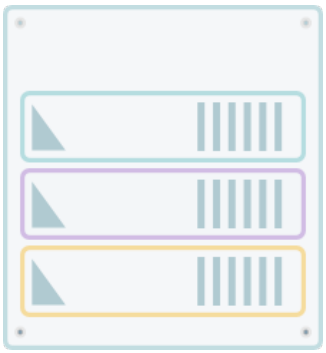
CSS, HTML UND JAVASCRIPT MIT {STIL}

Apr 2012

HTML5 Local Storage



Local Storage stellt Speicher auf dem System des Benutzers zur Verfügung – z.B. um ein einfaches Umschalten der Sprache bei mehrsprachigen Seiten zu realisieren.

Die Größe des lokalen Speichers hängt vom Browser ab, meist 5 bis 10 MB.

WebStorage-Objekte halten Benutzerdaten auf dem Client vor – entweder für die Dauer der Session (Session Storage) oder ohne definiertes Ende (*Local Storage*). Die User-Daten werden nicht mehr wie Cookies mit jedem HTTP-Request auf den Server übertragen.

Local Storage und Session Storage bilden keine Datenbank, sondern basieren auf einfachen Schlüssel-/Wertpaaren. Der Schlüssel (key) ist ein String.

Beide Arten von Speicher arbeiten mit derselben Schnittstelle.

Wie »persistent« – also wie haltbar Local Storage tatsächlich ist – lässt sich nicht voraussagen.

Einfache String-Verarbeitung

Die Werte können beliebige Datentypen vom String über Boolean, Integer oder Float sein, werden aber in allen Fällen als String gespeichert. Damit gespeicherte Zahlen wieder als Zahlen behandelt werden, müssen sie mit parseInt() oder parseFloat() in die Welt der Zahlen zurück geholt werden.

Local Storage

localStorage.clear()	löscht den Speicher
localStorage.getItem(key)	gibt null zurück, wenn der key nicht existiert
localStorage.removeItem(key)	löscht einen Eintrag
localStorage.setItem(key, value)	erzeugt einen Eintrag oder überschreibt den Eintrag ohne Warnung, wenn der key schon existiert

Beispiel HTML5 Local Storage

Spieler Kleinen Text eingeben
Punkte
Verein Kleinen Text eingeben
Werte übernehmen Local Storage löschen

Spieler
Punkte
Verein

Werte übernehmen überträgt die Formulareingaben in die Tabelle darunter. Bei einem erneuten Laden der Seite (auch wenn der Browser zwischenzeitlich geschlossen wurde) holt *Reload* die Daten aus dem Local Storage zurück und überträgt sie auch in das Formular.

Die Formulardaten bleiben im HTML5 Local Storage gespeichert, auch wenn der Benutzer das Browserfenster oder den Browser schließt und können zu jeder Zeit wieder abgerufen werden. Die Werte werden nicht wie bei HTTP-Cookies auf den Server übertragen. Die gespeicherten Daten haben kein Ablaufdatum.

Da sich die App nicht um die Verwaltung der Benutzer kümmern muss, ist die Implementierung außerordentlich einfach.

Werte werden mit der window-Methode window.localStorage.setItem in den Local Storage eingetragen und mit window.localStorage.getItem aus dem Local Storage gelesen.

```
<form action="#" method="post" id="webstorage">
   <input type="text" id="val1" value="" />
   …
   <input type="submit" id="submit" value="Werte übernehmen" />
</form>

<script type="text/javascript">
window.localStorage.setItem('Feld1',document.getElementById('val1').value);
</script>
```

Methoden für Local Storage

Die Verarbeitung von Local Storage und Session Storage ist dieselbe – nur das Objekt unterscheidet Local Storage von Session Storage. localStorage und sessionStorage sind Objekte von Window.

window.localStorage.clear()	Löscht die Datenbank
window.localStorage.getItem(fieldname,value)	liest Werte aus einem Datenbankfeld *fieldname*
window.localStorage.key()	Schlüssel am angegebenen Index
window.localStorage.setItem(fieldname,value)	schreibt Werte in ein Datenbankfeld *fieldname*
window.localStorage.removeItem(fieldname)	löscht den Wert von fieldname
window.localStorage.length()	
window.localStorage.remainingSpace()	Verfügbarer Speicherplatz für das Storage-Objekt

Suchbegriff(e) eingeben und

Window: Browser, Screen, Location

HTML5 Javascript – Storage & Canvas

Javascript DOM

DOM Manipulationen

Webseiten durch Events steuern

Javascript Events

Javascript für Formulare

Eine Javascript Library: jQuery

Fehlersuche, Performance und mehr

Mehr Javascript bei molily