



Scripting languages II (LS2)

Class 12

Cookies

A cookie (or *magic cookie*) consists into a file created on client-side to store data received by a computer, then sent back without changing it. When a user visits a website, the website may send a cookie to his computer. The computer stores it inside the web browser.

Cookies purposes

Contrarily to popular beliefs, cookies aren't always a bad thing as its purpose often is to help keeping track of a user's visits and activity on a given website. For instance, a cookie maybe used to keep track of a shopping cart's items as the user keep on shopping. Without cookies, the user's shopping cart would be deleted every time a new link would be clicked.

Some websites may also use cookies to keep a record of a user's most recent visit, name, or to record login information, which most people find useful.

Types of cookies

Different types of cookies have different purposes.

Session cookies

Session cookies are strictly used during a user's current navigation on a website. The cookie is deleted as soon as the session is over.

Tracking cookies

Tracking cookies (persistent cookies) are used to create long-term records of a user's visits to a given website.

Authentication cookies

An authentication cookie tracks if a user is logged in or not, and if so, under what name.

Potential treats

Rest assure, cookies do not transfer viruses or malware and has no way to affect how a computer. Although, some viruses and malware can disguise themselves as cookies. You may want to read about *supercookies* which may cause security concern, *zombie cookies* which recreate itself after being deleted, making them difficult to manage. Tracking cookies may also cause security concerns as well since they make it easier to watch where a user is going and what he or she is doing online.

Setting a simple cookie

In order to make sure there is at least one cookie set to your browser so the following examples work, let's rapidly see how to set a cookie. We will cover this aspect in details further in this document.

To set a cookie using JavaScript, simply use the **.cookie** method, followed by a name/value pair between quotes.

```
<script type="text/javascript">  
    document.cookie = "name=John Doe";  
</script>
```

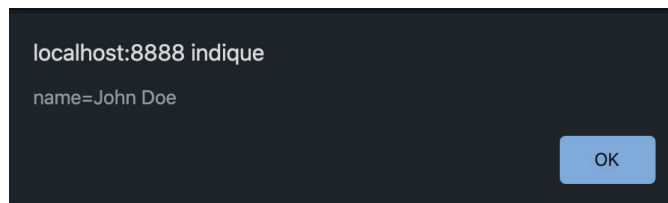
Reading existent cookies

To read and display the information of the various cookies installed on a browser is pretty easy using the **.cookie** method. Although, know that you might not be able to read all the cookies since some of them are set in a secure way.

```
<script type="text/javascript">  
    alert(document.cookie);  
</script>
```

Explanation :

In the above example, the contents of the cookies installed on a given browser shows in an alert window.



```
<h2> </h2>  
  
<script type="text/javascript">  
    $("h2").html(document.cookie);  
</script>
```

Explanation :

In the above example, the contents of the cookies installed on a given browser shows between the `<h2>` `</h2>` tags so it is shown on screen.

name=John Doe

Setting a cookie

When resetting a cookie using the same name, the cookie is updated, the new value replaces the existing one. When a new cookie is set, existing cookies remain and each new cookie is added to the browser cookie folder.

Expiry date

In order for a cookie to be deleted after a certain period of time or on a certain date an expiry attribute must be defined.

```
<script type="text/javascript">
    document.cookie = "userName=John Doe;
                        expires=sat, 31 Dec 2033 12:00:00 UTC;
                        path=/;
                        domain=mysite.com";
</script>
```

Explanation :

In the above example, a cookie is set for *userName*, it will expire at noon on December 31st 2033 and the cookie will be saved in the default location of the browser.

Expiry time

It is possible to set a cookie to be active within a certain limit of time using the attribute *max-age* followed by a number indicating the number of seconds.

```
<script type="text/javascript">
    document.cookie = "userName=John Doe; max-age=3600";
</script>
```

Explanation :

In the above example, the cookie would be active 3600 seconds (60 minutes) from the moment of its creation.

Domain

Specifies those hosts the cookie will be sent to (defaults : current document location (not including subdomains)). If a domain is specified, subdomains are always included.

Path

Indicates a URL path that must exist in the requested resource before sending the Cookie header. The "/" character is interpreted as a directory separator and sub directories will be matched as well (e.g. *path=/docs*, *"/docs"*, *"/docs/Web/"*, or *"/docs/Web/HTTP"* will all be matched).

js-cookie

JS-COOKIE is simple, lightweight JavaScript application that allows handling cookies easily. It works with all browsers and has been heavily tested and it can be downloaded to be installed within a website's structure or use a simple CDN.

js-cookie CDN

<https://cdn.jsdelivr.net/npm/js-cookie@2/src/js.cookie.min.js>

Setting a cookie

To create a cookie valid across an entire site :

```
Cookies.set('name', 'value')
```

To create a cookie that expires 7 days from now, valid across the entire site :

```
Cookies.set('name', 'value', { expires: 7 })
```

To create an expiring cookie valid to the path of the current page :

```
Cookies.set('name', 'value', { expires: 7, path: ' ' })
```

Reading a cookie

Read cookie:

```
Cookies.get('name');    // Would show associated value
```

Read all visible cookies:

```
Cookies.get();
```

Removing a cookie

```
Cookies.remove('name');
```

IMPORTANT! To delete a cookie which isn't relying on the default attributes, you must pass the exact same path and domain attributes used to set the cookie :

```
Cookies.remove('name', { path: ' ', domain: 'yourdomain.com' })
```