

The background of the slide is a light gray gradient. It is decorated with numerous realistic water droplets of various sizes. Some droplets are large and prominent, while others are small and subtle. They are scattered across the slide, with a higher concentration in the top-left and bottom-right corners, framing the central text.

JAVA 簡介

第 1 章

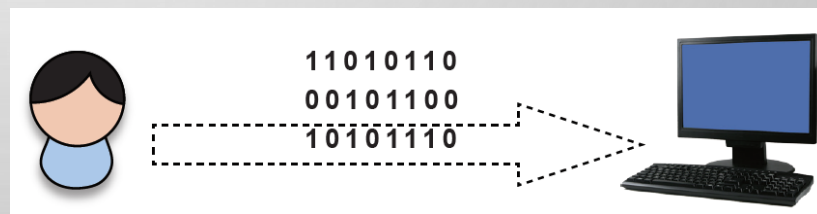
0 與 1 的語言世界

- 電腦是由眾多電子零件所組成，電腦中的運算核心：中央處理器（CPU）更是由數十億個電晶體所組成，而這些電晶體元件，有些設計製造成記憶體元件，用來儲存運算中的資料。



■ 機械語言

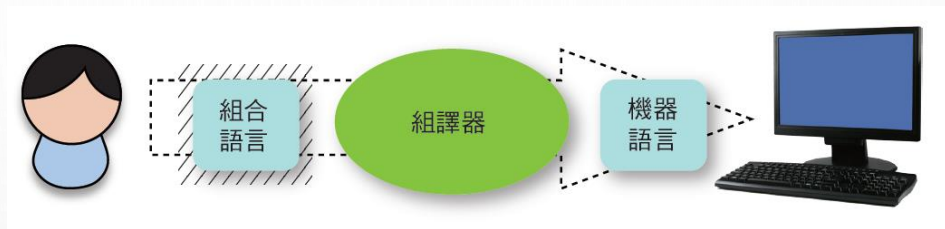
- 數十億個電晶體元件組成了電腦的核心，控制電腦運作的方式自然也是經由 0 與 1 的數列所組成，所以在電腦的世界，便是利用 0 與 1 進行溝通。
- 為了區分這種單純經由 0 和 1 與電腦溝通的程式語言，學界就將其稱為**機械語言**（**MACHINE LANGUAGE**），而這也是電腦語言發展中的第一代程式語言。



■ 組合語言

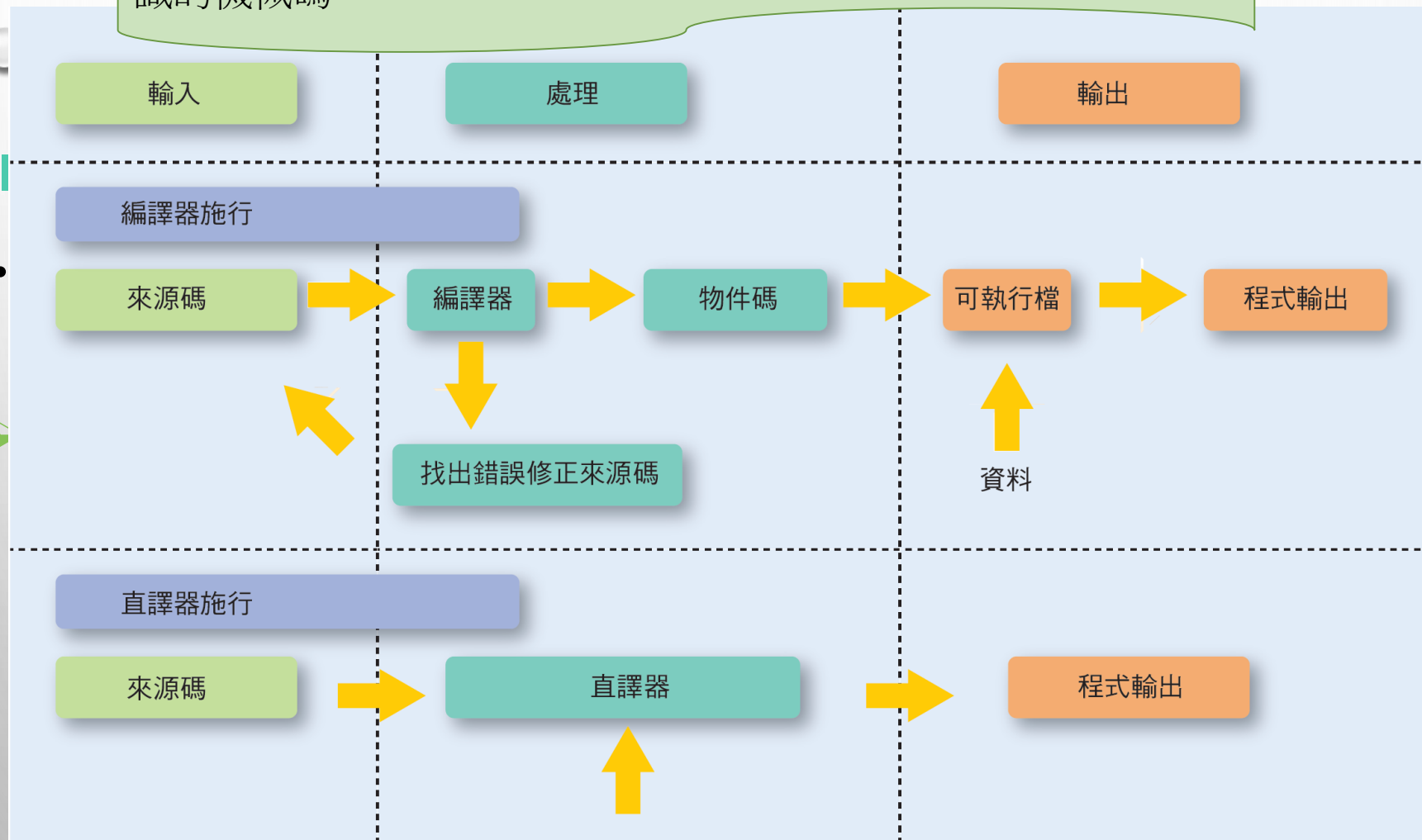
- 由於以機械語言來開發程式極為不便，因此工程師便將機械語言中 0 與 1 所組成的序列，使用特定的英文或數字來代表，將一連串只有 0 與 1 的數字，變成一句句對人類而言較容易看懂的語句，這些用來取代的符號稱之為「助憶符」（**MNEMONICS CODE**），而採用助憶符所撰寫的程式語言，稱為「組合語言」（**ASSEMBLY LANGUAGE**），這便是程式發展過程中的第二代程式語言。

■ 低階語言



- 組合語言只是將原本不易閱讀的機械語言改寫為較易辨識的組合語言，但它的開發跟設計方式還是非常接近機械語言，程式設計人員仍須對處理器與硬體運作方式有一定瞭解，設計出的程式碼所能適用的CPU也仍有相當局限，難以跨越廠牌鴻溝。
- 相較於機械語言，組合語言雖然已經較容易撰寫，但若對電腦結構不夠瞭解，還是無法深入撰寫組合語言，且組合語言對不同CPU的程式移植性不佳，於是，為改善這些缺點，便發展出第三代程式語言：高階語言（HIGH-LEVEL LANGUAGE）。

編譯器與直譯器的目的，都是翻譯來源碼，並轉換成電腦可辨識的機械碼。



■ 編譯器與直譯器

- 編譯器的功能當然不限於將原始碼轉換成執行檔，在編譯原始碼的過程中，編譯器也會檢查程式碼語法，一旦發現原始碼有語法錯誤，編譯器會標示出可能的錯誤位置，提醒程式設計人員除錯（**DEBUG**）。
- 「直譯器」則是透過一套專屬的程式，直接下達命令（**COMMANDS**）（或稱為指令），和電腦互動。

高階語言

- 在編譯器及直譯器發展出來後，程式設計不用再相依於處理器的架構，或採用何種低階語言設計，也因此大型的程式開發都是採用高階語言來進行，以提高程式的易維護性與可移植性。

■ 高階語言的先驅：COBOL 與 FORTRAN

COBOL

- 中文稱為：通用商業導向語言/ 面向常規業務型語言/ 常規商業信息處理語言
- 是最早的高階程式語言，正式發布於1960年，屬於第三代的商用程式語言，主要應用在財會工作、統計報表、計劃編制、情報檢索、人事管理等數據管理及商業數據處理等領域。

FORTRAN

- 第三代程式語言，於1953年由IBM公司所發展，其命名方式是由「公式轉換器」(formula translator, FORTRAN) 的簡稱而來，當初主要是為了進行科學、數學及工程計算所發展。

■ 結構化與模組化語言：Ada Basic

使用Ada語言顯示Hello World! 的程式範例。

- **ADA 程式語言**於1980年問世，它的命名是為了紀念英國數學家設計師 **ADA LOVELACE** 夫人，而其發展起源於一項美國軍方的計畫，主要目的在

- AI

```
with Ada.Text_IO;
```

```
procedure Hello is  
begin
```

```
  Ada.Text_IO.Put_Line("Hello World!");  
end Hello;
```

■ 結構化與模組化語言：ADA、C、BASIC、Visual Basic

- **C 語言**是由AT&T貝爾實驗室（BELL LABS）所開發，於1972年正式推出，最初用於移植與開發UNIX 作業系統。

```
# include <stdio.h>
int main(void)
{
    printf("Hello World\n");
    return 0;
}
```

使用C語言顯示Hello World的程式範例。

物件導向概述

- 結構化與模組化程式語言是大型軟體系統的敲門磚，但最後集所有功能於大成的，則是物件導向（**OBJECT-ORIENTED, OO**）語言。

➤ 目前流行的幾種物件導向程式語言：

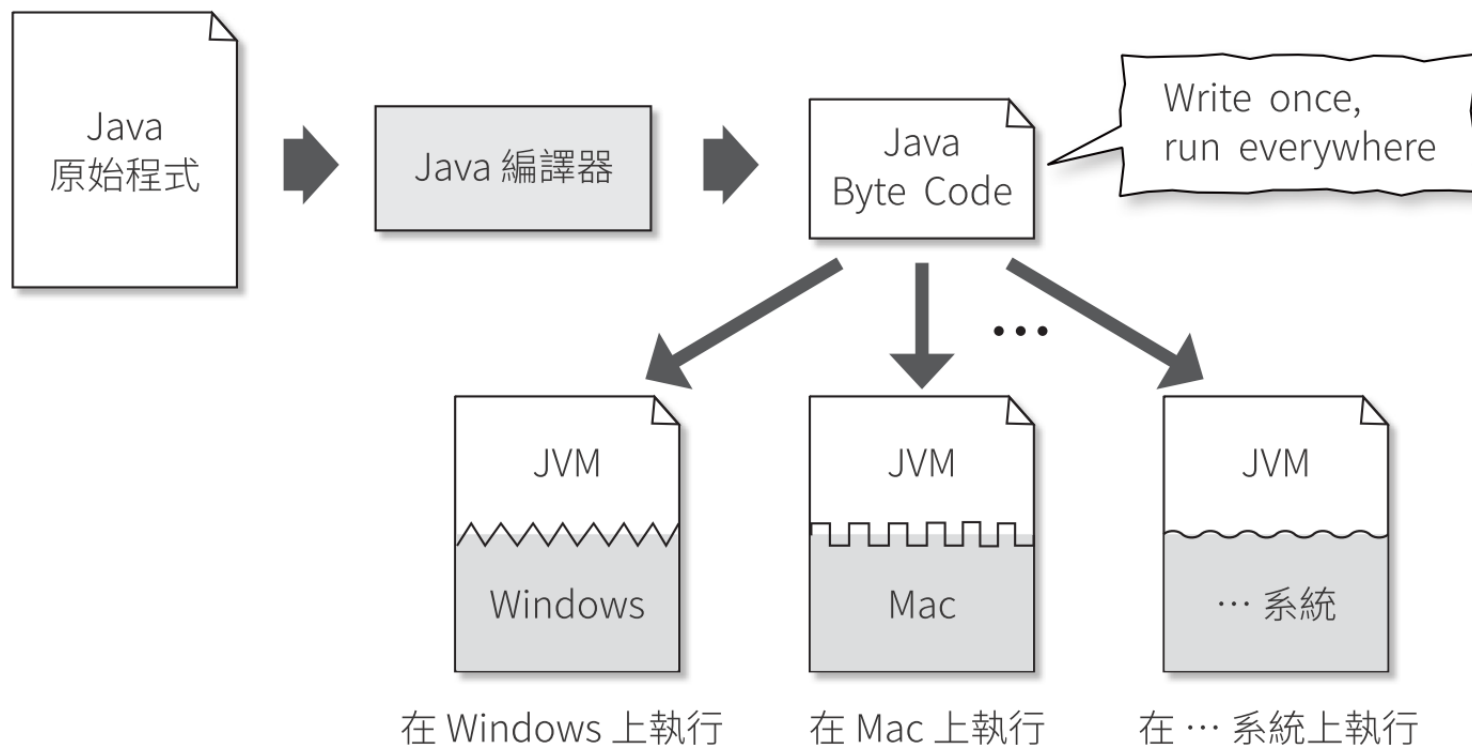
一、C ++

- 1980年代由貝爾實驗室開發而成，是由C語言整合了物件導向的概念演進而來。

二、Java

- 1995年由昇陽（Sun Microsystems）公司所發布，目前該公司已隸屬於甲骨文（Oracle）旗下。

1-1 JAVA 程式語言的特色



1-1 JAVA 程式語言的特色

- 網路功能 (NETWORKING)
- 物件導向
- 開發工具隨處可得

1-2 JAVA 平台簡介

- **JAVE ME (JAVA MICRO EDITION) :**

泛指消費性裝置上的 JAVA 執行環境，
像是功能手機、機上盒等等

- **JAVE SE (JAVA STANDARD EDITION) :**

這是一般應用的執行環境，
像是常見的個人電腦上所提供的JAVA執行環境就屬於這一種

- **JAVE EE (JAVA ENTERPRISE EDITION) :**

這是為了商業運用所建構的平台，
提供大規模運算所需的能力

1-3 ANDROID 與 JAVA

- 開發 ANDROID 智慧手機應用程式 (ANDROID APP, 或簡稱 APP) 所用的程式語言就是 JAVA

Android Studio

The Official IDE for Android

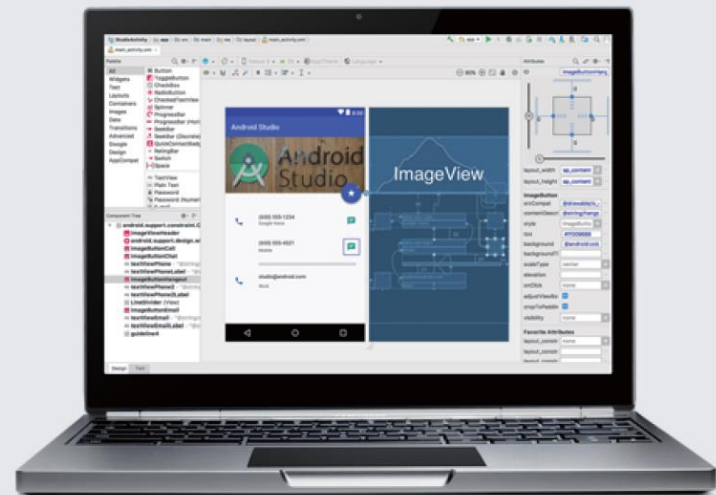
Android Studio provides the fastest tools for building apps on every type of Android device.

World-class code editing, debugging, performance tooling, a flexible build system, and an instant build/deploy system all allow you to focus on building unique and high quality apps.

DOWNLOAD ANDROID STUDIO
3.1.1 FOR WINDOWS (758 MB)

> [Read the docs](#)

> [See the release notes](#)



- 開發 ANDROID APP 需使用 JAVA 程式語言外, 也要使用 ANDROID SDK (SOFTWARE DEVELOPEMENT KIT), 其中有許多 ANDROID 平台特有的類別庫、API 等等