



網站基礎與建構開發環境

Python Django網站框架與資料庫於伺服器端程式設計實務| 許季秦 | 113.03.01

學習框架之前，應先掌握的基本概念和技能



基本的程式設計概念

HTML/CSS

JavaScript

資料庫基礎知識

Python 語言

版本控制系統（如 Git）

命令列/終端機使用

基本的程式設計概念

對變數、資料型別、迴圈、條件語句等基本程式設計概念有基本的了解。

基本的程式設計概念



■變數和資料型別：

- 了解資料型別，例如整數、浮點數、字串等，對於有效管理和操作資料至關重要。

■運算子和運算式：

- 運算子用於執行各種運算，運算式是由運算子(operator)和運算元(operands)組成。

■控制流程結構：

- 循序控制流程
- 條件式控制流程（例如 if、else）
- 重複式控制流程（例如 for、while）

基本的程式設計概念



■ 函式和方法：

- 函式是一組執行特定任務的程式碼，
- 方法是與特定物件相關聯的函式。
- 程式碼分解成函式或方法有助於提高代碼的可讀性和重用性。

■ 陣列和列表：

- 一種組織和存儲資料的方式。
- 陣列和列表使開發者能夠有效地處理和操作多個資料元素。

■ 物件導向程式設計（ OOP ）：

- 強調使用物件（具有屬性和方法的實例）來組織和設計程式。
- 類別、繼承和封裝是 OOP 中的基本概念。

基本的程式設計概念



■ 例外處理：

- 處理程式執行過程中錯誤和意外情況的方式。使用 `try`、`except` 區塊來捕獲和處理異常。

■ 檔案操作：

- 瞭解如何讀取和寫入檔案是重要的，這是將資料永久保存和檢索的一種方式。

■ 演算法和資料結構：

- 演算法是解決問題的程序或方法，而資料結構則是組織和存儲資料的方式。

HTML/CSS

HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) are fundamental technologies used to create and design web pages.

HTML (Hypertext Markup Language)



■HTML (超文本標記語言)

- HTML是用於定義網頁結構的標記語言
- 使用一系列的標籤 (tag) 來構建網頁各個不同的部分

CSS (Cascading Style Sheets)



■CSS (層疊樣式表)

- CSS用於定義網頁元素的樣式和外觀
- 能夠控制字體、顏色、間距與佈局等方面
- 層疊性(Cascading)表示樣式的繼承性，即樣式可以層層堆疊
- CSS可以與HTML分離，以外部樣式表的形式存在，透過連結(link)引入HTML文件

JavaScript

是一種用於網頁開發的動態程序語言，它主要用於瀏覽器中的客戶端腳本，可以在網頁上執行交互式的操作。

JavaScript 基本概念



■ 動態程序語言:

- 動態程序語言，可以在執行時修改和操作網頁的內容、結構和樣式。

■ 客戶端程序:

- 主要用於瀏覽器端，與用戶互動，動態更新頁面，檢驗表單資料與處理事件等。

■ 語法和結構:

- 與 HTML 和 CSS 一起工作。可以直接內嵌在HTML中，也可以作為外部程序文件連結到HTML中。
- 基本語法包括變數宣告、條件語句、迴圈與函數等。

JavaScript 基本概念



■ 事件驅動:

- 響應各種事件，如使用者的點擊、輸入等，並且可以執行相應的程式碼。

■ 物件導向:

- 物件導向的語言。它使用物件（Objects）來組織代碼，允許創建和使用自定義物件。

■ 非同步編程:

- 支援非同步編程，例如使用 Call Back 函數、Promise 物件，以及 async/await 語法，允許處理非同步操作，如資料請求。

JavaScript 基本概念



■ 跨平台:

- 原本是瀏覽器端的程序語言，但隨著 Node.js 的出現，JavaScript 現在也能在伺服器端執行。

■ 第三方庫和框架:

- 當今有許多JS的框架和函式庫，如React、Angular、Vue等，用於簡化和加速前端開發。

資料庫基礎知識

Django 通常與資料庫一起使用，主要支援的是 SQLite、MySQL、PostgreSQL 等。了解基本的資料庫概念，如資料表、查詢、索引等，將對學習 Django 中的資料庫操作非常有幫助。

資料庫基礎知識



■ 資料庫 (Database) :

- 一個資料庫是組織化的資訊集合，通常以表格的形式存儲，允許有效地管理和檢索資料。

資料庫基礎知識



■ 資料表 (Table) :

- 資料庫中的一個結構，存儲特定型別的資料。它由列 (Row) 和欄 (Column) 組成。

■ 列 (Row) :

- 資料表中的一條記錄，包含了各種屬性的值。
- 每一列通常代表資料庫中的一個實例或項目。

■ 欄 (Column) :

- 資料表中的一個字元或字元組合，代表特定型別的資料。
- 每一欄通常代表了一個屬性。

資料庫基礎知識



■主鍵 (Primary Key) :

- 資料表中用來唯一識別每一行的欄位。
- 確保每個記錄都有唯一的標識。

■外鍵 (Foreign Key) :

- 一個欄位，它與另一個資料表的主鍵相關聯，用於建立不同表之間的關聯。

■索引 (Index) :

- 索引是一種資料結構，用於加速資料檢索操作。通常建立在一個或多個欄位上。

資料庫基礎知識



■ SQL (Structured Query Language) :

- 用於管理和操縱資料庫的語言。
- 包括了查詢 (Query)、插入 (Insert)、更新 (Update) 和刪除 (Delete) 等操作。

■ 資料庫管理系統 (Database Management System, DBMS) :

- 一個軟體，負責管理資料庫的建立、修改、查詢和維護。
- 常見的例子包括MySQL、PostgreSQL和SQLite。

■ 正規化 (Normalization) :

- 一個資料庫設計的過程，目的是減少資料冗餘並提高資料的一致性。
- 通常涉及將資料分解成不同的資料表。

Python 語言

Django 是用 Python 語言編寫的，需要對 Python 有基本的了解。這包括語法、函式與模組等。

Python 語言



■簡潔易讀：

- 語法設計強調簡潔和易讀性。

■動態類型：

- 是一種動態類型的語言，可以在執行時更改變數的資料型別。

■物件導向：

- 支援物件導向程式設計（OOP），允許開發者使用類別、物件、封裝和繼承等概念組織和設計程式碼。

Python 語言



■ 廣泛的標準函式庫：

- 豐富且強大的標準庫，包括各種模組和套件，用於處理文件、檔案、資料結構與網路通信等多個領域。

■ 直譯式：

- 直譯式語言，可以逐行執行程式碼，更容易進行測試。

■ 跨平台：

- 跨平台的語言，可以在不同的作業系統上運行，包括Windows、Linux和Mac。

Python 語言



■ 豐富的社群支持：

- 活躍且龐大的社群，可以輕鬆找到大量的資源、文件和支援。

■ 動態記憶體管理：

- 提供了動態記憶體管理，開發者無需操心手動分配和釋放記憶體。

■ 多用途：

- 適用於多種應用領域，包括Web 開發、資料科學、人工智慧、機器學習、自動化與遊戲開發等。

版本控制系統（如 Git）

在團隊中協作開發時，版本控制是必不可少的Git 將有助於更有效地管理程式碼版本。

版本控制系統（如 Git）



- 版本控制系統（ Version Control System，縮寫為VCS）是一種記錄和管理項目中文件變更的系統。
- 允許開發者或團隊追蹤項目的變更，協同工作，並隨時回溯到先前的狀態。
- Git 是目前最廣泛使用的版本控制系統之一。

命令列/終端機使用

使用終端機或命令列介面是開發過程中的一個重要技能，特別是在安裝套件、執行程序和進行版本控制時。

一種文字介面，透過鍵入指令來與電腦系統進行互動。

命令列/終端機使用



■ 指令輸入：

- 可以在命令列中鍵入指令。
- 指令是一些控制電腦系統行為的文字指令。

■ 路徑和檔案系統：

- 可以瀏覽電腦檔案系統，查看和操作檔案和目錄。

■ 執行程式：

- 可以直接執行可執行檔或程序文件，只需在命令列中輸入檔案的路徑或指令。

命令列/終端機使用



■ 套件管理：

- 可以透過命令列來安裝、更新和管理軟體套件。

■ 網路工具：

- 可以使用命令列進行簡單的網路操作。

■ 版本控制：

- 可以使用命令列來與版本控制系統進行互動。

命令列/終端機使用



■ 虛擬環境：

- 使用命令列設置和管理虛擬環境，這對於隔離和管理專案依賴性非常有用。

■ 系統管理：

- 系統管理員可透過命令列執行系統管理任務，例如設置網路配置、檢查硬體信息、管理用戶和權限等。

