Cookies have an expiration date implicitly or explicitly set which controls how long they last (subject to the user agent actually enforcing it). A cookie may persist only for the duration of the session (or an even shorter period).

If a cookie is valid, it will be passed along with the HTTP request to the domain that it originated from. Only the domain that set the cookie can read the cookie (though there are ways to exploit this, such as cross-site scripting).

- If you want a cookie to expire at a specific time, set an expiration date on it using the client or server-side language of your choice.
- If you want the cookie to expire when the session ends, don't set an expiration date.

From the RFC (emphasis mine):

> The cookie setter can specify a deletion date, in which case the cookie will be removed on that date.
>
> **If the cookie setter does not specify a date, the cookie is removed once the user quits his or her browser.**
>
> As a result, specifying a date is a way for making a cookie survive across sessions. **For this reason, cookies with an expiration date are called persistent.**
>
> As an example application, a shopping site can use persistent cookies to store the items users have placed in their basket. (In reality, the cookie may refer to an entry in a database stored at the shopping site, not on your computer.) This way, if users quit their browser without making a purchase and return later, they still find the same items in the basket so they do not have to look for these items again. If these cookies were not given an expiration date, they would expire when the browser is closed, and the information about the basket content would be lost.