

Chapter 4. Cookies & Browser Data

cookies

- It is small piece of data stored in file on your computer.

- The information present in cookies is accessed by the web browser.

- Cookies remembers the information about the user in the following ways:-

step 1: when the user visits the web page his/her name can be stored in cookie

step 2: Next time when the user visits the page, the cookie remembers his/her name.

- cookies are saved in name-value pair

eg:- username = AAABBB

- cookie is small text file which contains following data:

- A name-value pair containing the actual data

- An expiry date after which it is no longer valid

- The domain & path of the server it should be sent to.

There are two types of cookies

① session cookies.

② persistent cookies.

1) Session Cookies

It is cookie that remains in temporary memory only while user reading & navigating the web site.

The cookie is automatically deleted when the user exits the browser application.

2) Persistent Cookies

It is cookie that is assigned an expiration date.

A persistent cookie is written to the computer's hard disk & remains there until the expiration date has been reached; then it is deleted.

Creating Cookies

It is a simple technique

for creating a cookie we need to assign cookie value to window.document.cookie.

For instance

document.cookie = "Username=Payal";

Name value :=; Delim

Thus the name value pair separated by = sign & terminated by delimiter like ; (semicolon), the cookie can be assigned to document.cookie. Instead of window.document.cookie we can simply write document.cookie

Write a javascript for creating a cookie for the user name.

```
<html>
<head>
<script type="text/javascript">
    function AC()
    {
        with (document.form1)
        {
            document.cookie = "Username" +
            username.value + ";" alert (" cookie
            is created");
        }
    }
</script>
</head>
<body>
<form name="form1">
Enter Name: <input type="text" name=
"username"/>
<input type="button" value=" Create
cookie " onclick="AC()"/>
</form>
</body>
</html>
```

Reading a cookies value

The document.cookie string will keep a list of name = value pairs separated by semicolons, where name is name of a cookie & value is its string value

- using split() for the string of cookies is break into key & values.
- The split method finds the = character in cookie, & then takes all the characters to the left of the = & store them into array array[0].
- Next the split() method takes all the characters from the right of the = upto; but no including the semicolon, & assign those characters to array array[1].
- Thus we can obtain the name value pair array[0] & array[1] respectively.

eg:-

```

<html>
<head>
<script type="text/javascript">
function AC()
{
with(document.form1)
{
value = document.cookie.split('=');
}
document.write("value: "+
value);
}
}
</script>
</head>
<body>
<form name="form1">
<input type="button" value="Read
cookie" onclick="AC()">
</form>
</body>
</html>

```

Deleting Cookies

- cookies get deleted automatically when the browser session ends or its expiration date is reached.
- Hence by setting previous expiry date we can delete the cookie.

eg:-

```

<html>
<head>
<script type="text/javascript">
function AC()
{
expiryDate = new Date(),
expiryDate.setMonth(expiryDate.
getMonth()-1)
with(document.form1)
{
document.cookie = "Username" + username.
value + ";";
document.cookie = "expires = " + expiryDate.
toUTCString() + ";";
}
}
}
</script>
</head>
<body>
<form name="form1">
Enter name: <input type="text" name="user
name">
<input type="button" value="Delete cookie"
onclick="AC()">
</form>
</body>
</html>

```

- to delete cookie we are simply setting expiry date of the cookie to a time in past

- We take date variable expiryDate. The getMonth() returns the system provided month. We just set this month to past month by subtracting one from current month.

- The new expiry date is assigned to the cookie

- Then the cookie is written by setting a new expiry date. This is how the cookie get deleted.

Setting Expiration Date of cookie

- We can set the expiration date of a cookie by extending the life of a session. Then, we need to save this extended expiry date within the cookie. This can be done by setting the expires attribute to a date & time

1) getMonth() :- This method returns the current month based on the system clock of the computer running the Javascript

2) setMonth() :- This method assigns the month to Date variable

3) toGMTString() :- This method

returns the value of the Date variable to a string that is in the format of Greenwich Mean Time, which is then assigned to the cookie.

eg:- set new expiry date

```
<html>
<head>
<script type="text/javascript">
function AC()
{
    expiryDate = new Date();
    expiryDate.setMonth ( expiryDate.getMonth() + 1);
    with (document.form1)
    {
        document.cookie = "username" + username.value + ";";
        document.cookie = "expires" + expiryDate.toUTCString() + ";";
    }
}
</script>
</head>
<body>
<form name="form1">
Enter name: <input type="text" name="username">
<input type="button" value="set New Expiry Date" onclick="set AC()">
</form>
</body>
</html>
```

Browser

- It is possible to open a new browser window from currently running JavaScript. One can determine the size, location of this window, toolbar or any other style that normally the browser windows have.
- Once the new browser window is set up, it is also possible to change the contents within that window dynamically.

Opening Window

- It is possible to open a new window from a JavaScript by simply clicking a button for that purpose the window object is used.
- This window object has various useful properties & methods.
- To open a new window we use open() method of window object.

Syntax

window.open(URL, name, style)

where,

URL:- specifies the URL of webpage that will appear in new window. This is optional parameter name:- The name that could be assigned to window. This is optional parameter

style:- various parameters such as toolbar, scrollbar, location, height & width of window & so on. This is an optional parameter

Supplied By: SNJB

Course & Semester :		Roll No. :		Date : / / 201	
Name of Subject :		Marks obtained :			
Sign. of Supervisor :		Sign. of Sub. Examiner :		Total Marks	
Q. No. 1	Q. No. 2	Q. No. 3			

000558

style

fullscreen = yes | no | 1 | 0

height = pixels.

left = pixels

location = yes | no | 0 | 1

menubar = yes | no | 1 | 0

resizable = yes | no | 1 | 0

scrollbars = yes | no | 1 | 0

status = yes | no | 1 | 0

titlebar = yes | no | 1 | 0

top = pixels

width = pixels

Purpose

To display the browser in full screen mode

The height of window.min value is 100.

The left position of the window. Negative values not allowed.

To display the address field. opera only.

To display menu bar.

To determine if the window is resizable

To display scroll bars.

To add a status bar.

To display the title bar.

To display the browser toolbar

To determine the top position of window. Negative values not allowed

The width of window. Min value is 160.

Return value

A newly created window gets opened up on success & null on failure.

eg:- To open a new window

```
<html>
<head>
<script type="text/javascript">
    function AC()
    {
        var a = window.open("", "", width=100,
            height=100);
    }
</script>
</head>
<body>
<form name="form1">
<input type="button" value="open
window onclick="AC">
</form>
</body>
</html>
```

Giving the New Window Focus

A focus to a new window can be given using focus() method.

```
<html>
<head>
<script type="text/javascript">
    function AC()
    {
        var mywin = window.open("", "MyWindow",
            "status=0, toolbar=0, location=0, menubar=0,
            directories=0, resizable=0, width=200, height=
            200");
        this.focus();
    }
</script>
</head>
<body>
<form name="form1">
<input type="button" value="open window"
    onclick="AC">
</form>
</body>
</html>
```

* Window Position
we can set desired position for the

window.
using left & top attributes values the
window position can be set.

eg:-

```
<html>
<head>
<script type="text/javascript">
function AC()
{
var m=window.open("", "My Window",
"left=0, top=0, width=200, height=
200");
}
</script>
</head>
<body>
<form name="form1">
<input type="button" value="open
window" onclick="AC">
</form>
</body>
</html>
```

we set the style attributes as left=0
right=0 in open() method, the window
is created & positioned at the extreme
left & top location of screen.

Changing the contents of window:-

By writing some text to the newly created
window we can change the contents of a
window.

for eg:-

```
<html>
<head>
<script type="text/javascript">
function AC()
{
var m=window.open("", "My
Window", "width=200, height=200");
m.document.write("This line is written
in current window");
}
</script>
</head>
<body>
<form name="form1">
<input type="button" value="open window"
onclick="AC">
</form>
</body>
</html>
```

* Closing a Window

The most simple operation about the window is to close it.

It can be closed using the function `close()`.

Eg.:

```
<html>
<head>
  <script type="text/javascript">
    var m,
        function AC()
        {
          m=window.open("","MyWindow",
            "left=500,top=400,width=200,
            height");
        }
        function BC()
        {
          m.close();
        }
  </script>
</head>
<body>
  <form name="form1">
    <input type="button" value="open
    window" onclick="AC">
  </form>
  </body>
</html>
```

* Scrolling a Web page

We can scroll horizontally or vertically using `scrollTo()` function

```
<html>
<head>
  <script type="text/javascript">
    function AC()
    {
      window.scrollTo(300,0);
    }
  </script>
</head>
<body>
  <form name="form1">
    <input type="button" value="scroll" onclick =
    "AC">
  </form>
</body>
</html>
```

* Multiple Windows at a Glance

It is possible to open up multiple windows at time. It is simple to open multiple windows. Simply put opens method in a loop.

Eg.:-

```
<html>
<head>
  <script type="text/javascript">
    function AC()
    {
      window.open("","");
    }
  </script>
</head>
```



```

}
for (i=0; i<5; i++)
{
var m = window.open("", "myWindow"
+ i, "width=100, height=100");
}
</script>
</head>
<body>
<form name="form1">
<input type="button" value="open
window" onclick="AC3">
</form>
</body>
</html>

```

* Creating a Web Page in new window

- We can create a web page using window object with the help of write method.
- The only thing is that inside the write we have to write the contents of the web page with the help of html elements such as <head>, <body>, <title>, <h1> and so on.

SHRI H. H. J. B. POLYTECHNIC, CHANDWAD
CLASS TEST I / II (201 - 201)

Supplied By: SNJB

Course & Semester:	Roll No.:	Date:	7 / 201
Name of Subject:	Marks obtained:		
Sign. of Supervisor:	Sign. of Sub. Examiner:		
Q. No. 1	Q. No. 2	Q. No. 3	Total Marks

eg:-
<html> 000539

```

<head>
<script type="text/javascript">
function AC3
{
var m = window.open("", "myWindow", "width=
300, height=300");
m.document.write("<html>");
m.document.write("<head>");
m.document.write("<title> website </title>
");
m.document.write("</head>");
m.document.write("<body>");
m.document.write("<h3> Hello </h3>");
m.document.write("</body>");
m.document.write("</html>");
}
</script>
</head>
<body>
<form name="form1">
<input type="button" value="create webpage"
onclick="AC3">
</form>
</body>
</html>

```

Timers

- using window object it is possible to execute the code at specified time intervals.

- Two functions are supported by window object for handling timing events

1. setTimeout

This method executes fn after waiting for a specified number of milliseconds.

Syntax :: window.setTimeout(function, milliseconds);

fn to be executed
No. of milliseconds before execution take place

eg:-

```
<html>
<head>
<script type="text/javascript">
function AC()
{
  alert("Welcome User");
}
</script>
</head>
<body>
<form name="form1">
<input type="button" value="click me" onclick="
  setTimeout(A,1000);">
</form>
</body>
</html>
```

② setInterval

This fn executes the specific fn at every given time interval

Syntax:-

window.setInterval (fn, milliseconds);

fn to be executed
↓
length of time interval between each execution

eg:-

```
<html>
<body>
  <p id="ID"> </p>
  <script type="text/javascript">
    var t = setInterval ( A, 2000);
    function A()
    {
      document.getElementById ("ID").
      innerHTML += "<p>|Welcome to
      javascript|</p>";
    }
  </script>
</body>
</html>
```

Browse Location & History

1) Location :-

The window.location object is useful for finding out the current location or path of the web page.

Various Properties

① window.location.hostname

It returns the name of the host on which the webpage is running

② window.location.pathname

It returns the path name at which the web page is located. It includes folder & file name.

③ window.location.protocol

It returns the web protocol used as HTTP, File, HTTPS.

④ window.location.assign

It loads the new document.

eg:- to display the pathname of the web page using window.location object

```
→ <html>
  <body>
    <p id="ID"> </p>
    <script type="text/javascript">
      document.getElementById ("ID").innerHTML
      TML = "This web page is at path: " +
      window.location.pathname;
```

```
</script>
</body>
</html>
```

eg:- Write a Javascript to display the protocol of the webpage using window.location object

```
→ <html>
  <body>
    <p id = "ID" ></p>
    <script type = "text/javascript">
      document.getElementById("ID").innerHTML = "This webpage is using the protocol " + window.location.protocol;
    </script>
  </body>
</html>
```

2) History

The window.history object is used for displaying browser history.

There are two important methods of window.browse

a) window.history.back()

This method load the previous URL in the history list. It is like clicking back button on browser window.

b) window.history.forward()
This method loads the next URL in the history list. It is like clicking forward button on browser window.

eg:-

```
<html>
  <head>
    <script type = "text/javascript">
      function A()
      {
        window.history.back();
      }
      function B()
      {
        window.history.forward();
      }
    </script>
  </head>
  <body>
    <form name = "form1">
      <input type = "button" value = "Back" onclick = "A()">
      <input type = "button" value = "Forward" onclick = "B()">
    </form>
  </body>
</html>
```