

國中資訊教育教案

教案名稱：Let's play Python with Tobi

教學設計者：蔡淑君

一、雙向細目表(以融入領綱學習重點及國中資訊教育課程發展參考說明之學習重點，規劃與撰寫)

學習內容 學習表現	資 T-IV-2 資訊科技應用專題。
運 t-IV-4 能應用運算思維解析問題。	<ol style="list-style-type: none"> 1. 學生能分析 Python 程式語言的特性與優勢。 2. 學生能操作 PictoBlox Python 編輯介面。 3. 學生能運用 print() 函式，寫出簡單的 Python 文字輸出程式。 4. 學生能運用 PictoBlox 的 sprite.say() 函式讓角色說出指定文字。 5. 學生能運用 input() 函式，撰寫 Python 程式以接收使用者輸入。 6. 學生能理解變數 (Variable) 在程式中的功能，並能正確建立變數來儲存資料。 7. 學生能運用字串組合技術，使用 + 號連結變數與文字串。 8. 學生能分析並撰寫「單向選擇結構 (if)」，根據條件判斷執行特定程式區塊。 9. 學生能運用「雙向選擇結構 (if-else)」，處理條件成立與不成立時的不同反應。 10. 學生能正確使用 Python 的語法規範，包含半形冒號：以及正確的程式縮排。 11. 學生能切換 PictoBlox 中的角色造型與舞台背景，以視覺化呈現程式狀態。

二、教案概述

科目/領域別	資訊科技/科技
教學節數	4 節課
實施年級	9 年級
教學設備	電腦
專題摘要	本教學設計的核心理念在於「無縫接軌與跨域整合」。考量到九年級學生已具備 Scratch 的積木式程式設計經驗，以 PictoBlox 作為教學工具，其與 Scratch 高度相似的介面能有效降低學生從圖形化編程過渡到文字型編程 (Python) 的門檻。這種「由簡入繁，以舊學新」的策略，使學生能更

	<p>專注於 Python 的語法邏輯，而非學習新工具的複雜操作，從而建立學習程式語言的自信心。</p>	
<p>先備知識</p>	<p>學科準備度 Readiness of Domain/Subject</p> <ol style="list-style-type: none"> 輸入與輸出基礎：學生已學過輸入與輸出的概念，並能操作 Scratch 的「詢問」與「詢問的答案」積木。 角色與舞台互動：學生已學過如何使用 Scratch 的「說」積木，以及更換背景與角色造型。 變數概念：學生已具備 Scratch「變數」建立與存取的基礎概念（理解變數作為資料容器的性質）。 關係運算：學生已學過如何使用 Scratch 的「關係運算子」積木（如：等於、大於、小於），具備比較兩數值的基礎邏輯。 選擇結構：學生已學過 Scratch 的「如果...那麼...」以及「如果...那麼...否則...」積木，具備單向與雙向選擇的邏輯思維。 字串處理：學生已具備 Scratch「字串組合」積木的使用經驗，能理解將不同文字片段串聯起來的概念。 <p>英語準備度 Readiness of English</p> <ol style="list-style-type: none"> 句型結構：學生已學過帶有助動詞 would 的特殊疑問句（如：What code would you use?）。 動詞用法：學生已學過動詞 help 的特殊用法。 進階語法：學生已學過間接問句，這能幫助他們理解 input() 函式中的引導問題邏輯。 	
<p>總綱 核心素養</p>	<p>A 自主行動</p> <p>A1 身心素質與自我精進</p> <p>A2 系統思考與解決問題</p> <p>B 溝通互動</p> <p>B1 符號運用與溝通表達</p>	
<p>中／英文 使用時機 Timing for Using Chinese/ English</p>	<p>教師 Teacher</p> <p>英文</p> <ol style="list-style-type: none"> 課堂常規管理：打招呼、提醒坐下開機、下課關機與擺放滑鼠指令。 程式關鍵字發音與應用：引導學生唸出 if, else, print 及各種水果單字（如 pineapple）。 口頭正向回饋：給予學生鼓勵（如：Brilliant! Perfect! Excellent!）。 簡單操作指令：如 Look at the slide, Go back to page 28, Turn to page 29。 	<p>學生 Students</p> <p>英文</p> <ol style="list-style-type: none"> 和老師打招呼。 尋求協助：使用 Teacher, help me please.。 回報進度：小組完成任務時說 "Group X is finished!"。 朗讀程式：嘗試唸出 if, else, print 及水果單字。 <p>中文</p> <ol style="list-style-type: none"> 困難點提問 找不到 PictoBlox 的功能。 同儕討論

	中文 1. 資訊知識架構說明 2. 詳細任務說明	
教學方法/策略	示範法、講述法、實作法	
評量方法	口語評量、實作評量、檔案評量(蒐集方式：採取用 Google Classroom，讓學生共編自己的作業簡報)	

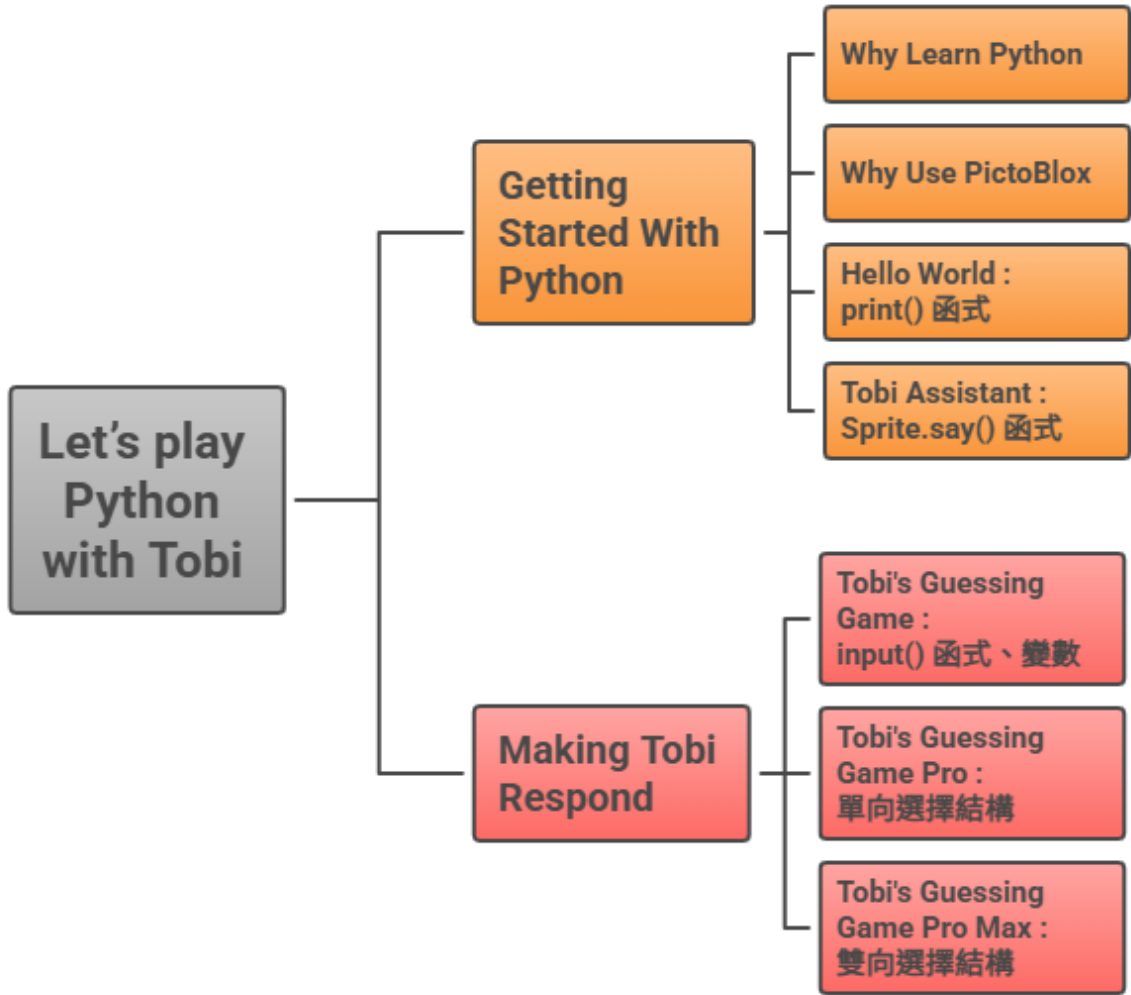
三、 評量方式

學習目標	超乎預期 (Exceeds)	符合期望 (Meets)	繼續努力 (Needs Improvement)
1. 分析 Python 特性與優勢	能流暢且完整地闡述 Python 特性，並提出獨到見解或舉例說明。	能清晰地陳述 Python 程式語言的主要特性與優勢。	未能完整陳述 Python 程式語言的特性與優勢。
2. 操作 PictoBlox 編輯介面	除了能成功開啟並進入介面，還能獨立解決操作時遇到的技術問題。	能成功開啟 PictoBlox 並順利進入 Python 編輯介面。	尚未能成功進入 Python 編輯介面進行操作。
3. 運用 print() 函式	能結合變數或其他基本語法，寫出多樣化或具互動性的文字輸出。	能運用 print() 函式正確寫出指定的文字輸出程式碼。	未能運用 print() 函式寫出正確的文字輸出程式。
4. 運用 sprite.say() 函式	能結合其他函式（如動作、切換造型），完成具備多樣化效果的程式。	能正確運用 sprite.say()，讓角色說出指定且完整的文字。	未能運用 sprite.say() 讓角色成功說出指定文字。

學習目標	超乎預期 (Exceeds)	符合期望 (Meets)	繼續努力 (Needs Improvement)
5. 運用 input() 函式	能主動協助同學 debug 關於 input() 函式在英語句型或語法上的錯誤。	能運用 input() 撰寫程式，準確呈現所要求的英語引導問題。	未能運用 input() 函式或無法撰寫出正確接收輸入的程式。
6. 理解並建立變數	能建立多個具意義名稱的變數，並熟練地在多輪問答中進行存取。	能正確建立變數來儲存使用者內容，並搭配輸出函式呈現複合字串。	未能運用正確的賦值語法建立變數，導致程式無法儲存內容。
7. 運用字串組合 (+)	能純熟運用 + 連結多個變數與字串，且注意字串間的空格與排版。	能正確使用 + 號連結固定字串與變數，解決 sprite.say() 報錯問題。	未能正確使用 + 號，或在使用字串組合時出現語法錯誤。
8. 單向選擇結構 (if)	能獨立根據流程圖邏輯寫出 if 判斷式，並能解釋運算子 == 的意義。	能正確撰寫 if 結構，使程式在條件成立時執行指定動作。	判斷邏輯不正確，或無法正確寫出 if 條件判斷式。
9. 雙向選擇結構 (if-else)	能靈活運用 if-else 處理多層邏輯，並能自行設計不同的答錯反應。	能整合 if 與 else 語法，正確處理「成立」與「不成立」的反應。	無法理解 else 的邏輯位置，或在雙向判斷上出現邏輯錯誤。
10. 語法規範 (冒號、縮排)	能精確使用半形符號，並能主動根據報錯紅線修正自己或同儕的縮排。	撰寫判斷式時能正確加上半形冒號並維持一致的縮排格式。	經常漏掉冒號、使用全形符號或因縮排錯誤導致程式無法執行。

學習目標	超乎預期 (Exceeds)	符合期望 (Meets)	繼續努力 (Needs Improvement)
11. 切換造型與背景	能根據不同的判斷結果（對/錯）設計具創意且準確的視覺回饋。	能在程式中運用指令，在指定時機切換角色造型與舞台背景。	無法將造型或背景切換指令正確放入邏輯判斷區塊中。

四、課程設計架構圖



Made with  Napkin

五、 教學活動步驟

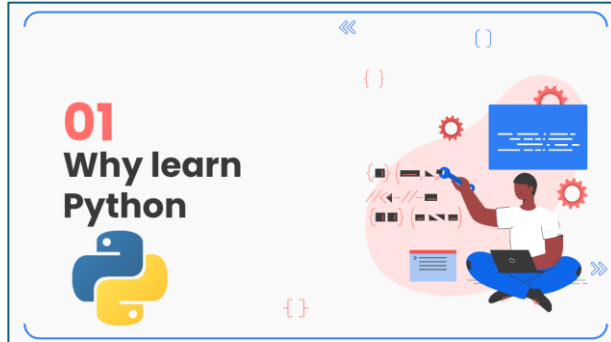
節次	教學流程 Teaching Procedures	時間 Time
第一節	<p style="text-align: center;">準備階段 Preparation stage</p> <p>T: Good morning/afternoon, everyone! 準備好開始今天的程式設計探索了嗎? Are you ready? Let's start!</p> <div data-bbox="311 616 976 987" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">  </div> <p>T: Look at the screen. 這是我們之前用過的 Scratch, NKNUBLOCK, 和 MIT App Inventor。大家已經有用block-based programming language 寫程式的經驗了。</p> <div data-bbox="311 1142 976 1514" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">  </div> <p>T: 接下來我們要進階到學習text-based programming language。雖然積木變成了文字，但核心邏輯是一樣的。Don't worry, it's simpler than you think!</p> <div data-bbox="311 1635 976 2007" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">  </div>	2 mins

發展階段 Development stage

一、 Why learn Python

1. 影片導入：Python 的威力

(1)T: 我們這次要學習的是名為Python的text-based programming language。請大家先看一段介紹影片。

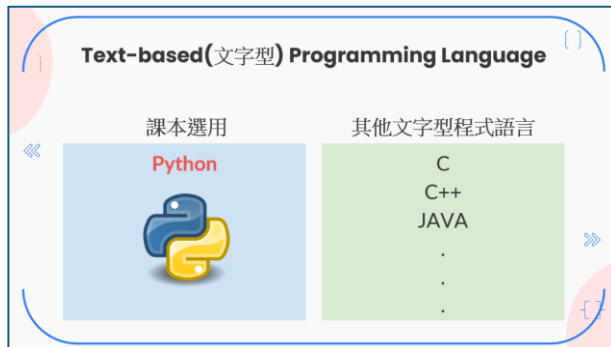


(2)教師播放影片 Getting Started with Python for Kids | Python with PictoBlox 0:26~1:00 打開字幕設定自動轉成繁體中文
(<https://www.youtube.com/watch?v=IKsNzvPUr8g>)



2. 核心優勢：簡單

(1)T: Text-based programming language有很多種，大家可能有聽過C、C++、Java，課本選擇Python讓大家學習，Why? 剛才的影片中有講到，對初學者來說，Python是最容易上手的Text-based programming language！簡單同時功能非常強大。



(2)T: 我們看這段 Syntax (語法) 的對比。假設我們要從 'Computer' 抓出前三個字母。其他語言要寫一長串，而Python is the simplest!

4 mins

Why do we learn Python?

It has **simple** syntax(語法).

Computer

語言 (Language)	程式語法 (Syntax)
C#	<code>string sub = "Computer".Substring(0, 3);</code>
JavaScript	<code>let sub = "Computer".slice(0, 3);</code>
Python	<code>sub = "Computer"[:3]</code>

3. 核心優勢：強大

T: 了解了How simple Python is, 接著我們要來看為什麼 Python 很強大。大家都有玩過LEGO blocks吧?



T: 我們可以想成Python擁有the most powerful LEGO baseplate。Look at the picture. 平常寫程式如果都要從零開始，就像要自己挖礦、打鐵、造零件去蓋房子，那太累了！但在 Python 世界，專家已經幫我們把零件都做好了。這些專業零件包，我們稱為Libraries (函式庫)。裝上數據Library，Python 就變成了NASA的軌道計算機。裝上推薦Library，Python就變成Netflix和Youtube的影片導遊。裝上AI Library，Python就變成ChatGPT的大腦。

Why do we learn Python?



4. 快問快答

(1)T: Now, it's Quiz Time! Q1: Python code is short and easy. Is Python simple or complex?

Quiz Time

Q1: Python code is short and easy.
Is Python **simple** or **complex**?

Ans: It is _____.

S: It is simple.

T: Correct! It is simple and easy to read.

(2)T: Q2: The experts made many 'Lego parts' for Python. What are they called?

Quiz Time
專家們

Q2: The experts made many **Lego parts** for Python.
What are **they** called?



Ans: They are _____.

S: They are Libraries.

T: Perfect! They are Libraries. With them, we can build anything!

二、 Why use PictoBlox?

- T:** 好，現在我們知道 Python 很棒了！但是，要怎麼寫 Python 程式呢？就像畫畫需要畫筆和畫布，寫程式也需要工具。編寫 Python 程式的工具有非常多種。



02
Why use PictoBlox

- T:** 例如課本介紹的Python官方的IDLE編輯器，會需要下載安裝到電腦裡。

5 mins



T:或是Google提供的Colab這種線上編輯器。不過，你們有沒有注意到，這些工具的介面，看起來都像我們平常在電腦裡用的「記事本」？如果一下子從Scratch那種充滿圖案和積木的介面，突然跳到只有文字的記事本，可能會覺得有點不習慣，需要適應一段時間。



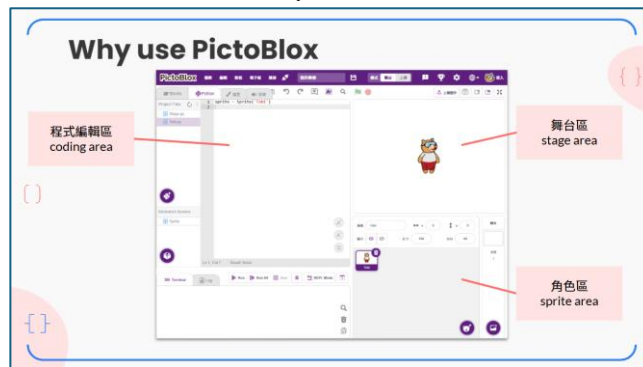
3. **T:** 這時候，PictoBlox 就派上用場了！它就像是為大家量身打造的橋樑。PictoBlox 的介面長什麼樣子呢？讓我們一起先看一小段影片來認識它！



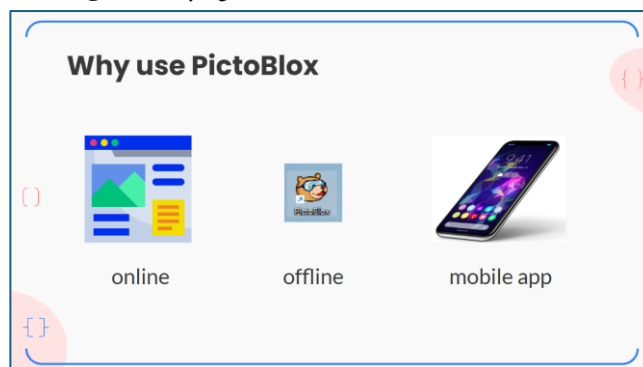
教師播放影片 Getting Started with Python for Kids | Python with PictoBlox 1:00~1:30 打開字幕設定自動轉成繁體中文 (<https://www.youtube.com/watch?v=IKsNzvPUr8g>)

4. **T:**從剛才的影片中，大家有沒有發現，PictoBlox 的介面是不是跟各位之前使用的 Scratch 和 NKNUBLOCK 長得很像呢？沒錯！它的右上方一樣有舞台區 (Stage Area)，右下方是角色區 (Sprite

Area)，而左邊則是我們寫程式的程式編輯區 (Coding Area)。這個熟悉的設計，就是為了讓你們能夠更輕鬆地，從積木式的思維無縫接軌到文字型的 Python 程式碼！



5. **T:** 除此之外，PictoBlox 還有許多厲害的擴充功能，可以讓我們未來不只寫程式，還能玩轉 AI、控制機器人，做出更多有趣的應用。而且，PictoBlox 同時提供線上版(Online)和離線版(Offline)，甚至還有手機平板版本(Mobile App)的程式編輯器，讓我們能夠更方便、隨時隨地編寫程式。所以，同學們在家如果想用手機寫程式，也完全沒問題喔！

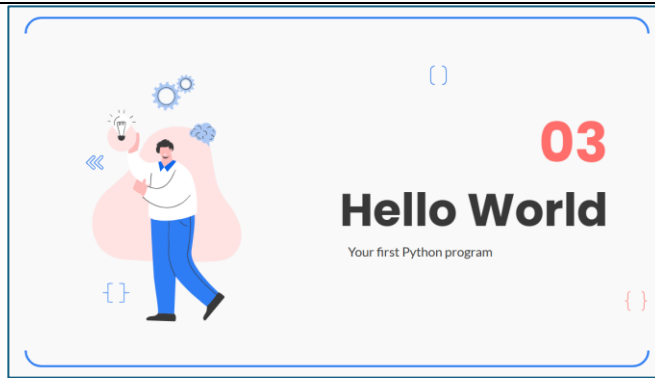


三、 Hello World

1. **T:** 剛剛我們認識了 Python 這個強大的程式語言，也了解了為什麼要用 PictoBlox 這個超棒的工具來寫程式。現在，老師已經迫不及待要帶大家親手寫出你們的第一個 Python 程式了！在文字型的程式語言世界裡，有一個非常可愛、幾乎是「不成文規定」的習慣，那就是當我們開始學習一個新的程式語言時，寫的第一個程式內容，通常會是印出「Hello World!」這句話。這個傳統其實有很有趣的歷史淵源喔，有興趣的同學可以當個小偵探，自己上網查查它背後的故事！

T: 今天，我們就要跟上這個有趣的傳統，讓大家親手寫出你的第一支 Python 程式，讓它對這個世界說聲「Hello!」。準備好了嗎？Are you ready? Let's give it a try! 讓我們一起動手試試看吧！

13 mins



2. **T:** 要印出**Hello World!** 這句話，要用到Python的**print** 這個function，function在Scratch稱為函式，**print**就是列印的意思，我們來看一下關於**print()** function的一些規定。首先，**print()** function是Python的內建函式，所以**print**一定要是小寫，不可以有大寫喔!**print** function的小括號 **()** 裡放的是我們要印出的內容。如果要印出的是**text**，就要用**quotation marks**包起來，用**single quote**或**double quote**都可以，例如 `print("Hello World")` 或 `print('Hello Python')`。如果要印出的是**numbers**或變數**variables**，就直接放進小括號裡，例如：`print(2025)`。

函式說明: **print()** function

- **print**是內建函式(function) · 必須全部小寫。
- 小括號 **()** 裡放的是我們要輸出的內容。
 - a. 如果是文字 (text) · 就要用引號 (quotation marks) 包起來 · 單引號 (single quote) 或雙引號 (double quote) 都可以 · 例如 `print("Hello World")` 或 `print('Hello Python')`。
 - b. 如果是數字 (numbers) 或變數 (variables) · 就可以直接放進小括號裡 · 例如 `print(2025)`。

3. **T:** 如果要印出的內容裡面有不同種類的資料要組合在一起，例如：變數**year**和文字 **happy new year!**。在Scratch用**字串組合**，在Python，要在**print()** function的小括號裡，用**English comma**，把不同的資料串接在一起：`print(year, "happy new year!")`。數字**2025**和文字 **happy new year!**也是用**English comma**串接在一起：`print(2025, "happy new year!")`。

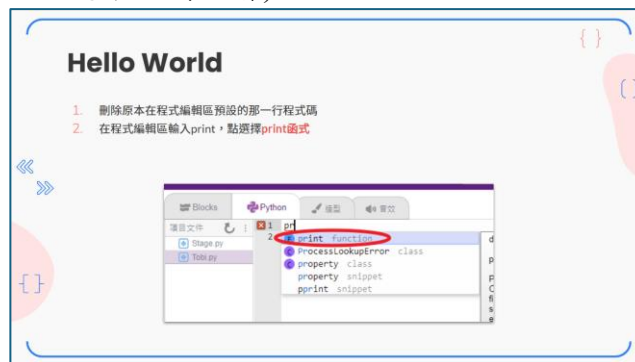
函式說明: **print()** function

- 要印出的內容裡面有不同種類的資料要組合在一起 · 例如：變數**year**和文字 **happy new year!** · 在Scratch用 **字串組合** · `year` `Happy new year!` · 在Python · 則要在**print()** function的小括號裡 · 用**英文的逗號 (English comma)**，把不同的資料串接在一起：`print(year, "happy new year!")`。
- 數字**2025**和文字 **happy new year!**也是用**English comma**串接在一起：`print(2025, "happy new year!")`。

4. **T:** 請大家在電腦桌面找到PictoBlox的圖示並打開。打開後如果有跳出更新訊息，直接按下跳過，然後點選Python編程(教師示範，簡報中也有步驟供同學忘記怎麼做時查看)



5. **T:** 刪除原本在程式編輯區預設的那一行程式碼，然後在程式編輯區輸入print，點選print函式(教師示範，簡報中也有步驟供同學忘記怎麼做時查看)



6. **T:** 在print() 中打上一個double quote"，按下Shift和Enter左邊的那個按鍵就可以打出double quote，這時候你會發現Pictoblox很聰明地幫你補了第二個double quote，接著我們在兩個double quotes 中輸入Hello World!，最後按下執行，就會看到終端印出Hello World! 這一行文字，到這邊恭喜同學們完成了人生中的第一行Python程式!(教師示範，簡報中也有步驟供同學忘記怎麼做時查看)



T: 完成你的程式後，別忘了將程式和終端畫面截圖貼在簡報中。



7. 學生實作

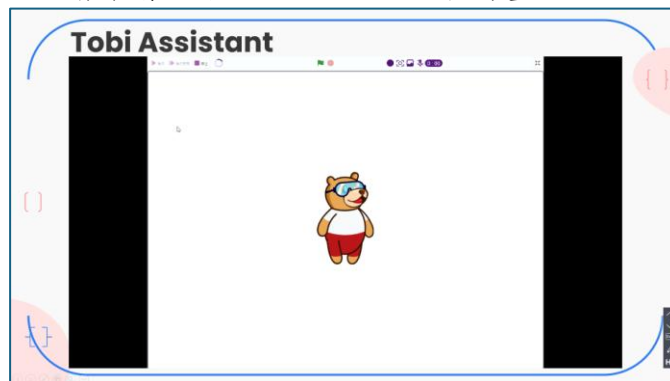
四、 **Tobi Assistant**

16 mins

1. **T:** 接下來要進行的活動叫做Tobi Assistant個人化程式助理任務，Tobi是誰啊？就像打開Scratch時，預設會有一隻cat的角色，打開PictoBlox，預設會有一隻bear的角色，這隻熊就叫做Tobi。



2. **T:** 接著來看一下Tobi Assistant的執行畫面。



3. **T:** 從剛才的影片中，看到了Tobi講了兩句話，請問第一句話，他說了什麼？

S: Hi, I'm Toby.

T: Good job! 再請問Tobi講的第二句話是什麼？

S: I can help you code.

T: Perfect!

4. **T:** 接下來，老師要示範如何完成這個任務囉! 首先選擇一個你喜歡的背景，按下Stage.py，這是舞台的程式，這時候你會看到「背景」頁籤，按下「背景」頁籤，接下來就跟Scratch的做法一樣。
(教師用PictoBlox示範，簡報中也有步驟供學生實作時查看)



5. **T:** 接著再選一個代表你的個人助理的造型，按下Tobi.py，這是Tobi這個角色的程式，這時候你會看到「造型」頁籤，按下「造型」頁籤，接下來就跟Scratch的做法一樣。



6. **T:** 現在我們要開始寫程式囉! 在Tobi.py的第2行打上sprite.，然後點選say這個函式，右邊會有提示告訴我們say這個函數中，第1個參數要放想說的訊息，第2個參數要放訊息停留的秒數



7. **T:** 再來我們要在say函式的括號裡打上一個雙引號，PictoBlox會自動加上另一個雙引號

Introduce Yourself with Tobi

{ }

1. 在say函式的括號裡打上一個雙引號(Shift+')
2. 軟體會自動加上另一個雙引號如下圖

```
1 sprite = Sprite('Shuchun')
2 sprite.say("")
```



8. **T:** 在雙引號中打上"Hi _____, I'm Tobi " 底線的部分是你的英文名字，例如:Lisa. 在第2個雙引號的後面打上逗號和2，表示這句話要持續2秒，一樣的作法，第3行讓你的角色說3秒"I can help you code."

Tobi Assistant

```
1 sprite = Sprite('Tobi')
2 sprite.say("Hi 你的名字, I'm Tobi!", 3)
3 sprite.say("I can help you code.", 3)
```

1. 在雙引號中打上Hi _____, I'm Tobi. 底線的部分是你的英文名字，例如:Lisa
2. 在第2個雙引號的後面打上逗號和3，表示這句話要持續3秒
3. 一樣的作法，第3行讓Tobi說3秒"I can help you code."
4. 請將你的程式和執行畫面分別截圖，貼在下一頁中



9. **T:** 完成程式之後，請按下執行測試。測試沒問題就把你的程式和執行畫面分別截圖，貼在簡報中。

Introduce Yourself with Tobi

{ }

程式截圖區

執行畫面截圖區

10. 加分題說明

(1)T: 進度快的同學，可以為你的作業爭取加分。有三道加分題給大家挑戰。第1題：目前Tobi只會說Hi ____, I'm Tobi! 請增加Tobi跟你打招呼的語句，例如:Nice to meet you. 可以到ChatGPT或Gemini查詢英文打招呼語句。

Tobi Assistant加分題1:
讓 Tobi 向你打招呼

- 目前Tobi只會說Hi __, I'm Toby!
- 請增加Tobi跟你打招呼的語句，例如:Nice to meet you.
- 可以到ChatGPT或Gemini查詢英文打招呼語句。

加分題1
程式截圖區

(2)T: 第2題：目前Tobi只會說I can help you code. 請增加Tobi助理可以幫忙你的事情。例如: I can answer your questions. 可以到ChatGPT或Gemini查詢相關的語句。

Tobi Assistant加分題2:
讓 Tobi 介紹他的一項能力

- 目前Tobi只會說I can help you code.
- 請增加Tobi助理可以幫忙你的事情。例如: I can answer your questions.
- 可以到ChatGPT或Gemini查詢相關的語句。

加分題2
程式截圖區

(3)T: 第3題：目前Tobi只會站在原地講話。可以利用sprite.goto() function 或是sprite.move() function 讓Tobi移動。

Tobi Assistant加分題3:讓 Tobi 移動

- 目前Tobi只會站在原地講話。
- 可以利用sprite.goto() function或是sprite.move() function讓Tobi移動。

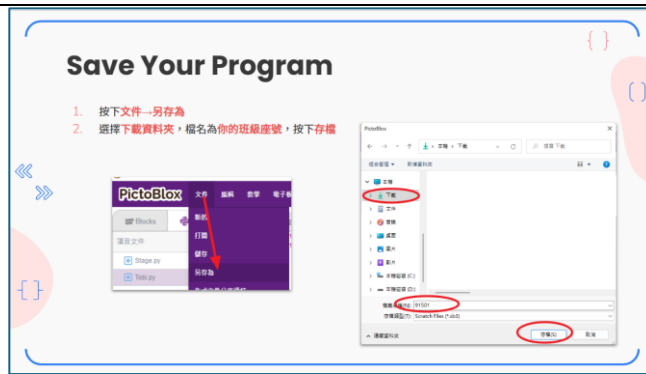
```

sprite.goto|
  goto          function  def gotoxy(x=0, y=0)
  gotoLayer     function
  gotoxy        function

sprite.move|
  move          function  def move(STEPS=10)
  
```

加分題3
程式截圖區

11. **T:** 下課前記得要將你的程式存檔，並上傳到Google Classroom，按下文件→另存為，選擇下載資料夾，檔名請打你的班級座號，按下存檔。



12. 學生實作

總結階段 Summary stage

5 mins

一、總結今日所學

1. **T:** 今天我們使用了PictoBlox來寫Python，學會使用`print()` function做輸出，還學會了如何讓角色說話。Now, let's do a quick quiz. 看看大家是否有將上課內容記起來。
2. **T:** First, about the `print()` function. If you want to print 'Good job!', what code would you use?



S: A

T: Good job! That's right! 括號裡的文字要用quotation marks來包住喔!

3. **T:** Next, how about printing '迎向2026'? What code would you use?



	<p>S: B</p> <p>T: <i>Excellent! 兩個項目之間要用English comma而不是Chinese comma喔!</i></p> <p>4. T: <i>Finally, about our sprite. If you want your sprite to say 'Hello' for 1 second, what code would you use?</i></p> <div data-bbox="450 461 1106 826" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p style="text-align: center;">About sprite</p> <p>想要角色說出"Hello"1秒鐘，應該要用哪一行程式碼?</p> <p>(A) <code>sprite.talk("Hello",1)</code></p> <p>(B) <code>sprite.speak("Hello",1)</code></p> <p>(C) <code>sprite.say("Hello",1)</code></p> <p>(D) <code>sprite.chat("Hello",1)</code></p> </div> <p>S: C</p> <p>T: <i>Perfect! 想要角色說話要用say，記得說話的內容後面，再加上秒數喔!</i></p> <p>二、教師最後叮嚀</p> <p>T: <i>Alright everyone, 程式檔案已經確實上傳的同學。Before you leave, please shut down the computer and put your mouse and chair back in place. See you next week!</i></p>	
<p>第二節</p>	<p style="text-align: center;">準備階段 Preparation stage</p> <p>一、老師指示入座與開機</p> <p>T: <i>Good morning/afternoon, everyone! Please find your seat and turn on your computer. Let's get ready for an exciting class!</i></p> <p>(學生進教室、就定位、開電腦中)</p> <p>T: (巡視，確認大部分學生就緒後) <i>Are all computers on? Good.</i></p> <p>二、複習上週所學</p> <p>T: <i>Last week, we explored Python and saw how simple but powerful it is. We used PictoBlox to write our first Python programs using the <code>print()</code> and <code>sprite.say()</code> functions.</i></p> <p>三、引入主題</p> <p>T: <i>Last time, Tobi was only talking to us. He couldn't hear our answers. 接下來的四節課，we are going to make Tobi smarter and smarter. 讓 Tobi 學會聽</i></p>	<p>5 mins</p>

我們的回答，還可以針對回答內容做出不同的回應。



T: 首先我們要進行一個活動叫做 'Tobi's Guessing Game'。透過這個活動，系統會問問題，而使用者可以輸入回答。大家準備好要跟 Tobi 互動了嗎？ Are you ready? Let's get started!

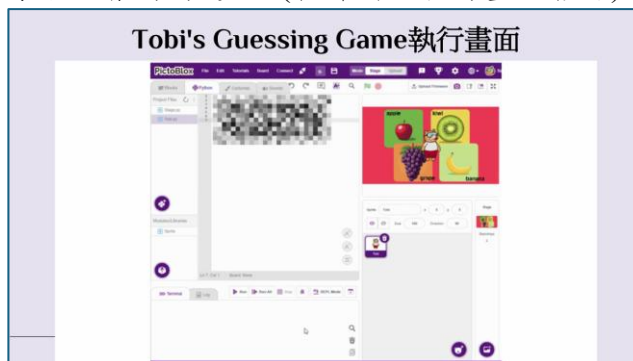


發展階段 Development stage

一、 Tobi's Guessing Game情境分析

1. 播放執行畫面影片

T: 請同學們仔細觀看Tobi's Guessing Game的執行畫面，為了讓同學看清楚，老師已經把速度調成0.5倍速，大家要仔細觀看，因為等下就會問問題喔!(教師播放執行畫面影片)



1 min

	<p>2. 分組討論情境分析</p> <p>T: 相信大家剛才都很專心觀看影片，接著就要請同學們把這三個步驟排序。You have 1 minute to discuss and answer.</p> <div data-bbox="422 318 1074 685" style="border: 1px solid black; padding: 10px; text-align: center;"> <p>Tobi's Guessing Game 1 情境流程圖</p> <pre> graph TD A[] --> B[] B --> C[] </pre> <p>問:What fruit do you think Tobi likes to eat? 印出"Your answer is"與使用者輸入的答案 使用者輸入答案</p> </div> <p>(學生討論並上傳答案)</p> <p>T: Alright, time's up! Let's see your answers. Group 1 的回答是「先問接著使用者作答，最後印出」。Correct! Good job, Group 1!</p> <p>T: Next is Group 2. Group 2 的答案有點可惜，第二和第三步驟相反了，必須要等使用者回答完，才有辦法印出回答的內容喔!</p> <p>(省略3~4組的回覆)</p> <p>T: 所以全對的有Group 1, Group 3 and Group 4. Well done! You get 3 points. Group 2 只有第一個步驟答對，So you have 1 point. Keep it up!</p>	3 mins
	<p>3. 將答案寫進作業簡報中</p> <p>T: 好，我們討論完答案了。Please log in to Google Classroom and open your assignment. 請完成 page 3 的這三個步驟排序。Once your whole group is finished, please raise your hand and say, 'Group X is done!'。</p> <p>(學生操作)</p> <p>T: OK, you all finished page 3. Group 2 and Group 3 were the fastest! So you get 3 points. Group 1 was the second fastest, so you get 2 points. And then Group 4 gets 1 point.</p>	5 mins
	<p>二、 Input() function(輸入函式)與Variables(變數)介紹</p> <p>T: 在剛才的執行畫面中，我們看到 Tobi 會提問，而使用者可以輸入回答。大家回想一下，在 Scratch 裡想要「詢問並等待使用者回答」時，Which block should we use?</p> <p>S: 詢問並等待</p>	7 mins

T: Excellent! 在Python裡對應的功能叫做 `input()` function。我們會把問題寫在小括號 `()` 裡面。Look at the slide. What should we use to wrap the text inside the parentheses? 括號裡的文字要用什麼包起來?

S: 引號

T: Very good! 記得喔，Strings (字串) 一定要用 quotation marks 包起來，電腦才讀得懂。再請問大家，在 Scratch 裡，有一個專門用來「儲存」使用者答案的積木，它叫什麼名字?

S: 詢問的答案

T: Perfect! 但在 Python 的世界裡，沒有自動幫你存好的積木。我們必須自己建立一個 Variable (變數) 來存放答案。

The slide is titled "Input() function(函式) and variables(變數)". It contains three lines of Python code: `1 sprite = Sprite('Tobi')`, `2 answer = input("What fruit do you think Tobi likes to eat?")`, and `3 print("Your answer is", answer)`. Below the code is a text box with the text "input() function的小括號中輸入要提問的內容". Underneath that is a Scratch-style question block: "詢問 What fruit do you think Tobi likes to eat? 並等待". At the bottom is a "詢問的答案" block.

T: Look at the slide，這行程式碼 `answer = input(...)` 就是建立一個名叫 `answer` 的變數，並把使用者輸入的答案存到這個變數裡。大家可以發現，這一行 Python 程式的作用，就等於 Scratch 裡的這兩行積木組合：先「詢問」，再把「詢問的答案」設定給變數 `answer`。

T: 變數非常強大，它儲存的不只有字串喔！變數也可以儲存數字，就像下面的例子：我們可以把變數 `name` 設為字串 "光明"，也可以把變數 `number` 設為數字 7。最後有個小提醒，大家在寫程式碼時，`=` 這個按鍵就在鍵盤 Backspace 的左邊，我已經幫大家圈起來了！現在，輪到你們來試試看，用 `input()` function 寫出自己的第一個互動程式囉！

Input() function(函式) and variables(變數)

```

1 sprite = Sprite('Tobi')
2 answer = input("What fruit do you think Tobi likes to eat?")
3 print("Your answer is", answer)

```

建立變數answer，並且把使用者輸入的答案存到變數answer裡

三、Tobi's Guessing Game任務講解

3 mins

1. 下載程式檔案

T: First, please log in to Google Classroom. 請大家先到作業區，找到 'Tobi's Guessing Game_學生作業檔案.sb3' 並點一下這個檔案的連結。點開後按下右上角的「三個點點」，點選「在新分頁開啟」。

2. 開啟程式檔案

3 mins

T: 這時候我們會看到「下載」按鈕，請按下它來下載這個作業檔案。Next, open PictoBlox on your desktop. 打開 PictoBlox 後，點選「檔案」接著按下「打開」。請點選電腦符號，到「下載」資料夾點選剛才的檔案，按下開啟。Is everyone ready? 大家都打開檔案了嗎？



3. 撰寫程式與測試

T: Now, let's code! 請參考簡報，在程式檔案中完成這三行程式碼。
 • Please use the input() function to ask Tobi's question. 寫完後請執行測試。如果執行沒問題，請將 Terminal (終端機) 的執行結果截圖，貼在簡報的黃色區域中。

T: A quick tip for the quotation marks: 如果找不到引號，請看簡報左下角，記得要按下Shift鍵加上引號鍵！完成後請先存檔上傳。

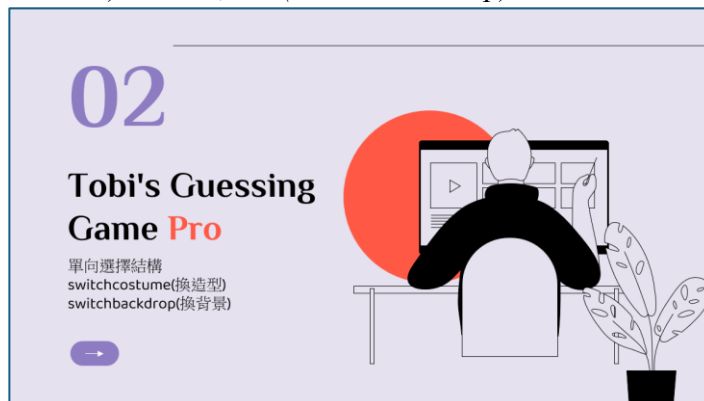


7 mins

四、Tobi's Guessing Game Pro介紹

T: 恭喜大家完成了 Tobi's Guessing Game 的程式！接下來我們要挑戰進階版的遊戲：Tobi's Guessing Game Pro。在這個版本中，我們不只要讓 Tobi 會聽，還要讓他學會做出反應！我們會學到 Selection Structure (單向選擇結構)，讓 Tobi 根據你的答案來換造型 (switch costume) 或換背景 (switch backdrop) 喔！

1 min



五、Tobi's Guessing Game Pro 情境分析

1 min

1. 播放執行畫面影片

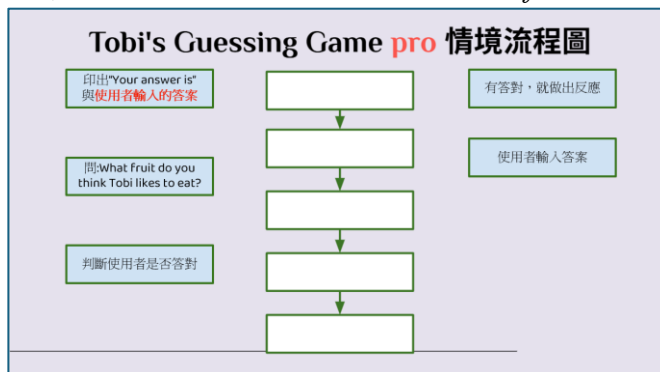
T: *What does the 'Pro' version do? Tobi's Guessing Game Pro 在做什麼呢？我們一起觀看執行畫面。請仔細看，當使用者輸入答案後，畫面上有什麼改變呢？影片播完後，老師有問題要問大家喔！（教師播放執行畫面影片）*



2. 分組討論情境分析

T: *Alright, 現在請看這張 Scenario Flowchart (情境流程圖)。畫面上這 5 個步驟被打亂了，請按照影片中發生的順序，將它們排入正確的格子裡。You have one and a half minutes to discuss. Go!*

3 mins



(學生討論並上傳答案)

T: *Alright, time's up! Let's see your answers. 除了 Group 2 之外，others 都答對囉！Well done! Group 2 的同學要注意：The computer needs the input first. 使用者還沒輸入答案，Tobi 是沒辦法印出結果或進行判斷的喔！*

3. 將答案寫進作業簡報中

T: *Now, please go back to your assignment. 請大家回到剛才開啟的作業簡報，turn to page 13 and sort these 5 steps. 將這 5 個步驟完成排序。Once your whole group is finished, please raise your hand and say, 'Group X is done!'*

3 mins

(學生操作)

	<p>T: <i>OK, I see Group 1 and Group 3 are done. Great! You all finished page 13. Good job!</i></p> <p style="text-align: center;">總結階段 Summary stage</p> <p>T: 大家今天表現得都很棒！我們今天一起完成了 <i>Tobi's Guessing Game</i>，成功學會了如何使用 <code>input()</code> 函式與變數來儲存答案。</p> <p>T: 我們也一起完成了 <i>Tobi's Guessing Game Pro</i> 的 <i>Scenario Flowchart</i> (情境流程圖)。大家現在都已經清楚 <i>Tobi</i> 變聰明的關鍵，就是先有「使用者輸入」，再進行「邏輯判斷」。</p> <p>T: <i>Next week, 我們會針對流程圖中的每個步驟進行問題分析與解題規劃。我們會正式學習 Python 的 Selection Structure (選擇結構)，並完成完整的程式，讓 Tobi 真正動起來跟我們互動！Looking forward to seeing your pro versions next week! 敬請期待，大家下課！</i></p> <p style="text-align: center;"><u>第二節結束 End of the second session</u></p>	3 mins
第三節	<p style="text-align: center;">準備階段 Preparation stage</p> <p>一、 老師指示入座與開機</p> <p>T: <i>Good morning/afternoon, everyone! Please find your seat and turn on your computer. Let's get ready for an exciting class!</i></p> <p>(學生進教室、就定位、開電腦中)</p> <p>T: (巡視，確認大部分學生就緒後) <i>Are all computers on? Good.</i></p> <p>二、複習上週所學</p> <p>T: <i>Last week 我們已經理清了邏輯：要讓 Tobi 變聰明，第一步必須先有 Input，也就是讓使用者輸入答案。沒有這個答案，後面的「判斷對錯」跟「更換造型」就無法發生。Remember, logic comes before coding. 邏輯永遠在寫程式之前。</i></p> <div data-bbox="311 1594 1002 1982" style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;">Tobi's Guessing Game pro 情境流程圖</p> <pre> graph TD A[Print "Your answer is" + user input] --> B[Ask: What fruit do you think Tobi likes to eat?] B --> C[Check if user's answer is correct] C --> D[If correct, make a response] </pre> </div>	5 mins

T: 我們也學會了兩個核心工具：首先是 `input()` function，它是 Tobi 的耳朵，負責聽使用者的回答；接著是 `Variable` (變數)，它是 Tobi 的大腦，負責把聽到的答案暫存起來。

Input() function(函式) and variables(變數)

```
1 sprite = Sprite('Tobi')
2 answer = input("What fruit do you think Tobi likes to eat?")
3 print("Your answer is", answer)
```

建立變數 `answer`，並且把使用者輸入的答案存到變數 `answer` 裡

name = "光明"

number = 7

接著要請你來試試看囉!

三、引入主題

T: 今天，我們要進入最精采的部分。我們要替 Tobi 設定一個「心目中的標準」，也就是建立一個變數 `fruit` 並設定為 `'pineapple'`。接著，我們會用 Python 的 `if` 選項結構，教 Tobi 如何把「看到的答案」跟「心裡的標準」做比對。Alright, let's get to work!

發展階段 Development stage

5 mins

一、 Tobi's Guessing Game Pro 問題分析與解題規劃

1. 問題分析與解題規劃1

T: 首先，我們要針對 Scenario Flowchart (情境流程圖)，討論每個步驟該如何轉化為程式碼。第一個要討論的是「判斷輸入的答案是否正確」這個核心環節。

T: To check if the answer is correct, we need a standard. 要判斷對錯，系統必須先知道「正確答案」是什麼。所以，我們要在程式中建立一個名叫 `fruit` 的變數，用來儲存 Tobi 真正喜歡吃的水果，也就是 `'pineapple'`。有了這個基準，我們才能拿變數 `fruit` 和使用者的輸入的 `answer` 進行比對。

T: 現在請各組討論：「建立一個變數 `fruit` 並將其值設定為 `'pineapple'`」這行程式碼該怎麼寫？Please discuss and send your answers. You have 2 minutes. Go!

(學生討論與回答)

T: Time's up! Let's see your answers. Group 3's answer is correct. Good job! 至於 Group 1 和 Group 2 的同學，remember, `'pineapple'` is a string. 別忘了字串一定要用 quotation marks (引號) 包起來喔！

T: Now, please go back to your assignment. Turn to page 14 and finish this line of code. Once you are done, please paste a screenshot of your code in the yellow area.

問題分析與解題規劃1

判斷輸入的答案是否正確

“建立一個變數fruit並將其值設定為pineapple”，才能用fruit和answer去比對是否相同。請將“”裡面敘述的這行程式碼寫在你之前的程式的最下方，並截圖貼在這裡。

程式碼截圖區

2. Single-way Selection Structure(單向選擇結構)if

2 mins

T: 有了基準答案 fruit 之後，我們就可以來判斷使用者的答案是否正確了！大家以前都用過 Scratch 的「如果...那麼 (If...then)」積木。

T:大家看簡報左上方，這個積木的六角形空缺是用來放 condition (條件式) 的；當條件成立時，要執行的動作都會被包在積木裡面，也就是我們說的 code block 1 (程式區塊 1)。

T:Python 的語法非常類似！我們首先寫下 if 指令，接著寫上 condition (條件式)，重點來了——最後一定要加上一個 semicolon (半形冒號)：。當你按下 Enter 換行時，你會發現系統自動幫你 indented (縮排) 了！在 Python 中，所有縮排的程式碼都屬於這個 if 結構的一部分。

T:Remember, the standard indentation is 4 spaces or 1 Tab. 縮排的標準是 4 個空格或一個 Tab 鍵，這是為了讓程式的層次結構更清楚喔！

Single-way Selection Structure(單向選擇結構)if

1. if 指令：意思是「如果」，一組選擇結構只會有一個if。

2. condition(條件式)：可以是判斷式、變數、布林值等。建議使用小括號將條件式包起來，以清楚呈現條件式的範圍。

3. semicolon (半形冒號)：用於表示if指令的條件式敘述結束。

4. 若條件式成立 (True)：執行程式區塊1。若不成立 (False)，則不會執行。

5. indentation (縮排)：只要語法中有「:」，下一行就要縮排。

(1) 為了呈現程式的內外層結構，Python 會使用「縮排」來表示，if 指令的下一行是選擇結構內層，換行時會自動縮排。

(2) 縮排的標準格式為4 spaces(空格)或 1個Tab。

3. Relational Operators(關係運算子)

5 mins

T: 了解了 if 的結構後，我們要來討論條件判斷裡不可或缺的

Relational Operators (關係運算子)。大家看簡報，這就是用來比較兩邊內容的方法。

T: 在 *Scratch* 裡，我們用一個 = 積木來判斷左右兩邊是否相等。但在 *Python* 語法中，我們必須使用兩個等號 == 才能判斷是否相同。*Quick question:* 為什麼 *Python* 判斷相等要用兩個 = 而不是一個呢？*You have 1 minute to discuss. Go!*

(學生討論與回答)

T: *Time's up! Let's see your answers. The answers of Group 1 and Group 4 are correct. Excellent!* 因為在 *Python* 中，一個 = 是 *Assignment (賦值)*，是用來將內容儲存到變數裡的；而兩個 == 才是 *Equal (相等判斷)* 喔！

T: 接著我們來看「大於 >」和「小於 <」的用法，這部分 *Python* 和 *Scratch* 是完全一樣的。

Relational Operators(關係運算子)1

運算子	說明	例1	對應Scratch積木
==	判斷兩者是否相等	answer == "apple" 判斷變數answer是否與字串apple相等	
>	判斷前者是否大於後者	num > 5 判斷變數num是否大於5	
<	判斷前者是否小於後者	num < 5 判斷變數num是否小於5	

T: *Next,* 我們來看一些比較特別的。在 *Scratch* 中要表達「不等於」有點麻煩，必須先用一個 = 積木，外面再套上一個「不成立(Not)」的積木。但在 *Python* 裡，我們只需要用 != 就可以表示囉！

T: 最後，如果我們要判斷「大於等於」或「小於等於」，在 *Scratch* 中必須用「或(Or)」來連接兩個積木。而在 *Python* 中非常直觀，直接寫成 >= 和 <= 就可以了！這就是 *Python Simple but powerful* 的地方。大家記住這些符號了嗎？

Relational Operators(關係運算子)2

運算子	說明	例1	對應Scratch積木
!=	判斷兩者是否不相等	answer != "apple" 判斷變數answer是否與字串apple不相等	
>=	判斷前者是否大於等於後者	num >= 5 判斷變數num是否大於等於5	
<=	判斷前者是否小於等於後者	num <= 5 判斷變數num是否小於等於5	

4. 問題分析與解題規劃2

4 mins

T: *Alright! 到這裡，老師已經介紹完 Single-way Selection Structure (單向選擇結構) 的用法，也認識了各式各樣的關係運算子。接著，就要考考大家是否掌握重點囉！*

T: *現在請各組討論：如果我們要寫出 'if (變數 fruit 的內容與變數 answer 相同):' 這行程式碼，正確的 Python 語法該怎麼寫？請討論後上傳你們的回答。You have one and a half minutes to discuss and answer. Go!*

(學生討論與回答)

T: *Time's up! Let's see your answers. Group 4's answer is correct. Perfect! 至於 Group 2 和 Group 3 的同學請注意，條件式的後面一定要加上 semicolon (半形冒號)：喔！還有 Group 1 要記得，在 Python 中要判斷是否 equal (相等)，必須使用兩個等號 == 喔！*

T: *Now, go back to your assignment and turn to page 18. Please finish this line of code. Once you are done, screenshot your code and paste it into the purple area.*



5. 問題分析與解題規劃3

4 mins

T: *接下來，我們要來討論情境流程圖中的最後一個環節：「有答對，就做出反應」。我們再看一次 Tobi's Guessing Game Pro 的執行畫面，這一次請大家仔細觀察：當使用者答對時，整個畫面發生了哪些變化？*

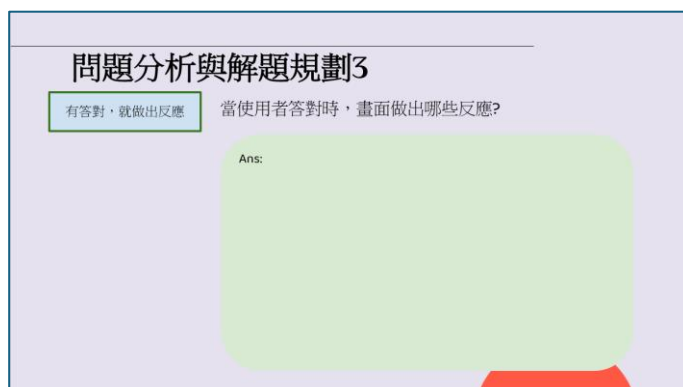
(教師播放執行畫面影片)

T: *請問大家，從剛才的影片中，你們看到當使用者答對時，系統做出了哪些反應？ You have 1 minute to discuss and answer. Go!*

(學生討論與回答)

T: *Time's up! Let's see your answers. Group 2 的答案最完整！當使用者答對時，背景會切換 (switch backdrop)，Tobi 的造型也會切換 (switch costume)，並且他會說出：「Correct! I like to eat pineapple!」 Perfect! Group 2 get 1 point!*

T: *Now, please turn to page 19 of your assignment. 請大家將剛才討論出的這三個反應，寫在簡報中的綠色區域中。*



6. 選擇回應時的背景與角色造型

(1) 教師說明與示範

T: *現在我們要完成今天最後一項任務：Personalize your game! 打造個人化的遊戲反應。*

T: *首先，要請大家在 PictoBlox 裡選擇一個自己喜歡的背景。當使用者答對時，系統就要 switch to the backdrop you chose，切換成你選的那個背景。接著，當使用者答對時，Tobi 也不能只是呆呆站著，請大家幫他選一個最帥或最漂亮的造型，讓他在答對時進行 costume switching。*

T: *大家可以先去右下角的背景區和造型區晃晃，Pick your favorite background and costume!*

(2) 學生操作(包含下載檔案與選擇背景和角色造型)

7. 問題分析與解題規劃4

T: *老師看到大家剛才都很認真地挑選自己想要的背景和造型。Very good! 現在我們要來進行一個邏輯大考驗。*

T: *大家看簡報左邊的流程圖。目前我們知道，在步驟「有答對，就做出反應」時要切換造型和背景。但是，為了確保遊戲每一次*

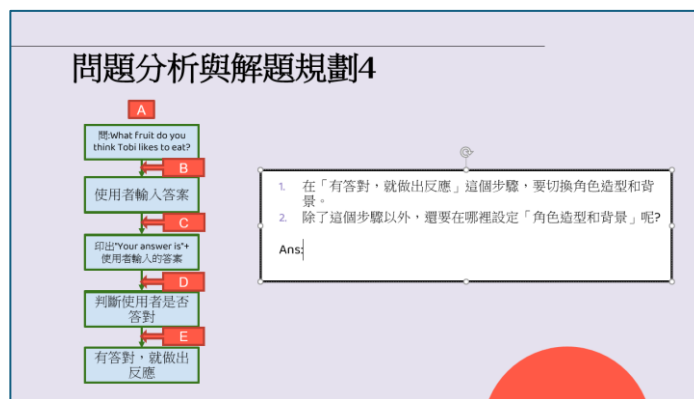
5 mins

3 mins

重新開始時，背景和造型都能回到最初的樣子，除了這個步驟以外，還有哪個位置必須設定「角色造型和背景」呢？請各組看著流程圖中的 A、B、C、D、E 位置進行討論。You have one minute. Go!

(學生討論與回答)

T: Time's up! Let's see your answers. Group 4 and Group 1 are correct. Perfect! 我們在程式最一開始 A 這個位置，就要設定好「預設」(Default) 的角色造型和舞台背景，這樣遊戲才不會一直停留在上一次答對的畫面喔！



8. 切換角色造型與背景(1)

7 mins

T: OK! 接下來我們要利用 Python 程式碼來正式接管造型與背景的切換了。請大家看老師的示範。

(教師打開 PictoBlox 操作示範)

T: 當我在程式編輯區輸入 `sprite.sw` 時，系統會自動彈出兩個關鍵函式：一個是 `switchcostume()` 用來換造型，另一個是 `switchbackdrop()` 用來換背景。

T: 在使用時有兩個重點：第一，小括號裡要填入你剛才記下的造型或背景名稱；第二，也是最重要的——這些名稱都是 String (字串)，所以一定要用 Double Quotes (雙引號) "" 包起來！

T: Now, please go back to your code. 請在程式的最上方加上這兩行「預設設定」的程式碼。完成後，take a screenshot and paste it into the yellow box on page 21. 請將截圖貼到簡報中的黃色區塊喔！

(學生實作)

切換角色造型與背景(1)

- 切換**造型**使用sprite.switchcostume()，小括號裡放的是**造型名稱**。
- 切換**背景**使用sprite.switchbackdrop()，小括號裡放的是**背景名稱**。
- **造型名稱**和**背景名稱**都是字串，所以要記得用double quotes""包起來喔!

請在你的程式最上面設定按下執行時的預設舞台背景和角色造型。

程式碼截圖區

9. 切換角色造型與背景(2)

4 mins

T: 大家現在都已經設定好遊戲開始時的「預設造型和背景」了。最後一步，我們要讓 Tobi 在答對時展現出驚喜的變化！請大家接續上堂課寫的 if 條件式，在下方完成「有答對，就做出反應」的步驟。

T: 重點提醒！大家看簡報第 2 點，在 if 條件式的下一行開始，所有反應動作都必須執行 Indentation (縮排)。請按一下鍵盤上的 Tab 鍵或輸入 4 個半形空格，讓程式知道這些動作是屬於答對後才要執行的喔！

T: 這部分的程式完成後，請將代碼截圖貼在下方的綠色區塊中，並記得將 PictoBlox 程式檔案存檔上傳。

切換角色造型與背景(2)

1. 現在你已經設定好預設的造型和背景了，接著請你完成「有答對，就做出反應」這個步驟，在if條件式下切換答對時造型和背景。

2. 記得要縮排呢!

3. 這部分的程式完成後，請截圖貼在下方綠色區塊，並將程式存檔上傳。

有答對，就做出反應

程式碼截圖區

總結階段 Summary stage

1 min

T: 現在 Tobi 已經會根據你的答案偷偷換造型、變背景，只是他現在還是個「省話一哥」。Next week, 我們會加上最後一個步驟 — 讓 Tobi 開口說話！到時候，大家就能完成真正的 Tobi's Guessing Game Pro 囉！已經上傳好檔案的同學就可以下課了，See you next week!

第三節結束 End of the third session

第四節

準備階段 Preparation stage

一、老師指示入座與開機

T: *Good morning/afternoon, everyone! Please find your seat and turn on your computer. Let's get ready for an exciting class!*

(學生進教室、就定位、開電腦中)

T: (巡視，確認大部分學生就緒後) *Are all computers on? Good.*

二、複習上週所學

T: *Look at the screen. 這是 Tobi's Guessing Game Pro 的 Program Flowchart。上週我們已經完成了「切換預設背景造型」、「建立變數 fruit」，以及最關鍵的「if 條件判斷」與「答對後的視覺切換」。現在，我們只差最後一個步驟，就能完成整個程式了！大家看流程图的最下面，當使用者答對時，除了換背景、換造型，Tobi 還要說出：「Correct! I like to eat ...」並加上使用者的答案。*



T: *現在我們來看看這行程式碼該怎麼規劃吧!*

發展階段 Development stage

一、字串組合

1. 發現問題

(教師打開 PictoBlox 操作示範)

T: *我們要讓 Tobi 說話。這個步驟也是在 if 的條件下執行的，所以大家要記得先按下 Tab 鍵進行縮排 (Indentation) 喔！大家還記得讓 Tobi 說話的指令嗎？*

學生回答: `sprite.say`

5 min

5 mins

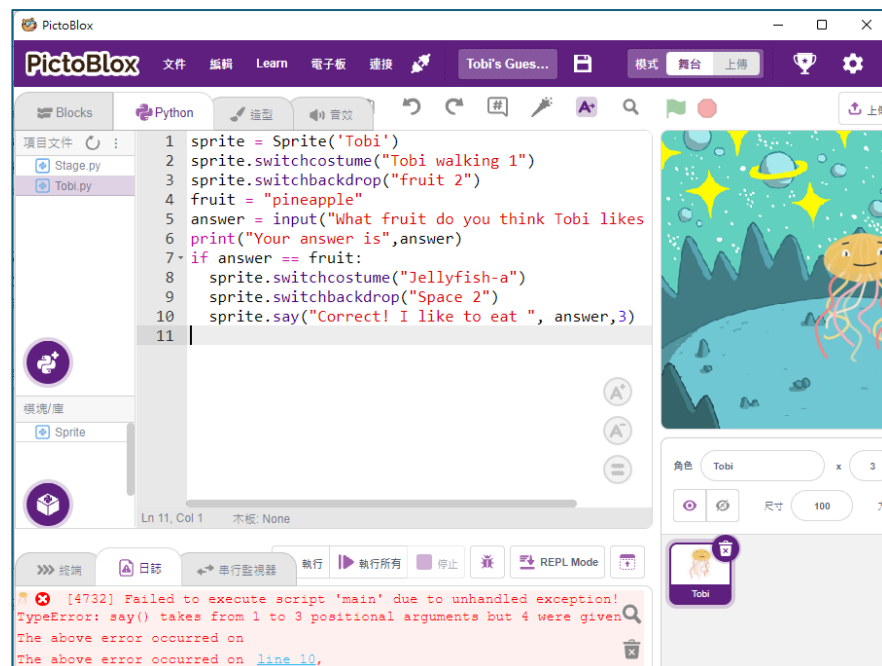
T: Good job! 我們先按照以前的習慣，輸入 `sprite.say("Correct! I like to eat ", answer, 3)`。記得這段台詞是 `String`，要用 `Double Quotes` (雙引號) 包起來；我們試著用 `English Comma` (半形逗號) 把台詞、變數 `answer` 和時間 `3` 隔開。

T: OK，我們來測試一下！按下執行，輸入正確答案 `pineapple`。Oh oh! 出現錯誤訊息了！

T: 首先我們可以先觀察，終端已經印出 "Your answer is pineapple"，而且背景和角色造型也都換了，那問題就是出在 `Tobi` 講話的這行程式。

T: 同時我們也可以看看下方日誌 (Log)，錯誤發生在第 10 行，訊息寫著：`TypeError: say() takes from 1 to 3 positional arguments but 4 were given`。這是什麼意思呢？當我重新輸入一次 `sprite.say`，系統提示告訴我們：括號裡只能放兩個位置，一個是 `MESSAGE` (內容)，另一個是 `SECS` (秒數)。

T: 但我們現在傳了台詞、變數、秒數，被系統誤認為是 3 個資料 (加上隱藏的 `self` 共 4 個)，所以報錯了。但我們一定要把 `answer` 放進台詞裡呀，該怎麼辦呢？



```
1 sprite = Sprite('Tobi')
2 sprite.switchcostume("Tobi walking 1")
3 sprite.switchbackdrop("fruit 2")
4 fruit = "pineapple"
5 answer = input("What fruit do you think Tobi likes
6 print("Your answer is",answer)
7
8 if answer == fruit:
9     sprite.switchcostume("Jellyfish-a")
10    sprite.switchbackdrop("Space 2")
11    sprite.say("Correct! I like to eat ", answer,3)
```

[4732] Failed to execute script 'main' due to unhandled exception!
TypeError: say() takes from 1 to 3 positional arguments but 4 were given
The above error occurred on
The above error occurred on [line 10](#).

2. 介紹字串組合

4 mins

T: 這時候，我們就要用到 Python 的大絕招：字串組合 (String Concatenation)！以前在 Scratch 有「字串組合」積木，在 Python 裡更簡單，直接使用 + 號 就可以把兩個字串像小火車一樣接在一起了。！

T: 因為變數 `answer` 裡面存的本來就是字串，所以我們把程式改成：`"Correct! I like to eat " + answer`。這樣對系統來說，加號兩邊合起來就是「一個」完整的訊息了！

T: Now, 請大家利用 + 號 來完成 Tobi 說話的這個步驟。完成後，請將這行程式截圖貼在 Page 24 的橘紅色區塊中。

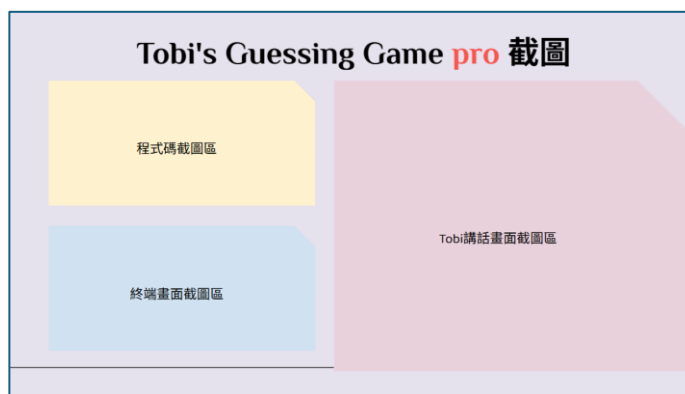


(學生操作)

二、完成 Tobi's Guessing Game Pro

3 mins

T: Good job！我們終於把整個 Tobi's Guessing Game Pro 的程式寫完囉！最後一項任務，請大家完整測試你的程式。確認沒問題後，turn to page 25，將你的程式截圖、終端畫面 (Terminal) 以及 Tobi 講話的畫面依序貼在對應的區塊中。



(學生操作)

三、Tobi's Guessing Game Pro Max

2 mins

1. 播放執行畫面

T: 完成了 Pro 版本之後，我們要挑戰更進階的 Pro Max Version 囉！這一次我們要學習 Selection Structure (雙向選擇結構)。



T: What does the 'Pro Max' version do? 到底它多了什麼新功能呢？我們一起來觀看執行畫面。請大家仔細觀察：當使用者輸入答案後，如果答錯了，畫面會有什麼改變呢？影片播完後，老師有問題要問大家喔！

(教師播放執行畫面影片)



2 mins

2. 情境分析

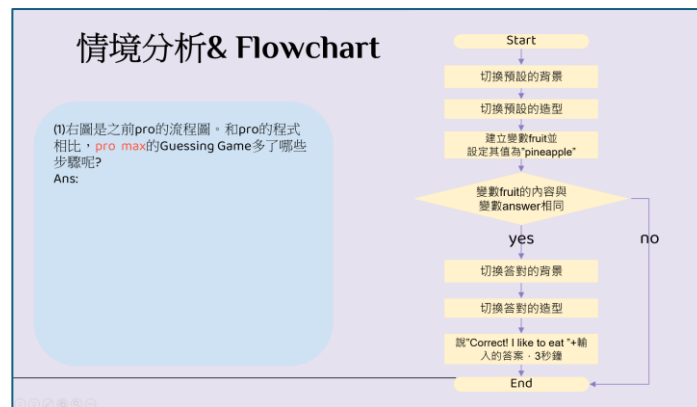
(1) 發現新步驟

T: Take a look at the right side. 這是我們之前 Pro version 的 flowchart。請問大家，和之前的程式相比，Pro Max 的 Guessing Game 多了哪些反應步驟呢？請各組討論一下。You have 1 minutes. Go!

(學生討論與回答)

T: Time's up! Let's see your answers. Group 2 and Group 3 are

correct. Excellent! 從剛才的影片中可以看到，當使用者答錯時，系統也多了造型切換以及背景切換的反應，不再只是靜悄悄地結束遊戲了！



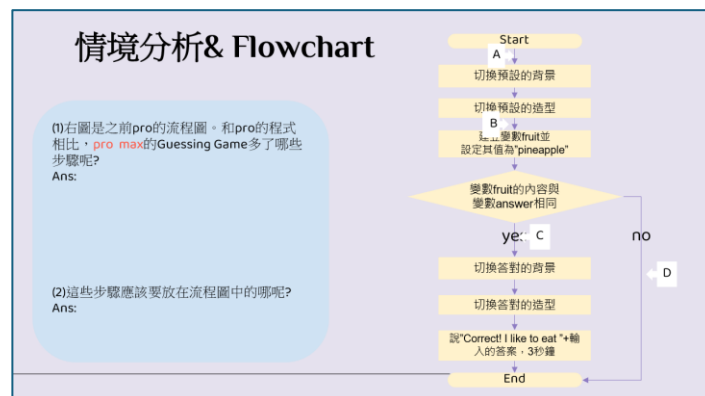
2 mins

(2) 邏輯定位

T: 那麼問題來了，剛才講到的這兩個「答錯反應」的新步驟，應該要放在流程圖中的哪一個位置呢？是 A、B、C 還是 D 呢？請各組再次討論。You have 1 minute. Go!

(學生討論與回答)

T: Time's up! Let's see your answers. All of you are correct. Perfect! 當判斷結果為 'no' (不相同) 時，程式就會走右手邊這條路線，所以這兩個新增步驟當然要放在 位置 D 這裡囉！



2 mins

(3) 完成作答

T: Now, please go back to your slides and turn to page 28. 剛才大家都討論出答案了，現在請把這兩題的分析結果寫在簡報第 28 頁的作答區。You have 2 minutes. Go!

3 mins

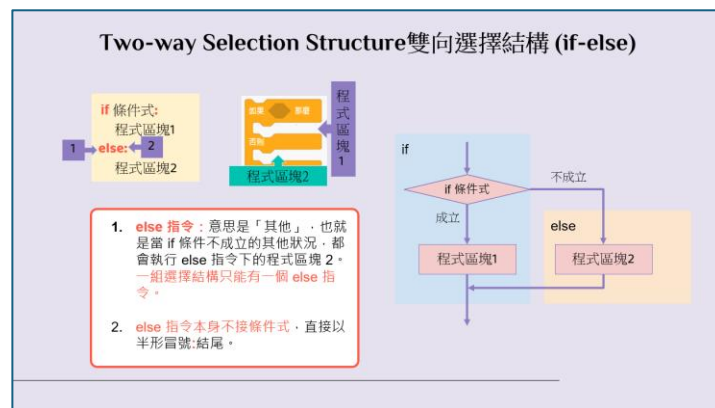
3. 雙向選擇結構 if-else

(1) 語法介紹

T: 從剛才看到的流程圖中，相信很多同學都想到了 Scratch 裡的「如果...那麼...否則...」積木。在 Python 中，對應的語法就是 *if-else* (雙向選擇結構)。

T: Look at the slide. *else* 的意思就是「其他」狀況，也就是大家在 Scratch 裡最熟悉的「否則」積木！它負責處理當 *if* 條件不成立時的所有剩餘情況。

T: 我們來看一下 Python 裡的 *else* 該怎麼寫。這裡有兩個非常重要的規則，大家一定要記住：第一，*else* 指令後面不接條件式：它很霸氣，直接以半形冒號：結尾。第二，必須要縮排：就像 *if* 一樣，*else* 下方要執行的程式區塊也必須向右縮進 (Indentation)，這樣電腦才知道哪些動作是屬於「否則」的部分。小提醒：在一組選擇結構中，只能有一個 *else* 指令喔！



4 mins

(2) 語法辨析活動

T: 現在來當個小偵探！請大家仔細觀察簡報 Page 30 的四個選項，哪一段程式的執行結果，才會和剛才 Pro Max 影片中的反應相符呢？請注意冒號和縮排的細節。 You have 2 minutes. Go!

Two-way Selection Structure 雙向選擇結構 (if-else)

下列哪一段程式的執行結果，才會和Tobi's Guessing Game pro max 執行畫面相符？

Ans:

A

```
if answer == fruit:
    sprite.switchcostume("Jellyfish-a")
    sprite.switchbackdrop("Space 2")
    sprite.say("Correct! I like to eat " + answer,3)
else:
    sprite.switchcostume("Jellyfish-b")
    sprite.switchbackdrop("Neon Tunnel")
    sprite.say("I don't like to eat " + answer,3)
```

B

```
if answer == fruit:
    sprite.switchcostume("Jellyfish-a")
    sprite.switchbackdrop("Space 2")
    sprite.say("Correct! I like to eat " + answer,3)
else:
    sprite.switchcostume("Jellyfish-b")
    sprite.switchbackdrop("Neon Tunnel")
    sprite.say("I don't like to eat " + answer,3)
```

C

```
if answer == fruit
    sprite.switchcostume("Jellyfish-a")
    sprite.switchbackdrop("Space 2")
    sprite.say("Correct! I like to eat " + answer,3)
else
    sprite.switchcostume("Jellyfish-b")
    sprite.switchbackdrop("Neon Tunnel")
    sprite.say("I don't like to eat " + answer,3)
```

D

```
if answer == fruit
    sprite.switchcostume("Jellyfish-a")
    sprite.switchbackdrop("Space 2")
    sprite.say("Correct! I like to eat " + answer,3)
else
    sprite.switchcostume("Jellyfish-b")
    sprite.switchbackdrop("Neon Tunnel")
    sprite.say("I don't like to eat " + answer,3)
```

(學生討論與回答)

T: Time's up! Let's see your answers. Group 1 is correct! 只有選項 B 的程式碼是完全正確的，Brilliant !

T: 我們來看看其他選項錯在哪裡：選項 A 的 else 下方沒有縮排，你們看第二行程式碼下面有紅線，就是 PictoBlox 在提醒語法錯誤；選項 C 的 if 和 else 後面都漏掉了重要的「冒號」，且下方也沒有縮排；而 選項 D 雖然有縮排，但同樣漏掉了 if 和 else 後面的冒號喔！

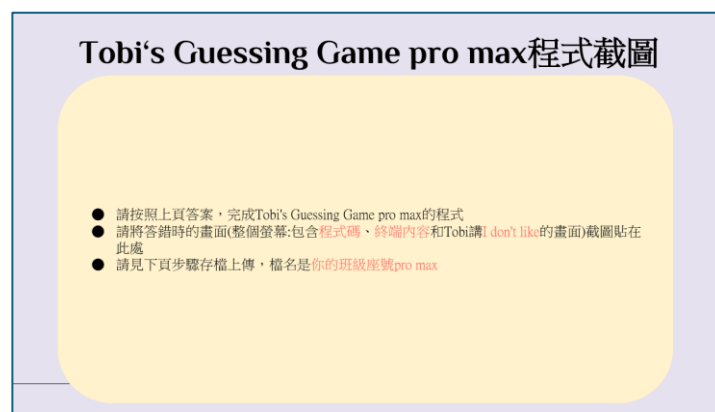
8 mins

4. 完成 Tobi's Guessing Game Pro Max

T: Now, please open your Pro version program. 請大家把剛才討論出的選項 B 正確語法，加進你的程式碼中，把最重要的「答錯反應」補齊！

T: 完成並測試沒問題後，請進行最後的截圖任務：截取答錯時的畫面（必須包含完整的程式碼、終端內容，以及 Tobi 說出 "I don't like to eat..." 的畫面），並貼在簡報 Page 31。

T: 最後，請將檔案存檔上傳，檔案名稱請標註：班級座號 + pro max。恭喜大家，你們馬上就要完成最頂級的遊戲版本了！大家動手試試看吧！ Go!



(學生開始操作，教師巡視指導)

5 mins

總結階段 Summary stage

T: Alright, everyone. 今天我們不只完成了 Tobi's Guessing Game Pro Max，更代表我們正式完成了整個「Let's Play Python with Tobi」單元了！大家給自己一個熱烈的掌聲！

T: 回想一下我們這幾週的旅程：第一階段，我們認識了 Python，學會用 `print()` 讓系統印出文字，並練習用 `sprite.say()` 讓 Tobi 第一次開口跟我們打招呼。

T: 第二階段，我們進入了遊戲開發核心。大家學會用 `input()` 接收玩家的回答，並用「變數」把答案存起來。

T: 最後階段，我們挑戰了最難的「邏輯判斷」。從一開始只能處理答對情況的單向選擇 (Pro 版)，到今天你們親手用 `if-else` 寫出了能同時處理對錯反應的雙向選擇結構 (Pro Max 版)。

T: 在這些過程中，你們學到了 Python 最關鍵的靈魂—冒號與縮排 (Indentation)，也克服了 `TypeError` 的 Debug 挑戰，學會用 `+` 號來組合字串。程式語言不只是一串代碼，它更是你們與電腦溝通、解決問題的工具。今天你們能讓 Tobi 根據對錯做出不同的動作與對話，這就是人工智慧與軟體開發的最基本邏輯。

T: 下課前，請確認你的 Pro Max 程式檔案已經命名正確並上傳完畢。
Shut down the computer, put your mouse and chair back in place. 期待日後看到大家用 Python 創造出更多有趣的作品！ See you next time!

第四節結束 End of the third session