

Aufgabe 1

In den Cookies werden einerseits die Sessiondaten gespeichert, sprich die ID unter der die Session auf dem Server gespeichert ist. In dieser wird der aktuelle Zustand der Applikation gespeichert. Es werden auch dauerhafte Einstellungen wie die Sprache oder ähnliches in den Cookies gespeichert.

Der LocalStorage wird für Nutzerspezifische Daten verwendet, sprich cachingspezifische Daten(Beispielsweise können Fonts oder Sounds gut damit gecached werden) und anderes, was über ein Schließen der Webseite hinaus verfügbar sein soll.

Der SessionStorage wird für das Caching der aktuell aufgerufenen Seiten verwendet (zumindest bei Amazon.de) und generell für Daten die nach dem Schließen nicht mehr relevant sind. Hier wird bei Facebook auch ein Log über die Navigationspunkte geführt.

Generell ist anzumerken, dass der WebStorage bisher noch unbegrenzt ist, es aber für Cookies eine Maximalgröße gibt.

Aufgabe 2

Cookies sind an das document Objekt im DOM gebunden und einfach als Simikolonseparierter String angebunden. Die Key-Value Paare sind per = getrennt, dementsprechend lässt sich beispielsweise eine Array von Key-Value Arrays so erzeugen:

```
function getCookies(){  
    return document.cookie.split(';').map(function(item){  
        return item.split('=');  
    });  
}
```

Nehmen wir an wir würden einen Wert hier drin geändert haben und den Cookie wieder speichern, so ginge dies über:

```
function setCookies(cookies){  
    return document.cookie = cookies.map(function(item){return  
        item.join('=');}).join(';');  
}
```

Die WebStorages hingegen sind deutlich moderner und lassen sich direkt als Key-Value Store nutzen. Folgendes Objekt abstrahiert den Webstorage:

```
var Storage = (function() {  
    function Storage(storage) {  
        this.storage = storage;  
    }  
    var store = {  
        setItem: function (key, value){  
            return this.storage.setItem(key, value);  
        },  
        getItem: function (key) {  
            return this.storage.getItem(key);  
        },  
        removeItem: function (key){  
            return this.storage.removeItem(key);  
        }  
    }  
    }  
    return store;  
})();  
  
var session = new Storage(sessionStorage);  
var local = new Storage(localStorage);
```
