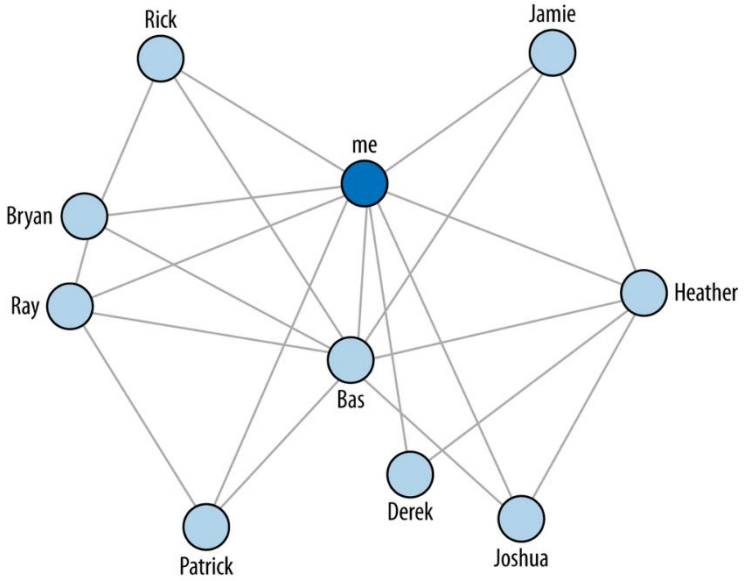
- 資料表示(Data representation)
  - 用適合的圖表、文字或圖片等表達與組織資料





如何教運算思維

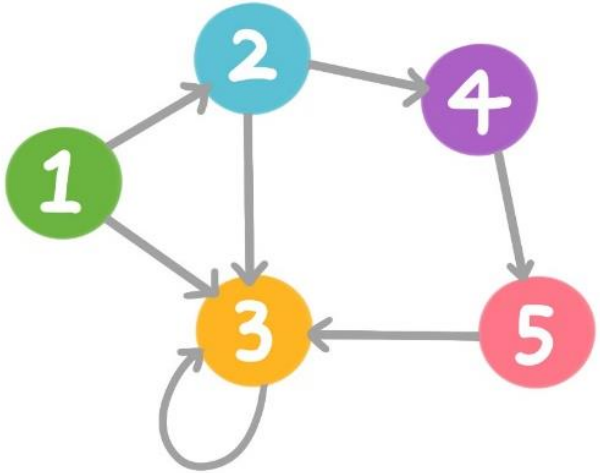
- 資料表示(Data representation)

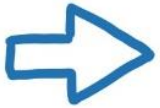


如何教運算思維

- 資料表示(Data representation)

若不表示成矩陣，如何寫程式(用運算)解決問題??

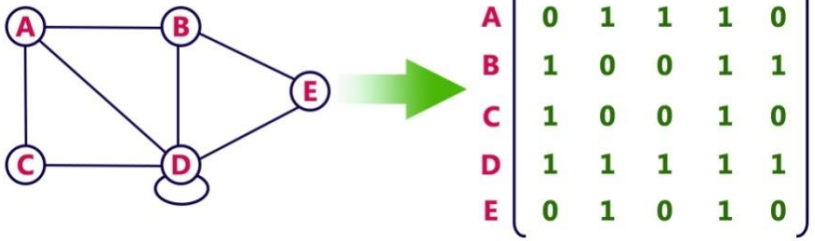




	1	2	3	4	5
1	0	1	1	0	0
2	0	0	1	1	0
3	0	0	1	0	0
4	0	0	1	0	1
5	0	0	1	0	0

如何教運算思維

- 資料表示(Data representation)



	A	B	C	D	E
A	0	1	1	1	0
B	1	0	0	1	1
C	1	0	0	1	1
D	1	1	1	1	1
E	0	1	0	1	0

- **Eigenvector Centrality**

Let x_i denote the score of the ith node. Let $A_{i,j}$ be the adjacency matrix of the network. Hence $A_{i,j} = 1$ if the ith node is adjacent to the jth node, and $A_{i,j} = 0$ otherwise. More generally, the entries in A can be real numbers representing connection strengths, as in a stochastic matrix.
For the i^{th} node, let the centrality score be proportional to the sum of the scores of all nodes which are connected to it. Hence

$$x_i = \frac{1}{\lambda} \sum_{j \in M(i)} x_j = \frac{1}{\lambda} \sum_{j=1}^{N} A_{i,j} x_j$$

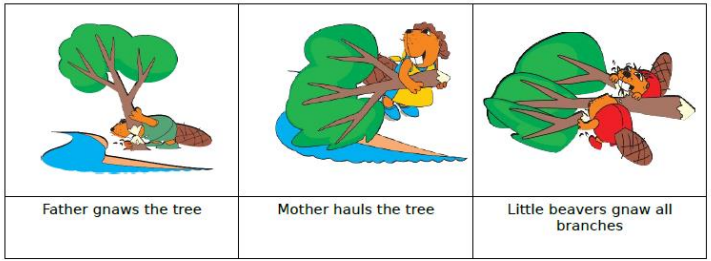
where $M(i)$ is the set of nodes that are connected to the i^{th} node, N is the total number of nodes and λ is a constant.

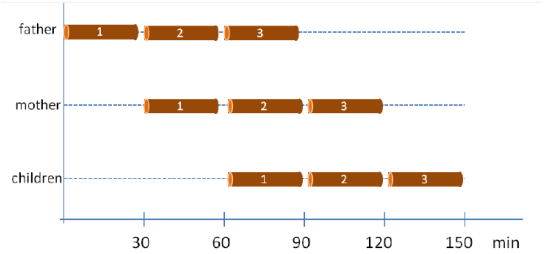
討論

- 資訊科技領域中有哪些有效的資料表示方法?

如何教運算思維

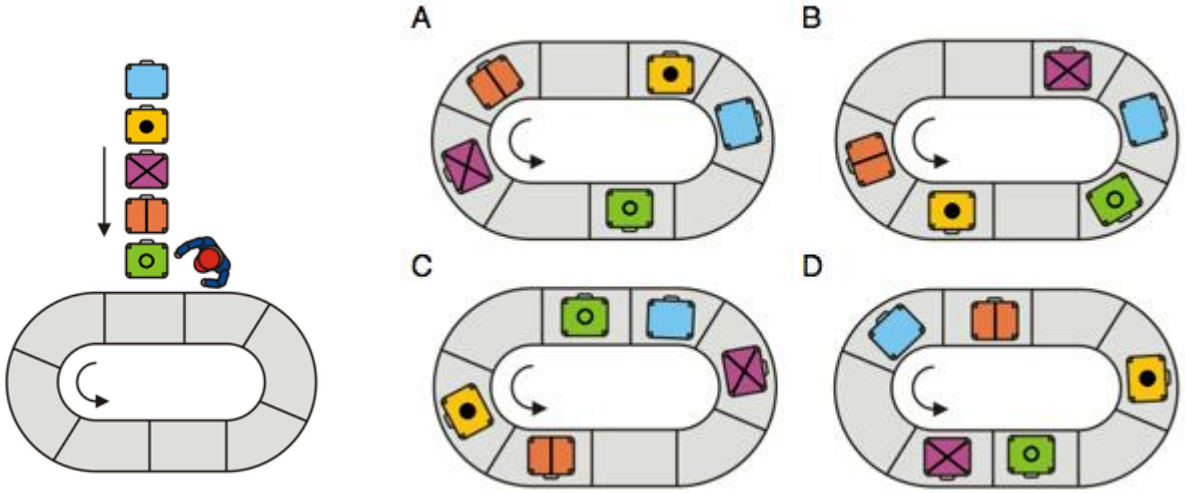
- 演算法思維(Algorithmic thinking)
  - 產出有序指令以解決問題或完成任務





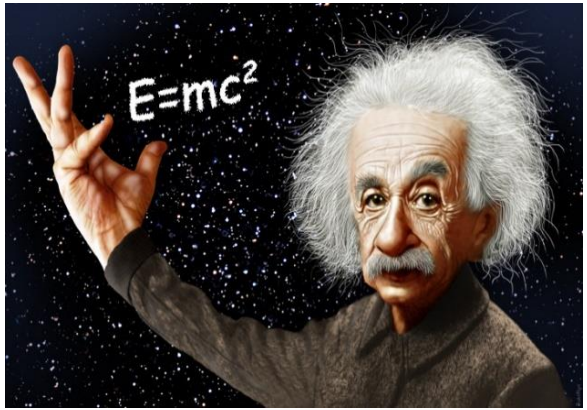
如何教運算思維

- 演算法思維(Algorithmic thinking)
  - 產出有序指令以解決問題或完成任務



如何教運算思維

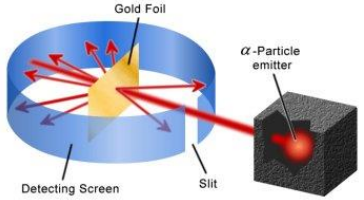
- 建模(Modelling)
  - 根據不同需求(為了容易瞭解、定義、量化、視覺化或模擬等)，將複雜的現象以簡化的方式表達
  - 可用以將抽象的概念視覺化
  - 可作為實驗結果闡釋的依據
  - 可作為預測的基礎

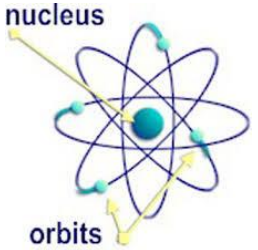


如何教運算思維

- 建模(Modelling)

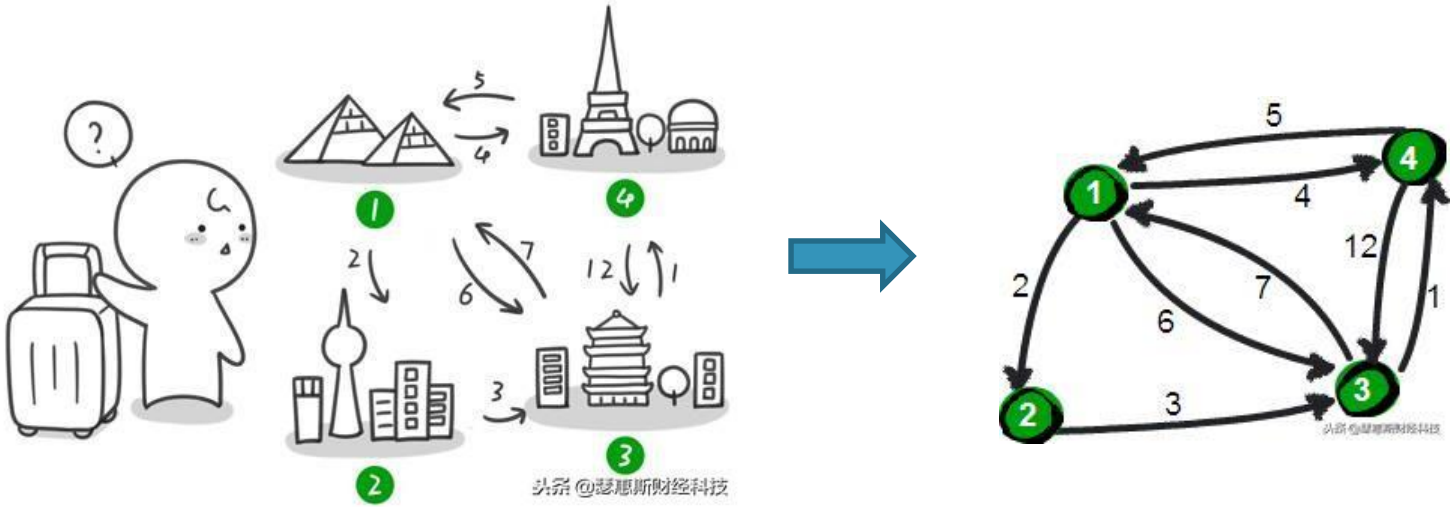
$$F = G\frac{Mm}{r^2}$$





如何教運算思維

- 建模(Modelling)



如何教運算思維

- 建模(Modelling)

如何教運算思維

- 在運算的領域，學生需要操作符號和數字等形式(formalism)，也需要將非形式化的(informal)且複雜的實體世界事物轉為簡化的抽象模型(model)。
- 形式化的建模與分析是實踐抽象思考與強化學生應用抽象的能力的重要方法。
- 建模是很重要的工程技術，可以幫助我們了解與分析大型而複雜的問題。
- 建模是實體事物的簡化，可以幫助理解與推論。學生必須練習透過抽象化的技巧建立模型來達成問題解決的目的。
- 學生必須在實體與抽象間作對應，並能解釋模型分析的意涵。
- 用資訊科技的知識找出有效的形式統整歸納出通則。

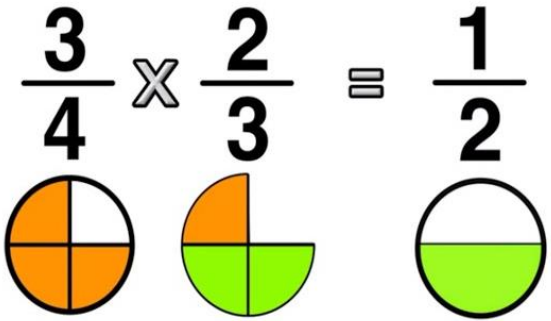
Kramer, J. (2007). Is abstraction the key to computing?. *Communications of the ACM*, *50*(4), 36-42.

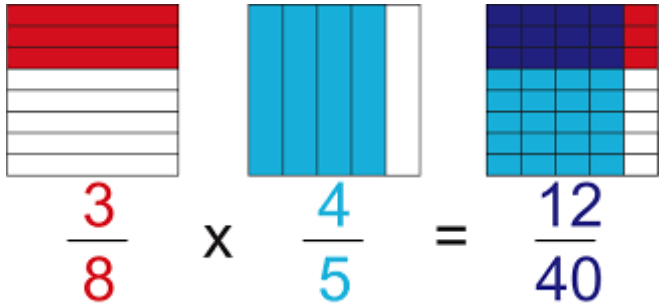
討論

- 用一句簡單的話說明運算思維
- 培養運算思維能做什麼?

教了運算思維之後...

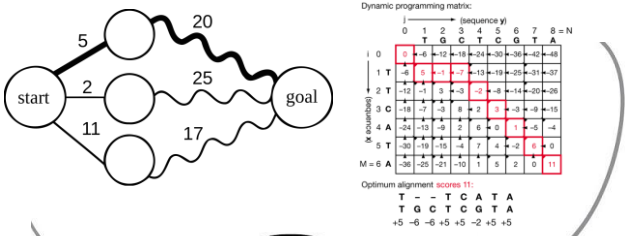
- 思考的表達影響概念的理解與解題的效能
- 若只透過規則的學習或操作，能真正理解概念並應用於解題嗎?





教了運算思維之後...

- 解決問題時，能透過有效的方式分析問題、表徵問題



教學設計

教學設計-選擇學習內容

- 選擇學習內容
  - 課綱中沒有明列人工智慧內容，那要教嗎?

章	節	對應課綱學習內容參考
第1章 人工智慧初探	第1節 關於人工智慧的那些事	資 H-IV-6 資訊科技對人類生活之影響
		資 H-IV-7 常見資訊產業的特性與種類
		資 H-V-3 資訊科技對人與社會的影響與衝擊
	第2節 人工智慧的演進	資 S-IV-1 系統平台重要發展與演進
		資 S-IV-4 網路服務的概念與介紹
		資 S-V-2 系統平台之未來發展趨勢
	第3節 人工智慧的道德議題	資 H-IV-1 個人資料保護
		資 H-IV-3 資訊安全
		資 H-IV-4 媒體與資訊科技相關社會議題
		資 H-IV-5 資訊倫理與法律
		資 H-V-1 資訊科技的合理使用原則
		資 H-V-2 個人資料的保護
		資 H-V-3 資訊科技對人與社會的影響與衝擊
	第4節 當代人工智慧發展	資 H-IV-6 資訊科技對人類生活之影響
		資 S-V-2 系統平台之未來發展趨勢
		資 D-V-1 巨量資料的概念
		資 D-V-2 資料探勘與機器學習的基本概念
		資 H-V-3 資訊科技對人與社會的影響與衝擊

第2章 人工智慧的運作原理	第1節 傳統人工智慧	資 A-IV-1 演算法基本概念
		資 D-IV-3 資料處理概念與方法
		資 A-V-1 重要資料結構的概念與應用
	第2節 機器學習	資 A-IV-1 演算法基本概念
		資 T-IV-1 資料處理應用專題
		資 A-V-1 重要資料結構的概念與應用
		資 A-V-2 重要演算法的概念與應用
		資 D-V-2 資料探勘與機器學習的基本概念
	第3節 類神經網路與深度學習	資 A-IV-1 演算法基本概念
		資 D-IV-1 資料數位化之原理與方法
		資 D-IV-3 資料處理概念與方法
		資 A-V-1 重要資料結構的概念與應用
		資 A-V-2 重要演算法的概念與應用
		資 D-V-2 資料探勘與機器學習的基本概念

教學設計-選擇學習內容

■ 選擇學習內容

七年級	八年級	九年級
演算法 (A)	**演算法 (A)**	**系統平台 (S)**
資 A-IV-1演算法基本概念 問題解決、流程控制	資 A-IV -2陣列資料結構的概念與應用 資 A-IV -3基本演算法介紹（搜尋、排序）	資 S-IV -1系統平台重要發展與演進 資 S-IV -2系統平台組成架構運作原理 資 S-IV -3網路技術的概念與介紹 資 S-IV -4網路服務的概念與介紹
程式設計 (P)	**程式設計 (P)**	**資料表示、處理及分析 (D)**
資 P-IV -1基本概念、功能及應用 資 P-IV -2結構化程式設計 循序.選擇.重複結構	資 P-IV -3陣列程式設計實作 資 P-IV -4模組化程式設計的概念 資 P-IV -5模組化程式設計與問題解決實作	資 D-IV -1資料數位化之原理與方法 資 D-IV -2數位資料的表示方法 資 D-IV -3資料處理概念與方法 資料整理與整合、壓縮、轉換
資訊科技應用 (T)		**資訊科技應用 (T)**
資 T-IV -1資料處理應用專題		資T-IV-2資訊科技應用專題 多媒體應用專題、程式設計應用專題
資訊科技人類社會 (H)	**資訊科技與人類社會 (H)**	**資訊科技與人類社會 (H)**
資H-IV-1個人資料保護 資H-IV-2資訊科技合理使用原則 資H-IV-3資訊安全	資H-IV-4資訊科技重要社會議題 資H-IV-5資訊倫理與法律	資H-IV-6資訊科技對人類生活之影響 資H-IV-7常見資訊產業的特性與種類

教學規劃-訂定教學目標

討論: 以下所列各主題的教學目標是否合宜?

程式設計-條件判斷

1. 能理解if條件判斷的概念、運作模式。
2. 能熟悉if條件判斷式的程式撰寫方法。
3. 能了解else, else if的差別與運用。
4. 能利用if條件判斷解決簡單的問題。
5. 能了解在什麼樣的情況下要使用if條件判斷。

演算法-分而治之

1. 能了解運算思維中，拆解問題、尋找規則的重要性
2. 能體驗運算思維中，演算法設計與抽象化的過程。
3. 能在解題過程使用運算思維，理解分治演算法的產出過程。
4. 能比較各組拆解問題的方式、設計出的解題流程有何差異，評估其優劣。
5. 能理解在特定情境下，使用分治演算法的必要性和可行性。

人工智慧簡介

認知：

1. 學生能了解人工智慧的演進
2. 學生能了解機器學習的概念
3. 學生能了解「分類、分群」的概念

情意：

1. 學生能主動察覺日常人工智慧的演變
2. 學生能主動察覺日常生活「分類、分群」的應用

技能：

1. 學生能運用「分類、分群」解決相關問題
2. 學生能批判性思考人工智慧對社會的影響

教學規劃-設計教學活動

AI人臉秀

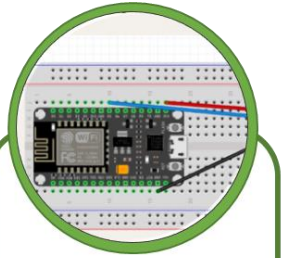
以同學生日為發想，使用Colab實作Deepfake技術，製作慶生影片。

蘋果分類

以分類蘋果為例，藉由Teachable Machine
實作影像辨識技術。

紀念T-shirt

為了製作校慶紀念T-shirt，可透過Python實作機器學習Kmeans，預測衣服版型。



智慧溫控妙管家

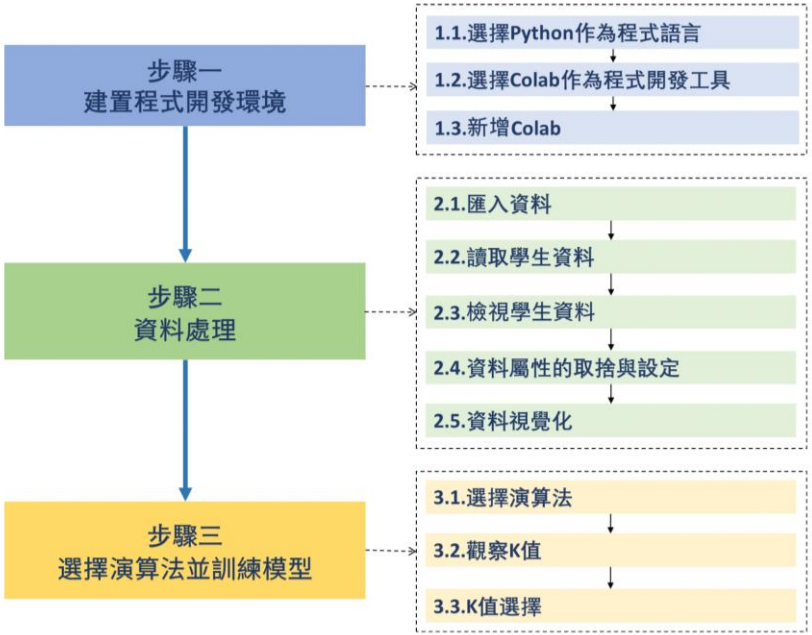
以開發板實作物聯網，將所偵測到的數據上傳至雲端。

當Siri遇上物聯網

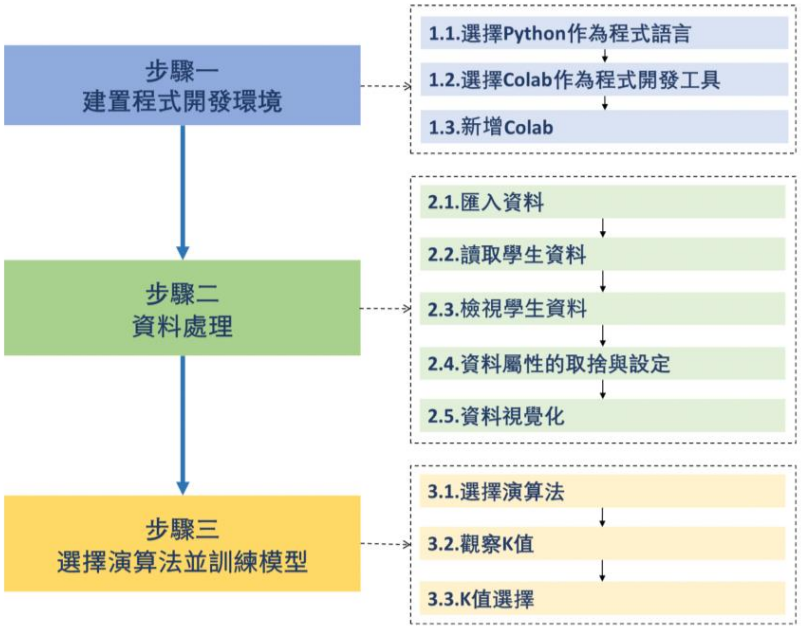
善用語音助理，進一步以更簡易的方式控制物聯網裝置。

紀念T-shirt

實作流程



紀念T-shirt



步驟三、選擇演算法並訓練模型

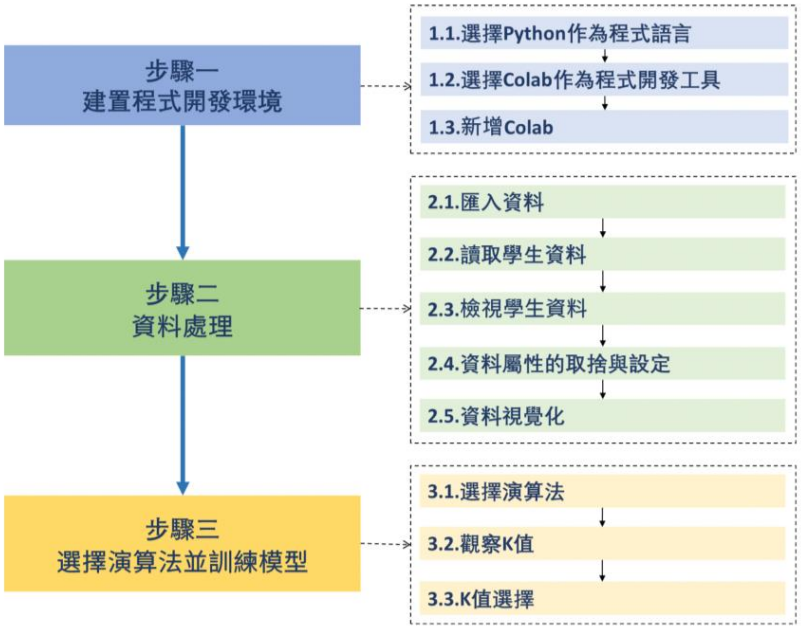
3.1. 選擇演算法

根據問題解析，此類問題適合使用非監督式學習，而此專題選用sklearn套件的K-means 函式庫做為機器學習演算法，程式範例如下。

```
# 使用 KMeans 演算法訓練機器
# 資料匯入                                    紅框部分：使用者需修改才能正確執行程式。
from google.colab import files
uploaded = files.upload()
import pandas as pd
df = pd.read_csv('data.csv', encoding='BIG5')

# Kmeans 演算法參數設定
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
student = df[[' 資料屬性 1 ',' 資料屬性 2 ']]
KM=KMeans( n_clusters = K , init='k-means++')

# 執行 Kmeans 演算法
KM.fit(student)

# 繪製圖形
plt.scatter(student.iloc[:,0],student.iloc[:,1],cmap='autumn',c=KM.predict(student))
centers=KM.cluster_centers_
plt.scatter(centers.T[0],centers.T[1],c='black',s=100,alpha=0.5)
plt.xlabel(' x 軸(資料屬性 1)標籤名稱，必須為英文方能呈現 ')
plt.ylabel(' y 軸(資料屬性 2)標籤名稱，必須為英文方能呈現 ')
plt.title('T-shirt')
plt.show
```

接下來我們針對程式碼一一做說明。

紀念T-shirt

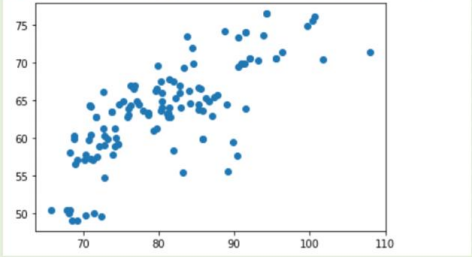


2.5. 資料視覺化

python可以引用Matplotlib函式庫，將資料視覺化，傳遞更易於理解的資訊。為初步了解訓練機器學習的資料屬性分佈情況，故利用scatter( )繪製散佈圖，呈現2個維度的資料關係，程式範例如下：

```
import matplotlib.pyplot as plt
plt.scatter(資料屬性1,資料屬性2)
```

```
<matplotlib.collections.PathCollection at 0x7fd336dc1810>
```



認識scatter(x,y)：

1.依據傳入x軸和y軸的資料，繪製散佈圖，x和y的值個數必須相同。

2.可使用參數整理如下，

c：符號顏色，例如，c='black'，設定所有符號顏色為黑色。

s：符號大小，例如s，='100'，設定符號大小為100點。

alpha：符號透明度，例如，alpha = '0.5'，設定符號透明度為0.5。

cmap：顏色配置，例如，cmap = 'autumn'，設定符號顏色配置為'autumn'。

linestyle：線條樣式，例如，linestyle='--'，設定線條樣式為'--'。

運算思維導向教材設計

- 教學設計步驟

選擇課綱學習內容 → 對應重要運算思維 → 對應主要學習表現

運算思維導向教材設計

選擇課綱學習內容	對應重要運算思維	對應主要學習表現
陣列資料結構的概念與應用	資料表示、抽象化	運算思維與問題解決
結構化程式設計	演算法思維、樣式辨識	運算思維與問題解決

運算思維導向教材設計

規劃教學內涵

提醒自己教學策略與教材設計應注意的原則

學習內容	運算思維	學習表現
資D-IV-1資料數位化之原理與方法 資D-IV-2數位資料的表示方法 資D-IV-3資料處理概念與方法	資料表示與處理 抽象化 演算法思維	資t-IV-4能應用運算思維解析問題。 資p-IV-1能選用適當的資訊科技組織思維，並進行有效的表達。

運算思維導向教材設計

選擇課綱學習內容 → 對應重要運算思維 → 對應主要學習表現

學習內容	運算思維	見
資D-IV-1資料數位化之原理與方法 資D-IV-2數位資料的表示方法 資D-IV-3資料處理概念與方法	資料表示與處理 抽象化 演算法思維	算思維解析 當的資訊科 ，並進行有

讓學生體會資料如何用符號表示，及有效符號表示對問題解決的影響

讓學生體驗資料表示的方法與策略

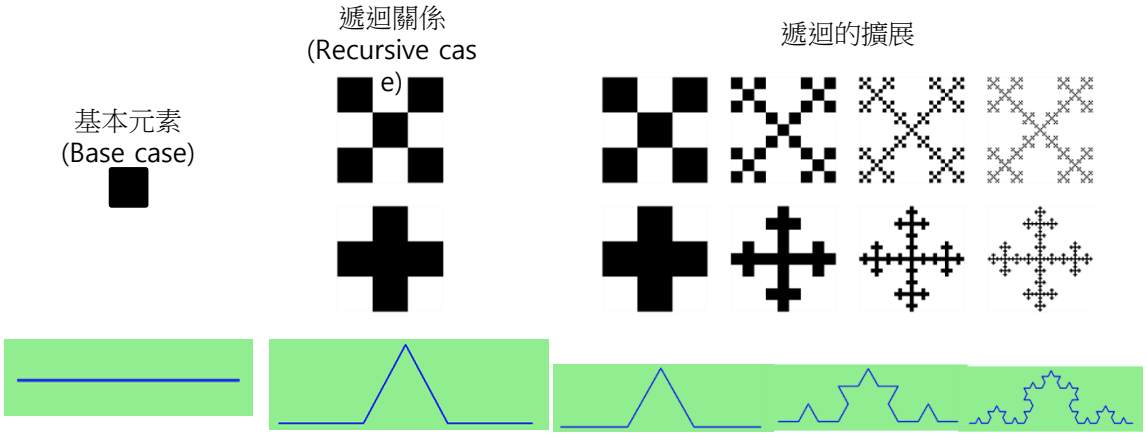
運算思維導向教材設計

選擇課綱學習內容 → 對應重要運算思維 → 對應主要學習表現

- 仔細確認所對應的重要運算思維內涵為何
  - 例如：動態程式規劃的學習需要對應什麼運算思維?若辨識動態程式規劃的演算法樣式是一個重點，則可透過不同的問題(例如旅行銷售員問題與背包問題)所用的動態程式規劃演算法來觀察歸納此一通則

運算思維導向教材設計

- 範例一 **樣式辨識(Pattern recognition)、問題解析(Decomposition)**
  - 透過視覺化圖形的辨識，了解**遞迴**的樣式(pattern)並透過解析找到基本元素與遞迴關係



運算思維導向教材設計

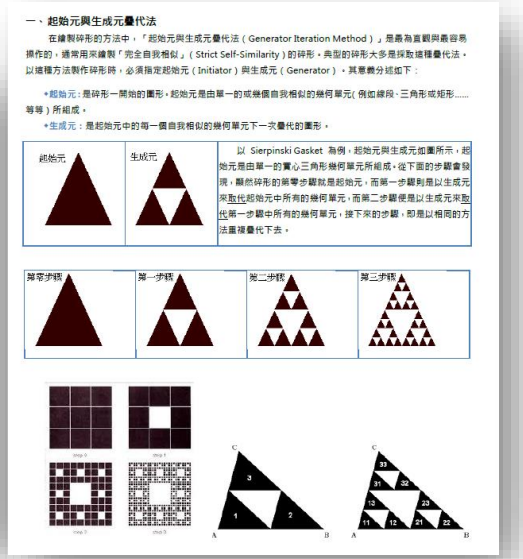
- 範例一**樣式辨識(Pattern recognition)**、**問題解析(Decomposition)**
  - 透過觀察與繪製，了解並體驗**遞迴**的樣式(pattern)

運算思維導向教材設計

- 範例一**樣式辨識(Pattern recognition)、問題解析(Decomposition)**
  - 循序引導與體驗運算思維


一、分組活動：尋找大自然界存在的神秘圖形規律

- 請討論以上圖形的特徵與規律(至少寫 3 個)。
- 請整理寫下各組分享的圖形特徵與規律


一、起始元與生成元疊代法

在繪製碎形的方法中，「起始元與生成元疊代法（Generator Iteration Method）」是最為直觀與最容易操作的，通常用來繪製「完全自我相似」（Strict Self-Similarity）的碎形，典型的碎形大多是採取這種疊代法，以這種方法製作碎形時，必須指定起始元（Initiator）與生成元（Generator），其意義分述如下：

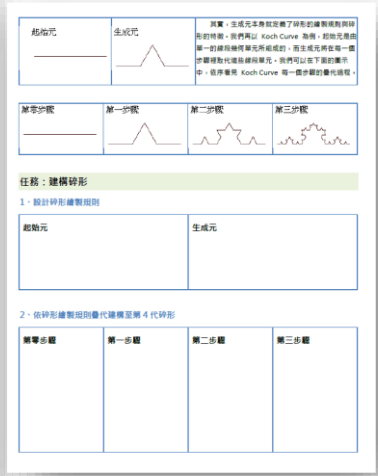
*起始元：是碎形一開始的圖形，起始元是由單一的或幾個自我相似的幾何單元（例如線段、三角形或矩形……等等）所組成。

*生成元：是起始元中的每一個自我相似的幾何單元下一次疊代的圖形。

以 Sierpinski Gasket 為例，起始元與生成元如圖所示，起始元是由單一的實心三角形幾何單元所組成，從下面的步驟會發現，顯然碎形的第零步驟就是起始元，而第一步驟則是以生成元來取代起始元中所有的幾何單元，而第二步驟便是以生成元來取代第一步驟中所有的幾何單元，接下來的步驟，都是以相同的方法重複疊代下去。

起始元　生成元

第零步驟　第一步驟　第二步驟　第三步驟


起始元　生成元

其實，生成元本身就定義了碎形的繪製規則與碎形的特徵，我們再以 Koch Curve 為例，起始元是由單一的線段幾何單元所組成的，而生成元則在每一個步驟裡取代這些線段單元，我們可以在下面的圖示中，依序看見 Koch Curve 每一個步驟的疊代過程。

第零步驟　第一步驟　第二步驟　第三步驟

任務：建構碎形

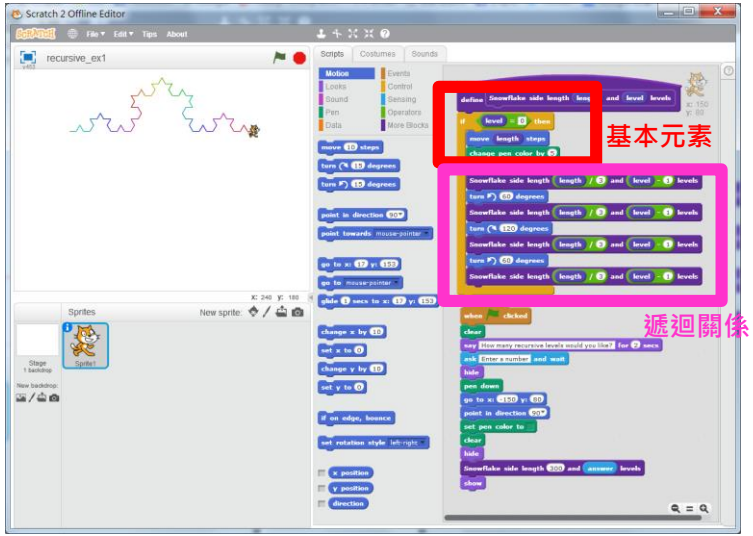
1．設計碎形繪製規則

起始元　生成元

2．依碎形繪製規則疊代建構至第 4 代碎形

第零步驟　第一步驟　第二步驟　第三步驟

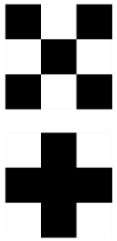
運算思維導向教材設計




```
def koch(t, order, size):
    if order == 0:
        t.forward(size)
    else:
        koch(t, order-1, size/3)
        t.left(60)
        koch(t, order-1, size/3)
        t.right(120)
        koch(t, order-1, size/3)
        t.left(60)
        koch(t, order-1, size/3)
```

基本元素

遞迴關係



運算思維導向教材設計

範例二 抽象化(Abstraction)、資料表示(Data representation)

- Data compression (Exploring Computational Thinking, Google for Education)

Activity 1: Smallest amount of data necessary (20 minutes)

Activity Overview: In this activity, students will use pattern recognition and data analysis to determine what is the smallest amount of data necessary to communicate to another person information such as a word, phrase, shape, or visual scene.

Activity:

1. Pair students into groups of two.
2. Have one student think of a word or phrase and write down one of the letters from that word, wait a second, then write another letter from that word, omitting some of the letters. All of the letters should be placed in the order of where they belong e.g. If I were thinking of "HELLO WORLD," I might write down "L," then later write down "LL," and eventually it might look like "H LLO W R D."
3. Have the second student try to guess the word or phrase as quickly as possible.
4. After the second student correctly guesses the word, students should switch roles.
5. After a couple of turns, ask students to try and do the same activity using a shape or a visual scene like "student walking to school," where each student draws the shape or visual one line or curve at a time.

Q1: On average how much information (turns/steps) from the first person was necessary to guess

Word	Phrase	Shape	Visual

https://docs.google.com/document/d/1PSD5sAiRoC96pPdbPQ7AzWBzH5lAWuH2aB-4QLpPMs8/edit#

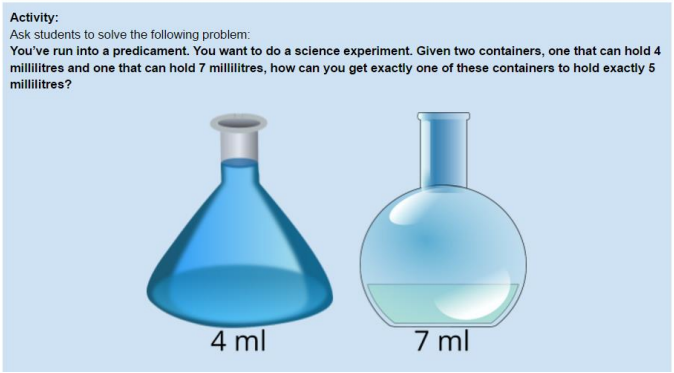
運算思維導向教材設計

- 範例二 **抽象化(Abstraction)、資料表示(Data representation)**
  - 教出壓縮策略與資料表示的精神
    - 用最少位元數精確表達資訊
    - 能夠解碼(解壓縮)

運算思維導向教材設計

範例三 **問題解析(Decomposition)、演算法思維(Algorithmic thinking)**

- Water Water Everywhere! (Exploring Computational Thinking, Google for Education)

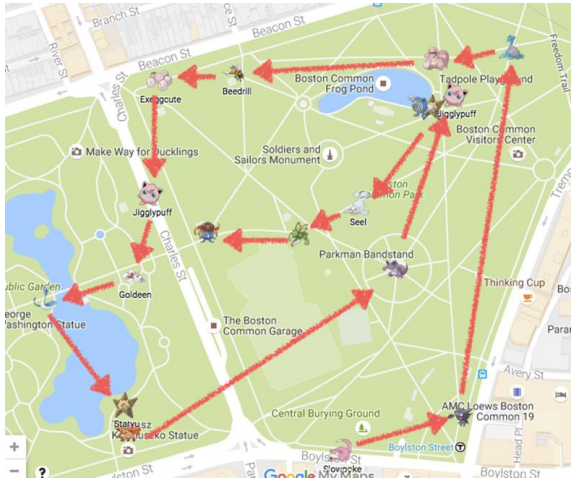


Instruction	Quantity in A (max 4)	Quantity in B (max 7)
START	0	0
Fill A	4	0
A -> B	0	4
Fill A	4	4
A -> B	1	7
Empty B	1	0
A -> B	0	1
Fill A	4	1
A -> B	0	5

https://docs.google.com/document/d/1jOGAD21D5BoVblYhs_eFRxdjQCbEaAFI9TiPuQgzdVg/edit

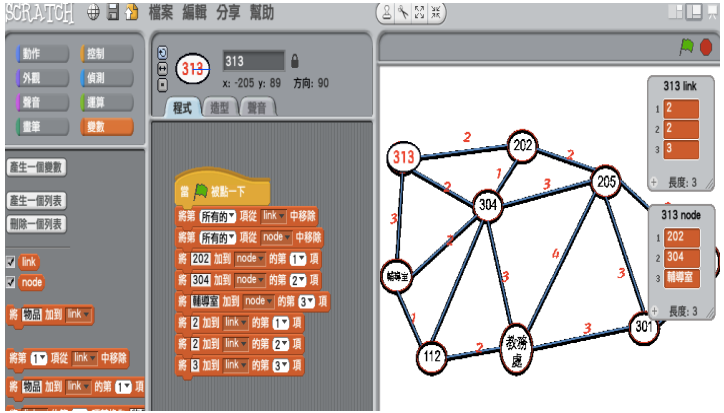
運算思維導向教材設計

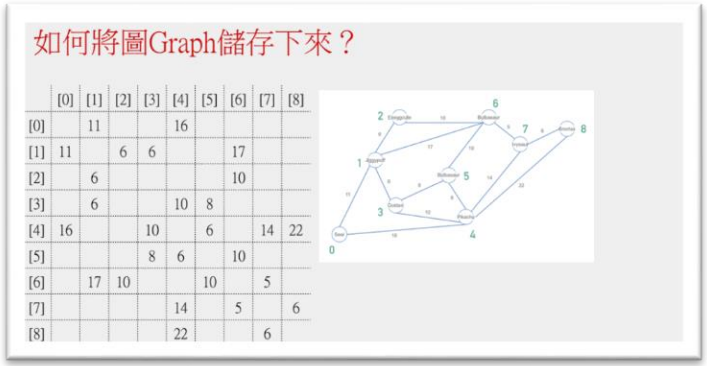
- 範例四 **演算法設計(Algorithm design)**
  - 透過視覺化的地圖與路徑，了解**最小成本的路徑**的演算法設計



運算思維導向教材設計

- 範例五 **資料表示(Data representation)**
  - 透過視覺化的地圖、花費、與陣列的對應，了解**陣列資料結構與圖（Graph）**的資料表示概念



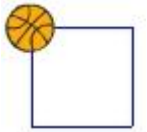


運算思維導向教材設計

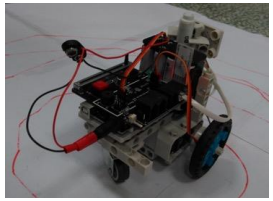
■ 範例六 **樣式辨識(Pattern recognition)、模擬(Simulation)**

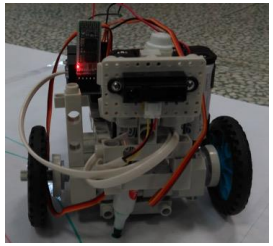
觀察實體掃地機器人

Scratch 軟體模擬



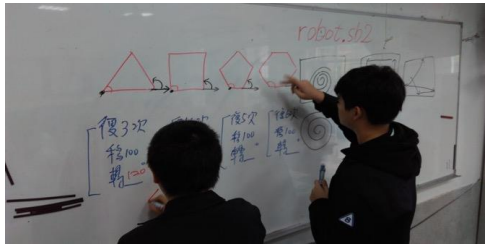
實作機器車螺旋式移動

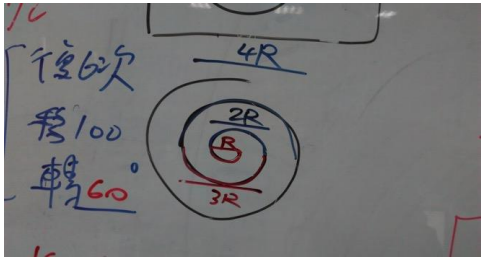




運算思維導向教材設計

- 範例六 **樣式辨識(Pattern recognition)、模擬(Simulation)**



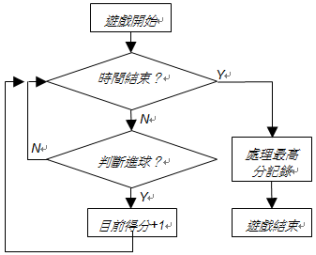


運算思維導向教材設計

範例七 投籃遊戲機學習--**問題分解（Decomposition）、演算法設計（Algorithm Design）**

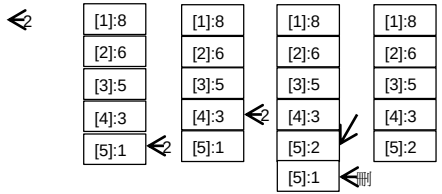
觀察實體投籃機

分析、討論投籃機的各項功能



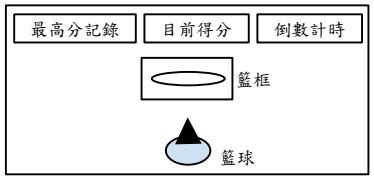
設計演算法

得分排序



運算思維導向教材設計

Scratch 軟體模擬



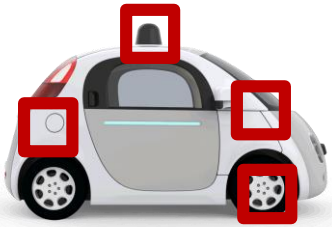
實作投籃遊戲機

運算思維導向教材設計

- 範例八 智慧避障車學習**問題分解（Decomposition）、模擬（Simulation）**

觀看影片

分析、討論
避障車的各項功能

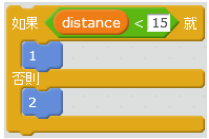


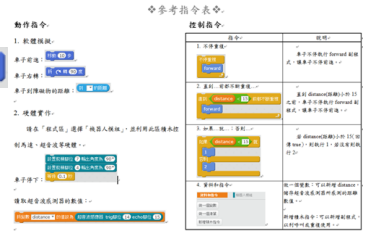
任務卡

請讓你的智慧自走車在遇到障礙物時停下來，避免撞擊到障礙物，然後右轉或左轉，繼續前進。

設計演算法

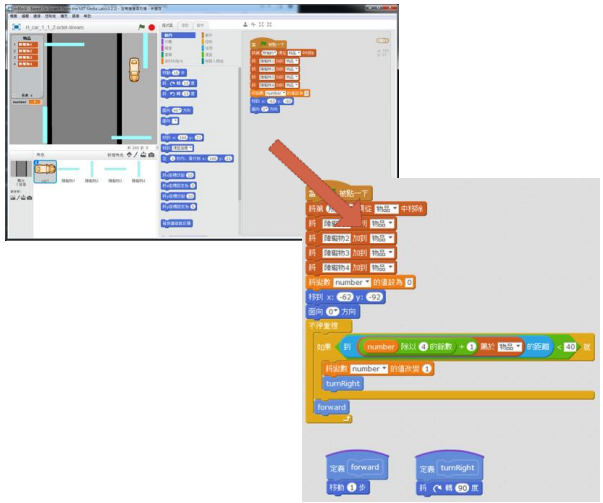
前進 移動 10 步
↓
遇到障礙物
↓
停下
↓
轉彎 將 轉 90 度
↓
前進 移動 10 步



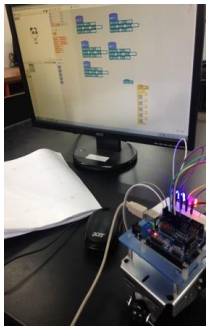


運算思維導向教材設計

mBlock 軟體模擬



實作智慧避障車



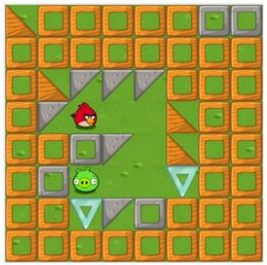
運算思維導向教材設計

- 範例九 **樣式辨識(Pattern recognition)**

80508001E_豐佳燕

二、迴圈-重複循環

1



(1)  要如何找到  ？

用箭頭畫出  路徑(畫在左圖)。

(2) 用積木來表示就像下圖，你發現有重複的地方嗎？試著找出重複指令的區塊，並把它「圈起來」，有(　　)個區塊？

運算思維導向教材設計

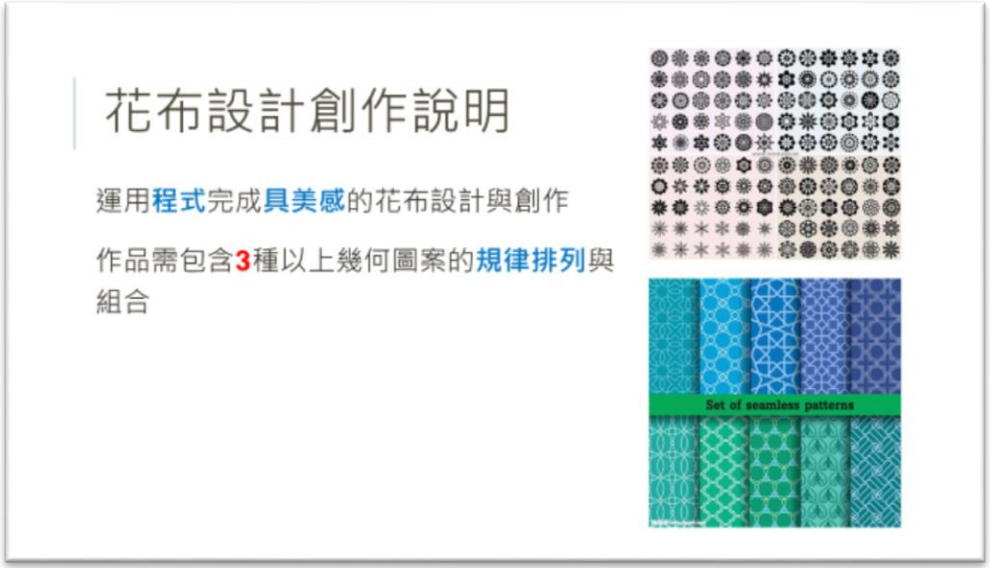
- 範例十、照片照騙-**資料表示（Data representation）、演算法思維（Algorithmic thinking）**
  - 透過影像處理，學習二維陣列的存取與程式設計

運算思維導向教材設計

- 範例十一 **資料表示(Data representation)**
  - 學習影像的資料表示、取樣、與量化的概念

運算思維導向教材設計

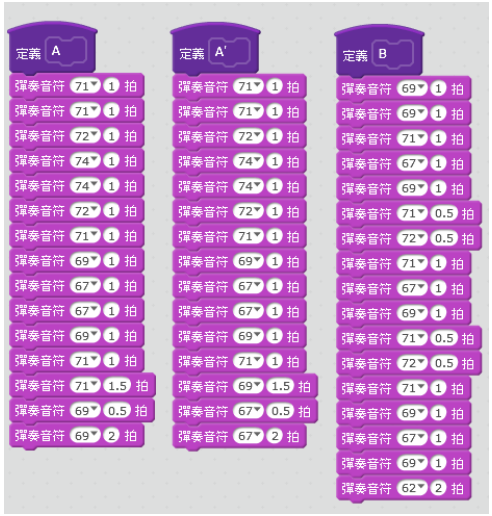
- 範例十二 **樣式辨識(Pattern recognition)**
  - 透過重複樣式的觀察，學習迴圈的概念

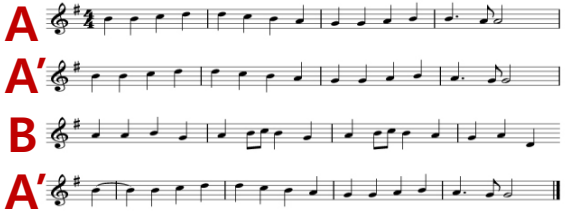


- 能夠解構圖形並尋找重複性規律
- 能善用自定函式與迴圈減少重複程式碼
- 能運用變數與運算式進行座標推算、圖形變化規律運算與RGB顏色數值運算

運算思維導向教材設計

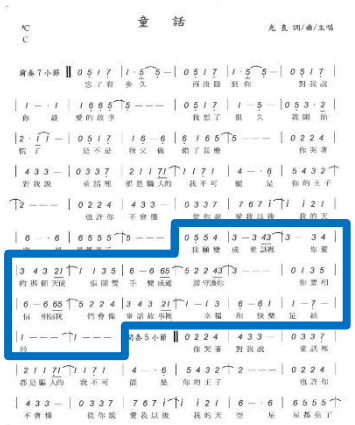
- 範例十三 **樣式辨識(Pattern recognition)、問題解析(Decomposition)**
  - 透過聽覺化音樂的辨識，了解**模組化**的樣式(pattern)





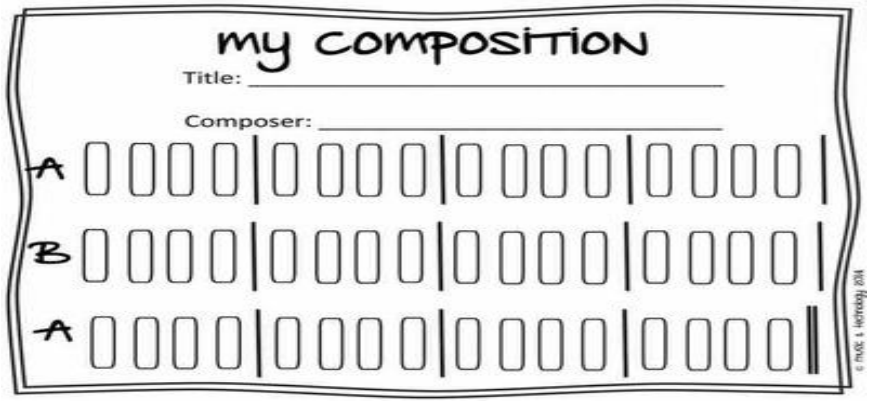
運算思維導向教材設計

- 範例十三 樣式辨識(Pattern recognition)
  - 透過聽覺化音樂的辨識，了解模組化的樣式(pattern)，進一步進行音樂創作



主歌Verse

副歌Chorus



運算思維導向教材設計

範例十四 資料表示與處理(Data representation and processing)

科技領域/資訊科技學習重點		科技領域核心素養
學習表現	學習內容	
資t-IV-4能應用運算思維解析問題。 資p-IV-1能選用適當的資訊科技組織思維，並進行有效的表達。 資p-IV-2能利用資訊科技與他人進行有效的互動。 資p-IV-3能有系統地整理數位資源。	資D-IV-1資料數位化之原理與方法 資D-IV-2數位資料的表示方法 資D-IV-3資料處理概念與方法 資T-IV-1資料處理應用專題 資T-IV-2資訊科技應用專題 資A-IV-1演算法基本概念 資A-IV-2陣列資料結構的概念與應用 資A-IV-3基本演算法的介紹 資P-IV-1程式語言基本概念、功能及應用 資P-IV-2結構化程式設計 資P-IV-3陣列程式設計實作 資P-IV-4模組化程式設計的概念 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	科-J-B1 具備運用各種科技符號與運算思維表達溝通的素養，並理解日常生活中科技與運算的基本概念，應用於日常生活。 科S-U-B1 具備精確掌握各類科技符號與運算思維表達的能力，能有效進行思想與經驗的表達，與他人溝通並解決問題。

運算思維導向教材設計

- 範例十四 **資料表示與處理、演算法思維**

  - **資料壓縮**
  - **活動 拼出間諜的長相**
    - 你拿到了敵方陣營的間諜拼圖，不幸被打散了，由於捉出間諜之時間有限，只能以十片拼圖將敵方陣營的間諜照片拼起來，你會選哪幾片拼圖拼出最像的間諜?為什麼?

運算思維導向教材設計

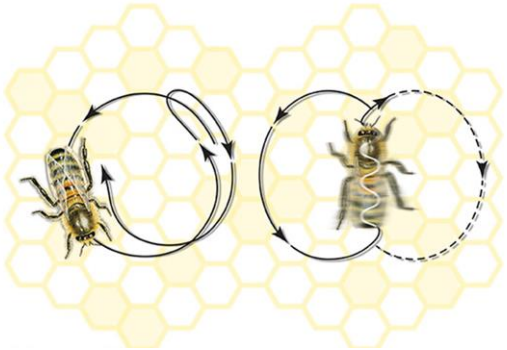
活動 黑白旗

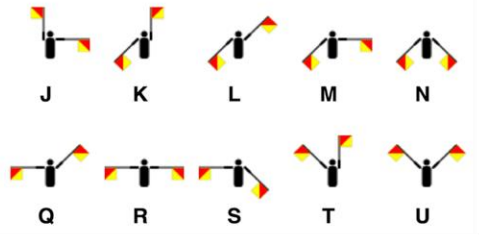
分成兩組，每組再分為發訊者與收訊者，各組先根據字庫，設計好編碼的原則，然後各組發訊者與收訊者帶開，由發訊者用黑白旗發出老師指定的訊息，收訊者負責解碼，看各組收訊者誰的解出來的訊號是正確的。

- 字庫: ABEFILMNOTUVXYZ

運算思維導向教材設計

- 範例十五、閃光傳情-**資料表示（Data representation）**
  - 透過旗語遊戲與蜜蜂的舞蹈，體驗資料表示對資訊傳遞的重要性，並理解資料表示的策略





討論

- 以上教材，不論插不插電，是否都有電腦科學?
- 有問題解決歷程的教學就是有運算思維嗎?
- 什麼樣的教材才能符合運算思維導向的教學?

Barefoot

Computing at School

Recommended for
ages 5-7

World Map Logic -

Logic and Programming

Logic

I know it's raining

I know if I wear my coat I do not get wet

I predict if I wear my raincoat today I will not get wet

Logic

I am hungry!

I know I am hungry

I know if I eat an apple I am not hungry

I am not hungry!

I predict if I eat an apple I won't be hu ngry anymore

Logic?

I am hungry!

?

I know I am hungry

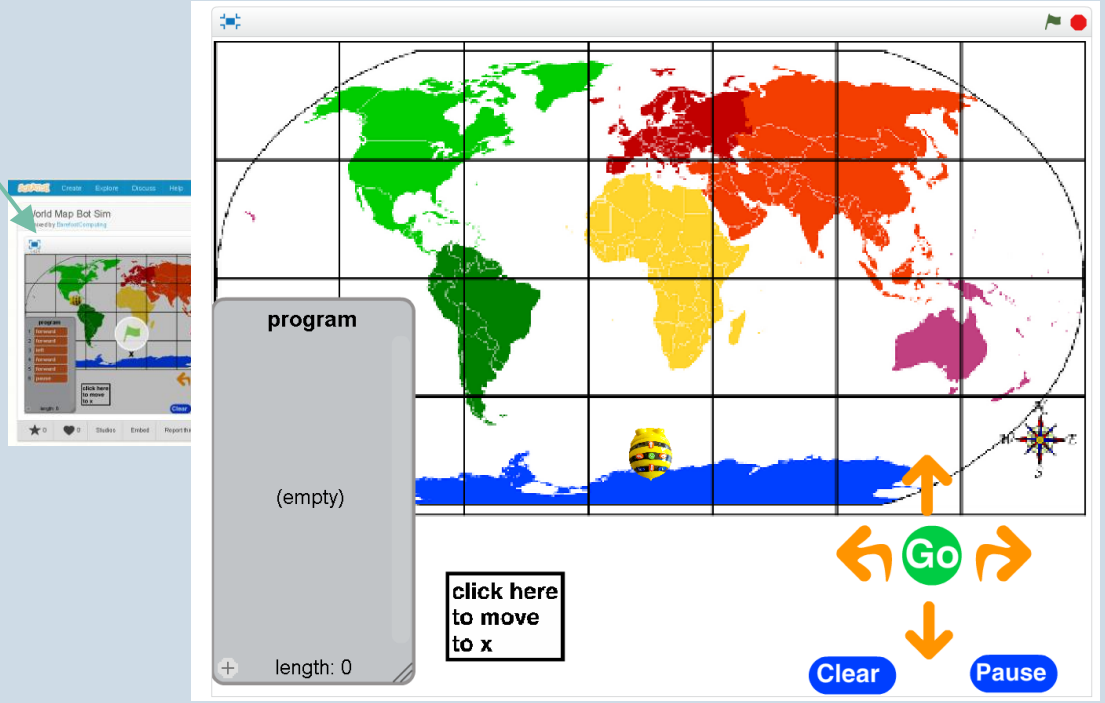
I know if I put my coat on I will not get wet

I predict if I put my coat on I will not hungry anymore

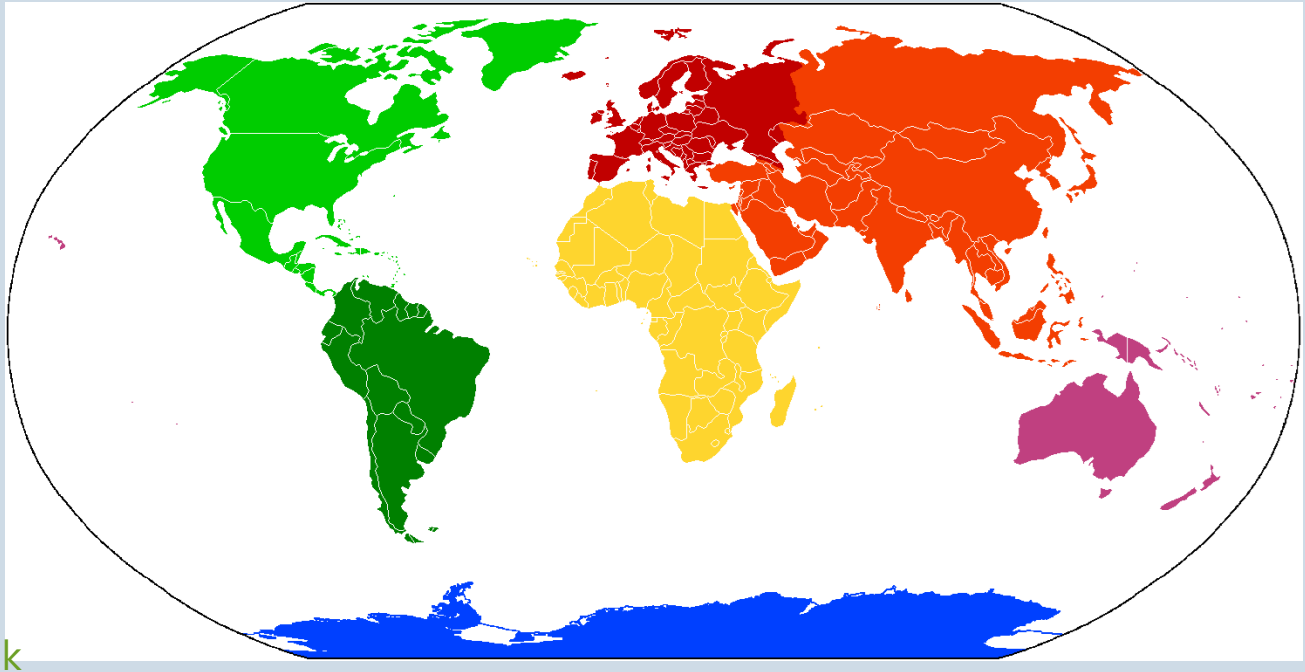
Today we are learning about...

- I can predict what a program will do
- I can explain why I think this

Logic


World Map Bot Sim
program
(empty)
length: 0
click here to move to x
Go
Clear
Pause

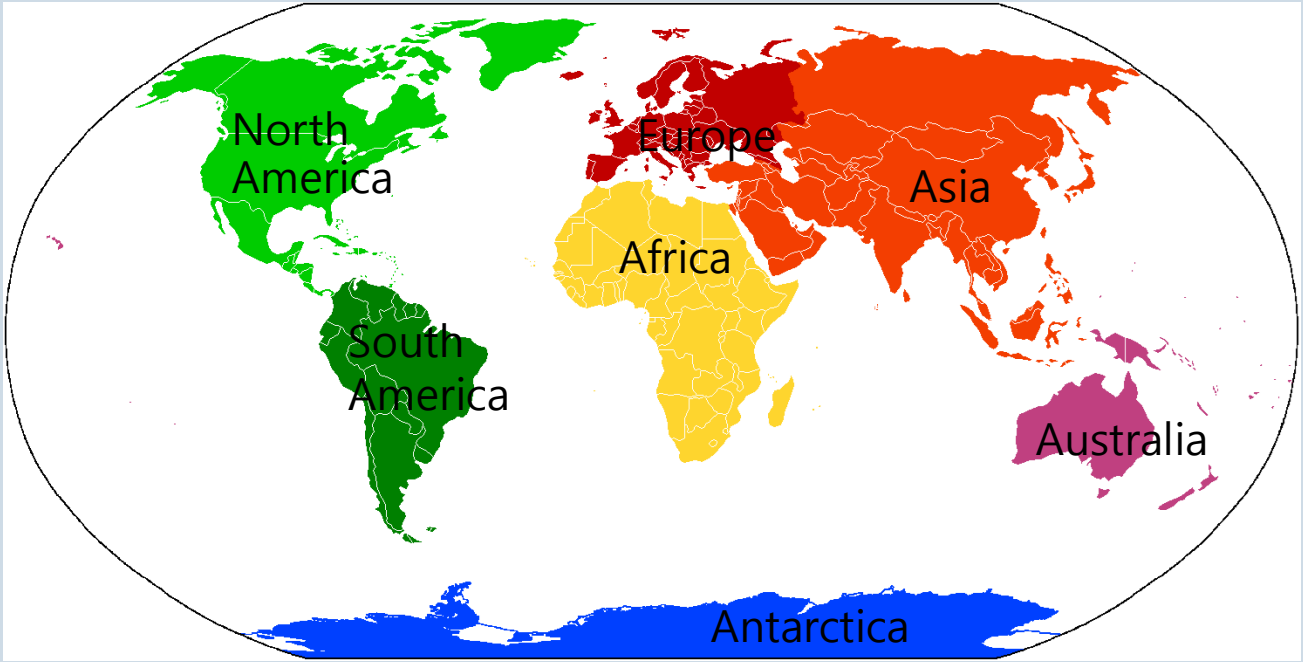
What is this?



Song

Continents link

What is this?


North America
Europe
Asia
Africa
South America
Australia
Antarctica

Europe

North America

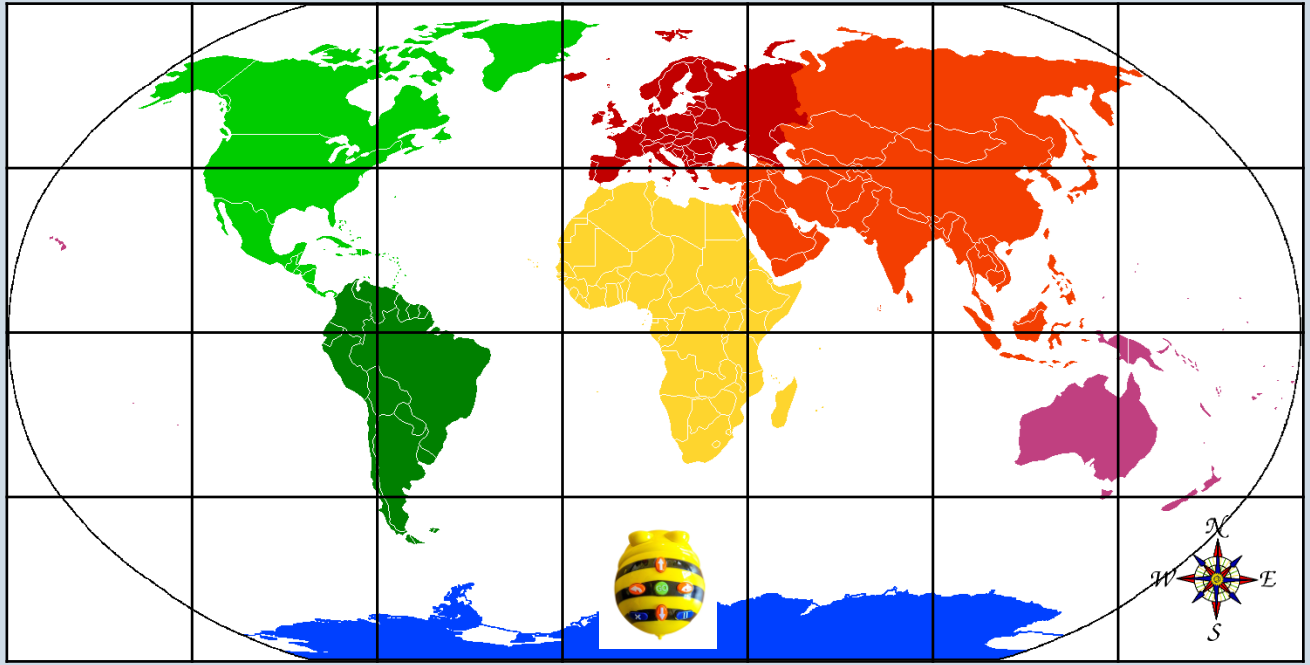
South America

Asia

Australia

Africa

Antarctica



Colours	Red	Pink	Dark Green
Yellow	Orange	Blue	Light Green

Europe

North America

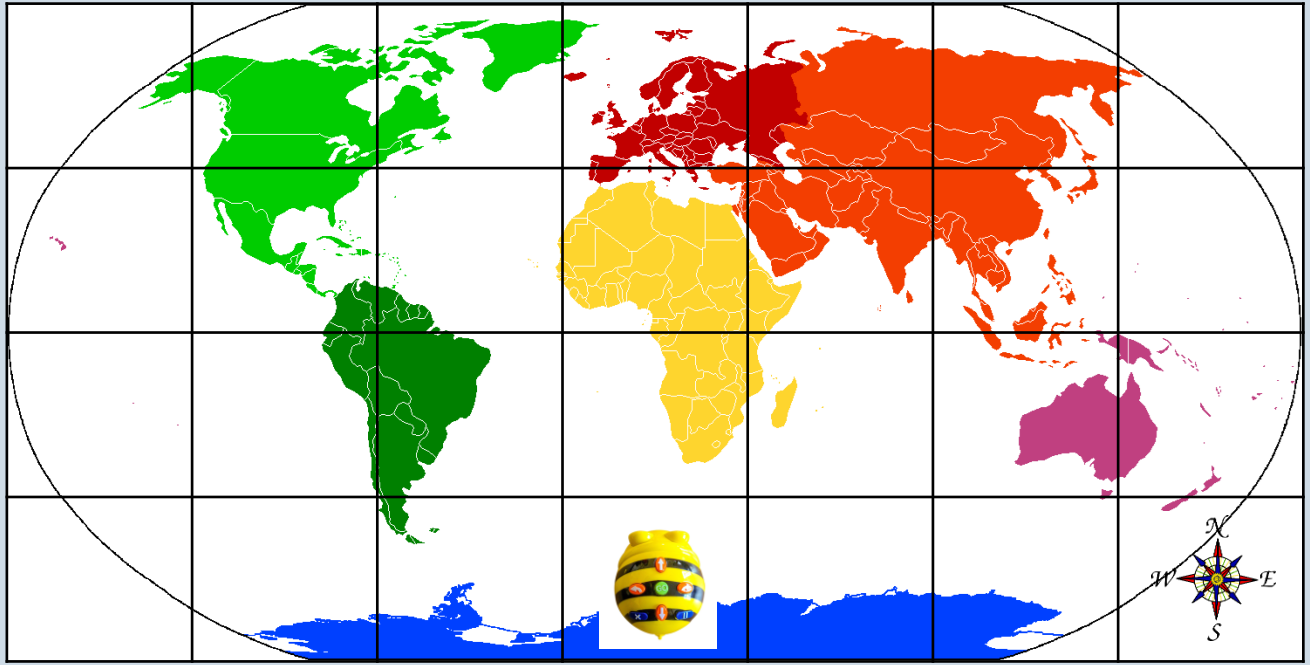
South America

Asia

Australia

Africa

Antarctica



Colours	Red	Pink	Dark Green
Yellow	Orange	Blue	Light Green

Europe

North America

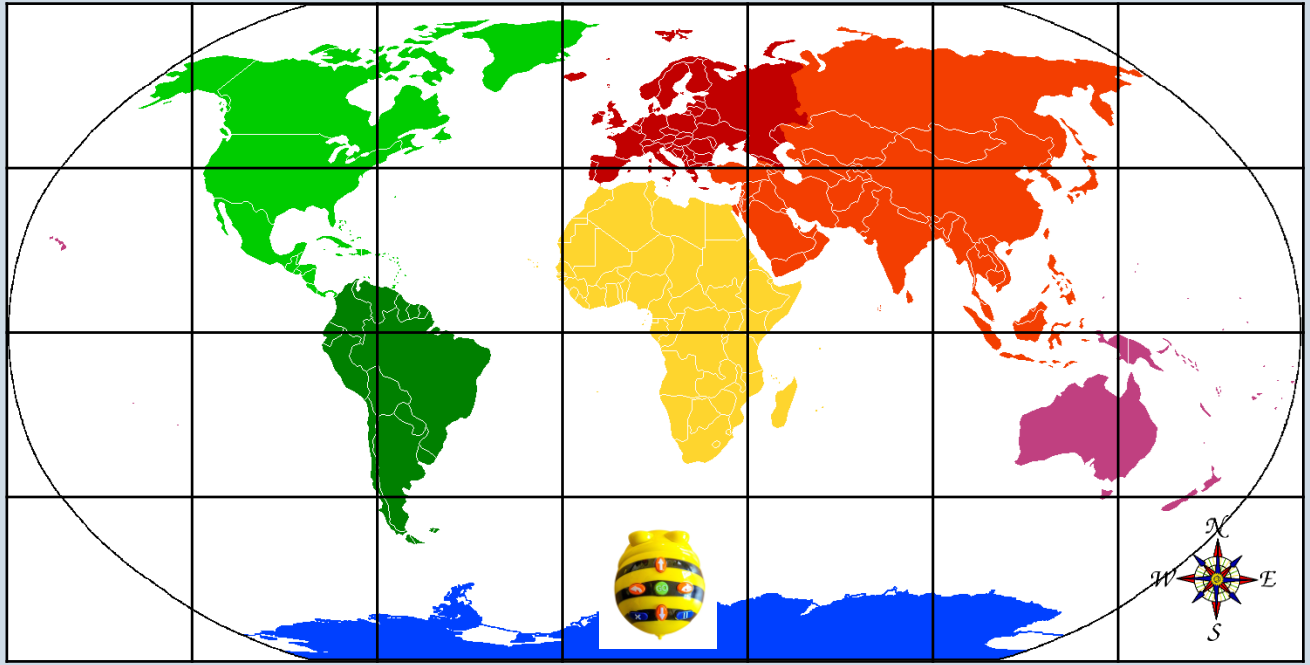
South America

Asia

Australia

Africa

Antarctica



Colours	Red	Pink	Dark Green
Yellow	Orange	Blue	Light Green

Europe

North America

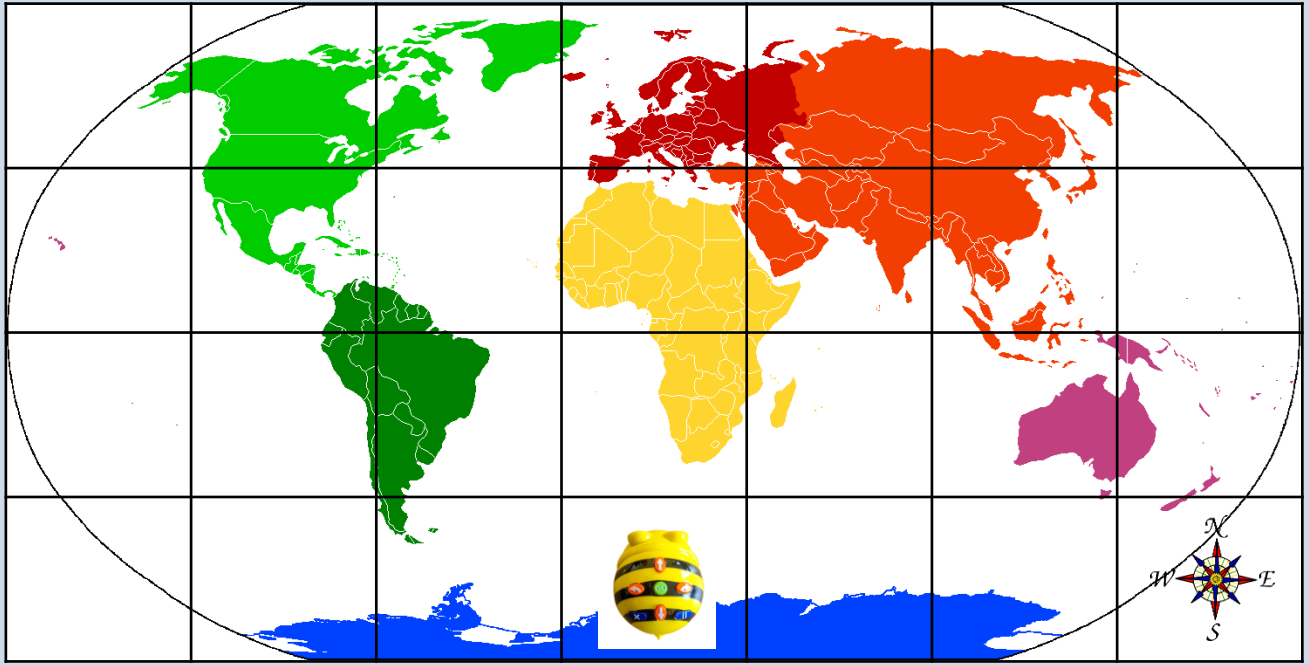
South America

Asia

Australia

Africa

Antarctica



Colours	Red	Pink	Dark Green
Yellow	Orange	Blue	Light Green

Europe

North America

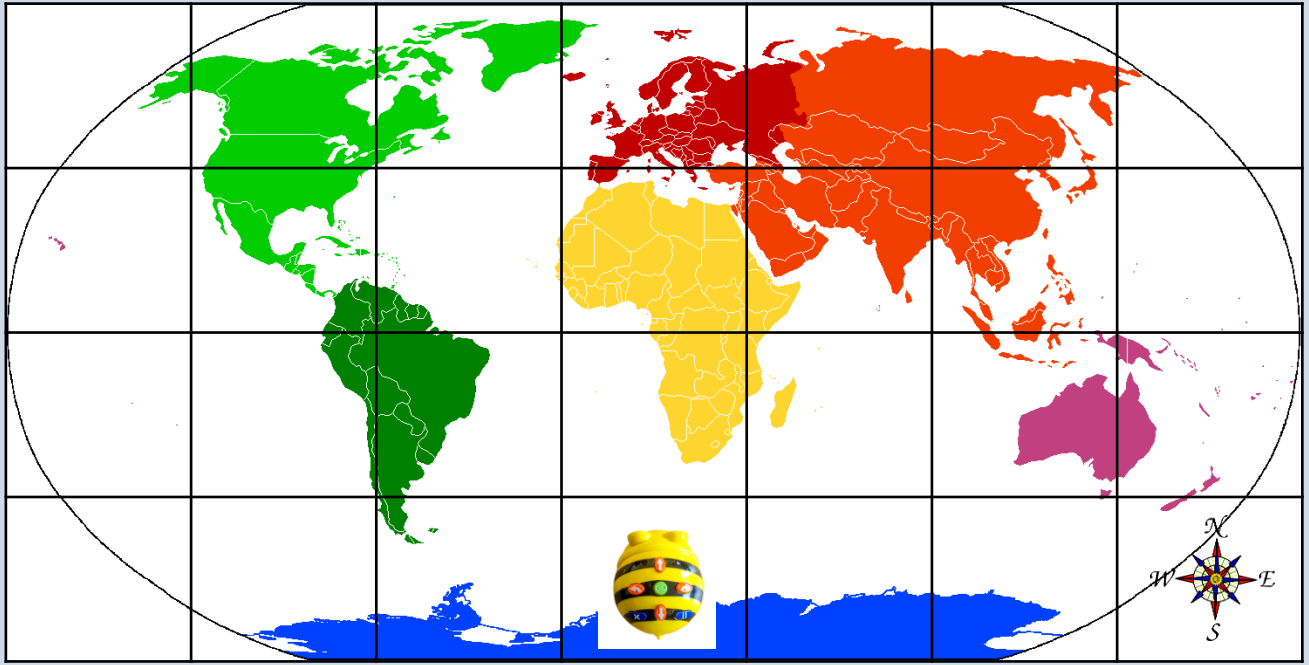
South America

Asia

Australia

Africa

Antarctica



Colours	Red	Pink	Dark Green
Yellow	Orange	Blue	Light Green

Europe

North America

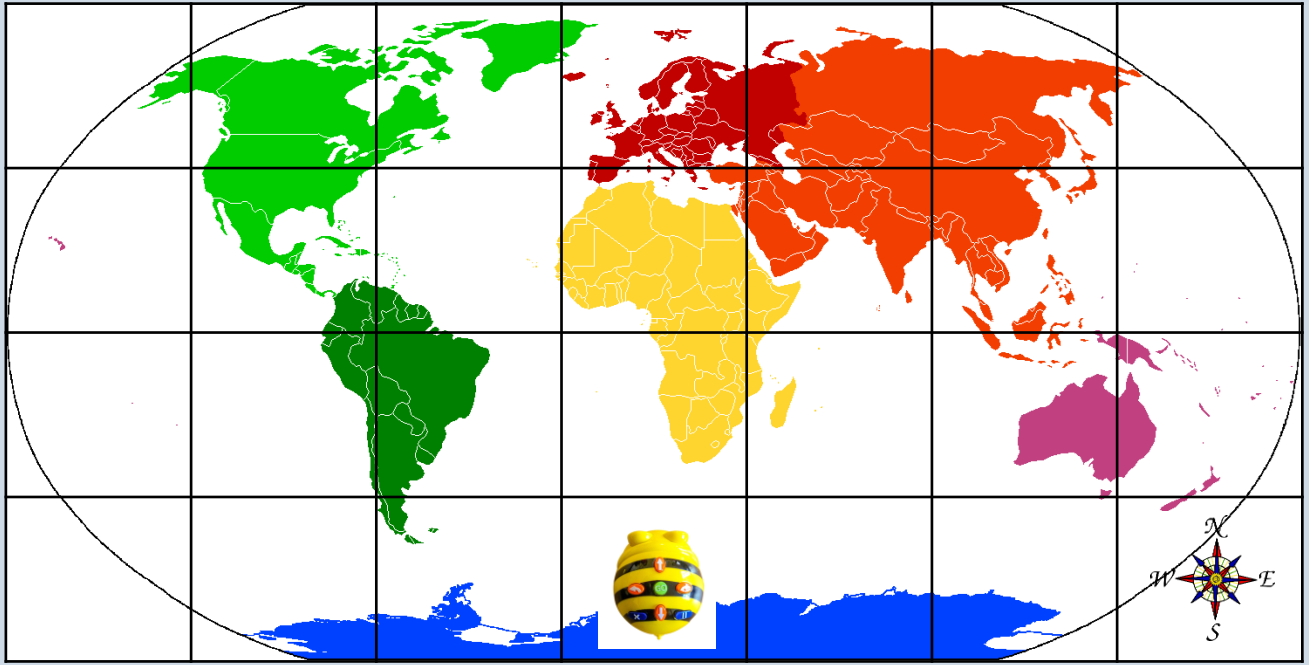
South America

Asia

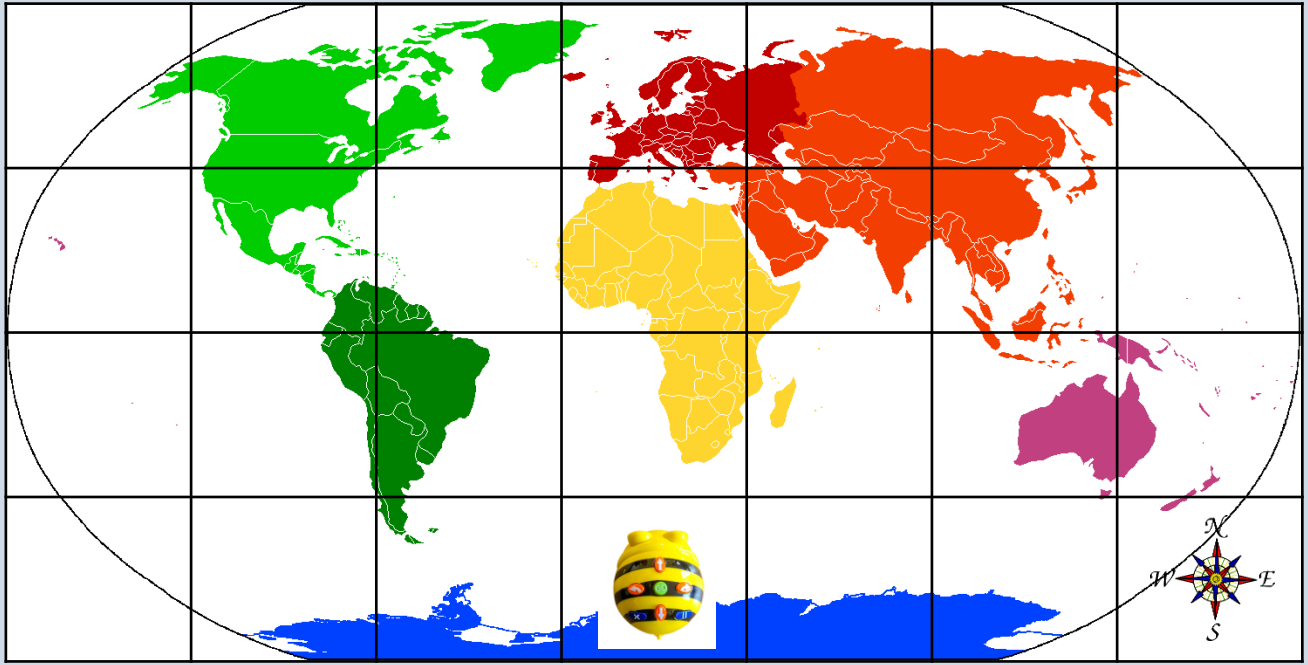
Australia

Africa

Antarctica



Colours	Red	Pink	Dark Green
Yellow	Orange	Blue	Light Green


N
W
E
S

討論

- 從這份教材學生可以學到什麼?

Barefoot

Computing at School

Recommended for
ages 7-11

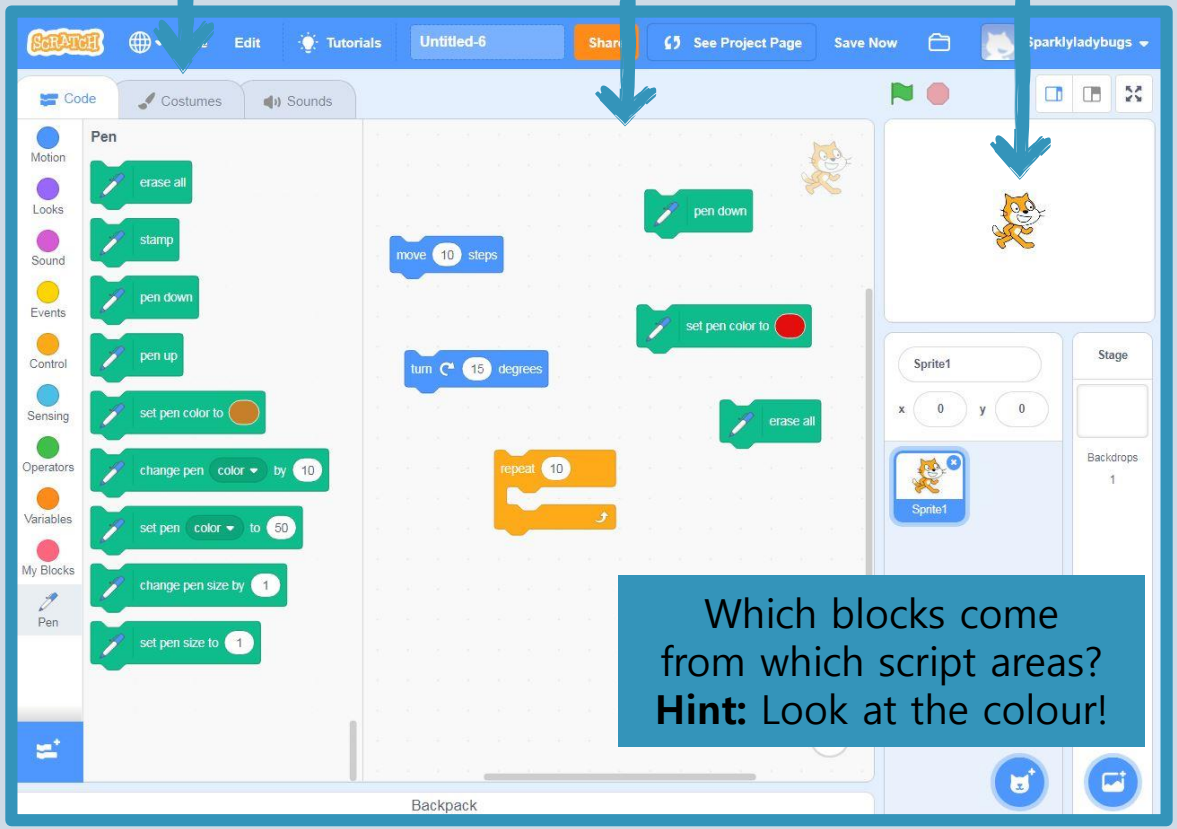
Shapes and Crystal Flowers

Use repetition in programs

Blocks Pallete

Scripting Area

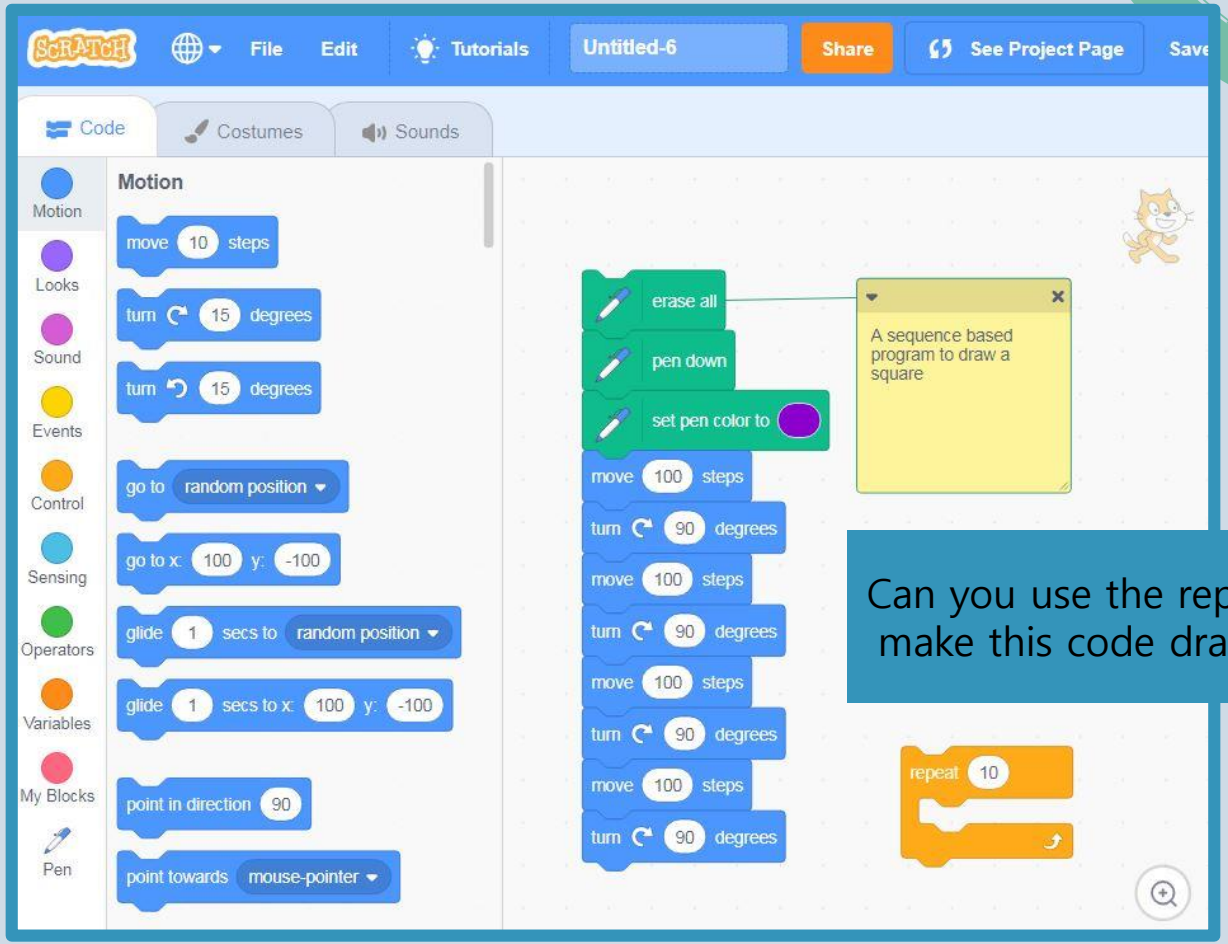
Stage



Which blocks come from which script areas?
Hint: Look at the colour!

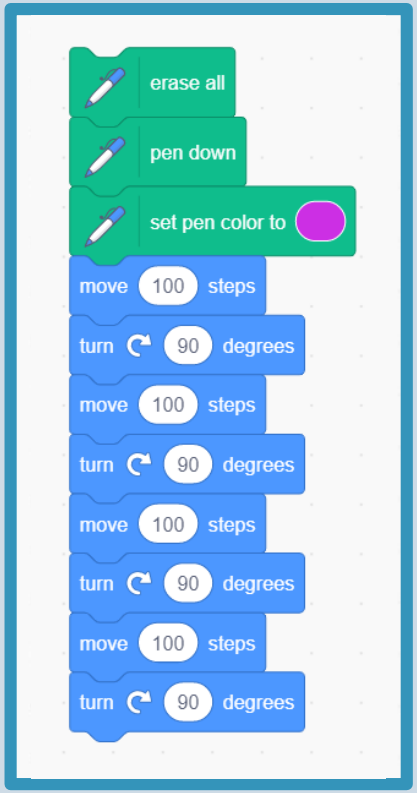
Helpsheet

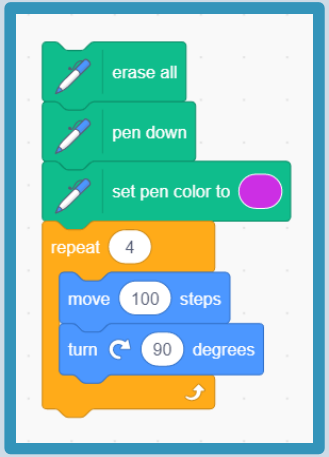
Helpsheet 2



Can you use the repeat block to make this code draw a square?

What is the difference?



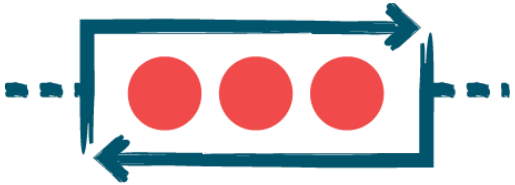


Challenge

Why are repeats (or loops) useful when we program?

Today we are learning about...

Programs & Repetition:



Repetition

- I can write a program that uses a repeat command
- I can explain what the repeats do in my program

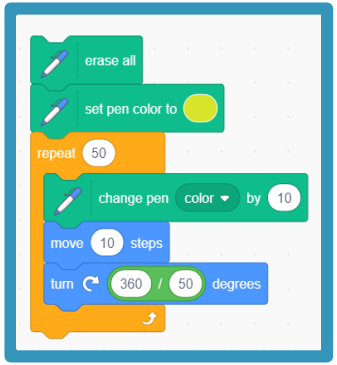
Images from Pixabay CC0 Public Domain

Images from Pixabay CC0 Public Domain

Challenge

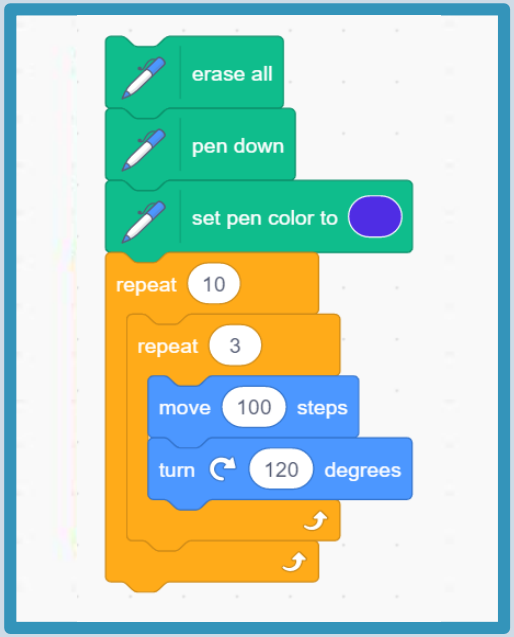
Drawing a Pentacontagon - 50 sided shape

Can you see a mistake?

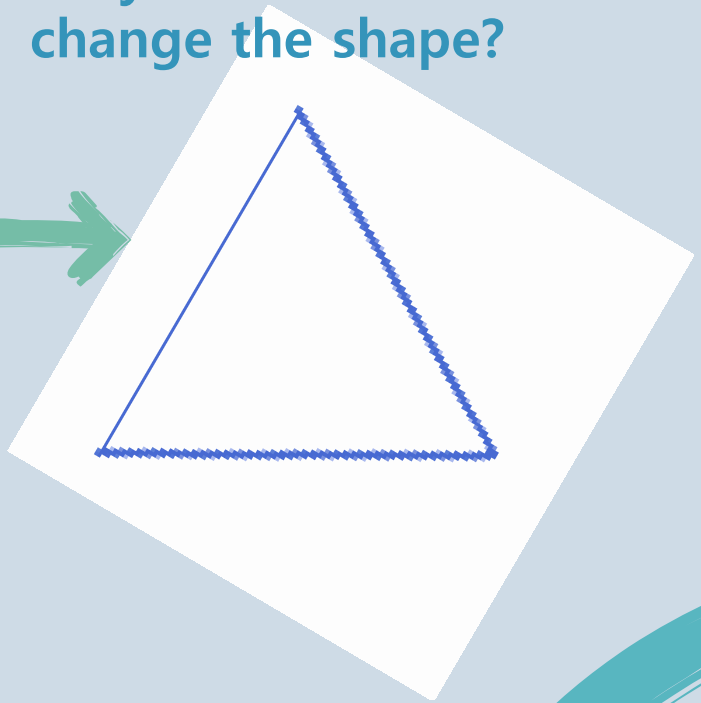


Why are repeats (or loops) useful when we program?

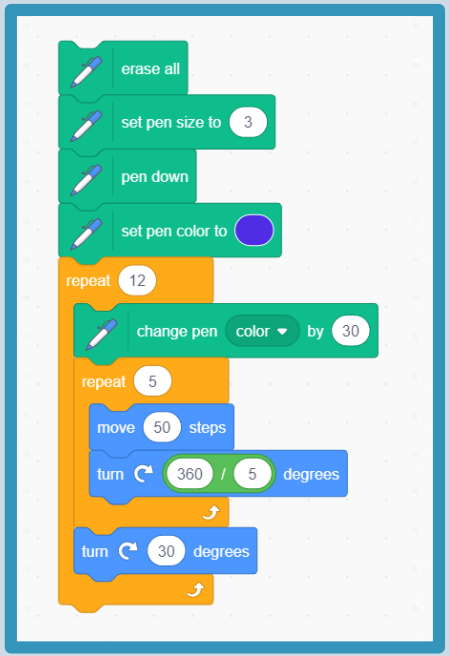
Nested Loop

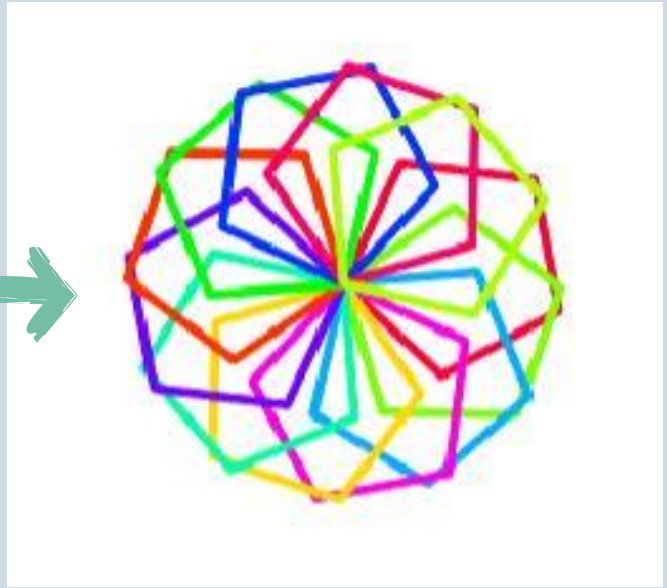


Why does this not change the shape?

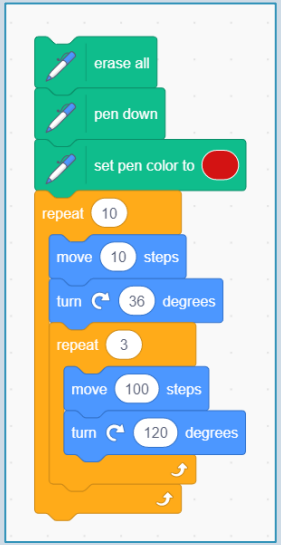


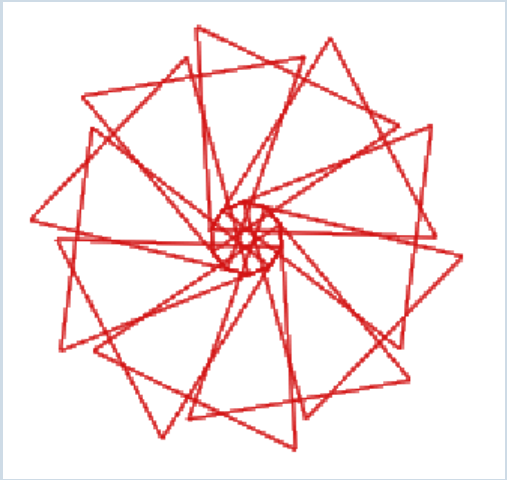
Nested Loop


erase all
set pen size to 3
pen down
set pen color to
repeat 12
change pen color by 30
repeat 5
move 50 steps
turn 360 / 5 degrees
turn 30 degrees



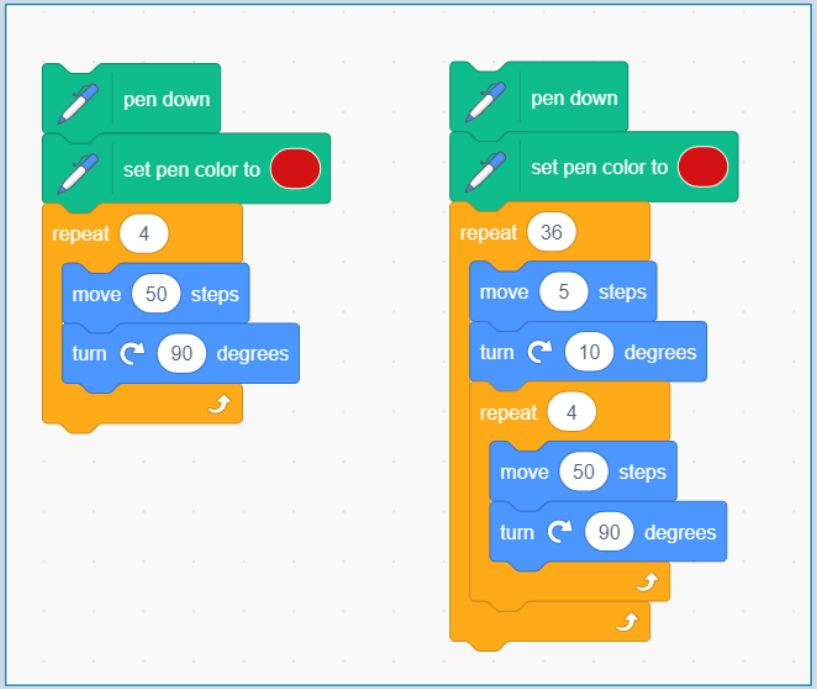
Nested Loop

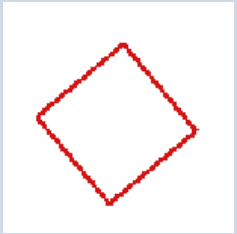


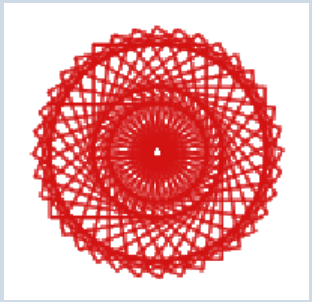


Challenge:

Nested Loop







Challenge:

討論

- 從這份教材學生可以學到什麼?

- 這樣教有什麼問題?

資訊科技教學策略-不插電的電腦科學

不插電的電腦科學

- 教學目標
  - 能瞭解二進位系統的概念
  - 能完成十進位、二進位、與各種進位法的換算
  - 認識各種電腦資料的表示
  - 能將資料不同的表示法應用於生活中之問題解決

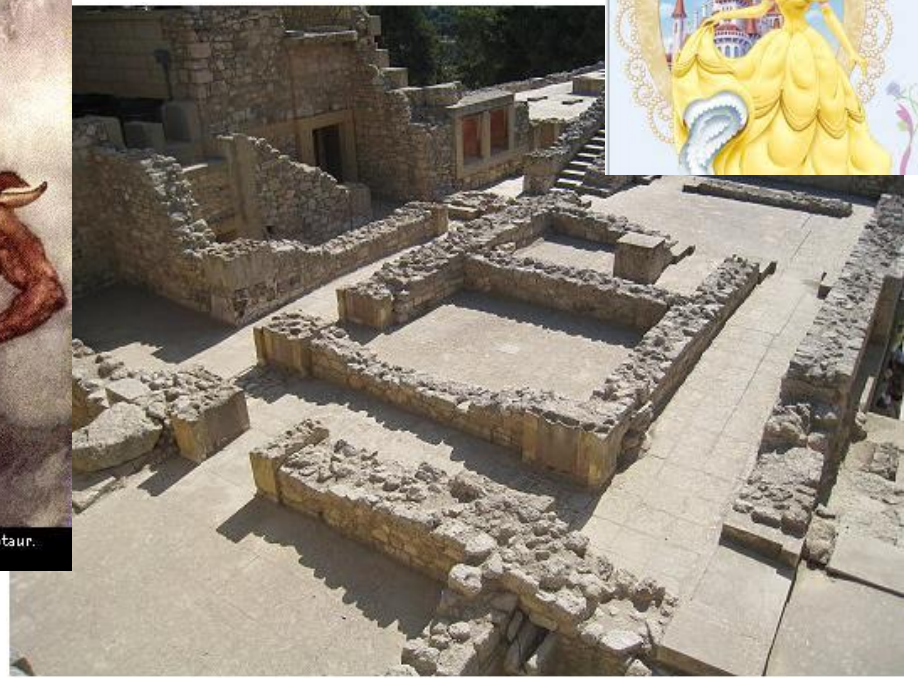
討論

- 二進位系統與電腦資料的表示教學內容的重要概念為何?

  ➔教進制轉換運算可以教到這些概念嗎?

希臘神話「黑白帆」-迷宮

W. Russell Flint, 1880–1969: Theseus and the Minotaur.
Photo © Malcar Förlag - GML.



希臘神話「黑白帆」

用二元化的符號表示
欲傳遞的資訊

表達數字

- 伸出你的手指頭
  - 從一數到二十，你會怎麼數?
  - 小叮噹會怎麼數數兒呢?

表達數字

■ **活動 聽聲辨數**

分成兩組，第一組先設定三個數字，利用【小叮噹的0&1】FALSH教學工具中的小叮噹二進位教學撥放其二進位的高低音，第二組同學解碼，再反過來由第二組發訊號，而由第一組解碼。

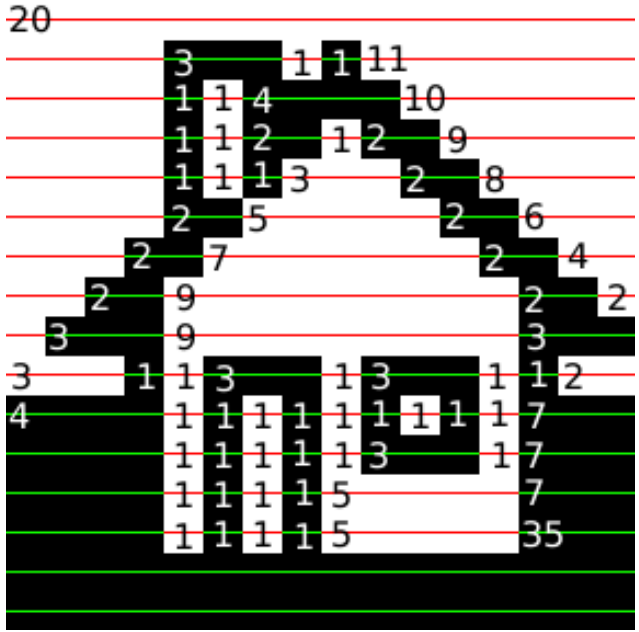
①	②	③

影像怎麼表示

- 傳話遊戲
  - 傳訊者要把嫌犯的名字全部傳給收訊者，嫌犯名字為:王壞人、林小壞。如何以最少的文字傳給收訊者?

影像怎麼表示

- Run-length coding



再壓多一點

- 各種符號出現的頻率不盡相同

a	b	c	d	e	f	g	h	i	j	k	l	m
8.2	1.5	2.8	4.3	12.7	2.2	2.0	6.1	7.0	0.2	0.8	4.0	2.4
n	o	p	q	r	s	t	u	v	w	x	y	z
6.7	7.5	1.9	0.1	6.0	6.3	9.1	2.8	1.0	2.4	0.2	2.0	0.1

- 知道這些有助於將文章編碼壓縮嗎?

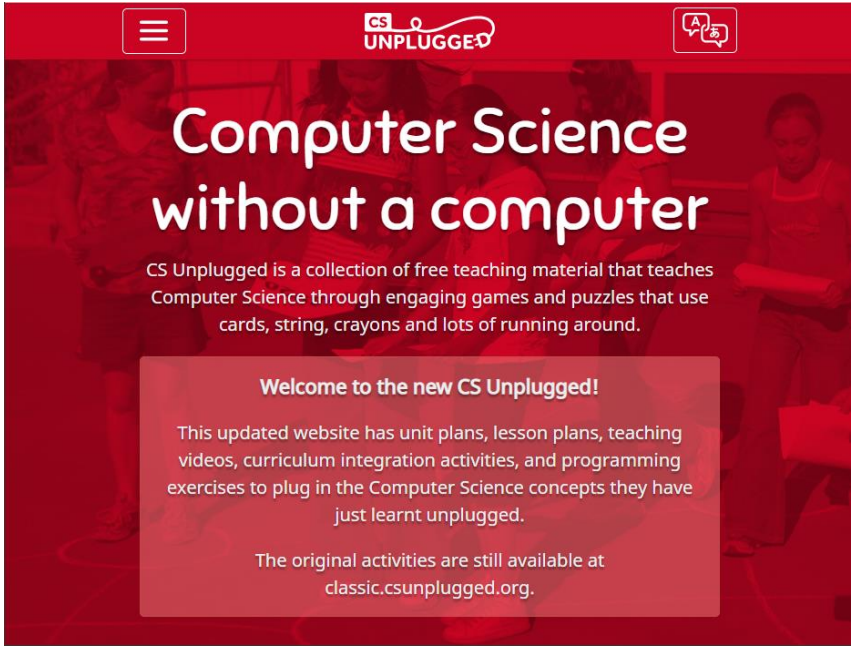
密碼學

- 你會解碼嗎?

usklp lfxyn mqmsn xisgl tysnq ywwyv zsfkk qnwks wrvqh vswiy rvqps wwsyv olool h swyv wswmq xsyvw lifpp svsnu ssxln lkcws vsusm msnxn qxvsl hsnxk qulkl fnxsv usr xs kluqv vswrq nilnu suqis ssnxv soloo lhssx wswis ybuqm rkfus wksrv qpsww syvae slxwx qnssx kluqm xsws kqzsk lussn wsmok suswq nxxvq fwisw swrvf xwksw rkywq vfhfn lybis ksyvh snsvl xfqnf kwsmo kstyf kwwsv lfsnx fmrkf tyswi lnwks iszsk qrrsm snxiy nsmlu efnsr qyvkl uqifp fulxf qnlyx qmlxf tysis wmsww lhswu styfw svlfx ynsms nlusx svvfo ksrqy vkswf nxsvs xwisk loskk spvln usnqy wsxyi fqnwk lrqww fofkf xsisp lfvsi swuqr fsws uvsxs wisus wrkln wsnyx fkfwl nxksw cwxsm sfnhs nfsyb ismil hysv v szsyf kksdv srvsn ivsuq nxlux wfkzq ywrkl fxlzs ukszq ksyvt yfswx wqyru qnnsw sxv qy zsvlw qyxel mrxqn fknsp lfxly uyniq yxsty susxv lfxvs iswus ksvlx zlues vuesv lfmr q wsvwq nrvfb usufv skszs isklr kywel yxsfm rqvxl nusnl xfqnl ksplf wlnxk qogsx iskl w lnuxf qnisk qyfwr efkfr rskyf msmss xrqyv vlfxm snlus vklkk flnus tyliv yrksw fuskl xqmol fxsnx vsism lyzlf wswml fnwxq yxsuq vvswr qniln uspyx yvsuq nusvn lnxus rvq rq wiszv llzqf vynuq iszfh snsvs ryfwt yfksw xfmrq wwfok slisu qisv

不插電的電腦科學

- 不插電的電腦科學是系列透過卡片、蠟筆等工具與大地遊戲等方式學習電腦科學概念的教學
- 聚焦於重要概念，而非低階操作
- https://csunplugged.org/



不插電的電腦科學

- Teaching London Computing 提供許多免費的教學資源
  - Queen Mary University of London 與 King's College London.
- https://teachinglondoncomputing.org
- Computing without computers 是Queen Mary University of London的 Paul Curzon 為新學者編寫的教材，介紹程式設計、資料結構、演算法等內容。透過隱喻等方式說明重要概念，而不是直接動手撰寫程式。
- https://teachinglondoncomputing.org/resources/inspiring-computing-stories/computingwithoutcomputers/

不插電的電腦科學

- 不用電腦學程式
  - 紙飛機團隊 譯

目錄

不插電的電腦科學

- 不用電腦學程式

第一部：程式

第一章. 不再混亂（簡介）

人類是一種討厭混亂的動物。我們嘗試有效規劃我們的生活、城市、家庭甚至有時是我們的桌面和房間。我們將衣服整齊地掛在衣櫥裡，我們利用郵遞區號和街道將城市分割成不同的區塊，且它們彼此相連。我們把書放在書架上，我們在超商排隊結帳，我們用盡一切方法把混亂的東西按照有秩序的方式重新規劃。

有很多種規劃事情的方法，而我們很自然地用不同的方法規劃不同的東西。當然更重要的是，在我們規劃好之後，我們希望用它們來做些什麼事。舉例來說：我們把地址按照郵遞區號規劃，是希望可以讓信件更容易被送到；而書店裡的小說通常會按照作者名字的字母開頭順序擺放，使顧客更快速找到想要的書。以我自己來說，我整理客廳的書櫃時會把令人印象最深刻的書放在最顯眼的位置，像是 Stephen Hawkin 的 *A Brief History of Time* 和 John Steinbeck 的書都會被我排在書櫃正中間，原因是我認為他們最能吸引到訪客的目光，而書店則是對於如何快速賣出書本更有興趣。

不插電的電腦科學

- 不用電腦學程式

同樣的任務，策略上的不同會影響達成方式是否更快、更簡單或更容易成功。這個概念可以透過 *Spit Not So* 這個遊戲來說明（取自 Berlekamp et al, 1982）。*Spit Not So* 是一個雙人遊戲，首先，在九張卡片上寫下 SPIT、NOT、 SO、 FAT、FOP、 AS、IF、IN、PAN。接著，把有字的那面朝上，然後每位玩家輪流選卡，當任一方獲得場上所有同一個字母的卡片（每個字母共有三張）則贏得比賽。舉例來說，如果你挑中了 SPIT、IF 和 IN，則你獲得了所有含有 I 的卡片並贏得了比賽。而如果全部的卡都抽完卻沒有任何一方獲得所有有相同字母的卡片，則雙方平手，因此這個遊戲會像是：

玩家 1: SPIT => 玩家 2: SO => 玩家 1: FAT => 玩家 2: NOT =>
玩家 1: FOP => 玩家 2: IF => 玩家 1: PAN

不插電的電腦科學

- 不用電腦學程式

在這個情況下玩家 1 贏得了比賽，因為他獲得了所有含有 P 字母的卡：SPIT、FOP 和 PAN！

多玩幾場你就可以更了解這個遊戲，然後翻去下一頁來看看可以怎麼排這些卡來讓你玩得更好（或是你能夠自己發現！）

在開始遊戲之前請先畫好下圖的九宮格，然後將字母填入格子之中。記住如何擺放字母是非常重要的，任一直線都必須要有一樣的字母，舉下圖來說：最底下那排都有 F

NOT	IN	PAN
SO	SPIT	AS
FOP	IF	FAT

不插電的電腦科學

- 不用電腦學程式

注意不要向你的對手透露你怎麼擺放你的字母，然後再像之前一樣進行這個遊戲。唯一的差別是這次每抽一張卡，你必須在方格內將抽到的那個字母打叉，然後將對手抽到的字母畫圈。舉例來說，用這個版本進行的遊戲會在以下這個情況結束：

NOT	IN	PAN
O		X
SO	SPIT	AS
O	X	
FOP	IF	FAT
X	O	X

不插電的電腦科學

- 不用電腦學程式

發現了嗎？當你的對手在玩 *Spit Not So* 時，你是在玩圈圈叉叉這個我們小時候就很會玩的遊戲！只要選擇如何排列字母，你就已經把這個很容易出錯的遊戲轉變成一個幾乎不可能輸的遊戲！這就是我所說的使用一個適當的規劃方法能給你很大的幫助！

不插電的電腦科學

- 不用電腦學程式

因此機器只是取代了進行運算的人之前所做的事情，而機器的速度和準確度代表著它們可以更快地處理更大的問題。儘管如此，它們本質上做的事情還是與之前一樣的。電腦是透過一行一行的程式語言的指令來完成任務，而進行運算的人也是依照程式去執行工作的，指令也是照著人類的語言去制定的。我們在進行多位數乘法的時候就是依照小時候學過的步驟，一步一步算出最後的結果，這些步驟就是演算法。演算法包含要做的動作以及執行這些動作的順序。除了算乘法以外，其實到處都可以看到演算法的蹤跡：例如說教你如何 DIY 書架的指令是演算法，破解魔術方塊的步驟也是演算法。事實上，在一系列牽涉到演算法設計的難題中，魔術方塊是其中最複雜的一項。

不插電的電腦科學

- 不用電腦學程式

某天當我要去商店買東西的時候，一台車開向我然後搖下車窗問：不好意思，請問你知道如何去 Kirkcroft 巷嗎？而我使用了演算法的方式回應他：

先下山；
在第一個路口左轉；
再往山上開；
然後在 Navigation 俱樂部後面第一個彎道左轉之後；
到了！

這為什麼是演算法呢？這是一連串的指令告訴對方如何採取行動，更精確來說，這是一連串的動作跟順序的巧妙安排。如果那個司機嘗試使用不一樣的順序（舉例來說在第一個步驟選擇上山不是下山），那麼他就不能到達正確的地方，而程式只是透過一連串電腦看得懂的指令讓電腦依序去執行而已。

不插電的電腦科學

- 不用電腦學程式

並不是列了一堆規則便可以叫做演算法。舉例來說，任何一個公園都可能有個看板寫著：

「請走在人行道上」
「禁止踏入草坪」
「禁止腳踏車，滑板，或溜冰鞋」
「公園每日於日落前關閉」
「禁止球類活動」

然而即使這也是一行行的規定所組成，這個卻不是演算法的案例。在飛機裡洗手間內的標語也是一樣的情況：

「禁止吸煙」
「禁止飲食」
「當安全帶警示燈亮起時請勿使用洗手間」

不插電的電腦科學

■ 不用電腦學程式

演算法告訴你如何一步一步執行任務，而上面這些規則是告訴你什麼可以做，什麼不能做。因此你應該可以發現演算法是在談論「如何」做，而不是做「什麼」。上面所提到的那些規則即使順序顛倒了，在讀的人看來意義仍然是一樣的，但在演算法裡順序是非常重要的，像是其他在飛機裡可以看到的指令：

緊急逃生發生時：

1. 拉開蓋子
2. 將門把推到開啟位置並鬆手
3. 向外把門推開

這些指令是關於「如何」逃生。當緊急事件發生的時候，沒有人會希望看到一張告示上面寫著我被允許做什麼，或不允許做什麼。因為在緊急情況發生的時候，我們需要的是如何最快離開飛機的方法。所以這些指示必須是清楚，不模糊，且非常精準的，不能夠有任何可能性讓我們對這些指令或是順序感到疑惑。換言之，我們要一個演算法。生活中充滿了演算法，而這些步驟都非常精準地告訴你如何按照順序完成指示，由此可見，順序是非常重要的。

不插電的電腦科學

- 不用電腦學程式

有一種童謠裡頭包含許多重複的循環，下面這首很明顯就是這樣：

十個小孩擠小床，小小孩兒擠向旁： “睡過去、睡過去”
擠呀擠、滾呀滾，一個小孩滾下床
九個小孩擠小床，小小孩兒擠向旁： “睡過去、睡過去”
擠呀擠、滾呀滾，一個小孩滾下床
八個小孩擠小床，小小孩兒擠向旁： “睡過去、睡過去”
擠呀擠、滾呀滾，一個小孩滾下床
（繼續）
只剩自己睡張床，小小孩兒開心躺： “晚安”

不插電的電腦科學

- 不用電腦學程式

1. 十個小孩擠小床
2. **當**（While）有其他小孩在床上（重複以下步驟）

   小小孩兒擠向旁：“睡過去、睡過去”

   全部小孩擠呀擠滾呀滾

   一個小孩滾下床
3. 小小孩兒說：「晚安」

討論

- 選擇一種程式結構，並設計一段中文的虛擬碼，教學生此程式結構。

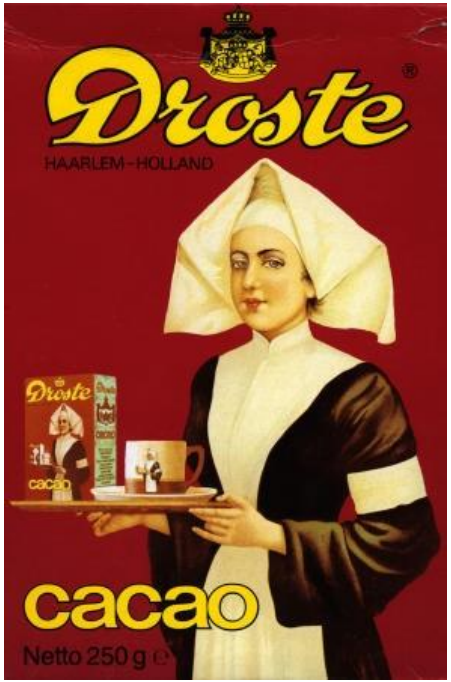
不插電的電腦科學

- 桌遊
  - 海霸

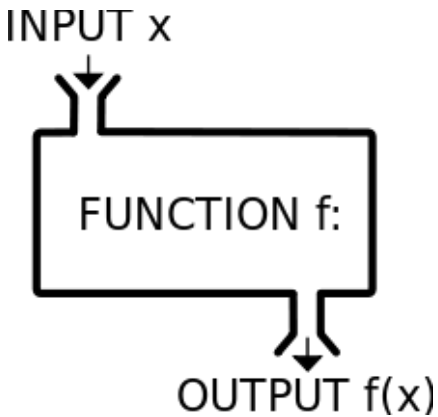
資訊科技教學策略-視覺化

視覺化

- 將抽象概念視覺化
  - 遞迴

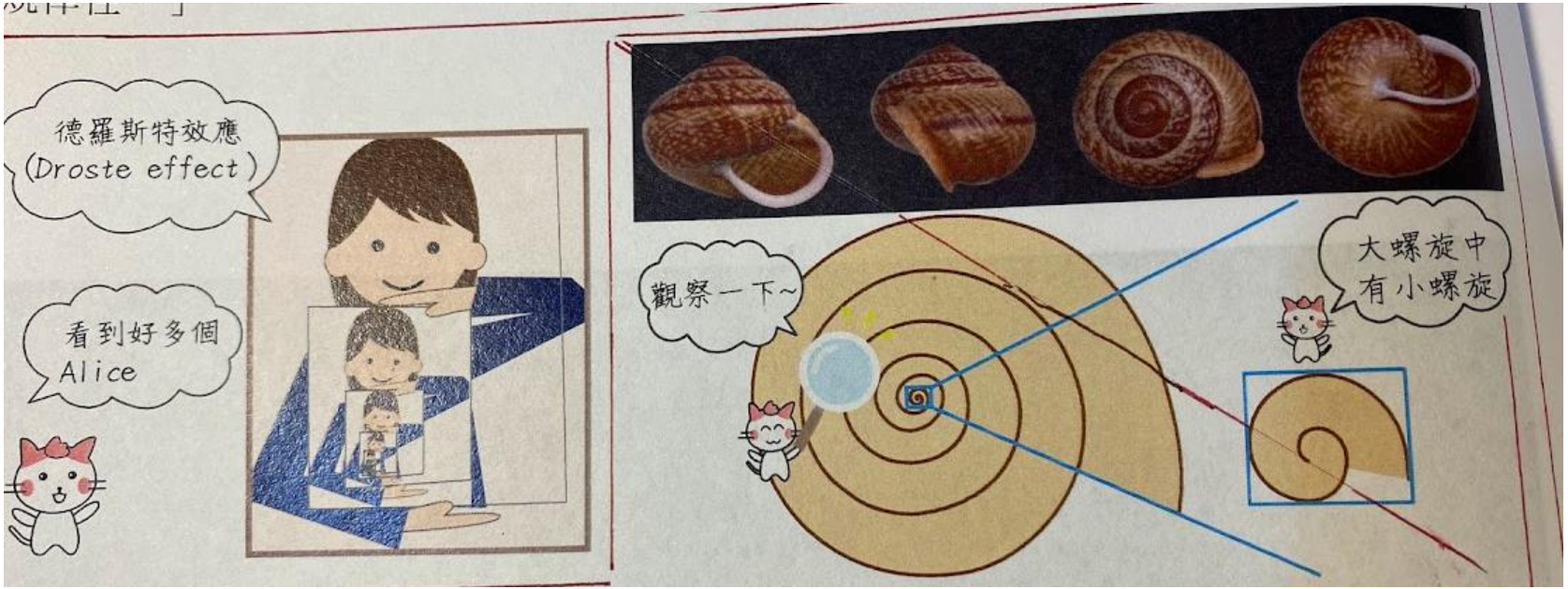


  - 函式



討論

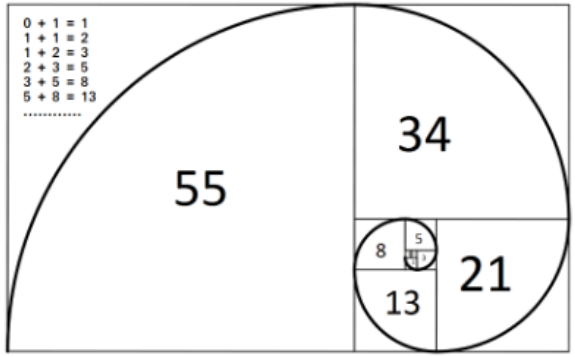
- 這是好的遞迴結構教材嗎?

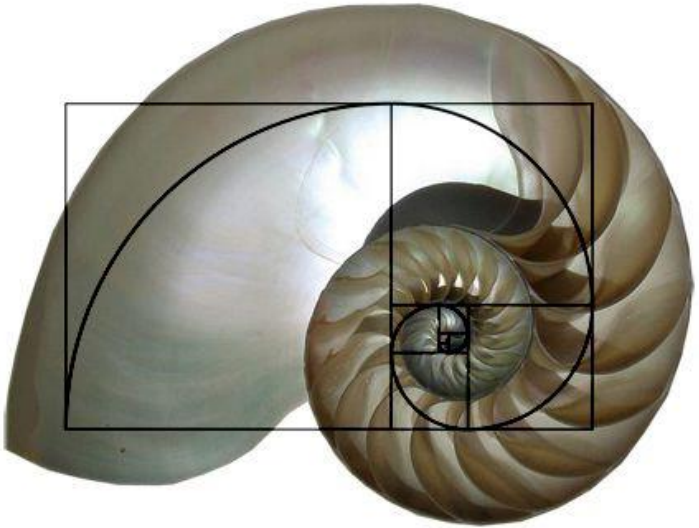


視覺化

■ 這是好的遞迴結構教材嗎?

Following is a simple program to print first n Fibonacci numbers.



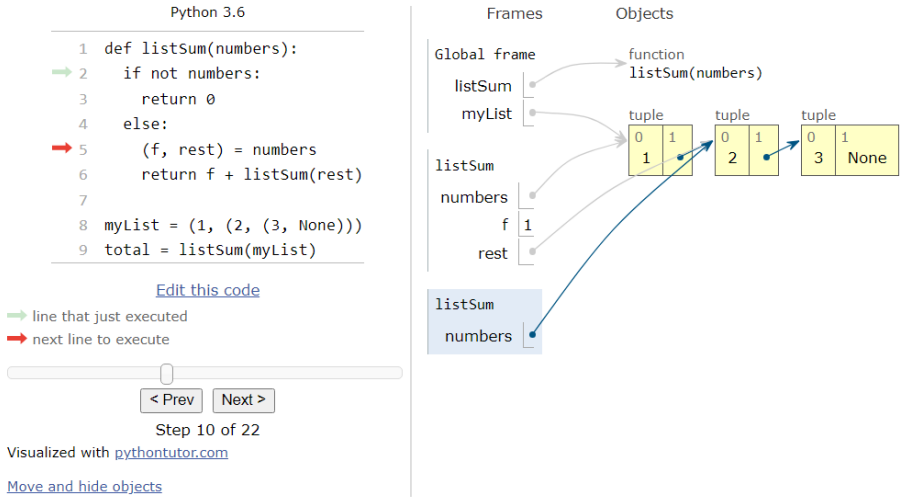


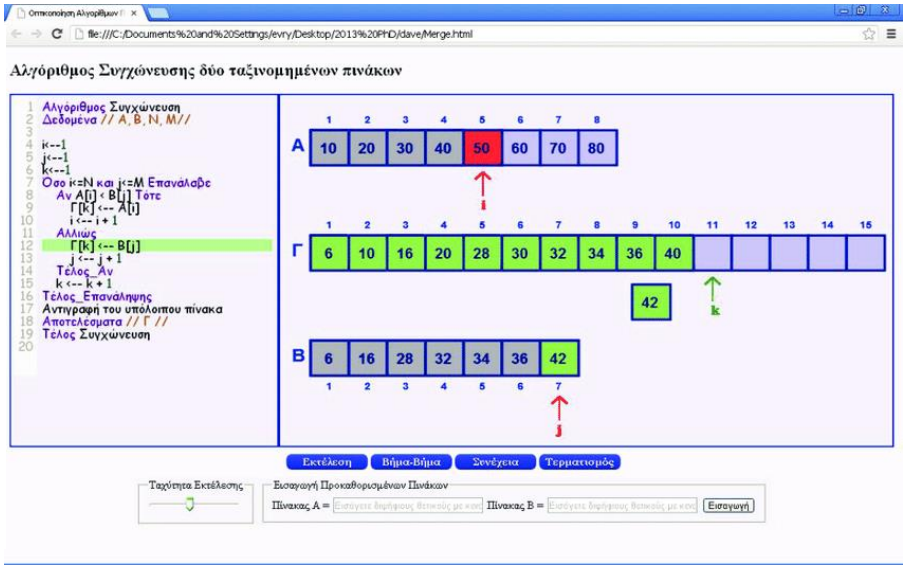
Examples :

```
Input : n = 3
Output : 0 1 1

Input : n = 7
Output : 0 1 1 2 3 5 8
```

視覺化

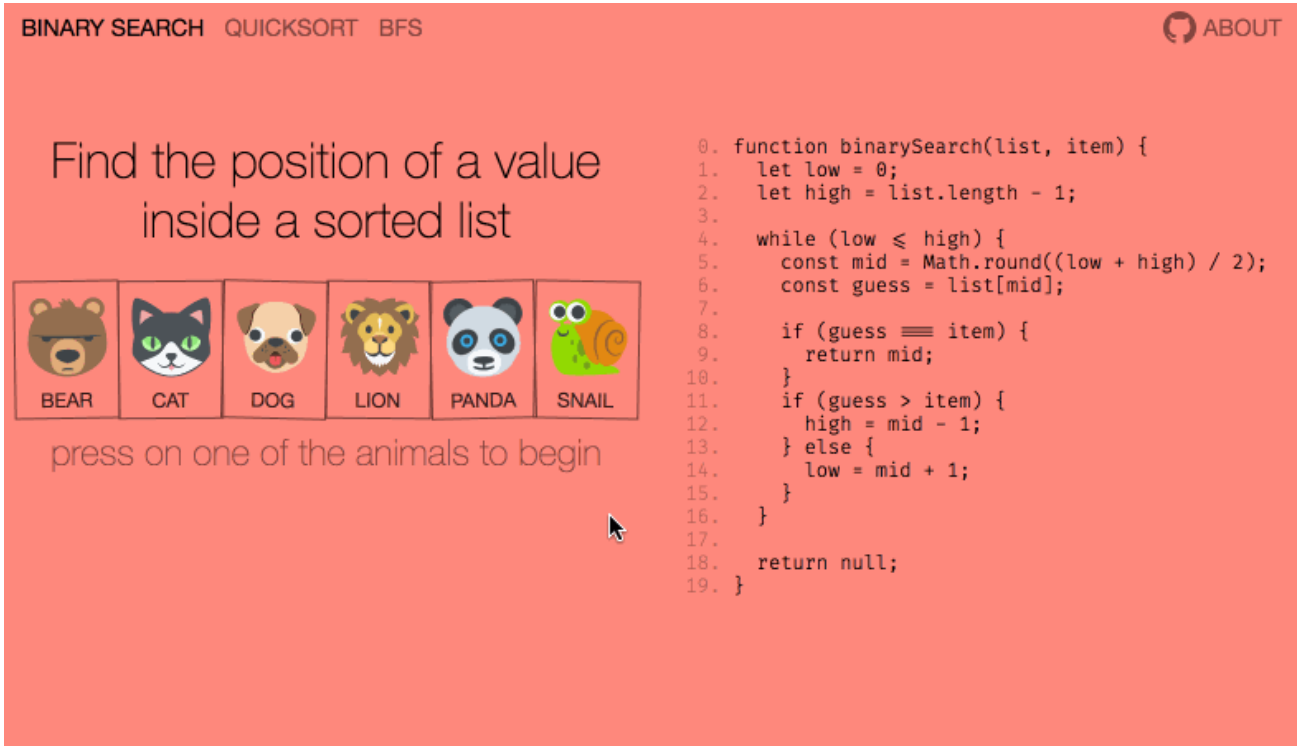
- 視覺化程式執行中的變數變化





https://pythontutor.com/

視覺化

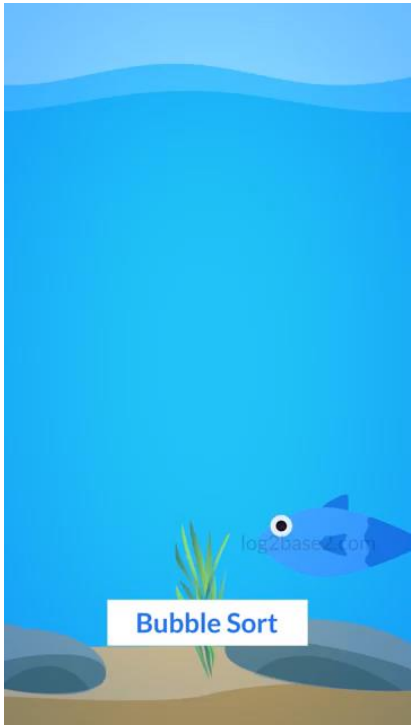

BINARY SEARCH
QUICKSORT
BFS
ABOUT
Find the position of a value inside a sorted list
BEAR
CAT
DOG
LION
PANDA
SNAIL
press on one of the animals to begin
0. function binarySearch(list, item) {
1. let low = 0;
2. let high = list.length - 1;
3.
4. while (low ⩽ high) {
5. const mid = Math.round((low + high) / 2);
6. const guess = list[mid];
7.
8. if (guess === item) {
9. return mid;
10. }
11. if (guess > item) {
12. high = mid - 1;
13. } else {
14. low = mid + 1;
15. }
16. }
17.
18. return null;
19. }

視覺化

- 視覺化演算法程序

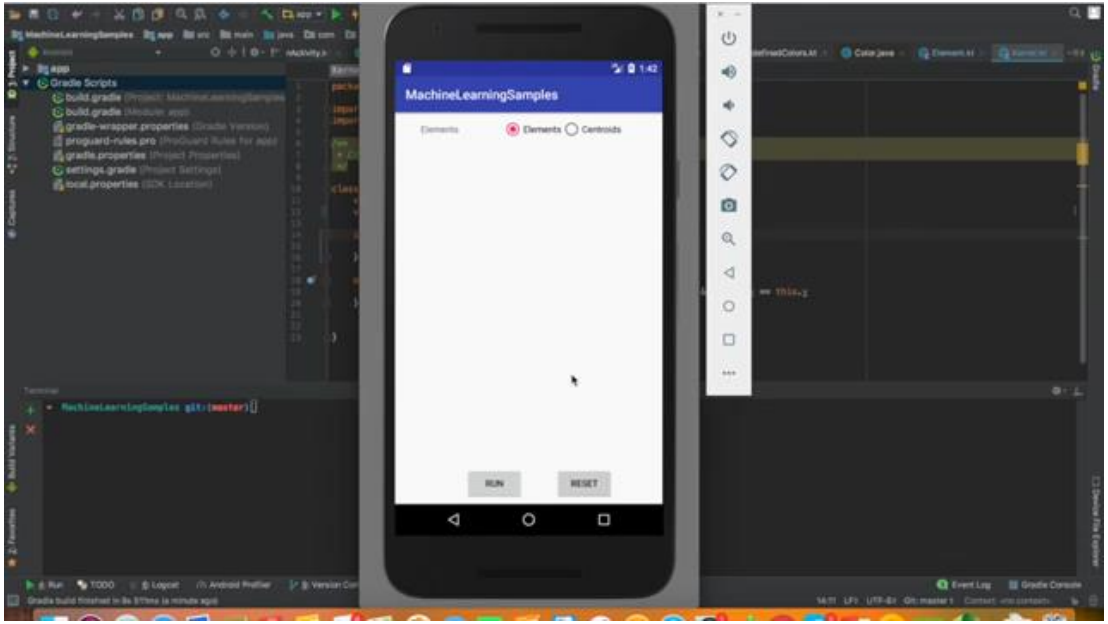
視覺化

- 視覺化排序演算法



視覺化

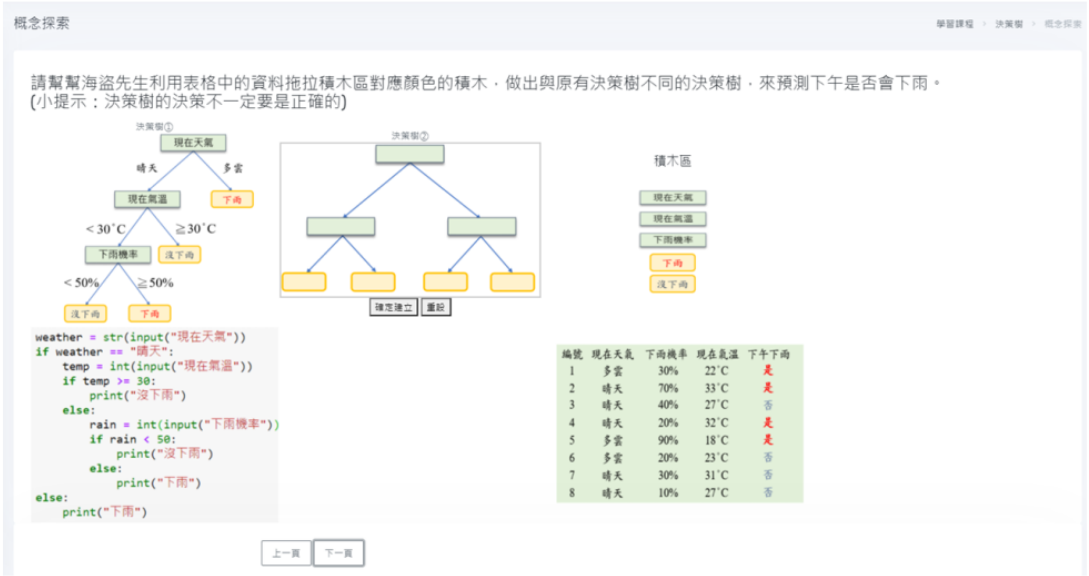
- 模擬



視覺化

■ 模擬

一、請操作平臺到下方的頁面，並於建立完決策樹後截圖至下方空白處。

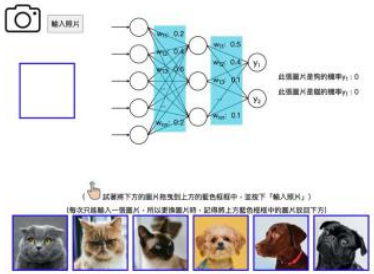


不能只提供視覺化工具，
重要的是思考的引導

二、承上題，你認為哪棵決策樹較好？

視覺化

C. 應用類神經網路



1. 請在資料搜集的頁面中，將圖片拖曳至訓練資料集並按下「資料搜集完成」，觀察頁面中呈現了什麼，並且記錄下來：

2. 請在資料搜集的頁面中，試著將「貓咪的圖片」拖曳到「狗的類別」中，按下「資料搜集完成」，觀察頁面中呈現了什麼，並且記錄下來：

3. 承上題，思考看看，如果將訓練資料集當中的類別標示錯誤，對於訓練類神經網路有什麼影響？

4. 請在訓練類神經網路的頁面中，按下「進行第一次迭代」且接續按下「讀取圖片」，觀察頁面中呈現了什麼，並且記錄下來：

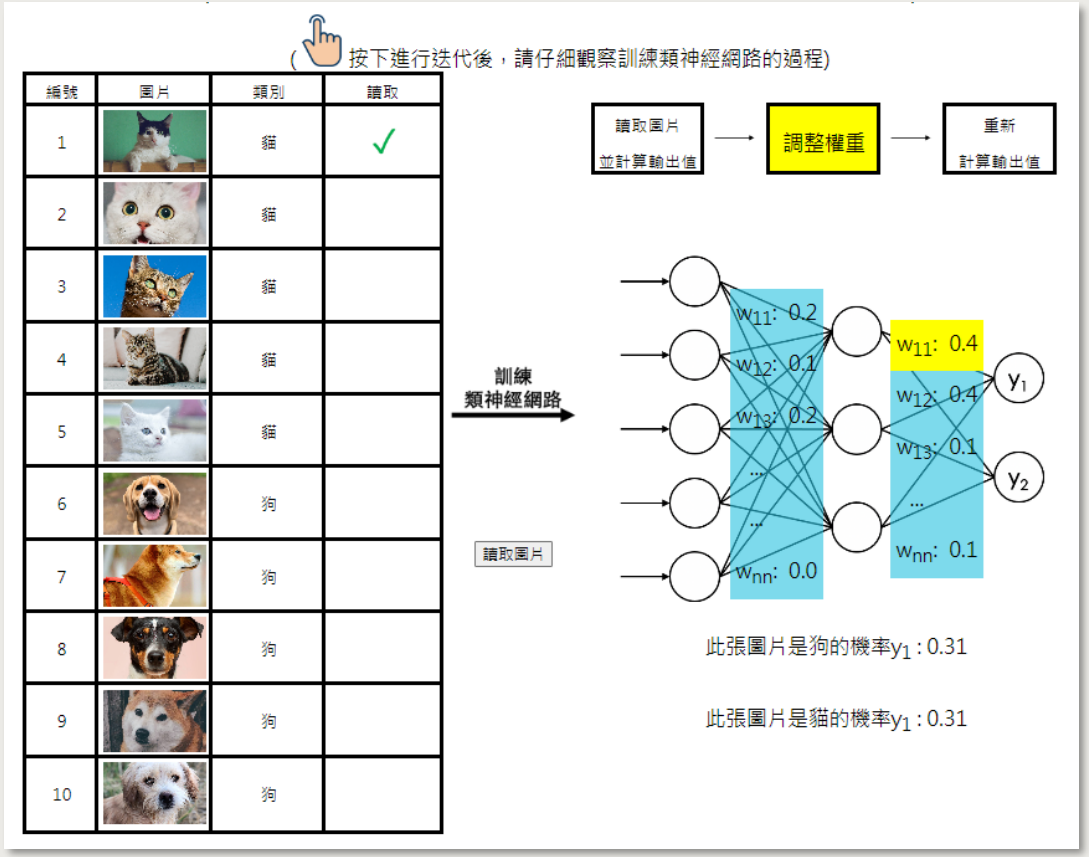
5. 請在訓練類神經網路的頁面中，按下「進行第一次迭代」且接續按下「讀取圖片」，觀察頁面中呈現了什麼，並且記錄下來：

6. 請在應用類神經網路的頁面中，將貓咪的圖片拖曳到左上角的方格中，並按下輸入照片，觀察頁面中呈現了什麼，並且記錄下來：

7. 承上題，思考看看，如果今天輸入了一個「很像貓咪的狗圖片」，類神經網路所輸出的機率值可能會有什麼狀況？

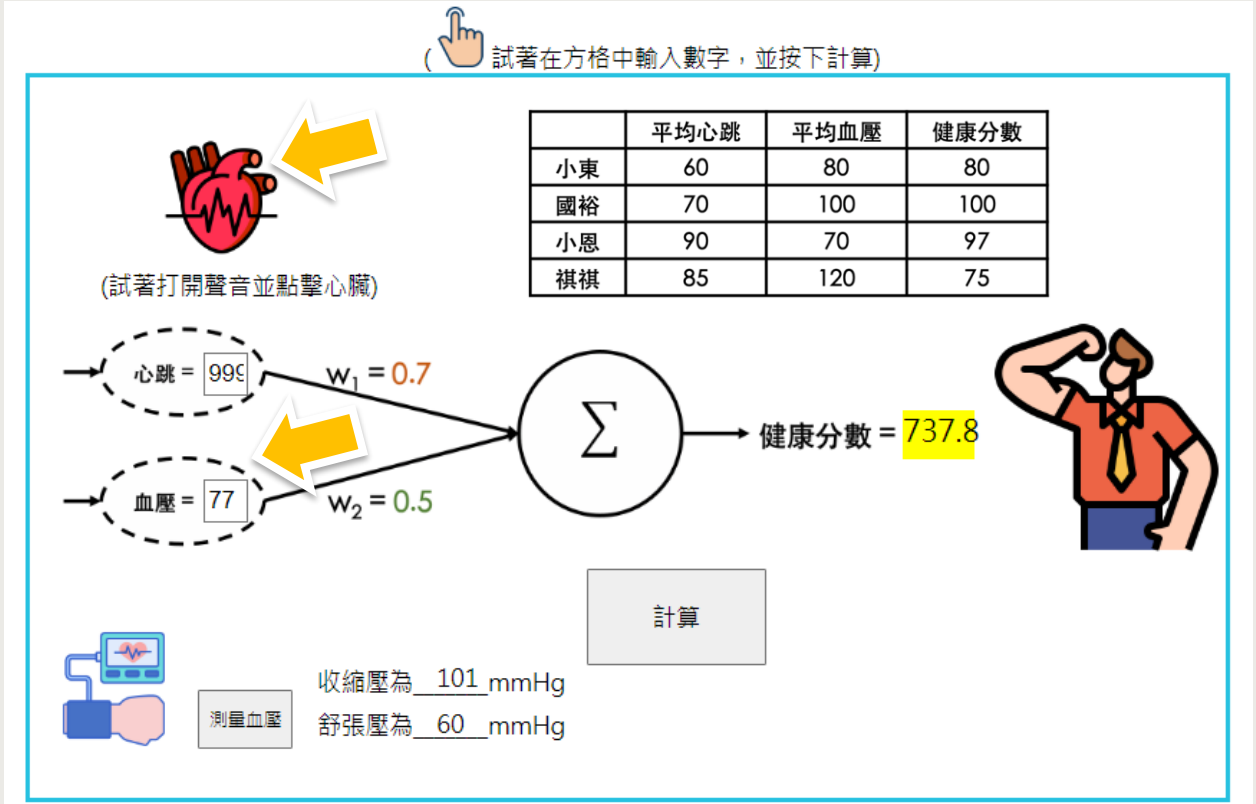
8. 請重新簡述，這三個頁面所學習到的相關概念：

不能只提供視覺化工具，重要的是思考的引導



以動畫模擬迭代

釐清概念



	平均心跳	平均血壓	健康分數
小東	60	80	80
國裕	70	100	100
小恩	90	70	97
祺祺	85	120	75

聽覺製造臨場感

手動調整參數

視覺化

運算思維導向程式設計視覺化輔助學習平台

條件

- 計算期末的總成績
- 計算百貨公司折扣
- 判別直角三角形
- 測謊機
- 電影售票機

回首頁

單元範例-例題5

程式視覺化 | 模擬執行

假日，美伢決定帶小新去看電影。來到電影院後，看著人工窗口大排長龍的隊伍，於是便決定去空無一人的電子售票機購票。
這台電子售票機的功能並不完善，每次只能販售一張電影票。
請同學試著寫出，能夠讓機器自動進行條件判斷的程式，亦即：消費者只要輸入年齡，電子售票機就會顯示相對應的票價。
開心電影院的票價： 60歲以上(敬老票)，票價為150元。 12歲到59歲(一般票)，票價為200元。 低於12歲，票價為170元。
(EX:輸入-10 輸出-票價為170)

程式碼

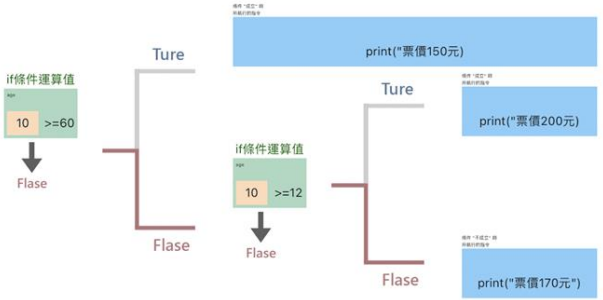
```
age =int(input("請輸入年齡："))
if (age>=60):
    print("票價150元")
elif(age>=12):
    print("票價200元")
else:
    print("票價170元")
```

程式解釋

程式語法

```
if 條件運算值1：
    條件1"成立"時所執行的指令
elif 條件運算值2：
    條件2"成立"時所執行的指令
else:
    條件1、2皆"不成立"時所執行的指令
```

變數記憶



執行結果

請輸入年齡： 10

票價170元

DFS

先看到的，後搜（找便利商店）

- 步驟一：
  - 從我家開始，（並標記為已走訪過）。
- 步驟二：
  - 相鄰我家的地點，(依逆時針方向)有博物館、加油站。
  - 選擇排在最後的加油站來出發。

先看到的，後搜（找便利商店）

- 步驟三：
  - 從加油站出發，並標記（已走過）。
- 步驟四：
  - 相鄰加油站，有四個地點。分別為醫院、我家（已走過）、主題樂園和科技公司。
  - 選擇排在最後的科技公司來出發。

科技公司
主題樂園
醫院
加油站
博物館

先看到的，後搜（找便利商店）

- 步驟五：
  - 從科技公司出發，並標記（已走過）。
- 步驟六：
  - 相鄰科技公司，有四個地點。分別為便利商店、加油站（已走過）、 大賣場和電影院。
  - 選擇排在最後的電影院。

？
科技公司
主題樂園
醫院
博物館

？該放什麼地點

先看到的，後搜（找便利商店）

- 步驟七：
  - 從電影院出發，並標記（已走過）。
- 步驟八：
  - 相鄰電影院的地點--無，而科技公司（已走過）。
  - 無路可走，只好循路，退回到科技公司。相鄰科技公司，未走訪的還有二處，便利商店和大賣場。
  - 選擇排在最後的大賣場出發。

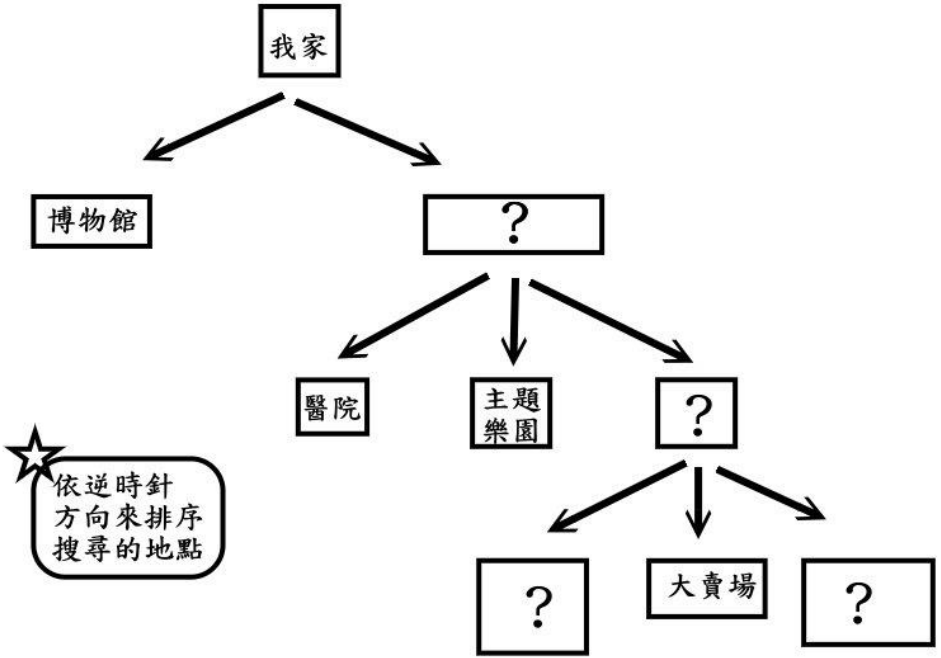
先看到的，後搜（找便利商店）

- 步驟九：
  - 從大賣場出發，並標記（已走過）。
- 步驟十：
  - 相鄰大賣場，只有科技公司（已走過）。
  - 無路可走，只好循路，退回到科技公司。
  - 相鄰科技公司，未走訪的有便利商店。
  - 選擇便利商店出發

先看到的，後搜（找便利商店）

- 步驟十一：
  - 從便利商店出發
  - 找到了

請問找到便利商店，經過哪些地點呢？



總結

1. 從起點出發，並設定已走過。
2. 持續走訪任一相鄰的地點(未走過)，若不是目標，設定已走過。若是目標，結束。反覆步驟2。
3. 若前面的相鄰地點都走過，則持續往後退，往其他相鄰地點(未走過) 。反覆步驟2。若都沒有符合目標，表示無此路徑存在。
4. 上述的方法為DFS演算法。

BFS

先看到，先搜尋（找便利商店）（依順時針方向）

- 步驟一：
  - 從我家開始。
- 步驟二：
  - 相鄰我家的地點
    有加油站和博物館。
    並把我家標記（已走過）。

先看到，先搜尋（找便利商店）（依順時針方向）

- 步驟三：
  - 從加油站開始。
- 步驟四：
  - 相鄰加油站的地點有科技公司、主題樂園、我家（已走過）、醫院。
  - 並把加油站標記（已走過）。

醫院
主題樂園
科技公司
博物館
加油站

先看到，先搜尋（找便利商店）（依順時針方向）

- 步驟五：
  - 從博物館開始。
- 步驟六：
  - 相鄰博物館的鄰居有學校、我家（已走過）、機場。
  - 並把博物館標記（已走過）。

機場
學校
醫院
主題樂園
科技公司
博物館

先看到，先搜尋（找便利商店）（依順時針方向）

- 步驟七：
  - 從科技公司開始。
- 步驟八：
  - 相鄰科技公司的地點有電影院、大賣場、加油站（已走過）、便利商店。
  - 並把科技公司標記（已走過）。

便利商店
大賣場
電影院
機場
學校
醫院
主題樂園
科技公司

先看到，先搜尋（找便利商店）（依順時針方向）

- 步驟九：
  - 從主題樂園開始。
- 步驟十：
  - 主題樂園無相鄰地點，略過。
  - 並把主題樂園標記（已走過）。

便利商店
大賣場
電影院
機場
學校
醫院
主題樂園

先看到，先搜尋（找便利商店）（依順時針方向）

- 步驟十一：
  - 從醫院開始。
- 步驟十二：
  - 醫院的鄰居有銀行、加油站（已走過）、學校。
  - 並把醫院標記（已走過）。

先看到，先搜尋（找便利商店）（依順時針方向）

- 步驟十三：
  - 從學校開始。
- 步驟十四：
  - 相鄰學校的地點有醫院（已走過）、博物館（已走過）、圖書館。
  - 並把學校標記（已走過）。

先看到，先搜尋（找便利商店）（依順時針方向）

- 步驟十五：
  - 從機場開始。
- 步驟十六：
  - 機場無鄰居，略過。
  - 並把機場標記（已走過）。

先看到，先搜尋（找便利商店）（依順時針方向）

- 步驟十七：
  - 從電影院開始。
- 步驟十八：
  - 電影院無鄰居，略過。
  - 並把電影院標記（已走過）。
- 步驟十九：
  - 從大賣場出發。
- 步驟二十：
  - 大賣場無鄰居，略過。
  - 並把大賣場標記（已走過）。

圖書館
學校
銀行
便利商店
大賣場
電影院
機場

先看到，先搜尋（找便利商店）（依順時針方向）

- 步驟二十一：
  - 從便利商店出發。
  - 找到囉。

圖書館
學校
銀行
便利商店

討論

- 設計一種視覺化的教學呈現，幫助學生理解複雜或抽象的概念或演算法

資訊科技教學策略-動手實作

動手實作

- Teaching Physics by doing experiments in lectures
  - Professor Walter Lewin, MIT
    - Potential Energy (勢能)

http://www.youtube.com/watch?v=CgqBg44azYk&feature=relmfu

44:00

- Simple Harmonic Oscillator (簡諧運動)

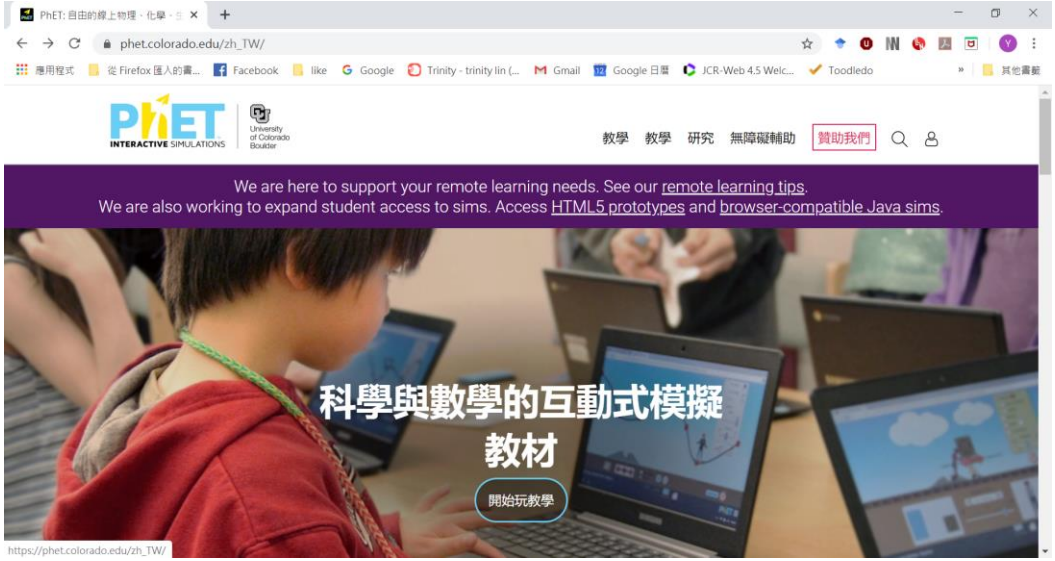
http://ocw.mit.edu/courses/physics/8-01-physics-i-classical-mechanics-fall-1999/video-lectures/lecture-10/

45:00

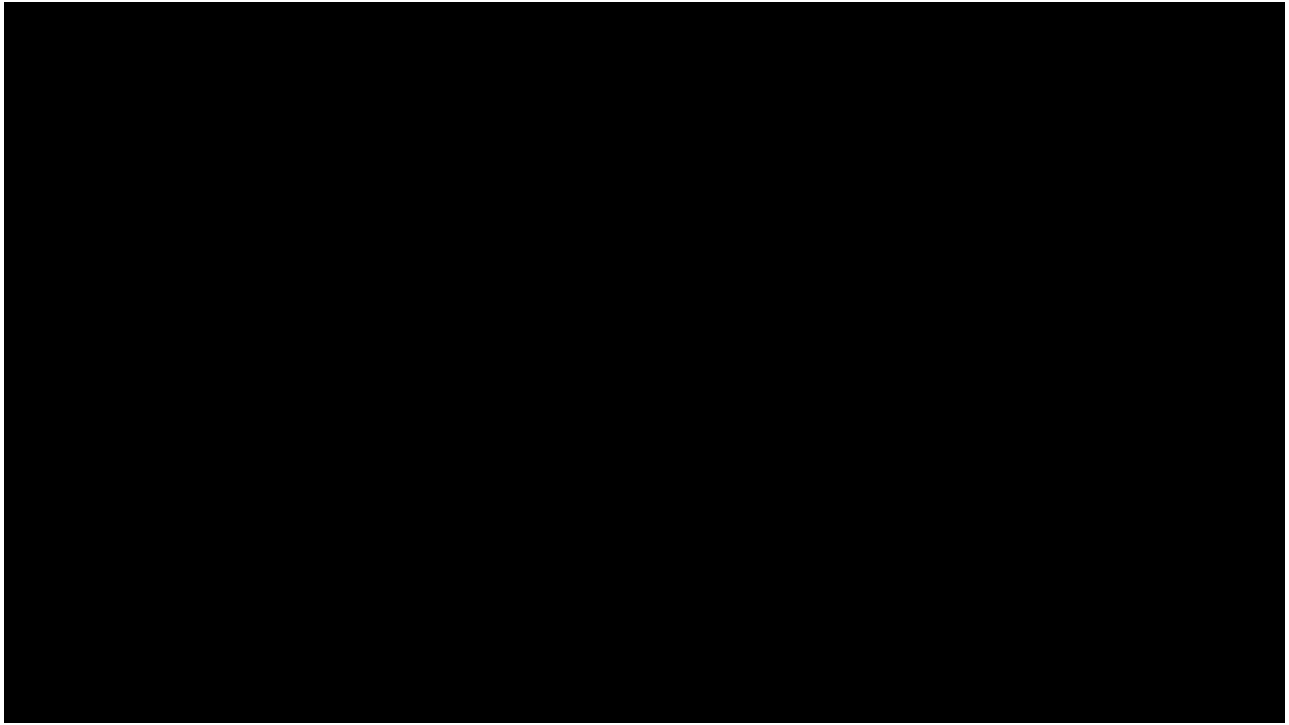
動手實作

■PhET 模擬教學

– https://phet.colorado.edu/

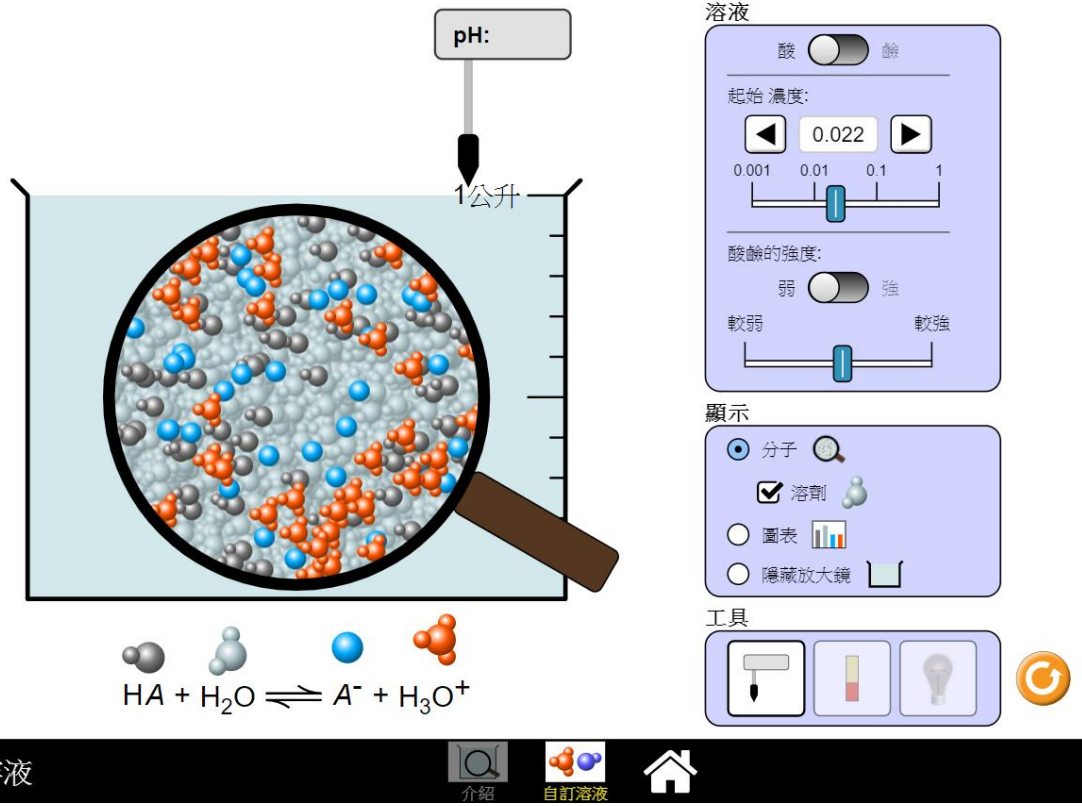


動手實作



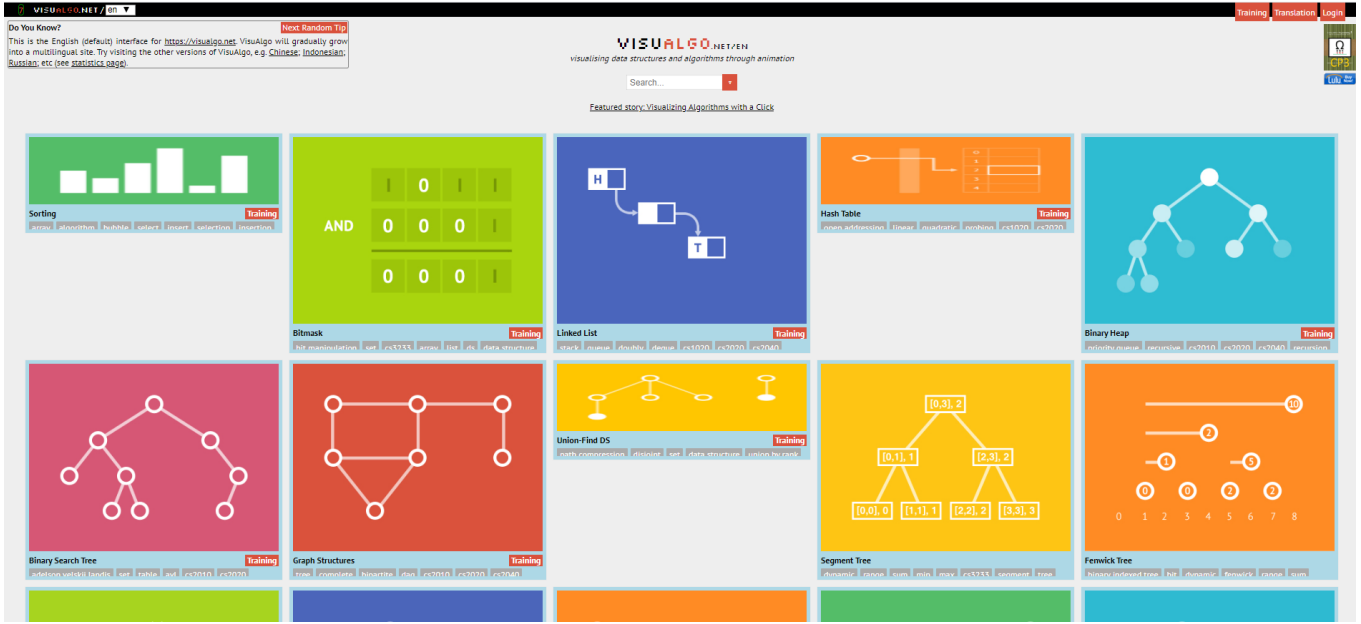
動手實作

■PhET 模擬教學



動手實作

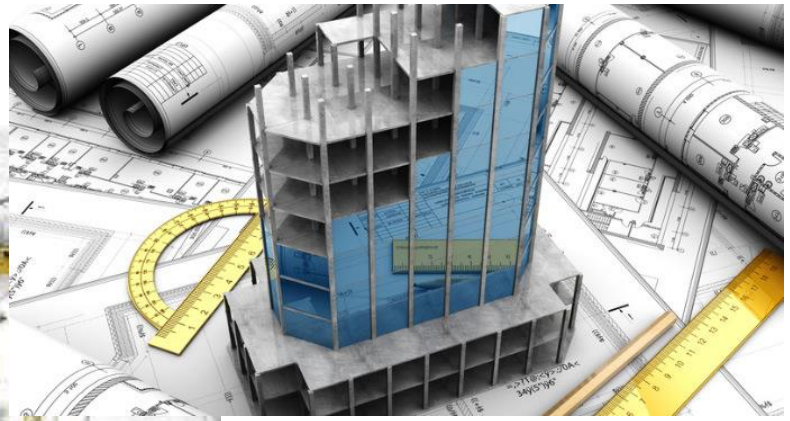
- VISUALGO
  - https://visualgo.net/



資訊科技教學策略-STEM/STEAM教學

STEM/STEAM教學

- 建築
  - 建築學
  - 美學
  - 物理學
  - 結構力學
  - 工程力學
  - 材料力學



STEM/STEAM教學

- 西北大學發展之CT-STEM教學
  - 美國國家科學基金會(National Science Foundation, NSF)與西北大學執行了一個Computational Thinking in STEM (CT-STEM)計畫，設計了一系列STEM的課程來教導高中生運算思維，同時也舉辦了許多教師工作坊，以進行CT-STEM教學之訓練，每個課程單元皆包含STEM四種領域的知識，以及運算思維元素
  - https://ct-stem.northwestern.edu/

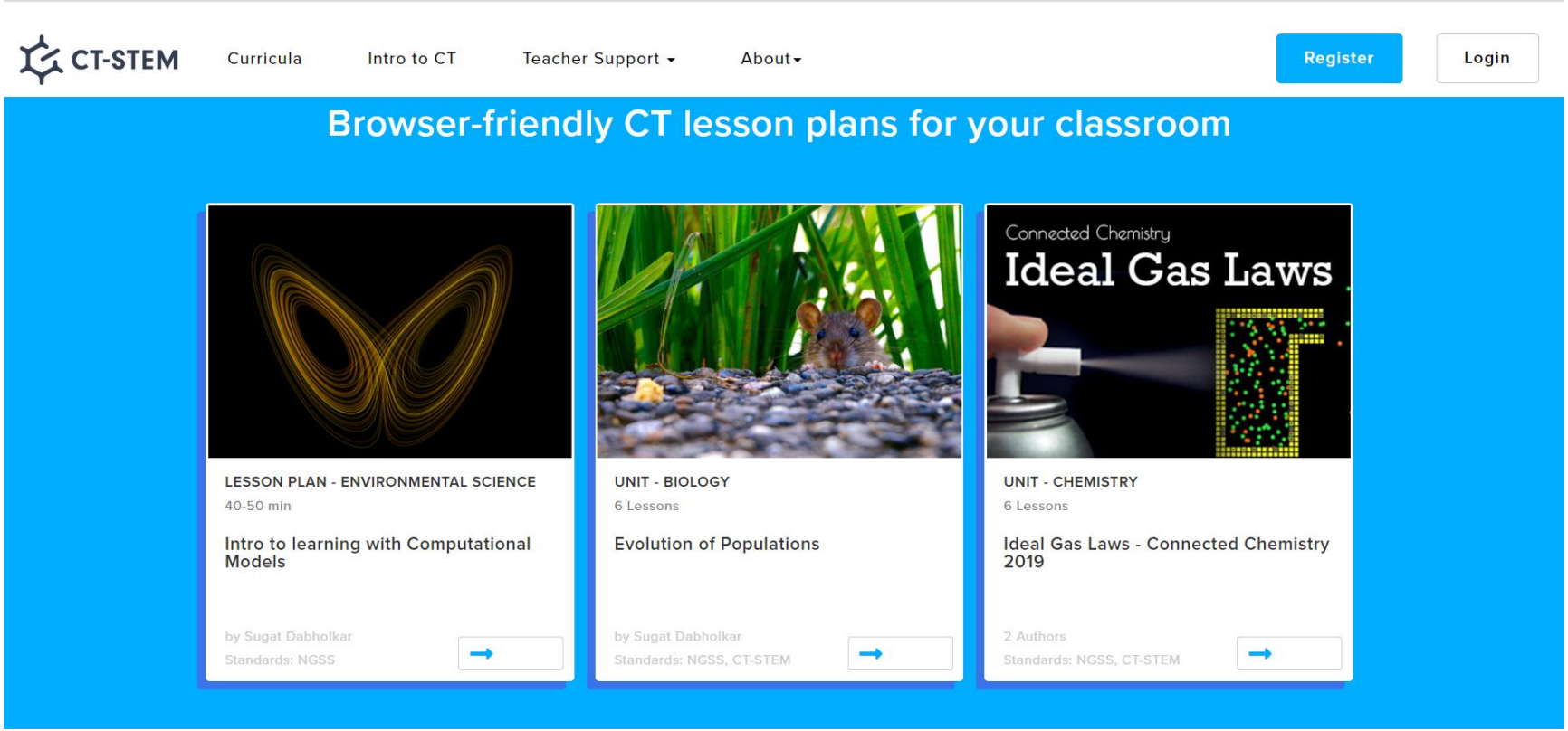
STEM/STEAM教學

■ 西北大學發展之CT-STEM教學範例

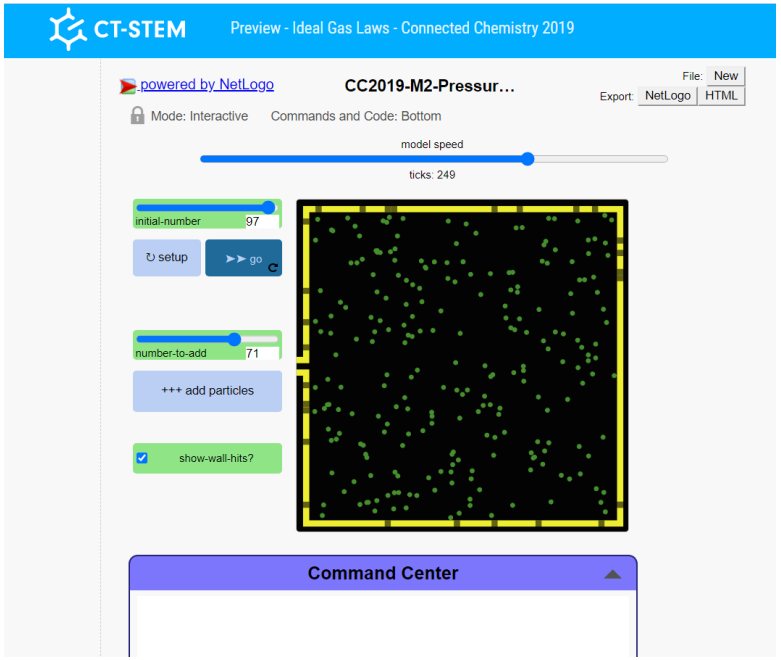
DA (Data Analysis), MS (Modeling & Simulation), CPS (Computational Problem Solving), ST (Systems thinking)

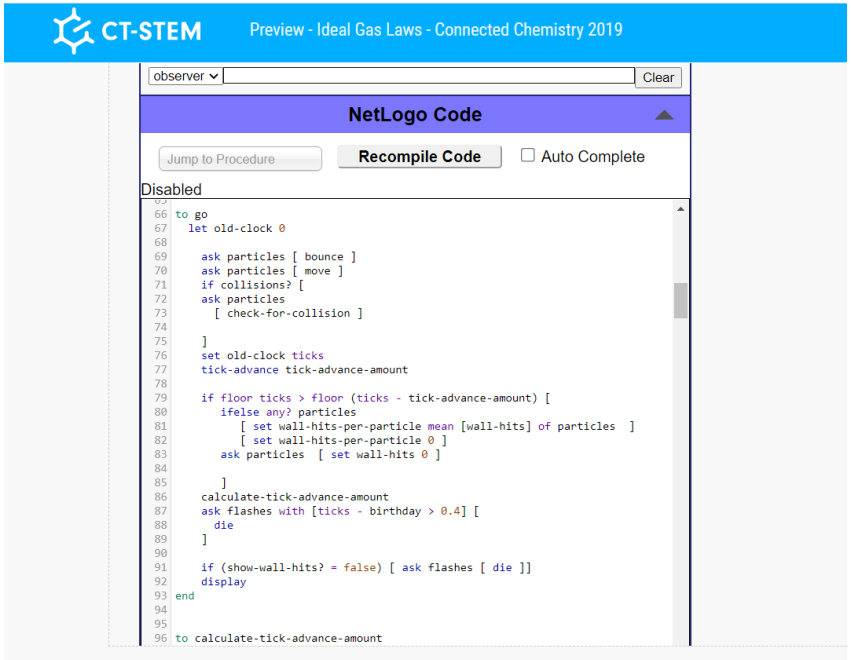
單元名稱 知識領域	The Physics of Angry Birds	Determination of Ohm's Law from Physical Data	Investigation of Resonance Using Computer Simulated Experiments	Modifying Code Introduction to Netlogo: A Newton's Law of Gravity Simulation
S	動量守恆、牛頓力學	歐姆定律、理解原子分子	駐波、諧音	牛頓力學
T	使用試算表，推估方程式	使用麵包板、電阻器	電腦	電腦
E	使用電腦模擬產生資料	使用試算表	使用JAVA模擬與設計	電腦模擬，Netlogo簡易coding
M	計算方程式	計算資料，推出歐姆定律	計算方程式	公式理解與計算
CT	DA, MS, CPS	DA, MS	DA, MS, ST	DA, MS, CPS

STEM/STEAM教學



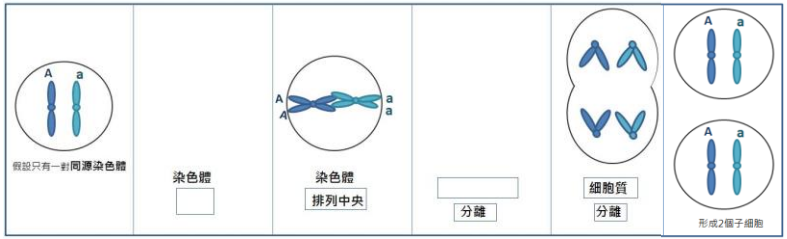
STEM/STEAM教學

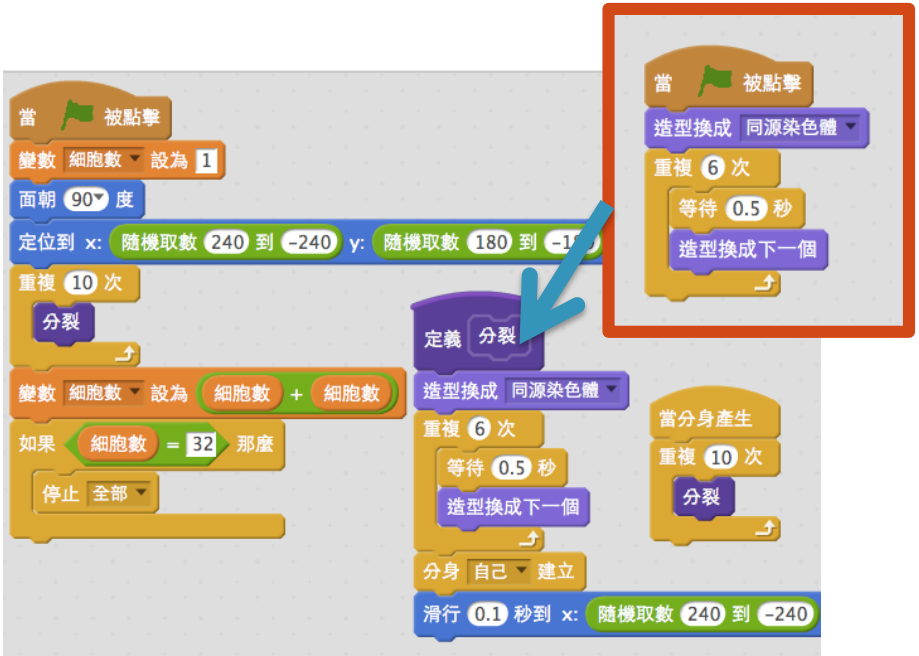




STEM/STEAM教學

- 細胞分裂





r work

data 換算方法

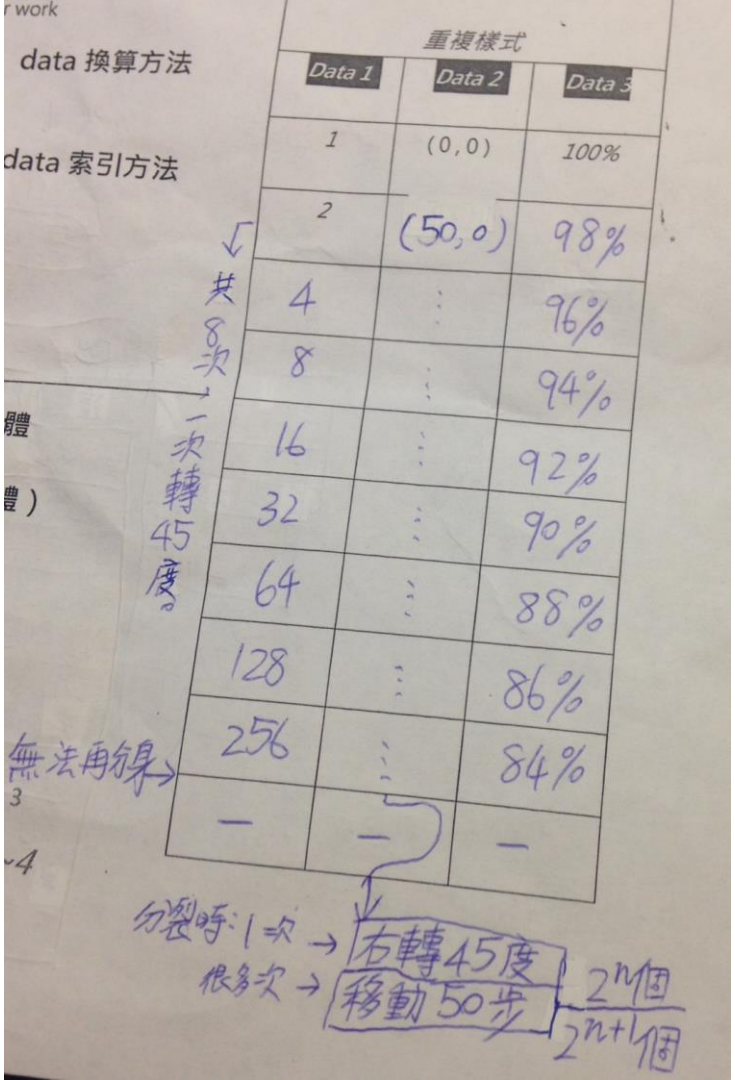
data 索引方法

體

體)

3

~4



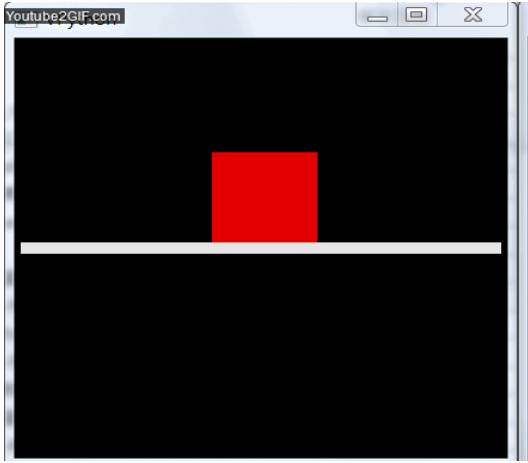
STEM/STEAM教學

- 模擬與建模

- **2-11:請觀察速率,時間,距離之間的關係（選擇題）**
  - (A) 相同距離的情況下，速率與時間成正比
  - (B) 相同距離的情況下，速率與時間成反比
  - 答案：_____
- **2-12:想一想，攀岩高手爬一趟的平均速率為多少？**
  - ✧ 平均速率=總距離（上山+下山）／總時間（上山時間+下山時間）
  - ◆ 平均速率=（３０＋３０）／（６＋３）＝６０／９　＝２０／３
- **2-13:請觀察平均速率的時間與距離**

速率(藍球)第_____小時，距離經過了 40.00 公里

速率(藍球)第 8 小時，距離經過了_____公里



while等速.py - D:\temp\while等速.py

File Edit Format Run Options Windows Help

```
#while等速
from visual import *

v = 0.05          #木塊速度 = 0.05 m/s
dt = 0.001        #畫面更新的時間間隔, 單位
t = 0             #模擬所經過的時間 , 單位為


floor = box(pos=(0, 0, 0), length=0.2, 
cube = box( pos=(0,(0.05+0.005)/2,0),le

while (1):
    while(cube.pos.x < 0.07):
        rate(1000)    #每秒1000次的頻率被
        t += dt       #t在模擬世界所經歷的
        #print t
        cube.pos.x += v*dt   #位移=速度*時
        #print cube.pos.x
    print t
```

```
>>>
上山 (紅球)第1小時,距離走了5公里
上山 (紅球)第2小時,距離走了10公里
上山 (紅球)第3小時,距離走了15公里
上山 (紅球)第4小時,距離走了20公里
上山 (紅球)第5小時,距離走了25公里
上山 (紅球)第6小時,距離走了30公里
--------------------------------
下山 (綠球)第1小時,距離走了10公里
下山 (綠球)第2小時,距離走了20公里
下山 (綠球)第3小時,距離走了30公里
--------------------------------
平均速率 (藍球)第1小時,距離走了6.67公里
平均速率 (藍球)第2小時,距離走了13.33公里
平均速率 (藍球)第3小時,距離走了20.00公里
平均速率 (藍球)第4小時,距離走了26.67公里
平均速率 (藍球)第5小時,距離走了33.33公里
平均速率 (藍球)第6小時,距離走了40.00公里
平均速率 (藍球)第7小時,距離走了46.67公里
平均速率 (藍球)第8小時,距離走了53.33公里
平均速率 (藍球)第9小時,距離走了60.00公里
>>>
```

教學活動

根據科學建模歷程(Sins et al., 2005b)與物理建模歷程(Ramaila, 2014)發展出適合本研究之STEM程式設計教學建模學習程序。學習者於學習活動階段與專題製作階段，經歷**「分析」、「演算法設計」、「程式化」與「解釋」**四個建模程序。

■ **題目 2**

有一架噴射機在跑道上，如果從靜止開始滑行 5 秒後，達到 180 公里/時的速度而起飛，則此飛機在跑道上滑行時，平均加速度量值為多少 公尺/秒 2 ，前進多少公尺。請用程式模擬噴射機在跑道上速度與位置。

✧ **小提示：**

◆ 平均加速度：單位時間內的速度變化量。

◆ 公式 1： $\bar{a} = \frac{\Delta v}{\Delta t} = \frac{v_2 - v_1}{t_2 - t_1}$

◆ 公式 2： $\triangle x = \frac{1}{2}(v_{end} + v_{start}) * t$

◆ 公式 3： $180(\text{km/h}) = \frac{180*1000}{60*60} = 50\ (\text{m/s})$

2-1:想一想，如何從空白的畫面，畫出題目要求的動畫？

2-2:想一想，要如何完成任務呢：

Q：需要知道哪些條件？

A：噴射機初始速度是靜止，為 0

噴射機在跑道上起飛前的速度是 50 m/s

時間從 0 開始經過 5 秒

2-3:根據以上的條件，我們可以先設定畫面與變數，噴射機初始位置假設在(-125.0,0.5,0.0)：

```
from visual import *
#變數設定-------------------------------------
v_start=0  #m/s  #初速度
v_end=50   #m/s  #末速度
t_start=0        #初時間
t_end=5          #末時間
dt = 0.001
#畫面設定-------------------------------------------------------------------
scene = display(width=2000, height=800,background=(0.5,0.5,0))       #設定畫面
floor = box(length=250, height=0.01, width=4, color=color.blue)      #跑道
ball = sphere(pos=(-125.0,0.5,0.0) , radius = 0.5,  color=color.red)#噴射機
```

While 單元加速度之學習單

分析

教學活動

While 單元加速度之學習單

演算法設計

2-4:已經知道噴射機的初始速度、在跑道上起飛前的速度、時間，請計算噴射機的加速度：

✧ 小提示：

(A) 公式 1： $\bar{a} = \frac{\Delta v}{\Delta t} = \frac{v_2 - v_1}{t_2 - t_1}$

(B) 公式 2： $\triangle x = \frac{1}{2}(v_{end} + v_{start}) * t$

```
#計算加速度與位置
a = (v_end - v_start)/(t_end-t_start) #加速度
print "噴射機的加速度是",a
x = (v_end+v_start)*t_end/2           #噴射機從靜止到起飛前的距離
print "噴射機從靜止到起飛前的距離",x
```

2-5:已經知道噴射機加速度與時間，想一想怎麼轉換程式碼顯示動畫？

因為 vpython 顯示動畫是 3D 空間，速度是有方向與大小的，在這邊先將速度轉換成 3D 向量。

```
#速度有方向與大小,在vpython中用向量表示
ball.v = vector(0.0, 0.0 , 0.0) #噴射機初始速度向量
```

2-6:已經知道噴射機加速度與時間，想一想如何用程式碼模擬噴射機起飛前的狀態：

當噴射機的位置小於等於噴射機起飛前的位置時

1. 噴射機每秒結束後都會有新的速度值
2. 噴射機每秒都會前進

```
#物體運動-------------------------------------------------------------------
while ball.pos.x <= x:                          #當噴射機的位置小於等於起飛位置時
    rate(1000)                                  #每秒動1000次
    ball.pos = ball.pos + ball.v * dt * 0.5     #噴射機每秒的距離 (V+Vo)*t/2
    ball.v.x = ball.v.x + a*dt                  #噴射機每秒速度 V=Vo+a*t
```

教學活動

While 單元加速度之學習單

程式化

```
# -*- coding: utf8 -*-

#飛機起飛
from visual import *

#初始速度
v_start=0

#終止（起飛那一瞬間）速度
v_end=50

#初始時間
t_start=0

#終止（起飛那一瞬間）時間
t_end=5

#設定間隔時間長短以便觀察
dt=0.01

#設定畫面
scene=display(width=2000, height=800, background=(0.5,0.5,0))

#以地板代表跑道
floor=box(length=250, height=0.01, width=4, color=color.blue)

#以球代表飛機
airplane=sphere(radius=0.5, color=color.red)

#設定飛機初始位置x,y,z都要設定
airplane.pos.x=0
airplane.pos.y=0.5
airplane.pos.z=0


#設定飛機初始速度
airplane.velocity = vector(0,0,0)


#利用公式計算加速度(末速-初速)/(末時間-初時間)
a=(v_end-v_start)/(t_end-t_start)
print "加速度",a

#利用公式計算靜止到起飛的距離(末速+初速)*所花時間*0.5
x=(v_end+v_start)*5*0.5
print "從靜止到飛行前距離",x


#利用while迴圈一步一步畫出運行的過程
#飛機準備從第0秒開始繪製這5秒的過程所以時間歸零
t=0
#當 飛機的x位置 還未超過 滑行距離 的前提下要執行迴圈
while airplane.pos.x < x: #記得while語法後面有一個冒號

    #把動作切的很細0.01秒一步一步請電腦畫上去
    print "現在計算",t,"秒到",t+dt,"秒的情況"

    #初速就是當下飛機的速度
    v0=airplane.velocity.x

    #末速就是預算0.01秒的情況   末速=初速+加速度*這段小小的時間
    v=v0+a*dt

    #計算這0.001秒的移動距離   (末速+初速)*所花0.01秒*0.5
    dx=(v0+v)*dt*0.5

    #把計算結果印出來看一下
    print "初速",v0,"末速",v,"移動距離",dx
    print "從原點到目前距離",airplane.pos.x
    print "----------------------------------"

    #真正的去改變飛機的現況
    #1.改變飛機的x座標(距離累加上去)
    airplane.pos.x=airplane.pos.x+dx

    #2.改變飛機的速度(初速再加上這次的增加)
    airplane.velocity.x=airplane.velocity.x+a*dt

    #準備下一回合時間累增0.01秒例如3.45秒進到3.46秒
    t=t+dt

    #為了畫面更新速度
    rate(10000)
```

教學活動

While 單元加速度之學習單

解釋與評估

2-7:請問此題目噴射機的平均加速度量值為＿＿＿公尺/秒 **2** ，前進＿＿＿公尺。

2-8:試試看 **print ball.pos**，觀察噴射機每秒的距離。

2-9:試試看 **print ball.v.x**，觀察噴射機每秒的速度。

2-10 我可以完成任務　我不能完成任務　（請圈起來）

STEM/STEAM教學

- 作業

電腦＋物理課後作業

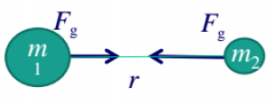
班級：_____座號：_____姓名：_____

```
#同步衛星與繞極衛星 (衛星的大小不符實際比例)
from visual import *

G = 6.7e-11 #重力常數
t = 0
dt = 0.0001

scene = display(width=800, height=800)#場景

earth = sphere(radius=6378.1, material=materials.BlueMarble)#地球半徑,球的材質
earth.mass  = 5.972e24#地球質量
mu = G*earth.mass#重力常數*地球質量

ge_satellites=[]#三個同步衛星
for i in range(0,3,1):
    ge_satellites.append(sphere(radius=1000, material=materials.rough))#增加同步衛星的半徑與材質
    ge_satellites[i].r = 42164#同步衛星與地球的距離
    ge_satellites[i].av = (mu/ge_satellites[i].r**3)**0.5

arrows=[]#繞極衛星
for i in range(0,3,1):
    arrows.append(arrow(shaftwidth=100 ))#增加繞極衛星軸寬度

PO_satellite = sphere(radius=1000, make_trail=True, retain=1000, material=materials.rough, color=(0.7,0.7,0.7))#繞極衛星的半徑顏色等資訊
PO_satellite.r = 6378.1+1500#繞極衛星與地球的距離
PO_satellite.av = (mu/PO_satellite.r**3)**0.5

earth.av = (mu/(ge_satellites[0].r)**3)**0.5

while True:
    rate(1000)
    t += dt#每秒
    for i in range(0,3,1):
        ge_satellites[i].x = ge_satellites[i].r*math.cos(-ge_satellites[i].av*t+i*120*pi/180)#同步衛星x軸
        ge_satellites[i].z = ge_satellites[i].r*math.sin(-ge_satellites[i].av*t+i*120*pi/180)#同步衛星z軸
        ge_satellites[i].pos = vector(ge_satellites[i].x, 0, ge_satellites[i].z)#同步衛星位置,y固定
        arrows[i].pos = ge_satellites[i].pos#同步衛星位置的值放置到繞極衛星位置
        arrows[i].axis = -0.85*ge_satellites[i].pos
    PO_satellite.y = PO_satellite.r*math.cos(-PO_satellite.av*t)#繞極衛星x軸
    PO_satellite.z = PO_satellite.r*math.sin(-PO_satellite.av*t)#繞極衛星z軸
```

1.請看物理課本，重力常數 G=_____

2.請看物理課本，萬有引力定律公式為：

_____



3. 請看物理課本，

m1、m2 為：_____

r 為：_____

Fg 為：_____

4. for i in range(0,3,1):

print i

i 會是：_____

5. 宇宙間的一切物體都是互相吸引的，兩個物體間的引力大小，跟他們的**質量的乘積**成_____比，跟他們的_____成反比。

6. 請觀察並寫下動畫中，同步衛星與繞極衛星與地球之間的關係。例如：垂直／水平繞著地球轉等，多描述你的觀察。

教學進度

週次	控制組	實驗組	課後練習	物理教師 授課進度
1	期初測驗(程式設計)			1-1 物理學簡介 1-2 物理量的單位 2-1 原子組成物質
2	期初測驗(物理+態度)			2-2 原子與原子核的組成 3-1 物體運動的軌跡
3	講述變數觀念與練習變數問題，學習程式設計變數的概念。	利用畫圖的問題，學會程式設計變數概念和能夠解決題目畫圖的問題。		3-1 物體運動的軌跡 3-2 牛頓運動定律

教學進度

週次	控制組	實驗組	課後練習	物理教師 授課進度
4	講述判斷式if else觀念與練習相關問題，學習程式設計判斷式if else的概念	利用物理位移問題，學會程式設計判斷式 if else 的概念和能夠畫圖顯示移動的位置。	虎克定律	3-2 牛頓運動定律
5	講述for迴圈觀念與練習相關問題，學習程式設計for迴圈的概念。	利用物理平均速率問題，學會程式設計 for 迴圈的概念和能夠藉由動畫了解平均速率的概念與每小時的位置。	萬有引力	3-3 克卜勒行星運動定律 4-1 重力
6			磁力線	4-2 電力與磁力 4-3 強力與弱力:自然界的基本作用力 5-1 電流的磁效應

教學進度

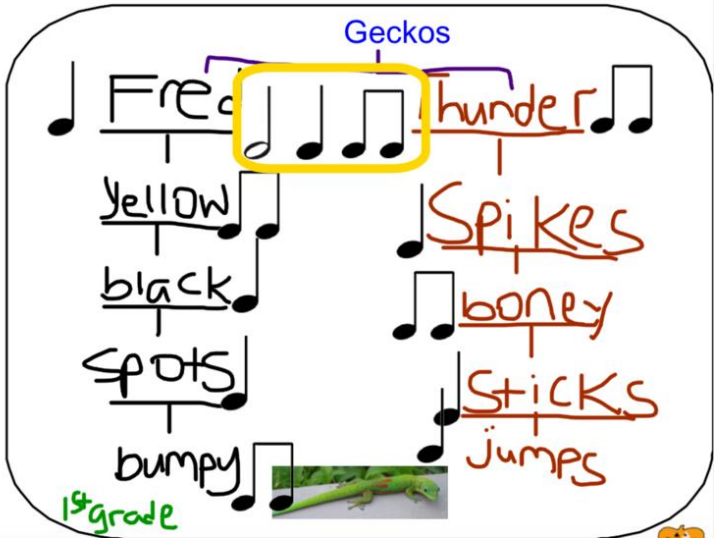
週次	控制組	實驗組	課後練習	物理教師 授課進度
7	講述while迴圈觀念與練習相關問題，學習程式設計while迴圈的概念。	利用畫圖讓物體位置移動，學會程式設計 while 迴圈的概念和能夠藉由動畫了解xyz 軸空間的概念。	平面波的反射、折射	5-2 電磁感應 6-1 波的性質
8		利用等速運動與等加速度運動問題，學會程式設計 while 迴圈的概念和能夠藉由動畫模擬等速與等加速度概念。	水波的干涉	6-2 光與電磁波 6-3 光
9	期中測驗(程式設計)			
10	期中測驗(物理+態度) 專題說明與分組			7-1 能量的形式

教學進度

週次	控制組	實驗組	課後練習	物理教師授課進度
11	程式設計專題製作	程式設計與物理專題製作		7-2 能量間的轉換與能量守恆
12				7-3 核能 7-4 能量的有效利用與節約
13				8-1 光電效應
14	專題發表			8-2 波力二象性
15	期末測驗(程式設計)			8-3 原子光譜
16	期末測驗(物理+態度)			9-1 星體觀測及哈伯定律 9-2 原子光譜
17	訪談			

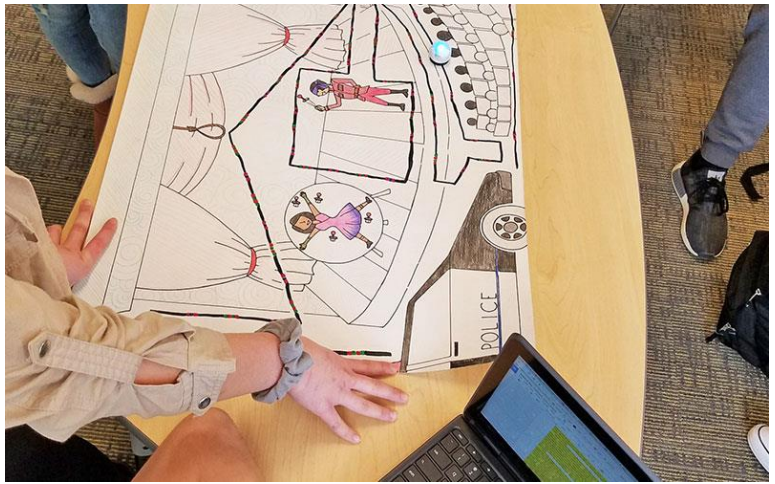
STEM/STEAM教學

- 音樂科技
  - http://amymburns.com/
  - http://amymburns.com/blogcc
  - 一年級學生在科學課中觀察兩隻壁虎，名叫弗雷德和雷霆，並為牠們作曲。學生必須建立一張思維圖（樹狀圖），顯示了每隻壁虎的特徵，接著選擇與特徵匹配的節奏作曲，並在課堂用樂器演奏。
- https://www.youtube.com/watch?v=ViQawYvAvnI



STEAM

- 用機器人Ozobot敘說謀殺之謎的故事
  - 程式設計與寫作是很類似的
  - 匹茲堡蒙圖爾高中十年級英語文學老師Gina Ligouri要求學生創作謀殺之謎故事，並需具有清晰的情節、人物、動機和結果。 學生們編寫好故事後，須製作視覺故事板，並使用 Ozobots 在板上移動並與學生分享他們的故事。



https://ozobot.com/blog/creator-of-the-month-ms-ligouris-murder-mystery-stories

討論

- 用今天課程提到的教學設計方法，設計一個教學活動，可以教學生量化與取樣的概念，要包含：
  - 學習內容
  - 學習表現
  - 教學目標
  - 簡單的教學活動描述