

Web Storages

Cookies, Local and Session Storage

Telerik Software Academy
Learning & Development Team
<http://academy.telerik.com>



Table of Contents

- ◆ Cookies
- ◆ SessionStorage
- ◆ LocalStorage
- ◆ Shims for localStorage
and sessionStorage



```
check None";  
function() {  
    el.getElementsByTagName("input").length = false;  
    if (el.getAttribut
```



WebStorages

Cookies, Local and Session



- ◆ WebStorages are places to store data
 - ◆ Save user settings, so next time he opens the application, they can be loaded
- ◆ Three common types of Web Storage
 - ◆ Cookies
 - ◆ Accessible only from a single document
 - ◆ localStorage
 - ◆ Accessible only from a single document
 - ◆ sessionStorage
 - ◆ Accessible only while the document is opened



Cookies

Ordner/Site	Name	Größe	Verändert	Status
admin.brightcove.com	analytics.sol	419	Samstag, 17. März 2012 23:2...	Nicht geschützt
admin.brightcove.com	Akamai_AnalyticsMetric...	100	Samstag, 17. März 2012 23:1...	Nicht geschützt
images-na.ssl-images-amazon.com	mercury.sol	69	Mittwoch, 28. März 2012 10:...	Nicht geschützt
images.tv.adobe.com	ATVPlayer.sol	159	Sonntag, 25. März 2012 23:4...	Nicht geschützt
mail.google.com	wakeup.sol	37	Freitag, 16. März 2012 23:37:08	Nicht geschützt
s.ytimg.com	soundData.sol	49	Mittwoch, 14. März 2012 20:...	Nicht geschützt
s.ytimg.com	videostats.sol	275	Montag, 26. März 2012 00:36...	Nicht geschützt
secureinclude.ebaystatic.com	ebayLSO.sol	131	Samstag, 17. März 2012 15:0...	Nicht geschützt
secureinclude.ebaystatic.com	ebayT.sol	39	Samstag, 17. März 2012 15:0...	Nicht geschützt
valde.sftcdn.net	rondavu.sol	95	Montag, 12. März 2012 18:37...	Nicht geschützt
www.paypalobjects.com	pplSsoTest.sol	48	Samstag, 17. März 2012 15:0...	Nicht geschützt
www.wdr.de	cookie.sol	45	Sonntag, 18. März 2012 22:3...	Nicht geschützt
admin.brightcove.com <settings LSO>	settings.sol	90	Samstag, 17. März 2012 23:1...	Nicht geschützt
images-na.ssl-images-amazon.com <settings LSO>	settings.sol	101	Mittwoch, 28. März 2012 10:...	Nicht geschützt

- ◆ Cookies are small pieces of data
 - Accessible from a concrete application
 - Stored in the user's browsers
 - i.e. different cookies for different browsers
 - Cookies can store only plain text

- ◆ Cookies are used to save some state of the user preferences and settings
 - If you have authenticated to the server once, it is not necessary to do so again
 - Cookies are attached to the headers of every HTTP request to the server
- ◆ Cookies can be read and set by JavaScript

- ◆ A cookie consists of three parts
 - A name-value pair that holds the cookie information
 - An expire date, after which this cookie is not available
 - A domain and path to the server, that the cookie belongs to

- ◆ Name-value pairs hold the cookie's data
 - ◆ The name is used to reach the data stored in the value
 - ◆ To read a cookie, you must search for the name
- ◆ Expire date
 - ◆ Used to give timeframe for the work of the cookie
 - ◆ If not set, the cookie is removed when closing the browser
 - ◆ To make a forever cookie, set the expire date after enough years

Working with Cookies

- ◆ Cookies can be accessed with JavaScript
 - ◆ Use `document.cookie` property
 - ◆ Thought cookies are not strings, they are used as strings

```
//sets a cookie
document.cookie =
'c1=cookie1; expires=Thu, 30 Apr 2013 21:44:00 UTC; path=/'
//sets another cookie
document.cookie =
'c2=cookie2; expires=Tue, 29 Apr 2013 11:11:11 UTC; path=/'
//reads all cookies
console.log(document.cookie);
```

Working with Cookies

- ◆ Read cookie (its information)

```
function readCookie(name) {  
    var allCookies = document.cookie.split(";" );  
    for (var i = 0; i < allCookies.length; i++) {  
        var cookie = allCookies[i];  
        var trailingZeros = 0;  
        for (var j = 0; j < cookie.length; j++) {  
            if (cookie[j] !== " ") {  
                break;  
            }  
        }  
        cookie = cookie.substring(j);  
        if (cookie.startsWith(name + "=")) {  
            return cookie;  
        }  
    }  
}
```

Cookies

Live Demo



localStorage

- ◆ **localStorage** is per document storage
 - ◆ Accessible through `document.localStorage`
 - ◆ Similar to cookies
 - ◆ Can store much larger amount of data
- ◆ Supported down to IE8
 - ◆ Needs a shim for IE7
- ◆ Saves data as string
- ◆ **localStorage** properties:
 - ◆ `setItem(key, value)`, `getItem(key)`
 - ◆ `removeItem(key)`, `length`

LocalStorage Example

- ◆ Local Storage

```
function saveState(text){  
    localStorage["text"] = text;  
}  
function restoreState(){  
    return localStorage["text"];  
}
```

- ◆ Same as

```
function saveState(text){  
    localStorage.setValue("text", text);  
}  
function restoreState(){  
    return localStorage.getValue("text");  
}
```

localStorage

Live Demo



- ◆ Session Storage
 - ◆ Similar to localStorage
 - ◆ Lasts as long as browser is open
 - ◆ Opening page in new window or tab starts new session
 - ◆ Great for sensitive data (e.g. banking sessions)
- ◆ Can store only strings

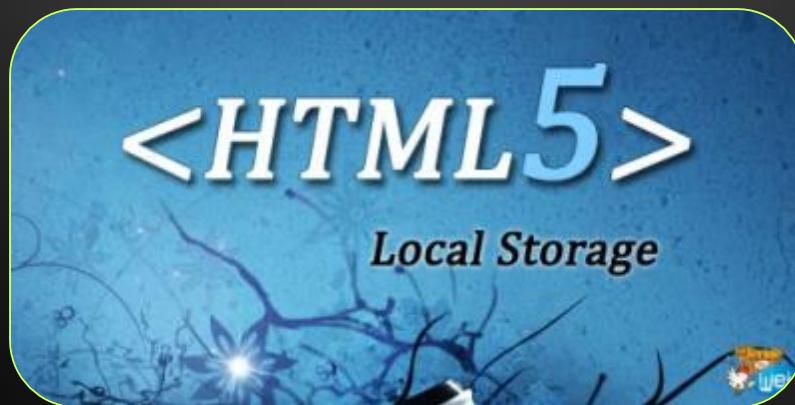
Session Storage Example

◆ Session Storage

```
function incrementLoads() {  
    if (!localStorage.counter) {  
        localStorage.setItem("counter", 0);  
    }  
    var currentCount =  
        parseInt(localStorage.getItem("counter"));  
    currentCount++;  
    localStorage.setItem("counter", currentCount);  
    document.getElementById("countDiv").innerHTML =  
        currentCount;  
}
```

sessionStorage Storage

Live Demo



Saving Object in WebStorages

- ◆ Local and session storage can only contain strings
 - ◆ If you try to save an object, its `toString()` method will be invoked
- ◆ To save objects into web storages, need to extend the Storage prototype

```
Storage.prototype.setObject =  
  function setObject(key, obj){  
    this.setItem(key, JSON.stringify(obj));  
};  
Storage.prototype.getObject =  
  function getObject(key) {  
    return JSON.parse(this.getItem(key));  
};
```



Saving Object in WebStorages

Live Demo

Questions?

- ◆ Create a simple number guessing game
 - The computer generates a random number with four different digits
 - The leftmost digit must not be 0 (zero)
 - For simplicity called abcd
 - At each turn the player enters a four-digit number
 - For simplicity called xyzw
 - When the game ends:
 - Ask the player for a nickname
 - Save the nickname inside the localStorage
 - Implement a high-score list

Homework (2)

- ◆ (cont.) Create a simple number guessing game
 - Sheep means that a digit from xyzw is contained in abcd, but not on the same position
 - If two such digits exists, the sheep are 2
 - Ram means that a digit from xyzw is contained in abcd and it is on the same position
 - If two such digits exists, the rams are 2
 - The game continues until the player guesses the number abcd
 - i.e. has 4 rams