HTML5 - WEB STORAGE

http://www.tutorialspoint.com/html5/html5_web_storage.htm

HTML5 introduces two mechanisms, similar to HTTP session cookies, for storing structured data on the client side and to overcome following drawbacks.

- Cookies are included with every HTTP request, thereby slowing down your web application by transmitting the same data.
- Cookies are included with every HTTP request, thereby sending data unencrypted over the internet.
- Cookies are limited to about 4 KB of data . Not enough to store required data.

The two storage's are **session storage** and **local storage** and they would be used to handle different situations.

The latest versions of pretty much every browser supports HTML5 Storage including Internet Explorer.

Session Storage

The *Session Storage* is designed for scenarios where the user is carrying out a single transaction, but could be carrying out multiple transactions in different windows at the same time.

Example

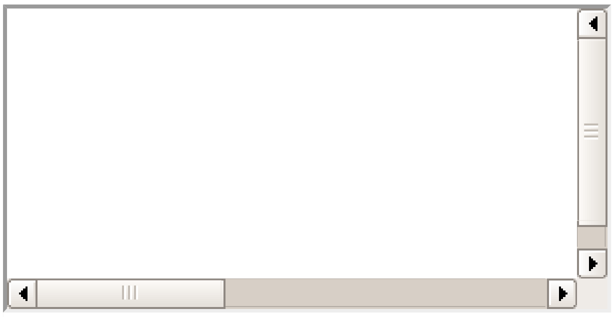
For example, if a user buying plane tickets in two different windows, using the same site. If the site used cookies to keep track of which ticket the user was buying, then as the user clicked from page to page in both windows, the ticket currently being purchased would "leak" from one window to the other, potentially causing the user to buy two tickets for the same flight without really noticing.

HTML5 introduces the *sessionStorage* attribute which would be used by the sites to add data to the session storage, and it will be accessible to any page from the same site opened in that window i.e session and as soon as you close the window, session would be lost.

Following is the code which would set a session variable and access that variable −

```
<!DOCTYPE HTML>
<html>
   <body>

      <script type="text/javascript">
         if( sessionStorage.hits ){
            sessionStorage.hits = Number(sessionStorage.hits) +1;
         }

         else{
            sessionStorage.hits = 1;
         }
         document.write("Total Hits :" + sessionStorage.hits );
      </script>

      <p>Refresh the page to increase number of hits.</p>
      <p>Close the window and open it again and check the result.</p>

   </body>
</html>
```

This will produce following result −



Local Storage

The *Local Storage* is designed for storage that spans multiple windows, and lasts beyond the current session. In particular, Web applications may wish to store megabytes of user data, such as entire user-authored documents or a user's mailbox, on the client side for performance reasons.

Again, cookies do not handle this case well, because they are transmitted with every request.

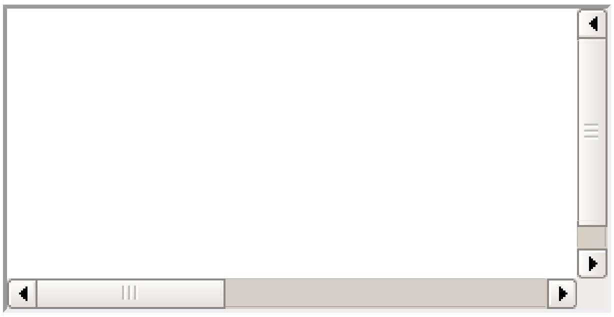
Example

HTML5 introduces the *localStorage* attribute which would be used to access a page's local storage area without no time limit and this local storage will be available whenever you would use that page.

Following is the code which would set a local storage variable and access that variable every time this page is accessed, even next time when you open the window –

```
<!DOCTYPE HTML>
<html>
   <body>

      <script type="text/javascript">
         if( localStorage.hits ){
            localStorage.hits = Number(localStorage.hits) +1;
         }

         else{
            localStorage.hits = 1;
         }
         document.write("Total Hits :" + localStorage.hits );
      </script>

      <p>Refresh the page to increase number of hits.</p>
      <p>Close the window and open it again and check the result.</p>

   </body>
</html>
```

This will produce following result –



Delete Web Storage

Storing sensitive data on local machine could be dangerous and could leave a security hole.

The *Session Storage Data* would be deleted by the browsers immediately after the session gets terminated.

To clear a local storage setting you would need to call **localStorage.remove**$'key'$; where 'key' is the key of the value you want to remove. If you want to clear all settings, you need to call **localStorage.clear** method.

Following is the code which would clear complete local storage −

```
<!DOCTYPE HTML>
<html>
   <body>

      <script type="text/javascript">
         localStorage.clear();

         // Reset number of hits.
         if( localStorage.hits ){
            localStorage.hits = Number(localStorage.hits) +1;
         }

         else{
            localStorage.hits = 1;
         }
         document.write("Total Hits :" + localStorage.hits );

      </script>

      <p>Refreshing the page would not to increase hit counter.</p>
      <p>Close the window and open it again and check the result.</p>

   </body>
</html>
```

This will produce following result −

Loading [MathJax]/jax/output/HTML-CSS/jax.js