在上面的代碼 allCookies 中是一個字符串，其中包含一個以分號分隔的所有 cookie（即key=value 對）列表。Cookie 是存儲在您計算機上的小型文本文件中的數據。每個鍵和值都可能被空格（空格和製表符）包圍：事實上，RFC 6265 要求每個分號後有一個空格，但一些用戶代理可能不遵守這一點。
;path=path（例如，'/'，'/mydir'）如果未指定，則默認為當前文檔位置的當前路徑。
;domain=domain（例如，'example.com' 或 'subdomain.example.com'）。如果未指定，則默認為當前文檔位置的主機部分。與早期的規範相反，域名中的前導點被忽略，但瀏覽器可能拒絕設置包含這些點的 cookie。如果指定了域，則始終包含子域。
;max-age=max-age-in-seconds（例如，60*60*24*365 或 31536000 一年）
;expires=date-in-GMTString-format 如果既沒有expires 也沒有 max-age 指定，它將在會話結束時過期。
;secureCookie 只能通過 https 等安全協議傳輸。在 Chrome 52 之前，此標誌可能與來自 http 域的 cookie 一起出現。
;samesiteSameSite 會阻止瀏覽器將此 cookie 與跨站點請求一起發送。可能的值為 lax,strict 或 none。
;samesiteSameSite 會阻止瀏覽器將此 cookie 與跨站點請求一起發送。可能的值為 lax,strict 或 none。
strict 值將阻止瀏覽器在所有跨站點瀏覽上下文中將 cookie 發送到目標站點，即使在遵循常規鏈接時也是如此。
none 值明確指出不會應用任何限制。cookie 將在所有請求中發送 - 跨站點和同站點。
cookie 值字符串可用於 encodeURIComponent()確保字符串不包含任何逗號、分號或空格（cookie 值中不允許使用這些字符）。
__Secure-向瀏覽器發出信號，它應該只在通過安全通道傳輸的請求中包含 cookie。
__Host-向瀏覽器發出信號，除了僅使用來自安全來源的 cookie 的限制之外，cookie 的範圍也僅限於服務器傳遞的路徑屬性。如果服務器省略了路徑屬性，則使用請求 URI 的“目錄”。它還表示域屬性不能存在，這會阻止 cookie 被發送到其他域。對於 Chrome，路徑屬性必須始終是原點。

Cookies 通常在 Web 應用程序中用於識別用戶及其經過身份驗證的會話。從 Web 應用程序中竊取 cookie 會導致劫持經過身份驗證的用戶的會話。竊取 cookie 的常見方法包括使用社交工程或利用應用程序中的跨站腳本(XSS) 漏洞。
擁有的 cookie 越多，每次請求在服務器和客戶端之間傳輸的數據就越多。這將使每個請求變慢。

訪問器屬性語法的原因 document.cookie是由於 cookie 的客戶端-服務器性質。當 Web 服務器將網頁發送到瀏覽器時，連接會關閉，服務器會忘記有關用戶的所有信息。當瀏覽器從服務器請求網頁時，屬於該頁面的 cookie 會添加到請求中。通過這種方式，服務器獲取必要的數據來"記住"有關用戶的信息。可以添加到期日期（以 UTC 時間表示）。默認情況下，關閉瀏覽器時會刪除 cookie。

cookie的特性，

1. 可以紀錄使用者訊息。
2. 儲存在客戶端。
3. 連線時會自動帶上，但過多的cookie可能會浪費流量、或是帶上無用之cookie。
4. 大小限制 4kb 左右。
5. 能夠設置過期時間。
6. 專屬於某網域(路徑)，也就是 google.com 的頁面不能存取 facebook.com 的cookie。