

» 什麼是 Web（全球資訊網）？

透過網路提供的機制

如今，我們的生活或工作早已少不了網路。廣義上的「全球資訊網」是指**透過網路提供或交換資訊、商業題材的機制**（圖 1-1）。

若從早期的 Web 技術發展史來看，Web 是 World Wide Web 的簡稱，而 World Wide Web 又可簡稱為 **WWW**，但簡單來說，Web 就是透過網路提供**超文件**的系統。

由於 Web 技術不斷地進化，現在大部分的資訊系統都可透過網路建構，本書也將根據 Web 技術的發展過程，解說目前的 Web 技術與系統。

透過超連結建立關聯性

組成網站的每一張網頁都是透過超連結或參照與其他網頁建立關聯性，而地球上的大量文件與資訊，也是從某個網站傳輸至另一個網站（從某個網頁伺服器移動至另一個網頁伺服器），藉此跨越大海與國界。這種稱為**超連結**的機制主要是利用以超文字標示語言（HyperText Markup Language，簡稱 HTML，請參考**2-3**）寫成的網頁建置。在以超文字製作的頁面植入超連結，**就能移動到其他的頁面**（圖 1-2）。

業務系統則是從選單畫面呼叫各種處理或程式，等到處理或程式結束後，再回到選單的構造，但網站則通常是透過連結切換頁面。

接著讓我們一起了解網路系統的構造。

圖 1-1

Web 的概要

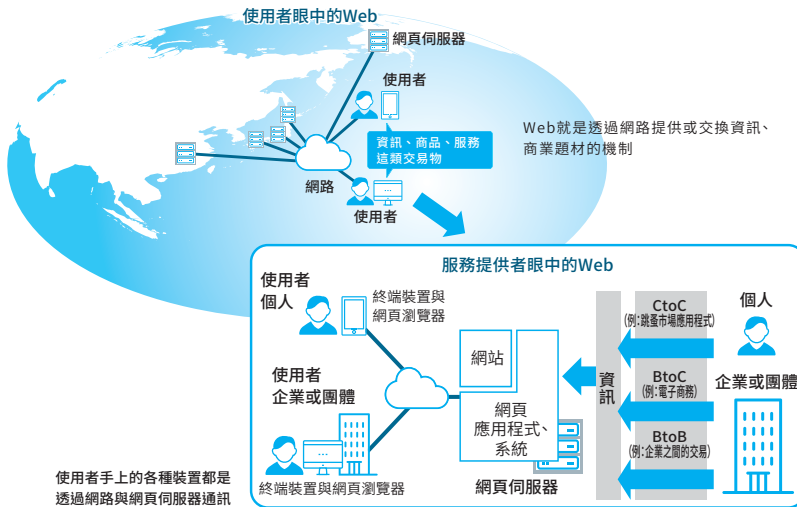
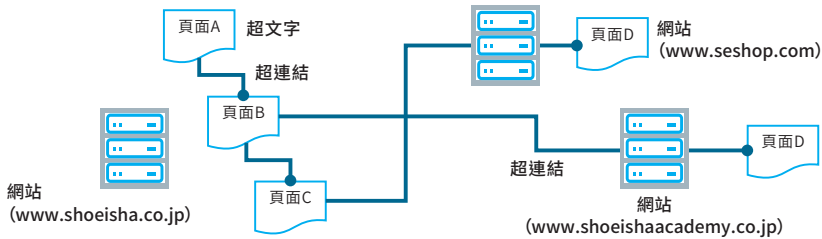


圖 1-2

超文字與超連結

超連結除了可串連網站內部的網頁，也能連接其他網站與網頁



●超文字之內(網頁之內)的連結設定

頁面C

在網頁的超文字植入超連結

頁面C的超連結

頁面D

頁面E是
seshop.com/index.html
這種不會與其他網站或網頁重複的獨特名稱

Point

- Web 就是透過網路提供或交換資訊、商業題材的機制。
- Web 可透過超文字與超連結這類機制移動至其他頁面或網站。

小試身手

網站的規模

要衡量網站的規模，可計算網站的總頁數。

假設將網站分成大、中、小三種規模，筆者認為可依照下列的總頁數分類網站。大規模的網站需要耗費不少心思經營與維護。

網站的規模與總頁數

規模	總頁數
小規模	100頁以內
中規模	100~1,000頁以內
大規模	超過1,000頁
超大規模	超過10,000頁

雖然企業或商用的網站很常超過1萬頁，但仔細一看就會發現，很多頁面都是在十年之前做好的，根本沒有人會瀏覽。上述的總頁數是以「還有人瀏覽的頁面」為計算基準。此外，明明是同一種商品，卻因為顏色不同而於不同的頁面刊載照片，導致總頁數超過1,000頁的情況，不能與介紹不同商品的頁數超過500頁的情況相提並論。

計算總頁數的範例

要計算網站的總頁數可使用 Google 的 site: 指令。比方說，要計算 shoeisha.co.jp 的總頁數可在 Google 的搜尋方塊輸入「site:shoeisha.co.jp」，就會顯示 23,000 頁這個數字。site: 指令只會計算 Google 找到的頁面，所以與實際的總頁數可能會有誤差，但還是能窺網站的規模，有機會的話請大家務必試用看看。

» Web 技術的變遷

越來越普及的應用領域

在第 1 章說明了網際網路的基本知識之後，本章要開始講解一些與技術有關的題目，不過在進入正題之前，先為大家介紹 Web 技術在過去十年的演變。

早期的資訊系統稱為 **SoR** (System of Record：記錄系統)，主要是由使用的機構或組織管理，而現在則以 **SoE** (System of Engagement：參與性系統) 為主流，越來越多串連組織或個人的系統出現。

從 SoR 到 SoE 的進化，也可說是以**閱讀為主的網站演化為收集各類資訊的網頁應用程式或系統**的過程。請大家先了解資訊系統在過去有如此大規模的進化過程 (圖 2-1)。

不斷改變的開發型態

這些技術背後的開發與維護除了會在本章解說，也會在第 8 章進一步說明，而這些開發與維護的技術也隨著時代一起演進。

早期是從零開始開發，後來則進化成利用範本或框架開發，最後再升級為利用既有的服務或 API 開發。從重視獨特性與專業性的機制演化為重視通用性、實用性的機制，而這也是盡可能不要寫程式的**低程式碼**或**無程式碼**的開發型態。之所以會出現這種開發型態，與終端裝置、網路快速演化的時代背景有關 (圖 2-2)。

為了靈活地應用資訊，就必須更重視各種資訊的流通以及系統的串連，如此一來，才能打造各類資訊於不同的系統之中流通的循環，市場才會跟著擴大。Web 技術就像是 一面照出時代樣態的鏡子。

下一節將為大家介紹更多的細節。

圖 2-1

從閱讀到應用資訊

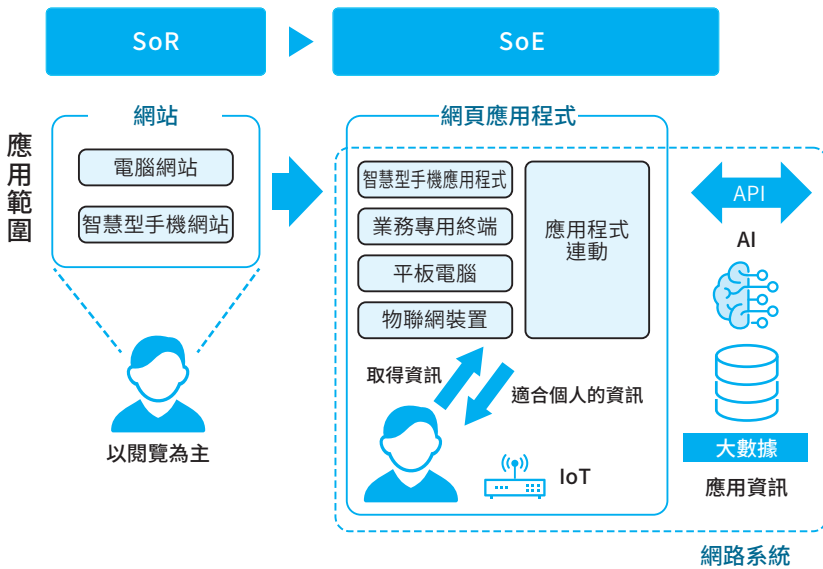
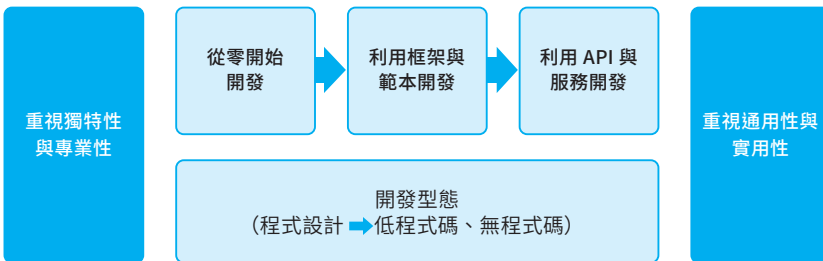


圖 2-2

開發型態的演化



比方說，在開發前台的畫面時，會使用 Angular、React、vue.js 這類框架（參考 2-10、8-4），不會從零開始製作。

Point

- Web 技術從以閱覽為主，演化為以應用資訊為主的潮流。
- 開發也不再是從零開始，而是利用框架或服務開發。

» 將用戶端與伺服器一分為二的思維

開發網頁的傳統技術

如 2-9 的 CGI，要讓網頁執行動態處理通常會使用腳本語言開發網路系統，而不是使用 HTML 這種標記語言。這時候使用的腳本語言包含在用戶端執行腳本語言的技術，與在伺服器端執行的技術。

接下來說明的網路系統技術都不會在網頁瀏覽器或不使用網路的業務系統使用。這是透過網頁瀏覽器與網路的網頁伺服器或其他伺服器連線或使用 API 的技術。

用戶端的腳本語言包含 JavaScript 或 TypeScript 這類語言，通常會在動態處理較為複雜的頁面使用。

至於伺服器端的腳本語言則包含 CGI、SSI、PHP、JPS、ASP.NET，越後面的語言難度越高，但功能也越強大，例如 JSP 或 ASP.NET 會在相對規模較大的網路系統使用。在此為大家將各項技術的特徵整理成圖 2-19。除了上述這些之外，當然還有很多語言，但以上是近年來較具代表性的使用語言。

可在網頁瀏覽器執行處理

圖 2-20 根據圖 2-19 整理了各項技術的定位，可以發現左側是與使用者介面相近的 HTML 或 CSS，也可透過 Node.js 在伺服器端使用 JavaScript。

技術的種類雖多，但不同的系統規模與功能的複雜度，適用不同的 Web 技術。

圖 2-19

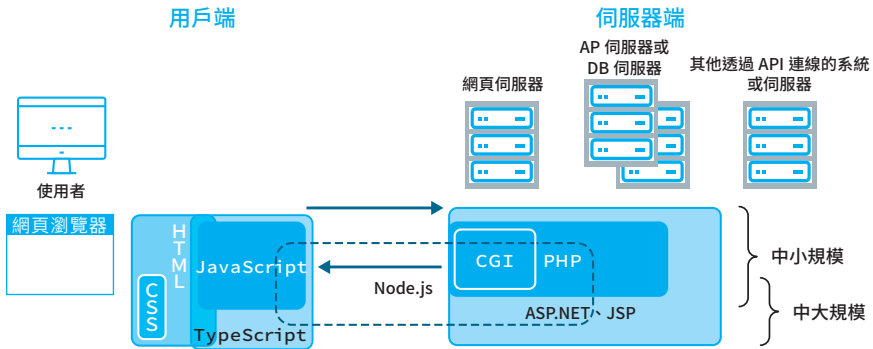
網頁專屬技術的概要

	網頁專屬技術	特 徵	開發單位
用戶端	JavaScript	<ul style="list-style-type: none"> 於用戶端使用的經典技術 語法與 HTML 或 CGI 相近，所以很容易閱讀 	Netscape
	TypeScript	<ul style="list-style-type: none"> 與 JavaScript 相容，所以也於大型 APP 使用 建議想開始學習腳本語言的人使用 	Microsoft
伺服器端	CGI (Common Gateway Interface)	目前已是動態頁面的基本框架	NCSA
	SSI (Server Side Include)	<ul style="list-style-type: none"> 將命令嵌入 HTML，就能製作陽春版的動態頁面 早期用於計算瀏覽次數與顯示日期，目前已不再使用 	NCSA
	PHP	與 HTML 檔案相容，常用於開發購物網站	The PHP Group
	JSP (Java Server Pages)	<ul style="list-style-type: none"> 如果是 Java 平台就會使用這種技術 若要開發大規模網頁就會使用 JSP 或 ASP 	Sun
	ASP.NET (Active Server Pages.NET)	可全面應用微軟技術的網路系統框架	Microsoft

- 有適用各領域的開發框架
- JavaScript 的範例：jQuery、vue.js、React、Angular ※ Angular 是由 Google 開發的框架
- CGI 的例子：Catalyst (Perl)
- JSP 的例子：Struts、SeeSea (都是 Java)
- 其他：Django (Python) 或 Ruby on Rails (Ruby)
- ※ () 括號內的是程式設計語言

圖 2-20

網頁專屬技術的定位



- TypeScript 也可用來開發大型系統
- 用戶端的技術必須透過網頁瀏覽器使用
- Node.js 是於伺服器建立 JavaScript 執行環境的平台，讓 JavaScript 或 TypeScript 得以在伺服器端執行
- ASP.NET 或 JSP 是能用來開發大規模、大範圍系統的平台

Point

- ✍ 制式的動態處理幾乎都使用相同的技術開發。
- ✍ 不同的系統規模與功能會使用不同的技術開發。

圖 2-27 Session 的概要

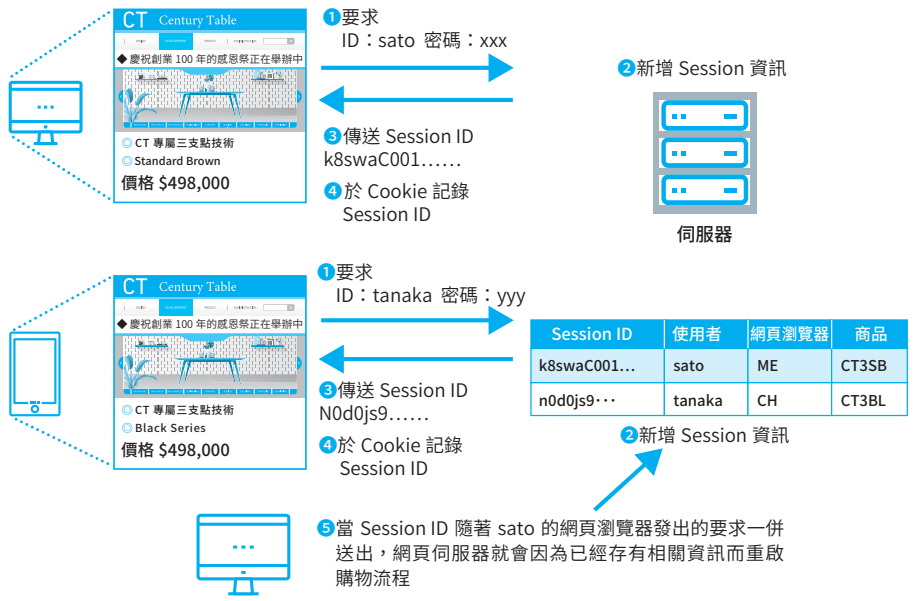
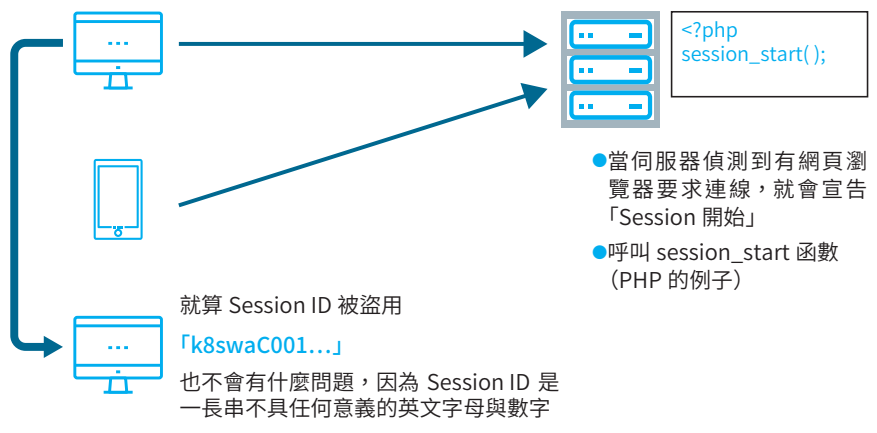


圖 2-28 Session 啟動與 Session ID 的範例



Point

- 伺服器端是利用 Session ID 管理網頁瀏覽器與伺服器之間的 連串處理。
- Session ID 本身是 串沒有任何意義的英文字母與數字。

小試身手

HTML 與 CSS

在第 2 章說明過 HTML 與 CSS 的內容。自 HTML 5 問世後，網頁基本上會以 HTML 定義內容與文章的架構，再以 CSS 定義外表與版面的編排與設計。

HTML 或 CSS 可利用 Windows 附屬的記事本、文字編輯器或 Word 製作，所以大家不妨參考圖 2-8，試著寫寫看 HTML。

圖 2-8 有 3 個 HTML 檔案與 1 個 CSS 檔案。在此要試著利用 2 個 HTML 檔案與 2 個 CSS 檔案觀察 CSS 造成的影響。

CSS 造成的影響

請試著在格式幾乎相同的 2 個 html 檔案，以及定義內容各不相同的 2 個 CSS 檔案撰寫簡單的程式碼。範例如下。請將這些檔案的副檔名設定為 .html 與 .css，再放在同一個資料夾裡。

公司簡介與徵才資訊 html

分別設定背景色與標題的 CSS

<p>◆事業內容</p> <ul style="list-style-type: none">• 出版業• 書籍銷售	<p>◆徵才重點</p> <ul style="list-style-type: none">• 對 Web 技術有興趣• 不需相關經歷	<pre>body{background: #eeeeee } h1{font size: 22px: color:blue }</pre>	<pre>body{background: #ffffff } h1{font size: 22px: font family: serif: color:black }</pre>
---	---	--	---

這次使用了 2 個 html 檔案與 2 個 css 檔案。雖然也可以在 HTML 檔案定義網頁的外觀，但是當網頁變多或是設計更加複雜時，就能體會到 CSS 有多麼方便。

» 現存系統難以雲端化的原因

雲端伺服器是虛擬伺服器

要讓現有的系統轉型為 Web 系統，最快的方式就是將系統移植到整備完善的雲端環境，但此時會遇到的障礙就是雲端服務提供的伺服器通常是**虛擬伺服器**。虛擬伺服器又稱 Virtual Machine (VM)。

若以實體伺服器為例，虛擬伺服器就是以虛擬的方式在一台伺服器之中建置多台伺服器的功能(圖 5-7)。在實體伺服器安裝虛擬環境專用的軟體就能建置虛擬伺服器，但雲端服務還是主機代管服務的使用者，通常都很難察覺自己使用的是虛擬伺服器。

確認是否為虛擬環境

VMWare、Hyper-V、OSS(開源軟體)的 Xen、KVM 都是非常有名的虛擬環境軟體。雖然也可在實體伺服器建置虛擬伺服器，但從圖 5-8 可以發現，在虛擬環境軟體建置虛擬伺服器比較簡單，而且還能透過監控軟體監控多台伺服器的運作狀況。

目前已有不少企業或團體的系統採用虛擬伺服器，但也有許多老舊的系統尚未移植到虛擬環境。

雲端服務業者或大量提供主機代管服務的網路供應商都因為有虛擬伺服器，才能有效率地維護伺服器。

需要更新的系統通常都是老舊的系統，所以得**確認這些系統是否已移植到虛擬環境**。假設已移植到虛擬環境，又採用同一套虛擬環境軟體，就能更有效率地移植至雲端服務或主機代管服務。

圖 5-7

虛擬伺服器的概要

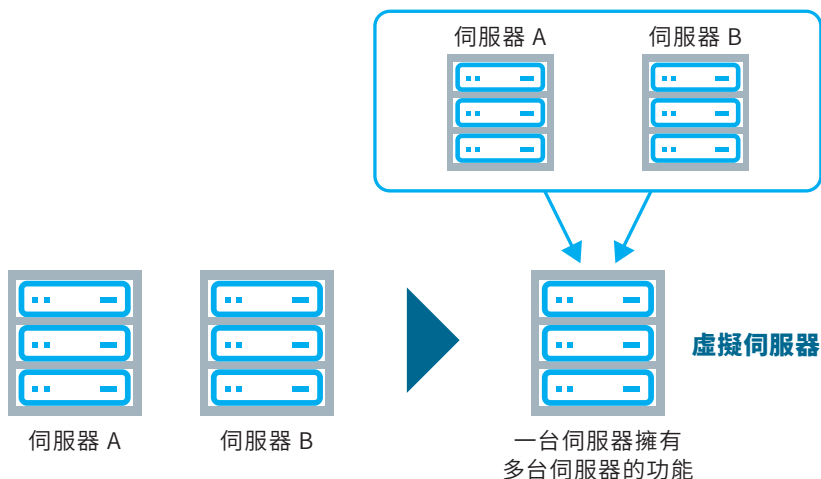
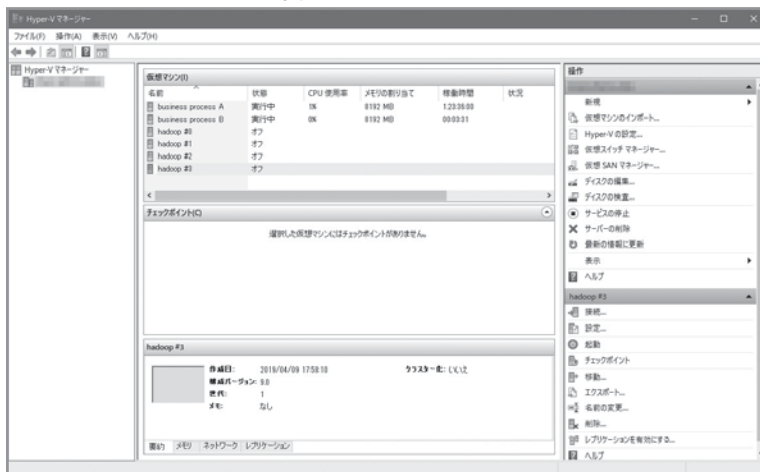


圖 5-8

虛擬伺服器的外觀

Hyper-V 管理者畫面



這是在 1 台實體伺服器建置 business process A、B、hadoop #0 ~ #3，共 6 台虛擬伺服器。

Point

- 雲端服務與主機代管服務通常是以虛擬伺服器提供服務。
- 要讓既有的系統雲端化，必須先確認是否已虛擬化。

小試身手

ping 命令

ping 命令是在業務系統或 Web 常用的命令，主要是從終端裝置確認另一端的裝置（此為網頁伺服器）是否連線。若以業務系統為例，會利用 ping 命令確認是否與伺服器連線。這項命令可於 Windows PC 與 Linux 的終端裝置使用，有機會請大家使用看看。若是 Windows 電腦，則需在命令提示字元模式下執行。

ping 命令執行之後的畫面

這個範例是於 ping 命令後面輸入 IP 位址。

左側為 Windows 電腦的畫面，右側為 Linux 電腦的畫面。雖然內容有些不同，但大致上都能取得相同的資訊。

在 Windows 執行 ping 命令的畫面

```
C:\>ping 182.22.59.229

182.22.59.229 に ping を送信しています 32 バイトのデータ:
182.22.59.229 からの応答: バイト数 =32 時間 =8ms TTL=53
182.22.59.229 からの応答: バイト数 =32 時間 =8ms TTL=53
182.22.59.229 からの応答: バイト数 =32 時間 =7ms TTL=53
182.22.59.229 からの応答: バイト数 =32 時間 =7ms TTL=53

182.22.59.229 の ping 統計:
    パケット数: 送信 = 4、受信 = 4、損失 = 0 (0% の損失)、
    ラウンドトリップの概算時間 (ミリ秒):
    最小 = 7ms、最大 = 8ms、平均 = 7ms
```

在 Linux 執行 ping 命令的畫面

```
[root@localhost ~]# ping 182.22.59.229
PING 182.22.59.229 (182.22.59.229) 56(84) bytes of data.
64 bytes from 182.22.59.229: icmp_seq=1 ttl=31 time=159 ms
64 bytes from 182.22.59.229: icmp_seq=2 ttl=31 time=160 ms
64 bytes from 182.22.59.229: icmp_seq=3 ttl=31 time=160 ms
64 bytes from 182.22.59.229: icmp_seq=4 ttl=31 time=159 ms
^C
--- 182.22.59.229 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 159.959/159.997/160.036/0.490 ms
```

即使直接在 ping 命令輸入網域名稱，也會得到相同的結果。之前曾在第 3 章的「小試身手」使用 nslookup 命令取得 IP 位址，大家可試著在 ping 命令後面輸入這些 IP 位址。

除了 ping 命令之外，網路相關命令還有很多種，例如 ipconfig（Windows 電腦的命令，Linux 的是 ifconfig）、tracert（Windows 電腦的命令，Linux 的是 traceroute）、arp（Windows 與 Linux 相同）。