

# 歡迎來到 1141 Scripting 程式語言課程！

各位同學大家好，歡迎大家來修習 114 學年度第一學期的「Scripting 程式語言」課程。我是授課老師朱孝國，很高興能在接下來的學期與大家一起探索 Scripting 程式設計的奧妙。

在我們正式深入程式碼的世界之前，第一堂課，我想先花點時間讓大家了解這門課的目標、我們的互動方式，以及這門課的核心價值。

## 課程相關資訊

### Line 課程群組

為了方便課程討論、問題交流與公告事項，請所有同學務必掃描 QR Code 或點擊連結加入課程的 Line 群組。

<https://line.me/ti/g/6S8wMK7s3h>

### 課程資訊

- 課程代碼：9004
- 課程名稱：Scripting 程式語言(四技進修部)
- 上課教室：E547-工程館 5F
- 學分學時：[選,3, 3]
- 修課班別：職四訊四選

## 授課教師

- 姓名： 朱孝國 老師
- 服務單位： 電算中心校務資訊組
- 校內分機： 2405
- 電子郵件： [peterju@ncut.edu.tw](mailto:peterju@ncut.edu.tw)

## 這門課，到底在學什麼？

### 1. 什麼是 Scripting Language (腳本語言)？

在程式設計的世界裡，語言大致分為兩類：

- **編譯語言 (Compiled Language)**，如 C++ 或 Java：這就像蓋大樓。你需要先有完整的設計藍圖（原始碼），經過嚴謹的施工（編譯），最終建成一棟堅固、高效能的獨立建築（執行檔）。執行效率高，但開發流程較長。
- **腳本語言 (Scripting Language)**，如 Python 或 JavaScript：這更像是寫食譜。你寫下一連串指令（程式碼），然後交給大廚（直譯器）即時地、一行一行地去烹飪。食譜可以隨時修改，過程非常靈活，開發速度快。

想像一下，每學期你都要手動去課程系統下載 5 門課、每門課 18 週的投影片，總共要點擊下載 90 次！

學了 Scripting，你可以寫一個程式自動登入系統、自動抓取所有 PPT，甚至自動幫你根據「課程名稱/週次」的格式重新命名。這就是 Scripting 的核心價值：將你從重複、繁瑣、無聊的工作中解放出來！

腳本語言不像 C++ 或 Java 這類編譯語言需要先將程式編譯為機器碼。腳本語言通常會由一個直譯器逐行解釋並執行，由於不需要編譯，腳本語言的開發迭代速度較快，但執行效能通常不如編譯語言。

腳本語言的核心價值，在於扮演「黏合劑」的角色：它極其擅長串接、自動化、整合各種現有的工具與服務，讓我們能快速解決問題。它通常不需要編譯，而是由直譯器直接執行，也可以說是一種直譯語言 (interpreted language)。

## 2. Scripting Language 的特點

- 易於上手：通常語法相對簡單，讓開發者可以專注於解決問題的邏輯，而不是底層細節。
- 跨平台性：因為由直譯器執行，許多腳本語言可以不經修改就在不同的作業系統上執行。
- 動態靈活：很多腳本語言支援動態型別，你不需要事先宣告變數的資料型別，讓開發更快速。
- 應用廣泛：
  - 自動化任務：用 Shell Script 或 Python 自動化處理文件、備份資料、維護系統。
  - 網頁開發：JavaScript 是前端互動的靈魂，而 Python, PHP, Ruby 則常用於後端伺服器。
  - 資料處理與分析：Python 在資料科學和機器學習領域是絕對的王者。
  - 遊戲開發：許多遊戲引擎使用腳本語言來控制遊戲邏輯。

常見的 Scripting Language

- 作業系統內建：BASH、CMD、PowerShell
- JavaScript：主要用於網頁開發，是當前最流行的腳本語言之一。
- Python：用途廣泛的腳本語言，從網頁開發到數據分析都有使用。
- Ruby：用於網頁開發和自動化任務，擁有簡潔的語法。
- PHP：一種專注於伺服器端網頁開發的腳本語言。

常見的控制應用程式語言則有

- LUA
- VBA

注意：上述分類並非絕對，一個程式語言可能擁有多種特性。分類只是幫助我們理解，切勿僵化地看待。

## 3. 為何我們這門課的主角是 Python？

在眾多腳本語言中，我們選擇 Python 作為這學期的主角，原因有三：

1. 語法簡潔：Python 的程式碼可讀性非常高，就像在讀英文一樣，對初學者極其友好。
2. 生態圈強大：它擁有無數現成的「輪子」（套件庫），從資料科學到網站開發，從網路爬蟲到人工智慧，幾乎所有問題都有人幫你造好了輪子，你只需要學會如何使用。
3. 獨特的型別特性：
  - 動態型別 (Dynamic Type)：你不需要像 C 語言一樣先宣告變數型別。
  - 
  - `x = 5`            # Python 自動知道 x 是整數
  - `print(type(x), x)`   # `<class 'int'> 5`
  - `x = "Hello"`   # 現在 x 又可以變成字串，非常靈活！
  - `print(type(x), x)`   # `<class 'str'> Hello`
  - 強型別 (Strong Type)：Python 不會在你沒允許的情況下隨便轉換型別，這確保了程式的穩定性。
  - `a = 5`
  - `b = "10"`
  - 
  - `# result = a + b`   # 這樣會直接報錯 (TypeError)
  - `# 它不會猜測你是要把 5 轉成字串還是把 "10" 轉成數字。`
  - 
  - `result = a + int(b)`   # 你必須明確地告訴它該怎麼做！
  - `print(result)`

## 課程要求與評分標準

在開始我們的學習旅程之前，請大家務必了解本課程的評分方式與課堂紀律。

### 成績評定(Evaluation)

- 期中考 (Midterm exam): 20%
- 期末考 (Final exam): 30%
- 作業 / 小考 (Assignments / Quiz): 35%
- 上課出席率及表現 (Attendance): 15%

## 課程要求(Discipline)

- **學術誠信**：作業及考試可參考筆記、教科書、網路、甚至是 AI，但切勿直接抄襲其它同學的程式碼。任何不誠實與投機的行為，一經發現，將依校規辦理，且該次成績以 0 分計算。
- **課堂參與**：出席率及上課的參與度是平時成績的憑據，老師每堂課都會點名。
- **額外加分**：上課認真、作業及考試特別用心的同學，將另有獎勵分數。