http cookie

Cookie 是伺服器（Server）傳送給瀏覽器（Client）的一小片段資料，並請瀏覽器保存起來，以便往後向相同的伺服器發送請求時，附上這 Cookie 的資料。
Cookie 常見用途
1.儲存和追蹤使用者行為
2.儲存用戶登入、購物車等伺服器所需的資訊
3.儲存使用者設定和偏好等
4.基本用戶登入流程範例
5.使用者透過瀏覽器登入畫面登入
伺服器端通過驗證後，可以將使用者「已經登入」的資訊附在 Cookie 中回傳，並請瀏覽器保存起來，往後，每當瀏覽器對伺服器發出請求時，會一併附上存有使用者「已經登入」狀態資訊的 Cookie 給伺服器，伺服器透過 Cookie 就能辨識這位使用者已經通過驗證了。

我自己用過 cookie 的經驗是在做使用者登入介面的時候，當使用者輸入帳號密碼成功登入後，便會回傳一段 TOKEN 給 client 端，並保存在 cookies，設定它的使用期限（假設 3 小時），則在這三小時內，只要使用者不做登出的動作（刪除 cookie），就算他 reload、關掉網頁，也不需要再重複輸入帳號密碼登入，因為 server 端是會去 cookie 找那段 token 存不存在、跟資料庫的 hashing code 比對成功，即表示登入成功。不過似乎這種作法不流行，因為不太安全。

這些資訊其實用戶是有機會可以在瀏覽器中修改的，因此使用者能透過串改 Cookie 上的＝內容，讓伺服器收到不正確的訊息 — 也就是說，以登入的例子來看，使用者可以在串改值，讓伺服器誤以為使用者已經通過認證。

Session
存放較敏感的資訊在客戶端是有安全上的疑慮，也因此改使用 Session 將使用者相關的敏感資訊存放在伺服器端 — 可能在記憶體或資料庫中 — 並創建一個相對應且獨特的 ID（Session ID），在回傳給客戶端的 Cookie 中一併附上，未來客戶端只要附上含有這個 Session ID 的 Cookie 給伺服器，伺服器就能匹配相對應的 Session — 也能找到需要的敏感資料了！

Cookies 和 Session 存在的目的都是幫助 Server 記住 Client 的狀態，差別在 Cookies 是將狀態資料存在 Client 端，而 Session 則是將資料存在 Server 端 。
Session 是用來彌補 Cookies 的不足:

Cookies size 限制: 根據 RFC-2965 的規定，一個 Cookies 最大可以是 4096 bytes，而一個 domain 最多只能有 20 cookies。
佔據網路流量: 由於每次 request 都必須將 cookies 的資料放在 headers，所以當 cookies 的資料變大時，request 也會跟著變大，過大的 request 會影響網路傳輸的速度。
Cookies 資料可被使用者竄改: 由於 Cookies 存在 browser 中，所以使用者可透過瀏覽器查看 cookies 的資料以及其結構，當使用者任意竄改 cookies 的資料時可能會導致資料外洩或者系統流程出錯。
根據以上限制，Cookies 只能儲存結構簡單，容量小且無意義的資訊，也就是說像是用戶資訊，購物車商品以及信用卡資訊都不適合放在 Cookies。
為了彌補 Cookies 的不足，我們會在 Server 端使用 Session 的機制去儲存這些狀態資訊，並產生一組 Session key 放入 Cookie 中，由於狀態資訊是直接存在 Server 端的，所以使用者無法讀取，使用者只能看到 session key 但不知道其結構，同時也能避免 Cookie 存入過多資料。
總結來說，Session 是一種比 Cookie 更安全的狀態管理機制，與 Cookie 不同的是，他將狀態資訊存在 Server 端，避免 client 端的 Cookie 資訊過載以及使用者任意修改 Cookie 內容，而為了確保使用者每次發的 request 都會讀取到相同的 Session 內容，在建立 Session 會產生 Session key 並放入 Cookie 中。