北科大互動系寒假體驗營

程式設計課程

講者：蔡姍珊

Hello World_

自我介紹

蔡姍珊

畢業於高雄高工資訊科

專長是網頁設計、資料分析

喜歡看動漫>///<

互動設計系媒體設計組三年級

作品：

Food Man互動展-拼圖遊戲開發

藍眼淚奇幻展-展出品航杭(程式負責)

載動科技旗下app網頁設計(進行中)

我不會設計喔＾ｏ＾

今天要聊甚麼呢!!

第1堂課

- 互動系學甚麼程式?
- 程式語言介紹

第2堂課

- 來寫一個小遊戲吧><
- Python遊戲實作

```
Hello World_
```

///

>=

互動系在學甚麼程式??

+ +/

大概有這些

C#

UNITY寫遊戲、APP(最常用到)、

Python

AI、生成式XXX之類的

C++

互動裝置會用到的Arduino

HTML/CSS

網頁前端設計

///

>=

甚麼是程式語言

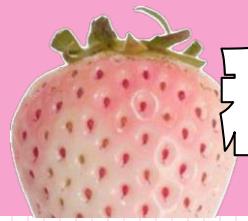
if{}
<<

++/

程式語言是一種用來與電腦溝通的工具。

透過它，人類可以撰寫指令來讓電腦執行各種任務。

程式語言的設計讓人類的思維邏輯能被轉化為電腦可以理解的語法結構，並最終執行出相應的操作。💻✨



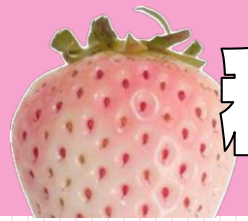
程式語言

流程導向

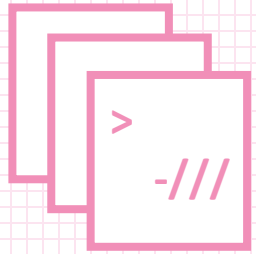
- 以程序和功能為中心，按照步驟或指令執行操作。
- 將問題拆解為一系列的操作流程，按順序完成任務。
- 強調 如何完成工作

物件導向

- 將程式設計視為由「物件」組成的世界，每個物件包含「屬性」。
- 將問題拆解為多個可互動的物件，模擬現實世界。
- 核心概念包括 封裝、繼承 和 多型。



程式語言

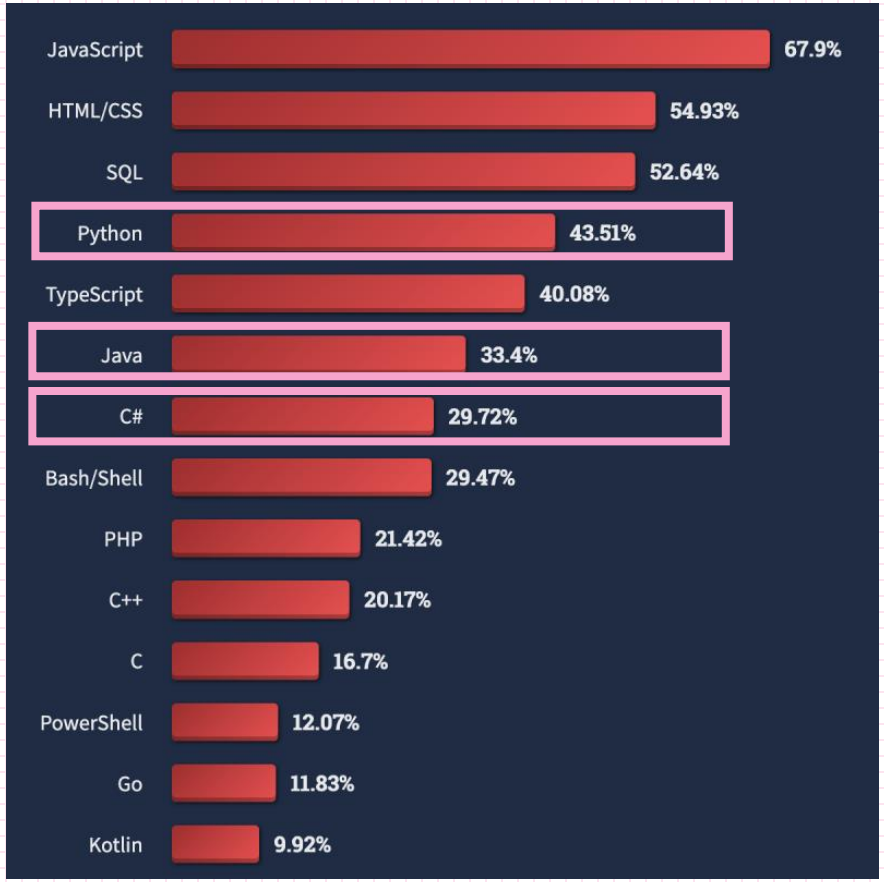


不同領域不同應用~

- 網頁開發（如：HTML、CSS、JavaScript）創建互動式網站。
- 遊戲開發（如：C++、C#、Unity Script）開發電腦或手機遊戲。
- 人工智慧與機器學習（如：Python）用於分析資料與訓練模型。
- 系統程式設計（如：C、Rust）開發作業系統或嵌入式系統。

程式語言

新手適合接觸哪個程式語言～～



General Purpose Language

Python

優點：

- 簡單易學，語法接近自然語言。
- 文件和教學資源豐富。
- 應用廣泛（如資料分析、人工智慧、網頁後端等）。

適合的人：

- 對程式設計完全沒有經驗，希望快速上手。

C++

優點：

- C++ 非常接近硬體，可以編寫高效能程式。
- 文件和教學資源豐富。
- 適合需要快速執行的應用（如遊戲引擎、嵌入式系統等）。

適合的人：

- 對底層開發有興趣，希望奠定扎實的基礎。

寫程式會不會

很難很無聊QQ

一起來學程式吧

不會啦寫程式很像在變魔法><

幾行字可以弄出酷東西!!!

很好玩!

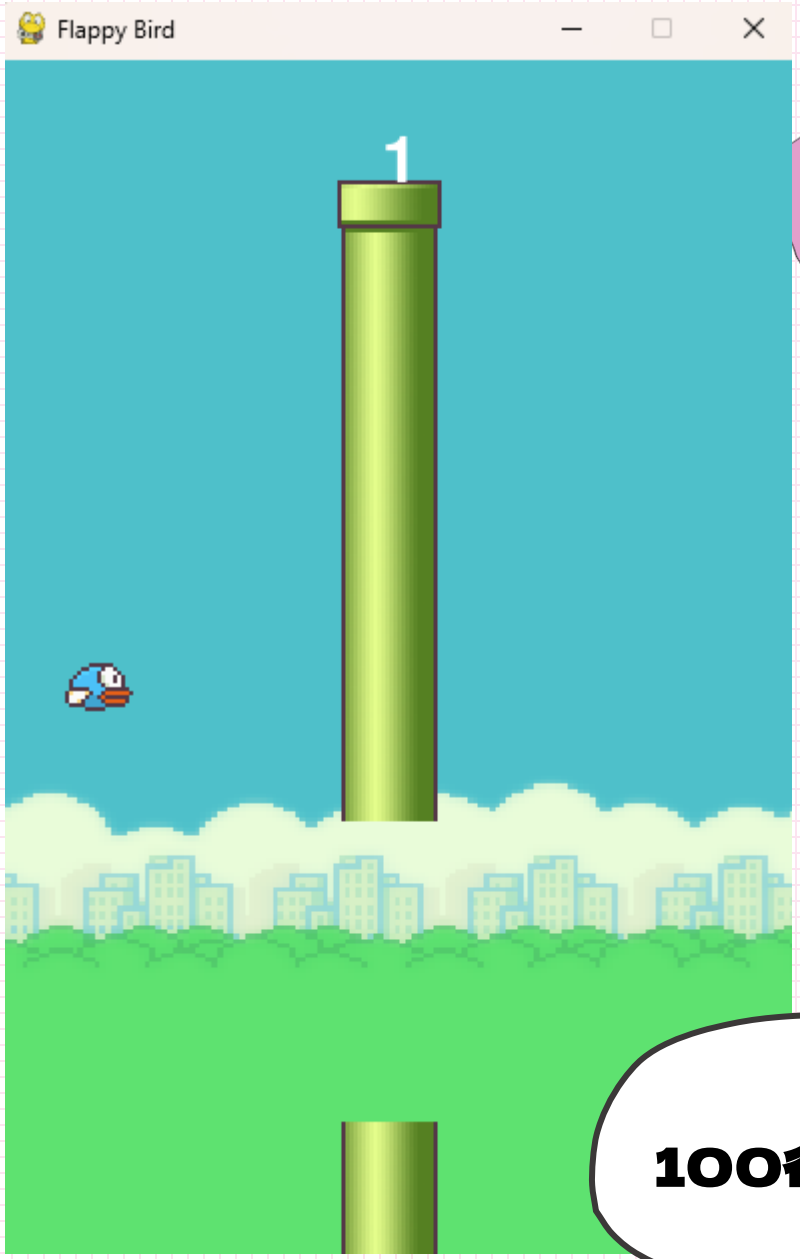
o.oǁǁ?!!



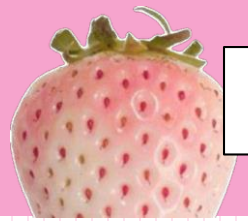
///
>=
用Python寫Flappy Bird


+ +/

```python
import pygame, sys, random

pygame.init()
screen = pygame.display.set_mode((400, 600))
pygame.display.set_caption('Flappy Bird')
clock, font = pygame.time.Clock(), pygame.font.SysFont(None, 48)
background = pygame.transform.scale(pygame.image.load(r'C:\Users\User\Downloads\background-day.png'), (400, 600))
bird_img = pygame.transform.scale(pygame.image.load(r'C:\Users\User\Downloads\bluebird-downflap.png'), (34, 24))
pipe_img = pygame.transform.scale(pygame.image.load(r'C:\Users\User\Downloads\pipe-green.png'), (52, 320))

class Bird:
    def __init__(self):
        self.rect = bird_img.get_rect(center=(50, 300))
        self.movement, self.gravity, self.jump_strength = 0, 0.25, -6

    def update(self):
        self.movement += self.gravity
        self.rect.y += self.movement

    def jump(self):
        self.movement = self.jump_strength

class Pipe:
    def __init__(self):
        y = random.randint(150, 400)
        self.top = pipe_img.get_rect(midbottom=(500, y))
        self.bottom = pipe_img.get_rect(midtop=(500, y + 150))

    def update(self):
        self.top.x -= 5
        self.bottom.x -= 5

    def draw(self):
        screen.blit(pipe_img, self.top)
        screen.blit(pygame.transform.flip(pipe_img, False, True), self.bottom)

def main_game():
    bird, pipes, score, game_over = Bird(), [Pipe()], 0, False
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT: pygame.quit(); sys.exit()
            if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE:
                bird.jump() if not game_over else main_game()

        if not game_over:
            bird.update()
            if bird.rect.top <= 0 or bird.rect.bottom >= 600: game_over = True

            for pipe in pipes:
                pipe.update()
                if pipe.top.right < 0: pipes.remove(pipe)
                if pipe.top.right < bird.rect.left and not hasattr(pipe, 'scored'):
                    pipe.scored, score = True, score + 1

            if pipes[-1].top.centerx < 200: pipes.append(Pipe())
            for pipe in pipes:
                if bird.rect.colliderect(pipe.top) or bird.rect.colliderect(pipe.bottom): game_over = True

        screen.blit(background, (0, 0))
        for pipe in pipes: pipe.draw()
        screen.blit(bird_img, bird.rect)
        screen.blit(font.render(str(score), True, (255, 255, 255)), (200, 50))
        pygame.display.update()
        clock.tick(30)

main_game()
```



100行內搞定!!

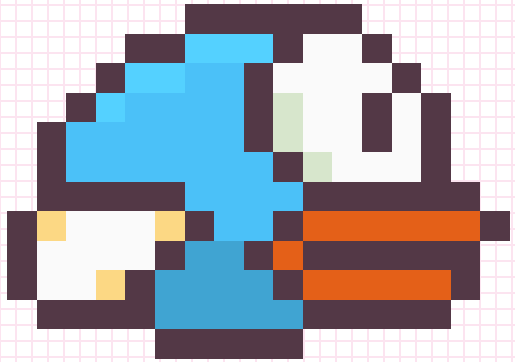


開始前先來個簡單的

```python
    # 輸入體重和身高
weight = float(input("請輸入您的體重（公斤）："))
height = float(input("請輸入您的身高（公尺）："))

# 計算 BMI
bmi = weight / (height ** 2)

# 判斷體重分類
if bmi < 18.5:
    category = "過輕"
elif bmi < 24:
    category = "正常"
elif bmi < 27:
    category = "過重"
else:
    category = "肥胖"

# 顯示結果
print(f"您的 BMI 是：{bmi:.2f}")
print(f"您的體重分類為：{category}")

```

```
請輸入您的體重（公斤）：60
請輸入您的身高（公尺）：1.65
您的 BMI 是：22.04
您的體重分類為：正常
```

Flappy Bird



程式碼連結

與我共用 › 愛拚才會營_程式課 › FlappyBird_game

類型 | 使用者 | 上次修改日期 | 來源

已使用 92% 的儲存空間 儲存空間耗盡後，你就無法建立、編輯及上傳檔案。現在購買 100 GB 儲存空間，1 個月只要 ~~$65.00~~ $0。

名稱

- img
- FlappyBird_game.py
- 程式碼_練習.docx

注意!!! 整個資料夾都要下載，圖片和程式碼在同個資料夾

程式碼解釋

```python
import pygame
import sys
import random
import os
```

★引入一些模組

```python
# 初始化 Pygame
pygame.init()

# 自動設置工作目錄為程式所在目錄
os.chdir(os.path.dirname(os.path.abspath(__file__)))
```

★初始化 Pygame

★設定工作目錄

```python
# 設定視窗尺寸
screen_width = 400
screen_height = 600
screen = pygame.display.set_mode((screen_width, screen_height))

# 設定視窗標題
pygame.display.set_caption('Flappy Bird')
```

★設定遊戲畫面的長寬

import模組

pygame ： 用來開發 2D 遊戲的模組。

sys：提供系統相關功能，例如退出程式。

random：提供隨機數，用於生成隨機的管道位置。

os：用於處理操作系統相關的功能，例如設定工作目錄。

設定畫面

pygame.display.set_mode 創建一個指定尺寸的視窗。

將視窗的標題設為 'Flappy Bird'。

程式碼解釋

```python
# 小鳥類別
class Bird:
    def __init__(self):
        self.image = bird_image
        self.rect = self.image.get_rect(center=(50, screen_height // 2))
        self.gravity = 0.25
        self.movement = 0
        self.jump_strength = -6

    def update(self):
        self.movement += self.gravity
        self.rect.y += self.movement

    def jump(self):
        self.movement = self.jump_strength

    def draw(self, screen):
        screen.blit(self.image, self.rect)
```

★ 定義小鳥類別

包含小鳥的初始化、更新動作、跳躍邏輯和繪製方法。

__init__方法

self.image: 小鳥圖片

self.rect: 小鳥的矩形，用於處理位置和碰撞

self.gravity: 重力加速度

self.movement: 垂直方向的移動速度

self.jump_strength: 跳躍時的速度

update方法

不斷更新小鳥的垂直位置，受重力影響逐漸下墜

jump方法

讓小鳥向上跳躍，設置速度為負值。

程式碼解釋

```python
# 管道類別
class Pipe:
    def __init__(self):
        self.image = pipe_image
        self.rect_top = self.image.get_rect(midbottom=(screen_width + 100, random.randint(150, 400)))
        self.rect_bottom = self.image.get_rect(midtop=(screen_width + 100, self.rect_top.bottom + 150))
        self.passed = False

    def update(self):
        self.rect_top.x -= 5
        self.rect_bottom.x -= 5

    def draw(self, screen):
        screen.blit(self.image, self.rect_top)
        screen.blit(pygame.transform.flip(self.image, False, True), self.rect_bottom)
```

★定義管道類別

管道類別用於生成和移動障礙物。

__init__方法

rect_top: 上方管道位置，隨機高度

rect_bottom: 下方管道位置，與上方管道保持固定間距

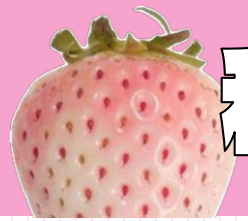
self.passed: 是否被小鳥穿過

update方法

讓管道向左移動

draw方法

在螢幕上繪製上下兩個管道，並翻轉下管道



程式碼解釋

類別

類別是一個藍圖，
用來定義物件的屬性（資料）
和行為（方法）。

類別的用途：

定義一個抽象概念（如小鳥、管道），並將其狀態和行為綁定在一起。

可以生成多個物件實例（如多隻小鳥或多根管道），每個實例都有獨立的屬性和狀態。

方法

函式 是一段可以執行的程式碼，通常用來完成特定任務或功能、避免重複程式碼。

函式的用途：

將功能模組化，像是處理事件、更新邏輯等。

在需要時重複執行同一段邏輯。可以接受參數或回傳結果。

程式碼解釋

```python
# 主遊戲函數
def main_game():
    bird = Bird()
    pipes = [Pipe()]
    score = 0
    game_over = False
```

★ 初始化

```python
while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT: pygame.quit(); sys.exit()
        # 使用 MOUSEBUTTONDOWN 檢測滑鼠點擊（或觸控）
        if event.type == pygame.MOUSEBUTTONDOWN:
            bird.jump() if not game_over else main_game()

    if not game_over:
        bird.update()
        for pipe in pipes:
            pipe.update()
            if pipe.rect_top.right < 0:
                pipes.remove(pipe)
            if pipe.rect_top.right < bird.rect.left and not pipe.passed:
                pipe.passed = True
                score += 1

        if pipes[-1].rect_top.centerx < screen_width // 2:
            pipes.append(Pipe())

        # 碰撞檢測
        for pipe in pipes:
            if bird.rect.colliderect(pipe.rect_top) or bird.rect.colliderect(pipe.rect_bottom):
                game_over = True

        if bird.rect.top <= 0 or bird.rect.bottom >= screen_height:
            game_over = True
```

★ 事件監聽

★ 遊戲邏輯

★ 結束條件

事件監聽

偵測事件，檢查玩家是否點擊滑鼠來跳躍，或重新開始遊戲

遊戲邏輯

更新小鳥和管道的位置

檢查小鳥是否碰到管道或超出螢幕邊界。

若小鳥成功穿過管道，增加分數

結束條件

如果小鳥飛出螢幕上方或碰到地面，結束遊戲

如果小鳥與任一管道發生碰撞，結束遊戲

管道類別用於生成和移動障礙物。

程式碼解釋

```
# 繪製畫面
screen.blit(background, (0, 0))
for pipe in pipes:
    pipe.draw(screen)
bird.draw(screen)
```

★ 繪製遊戲畫面

```
# 顯示分數
score_surface = font.render(str(score), True, WHITE)
score_rect = score_surface.get_rect(center=(screen_width // 2, 50))
screen.blit(score_surface, score_rect)
```

★ 顯示分數

```
pygame.display.update()
clock.tick(fps)
```

★ 更新

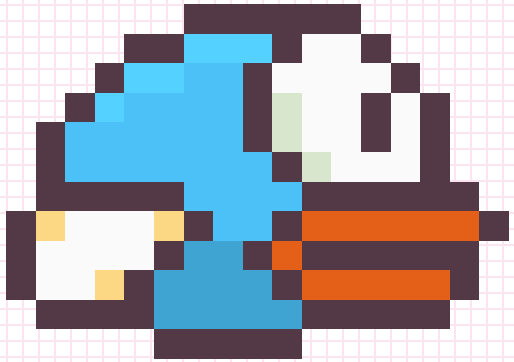
```
# 遊戲入口
if __name__ == '__main__':
    while True:
        main_game()
```

遊戲入口

遊戲從這裡開始，並不斷循環執行 main_game

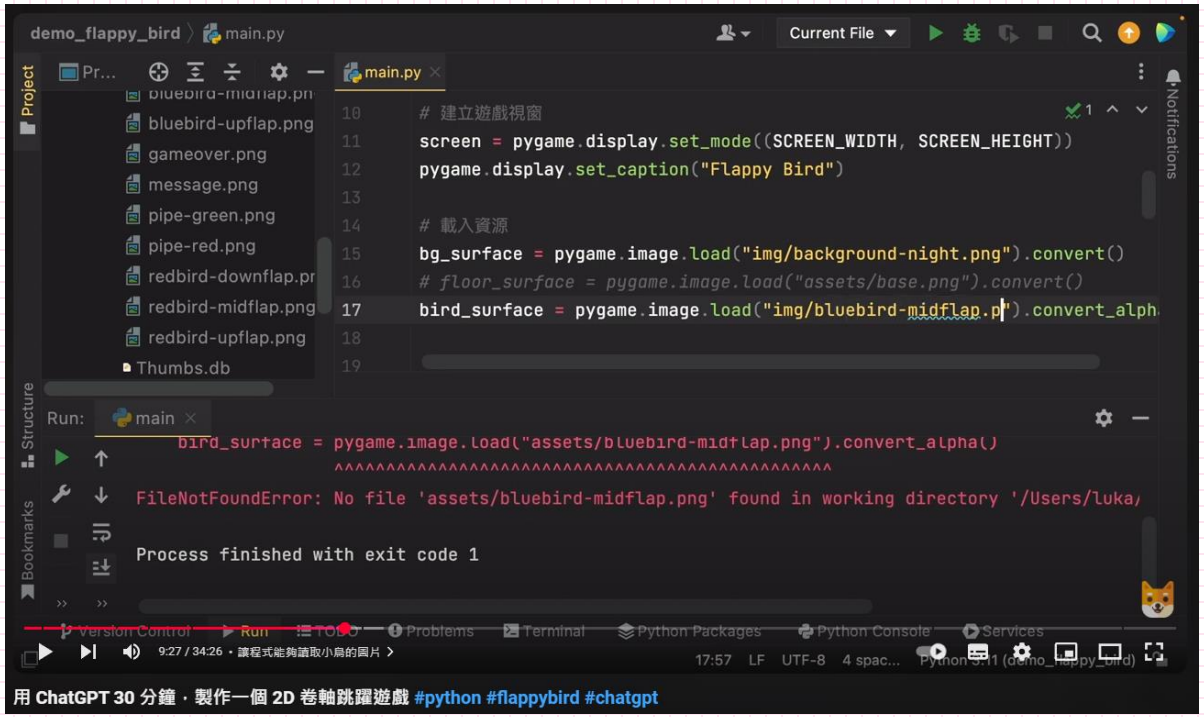
管道類別用於生成和移動障礙物。

Flappy Bird



程式碼完整版

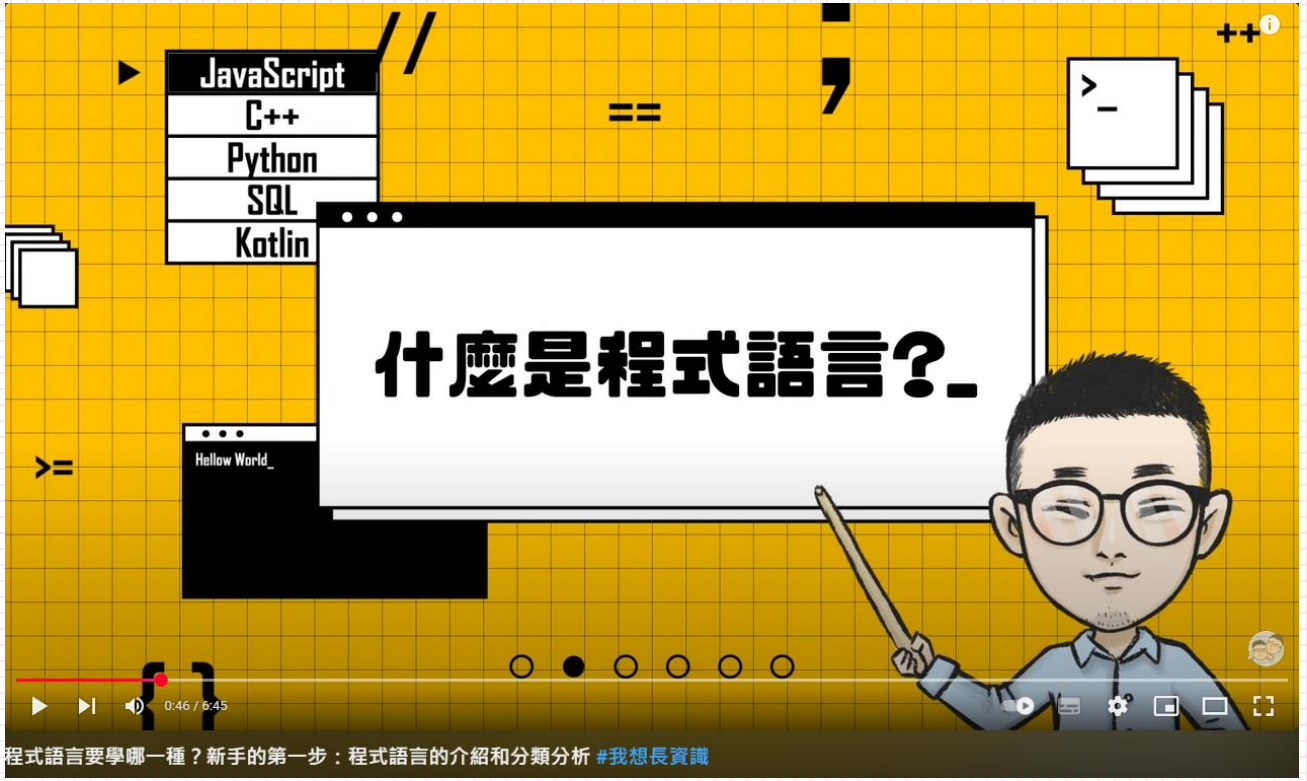
其他補充



用 ChatGPT 30 分鐘，製作一個 2D 卷軸跳躍遊戲 #python #flappybird #chatgpt

用 ChatGPT 30 分鐘，製作一個 2D 卷軸跳躍遊戲

其他補充



程式語言要學哪一種？新手的第一步：程式語言的介紹和分類分析

北 科 大 互 動 系 寒 假 體 驗 營

END

講 者：蔡 姍 珊

Hello World_