

Web Storage

HTML5 web storage is a local storage in client environment and it is a better than cookies. Unlike the cookie storage limitation of 4KB, we can store up to 10 megabytes in HTML5 local storage.

1. Web storage is the most secured and faster way of storing information in client environment.
2. The data is not included with every server request, but will be sent to server only when requested by server.
3. Using web storage it is also possible to store large amounts of data upto 10MB on client machine, without affecting the website's performance.

Web storage is supported by almost all the major browsers like Internet Explorer 8+, Firefox, Opera, Chrome, and Safari etc.

Note: Following programs will work only if HTTP protocol is issued and Page is HOSTED in web server like IIS

Checking if browser support this feature or not:

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title></title>
  <script>
    if (typeof (Storage) !== "undefined") {
      alert(" Web storage is supported.");
    }
    else {
      alert(" Web storage is not supported.");
    }
  </script>
</head>
<body>
  This program shows if Web Storage object is supported or not?
</body>
</html>
```

Web storage provides two objects to store data in the client:

- **LocalStorage** - stores data with no expiration date

- **SessionStorage** - stores data for one session

The localStorage Object:

- The localStorage object stores the data with **no expiration date**.
- The data will **not be deleted** when the browser is closed.
- Data will be stored in the form of key and value pair where key and values are string type.
- **localStorage** is scoped to the origin, however, meaning it's shared across all pages on the same origin.

The sessionStorage Object:

- The sessionStorage object is same as localStorage object, except that it stores the data for only one session. The data is deleted automatically when the user closes the browser window.
- Opening a page in a new tab or window will cause a new session to be initiated.

Properties and methods of Session Storage and Local Storage:

Property / Method	Description
setItem()	Stores a key-value pair in the respective Web Storage object.
getItem()	Retrieves value of a specified key from the respective Web Storage object.
removeItem()	Removes a specified key-value pair from the Web Storage object under consideration.
clear()	Removes all the key-value pairs from the Web Storage object.
key()	Takes a 0 based index and returns key name at that index.
length	Returns the total number of key-value pairs present in the Web Storage object.
remainingSpace	Returns the amount of storage space in bytes still available for storing data. This property is specific to IE. Other browsers may implement similar property.

```
<!DOCTYPE html>

<html lang="en" xmlns="http://www.w3.org/1999/xhtml">

<head>

  <meta charset="utf-8" />

  <title></title>

  <script type="text/javascript">

    function incrementCounter() {

      if (localStorage.counter) {

        localStorage.counter = Number(localStorage.counter) + 1;

      }

    }

  </script>

</head>

</html>
```

```
    }
    else {
        localStorage.counter = 1;
    }
    document.getElementById("lblCounter").innerHTML = localStorage.counter;
}
</script>
</head>
<body>
    <b>
        <label id="lblCounter"></label>
    </b>
    <br />
    <input type="button" value="Increment Counter" onclick="incrementCounter()" />
    <input type="button" value="Clear Stored Values" onclick="localStorage.clear()" />
</body>
</html>
```

Example 2: Auto save the contents of a text field, and if the browser is accidentally refreshed, restore the text field contents so that no writing is lost.

```
<script>
    function t1_onchange() {
        sessionStorage.setItem("t1",document.getElementById("t1").value)
    }
    function window_load()
    {
        if (sessionStorage.getItem("t1") != null)
        {
            document.getElementById("t1").value = sessionStorage.getItem("t1")
        }
    }
</script>
</head>
<body onload="window_load()">
    <form action="demo.html" >
```

```
<input type="text" id="t1" name="t1" value="" onchange="t1_onchange()"/>
<input type="submit" name="name" value="Submit" />
</form>
</body>
```

Event: storage

Event data	Description
key	Represents the key of the item that is being changed.
oldValue	The old value of the key (if any) being changed
newValue	The new value being assigned to the key
url	The url of the page that is changing the storage area
storageArea	Reference to the storage area

```
<script type="text/javascript">
$(document).ready(function () {
    window.addEventListener('storage', OnStorage);
});
function OnStorage(event) {
    alert("Storage event fired for key : " + event.key + " in page " + event.url);
    alert("Old Value - New Value : " + event.oldValue + " - " + event.newValue);
}
</script>
```