HTML - Web Storage

Web Storage

HTML Web storage is a mechanism used for storing structured data on the client side without sending it to the server. These two storage mechanisms are **session storage** and **local storage**. Both are collectively part of the HTML5 Web Storage API.

Need of Web Storage

HTML Web storage was introduced to overcome the following drawbacks of cookies:

- 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.

Types of Web Storage

HTML provides two types of web storage:

- Session storage
- Local storage

To use these two web storages (**session storage** or **local storage**) in your web application, you can access them through the **`window.sessionStorage`** and **`window.localStorage`** properties, respectively.

The Session Storage

The **session storage** is temporary, and it gets cleared when the page session ends, which happens when the browser tab or window is closed. The data stored in session storage is specific to each tab or window.

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, the session would be lost.

Example

Following is the code that would set a session **variable and access that variable** –

</> Open Compiler

```
<!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>
```

Local Storage

The **local storage** is designed for storage that spans multiple windows and lasts beyond the current session. It does not expire and remains in the browser until it is manually deleted by the user or by the web application. 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.

HTML5 introduces the **localStorage** attribute, which would be used to access a page's local storage area without a time limit, and this local storage will be available whenever you use that page.

Example

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

</> Open Compiler

```
<!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>
```

Delete Web Storage

Storing sensitive data on a 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.

However, to clear a local storage setting, we need to call **localStorage.remove('key'),** where 'key' is the key of the value we want to remove. If we want to clear all settings, the **localStorage.clear()** method can be called.

Example

Following is the code that would clear complete local storage –

Open Compiler

```
<!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>
```