4.1.1 Basic of Cookies

Q. What is cookie?

Q. Explain the basics of cookie.

The user information present in web pages can be stored by using cookie.Cookie is a small piece of information sent from the website and stored in a file on the user's computer. The information sent from the website is specific to a user.It is stored by the user's web browser, and web browser can access that information.

Cookie works as follows :

Storing information :

When user visits the website the basic information such as username and password can be stored in a cookie.

Accessing information :

When user visits to that website next time, the stored username and password can be accessed by the web browser which is remembered by the browser in the form of cookie. Cookies are saved in form of name-value pair as follows :

```
Username = "student";
```

When a browser requests a website, cookies belonging to that website are added to the request. In this way the website gets the required data to remember the user related information.Cookie is a small text file stored in the subfolder of the browsers installation folder.It contains following data.

- A name-value pair
- An expiry date for that cookie
- The website/Domain name

4.1.2 Types of Cookies

There are three types of cookies.

(a) Session cookie

(b) Persistent cookie

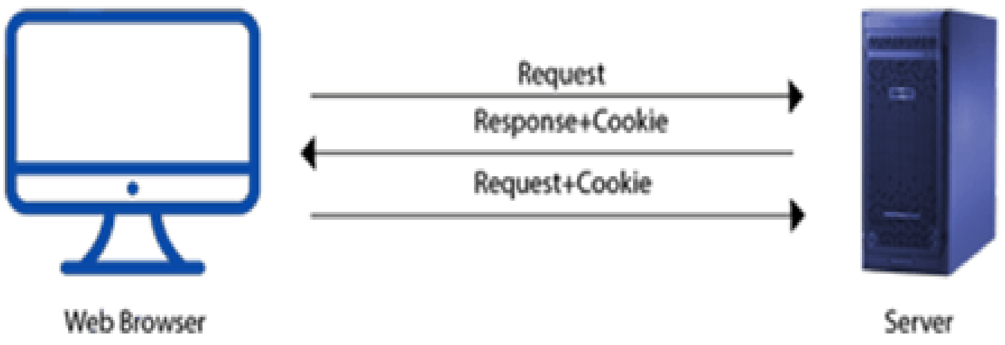
(c) Third party cookie

(a) Session Cookies

Session cookies are temporary cookies that remember user's online activities throughout a session. i.e. it remember the information when a user logged in until logout or when user close the web page. When a user gets logged out the session cookie gets expired.

(b) Persistent Cookies

Persistent cookies assigned an expiration date. It does not get expired until it reaches to the expiration date. It does not expire with users logged out and closing web page. It is stored in subfolder in folder of browser's installation. JavaScript can create, read and delete cookies using *document.cookie* property.



4.1.3 Creating a Cookie

Q.	How to create a cookie. Explain with example.
Q.	How to write a value to a cookie. Explain with example.

To create a cookie we need to assign a value to cookie by using document.cookie as follows :

```
document.cookie = "username = student";
```

Name   Value   Completion of Sentence (Semicolon)

As shown above, the value i.e. *'student'*is assigned to name i.e. *'username'* by using assignment operator '=' and sentence is terminated with a semicolon (;).The name value pair with double quote (" name = value ") is assigned to *document.cookie*.By default the above type of cookie is deleted when the browser is closed.You can also add an expiry date to the cookie in UTC time format,as follows :

```
document.cookie = "username = student;expires = Thu, 4 Jul 2019 11:30:00 UTC";
```

Name   Value   Delimiter (Semicolon)   Name   Value   Completion of Sentence

As shown above, if you want to assign more than one pair of name value to the cookie you need to separate the pairs with the delimiter, here semicolon (;) is used. Thus semicolon is used for the sentence completion and also for the pair separation. All the name value pairs are assigned in double quotes separated by the delimiter. The sentence completion semicolon is provided after the closing of the double quotes.

Program 1:

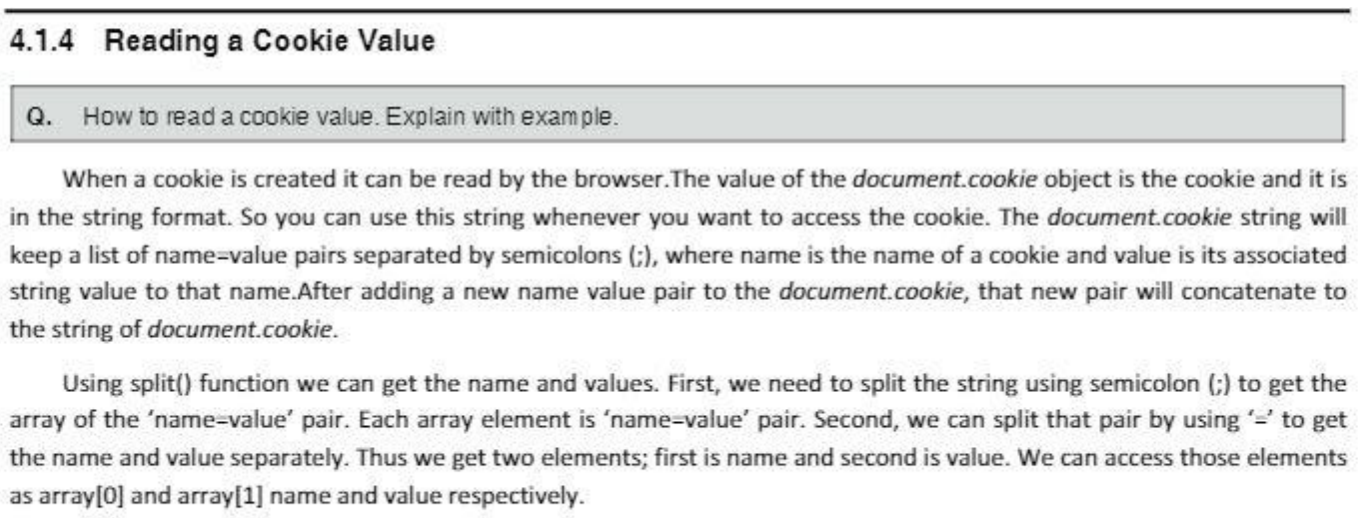
```
<!DOCTYPE html>

<html>

<head>

    <title> Button Element  </title>

            <script language="javasript" type="text/javascript">


function setCookie()

    {

        document.cookie="username=GPA";

        alert("Cookie is Created");

    }

 function getCookie()

    {

        if(document.cookie.length!=0)

        {

        alert(document.cookie);

        }

        else

        {

        alert("Cookie not available");

        }

    }


            </script>

</head>
```

```
<body>


<input type="button" value="setCookie" onclick="setCookie()">

<input type="button" value="getCookie" onclick="getCookie()">


</body>

</html>
```

Program 2:

```
<!DOCTYPE html>

<html>

<head>

   <title> Button Element  </title>

          <script language="javasript" type="text/javascript">


function setCookie()

   {

      var i=username.value;

      document.cookie="username="+i;

      alert("Cookie is Created");

   }

function getCookie()

   {

      if(document.cookie.length!=0)

      {

      alert(document.cookie);

      }

      else

      {

      alert("Cookie not available");

      }

   }


          </script>

</head>
```

```
<body>


Enter Username:<input type="text" id="username"> <br><br>

<input type="button" value="setCookie" onclick="setCookie()">

<input type="button" value="getCookie" onclick="getCookie()">


</body>

</html>
```



4.1.4 Reading a Cookie Value

Q. How to read a cookie value. Explain with example.

When a cookie is created it can be read by the browser.The value of the *document.cookie* object is the cookie and it is in the string format. So you can use this string whenever you want to access the cookie. The *document.cookie* string will keep a list of name=value pairs separated by semicolons (;), where name is the name of a cookie and value is its associated string value to that name.After adding a new name value pair to the *document.cookie*, that new pair will concatenate to the string of *document.cookie*.

Using split() function we can get the name and values. First, we need to split the string using semicolon (;) to get the array of the 'name=value' pair. Each array element is 'name=value' pair. Second, we can split that pair by using '=' to get the name and value separately. Thus we get two elements; first is name and second is value. We can access those elements as array[0] and array[1] name and value respectively.

Program 3:

```
<!DOCTYPE html>

<html>

<head>

    <title> Button Element  </title>

             <script language="javasript" type="text/javascript">


function setCookie()

    {

        document.cookie="username=GPA";

        alert("Cookie is Created");

    }

 function getCookie()

    {

        var res = document.cookie;

         var multiple = res.split(";");

         for(var i=0;i<multiple.length;i++)

         {

          var array= multiple[i].split("=");

        alert("Name="+array[0]+" "+"Value="+array[1]);

         }

    }


             </script>

</head>

<body>
```

```html
<input type="button" value="setCookie" onclick="setCookie()">

<input type="button" value="getCookie" onclick="getCookie()">


</body>

</html>
```

Program 4:

```
<!DOCTYPE html>

<html>

<head>

   <title> Button Element  </title>

            <script language="javasript" type="text/javascript">


function setCookie()

   {

      document.cookie = "username=" + document.getElementById("user").value;

      document.cookie = "password=" + document.getElementById("pass").value;

      alert("Cookie is Created");

   }

 function getCookie()

   {

       var res = document.cookie;

        var multiple = res.split(";");

        for(var i=0;i<multiple.length;i++)

        {

         var array= multiple[i].split("=");

       alert("Name="+array[0]+" "+"Value="+array[1]);

        }

   }


            </script>

</head>

<body>
```

```
Enter Username:<input type="text" id="user"> <br>

Enter Password:<input type="text" id="pass"> <br><br>


<input type="button" value="setCookie" onclick="setCookie()">

<input type="button" value="getCookie" onclick="getCookie()">


</body>

</html>
```

4.1.5. Writing a Cookie

To create a cookie :

```
document.cookie = “username = student”;
```

In above statement, we create a cookie with name ‘username’ and value ‘student. If we want change the value then we need to rewrite the statement as follows:

```
document.cookie = “username = admin”;
```

4.1.6 Deleting a Cookie

Q. How to delete a cookie. Explain with example.

If the cookie is temporary then the cookie will deleted automatically after closing the browser. If we want to delete the temporary cookie without closing the browser, we must update the expiry date of the cookie to the past date. Also to delete the persistent cookie we must update the expiry date of the cookie to the past date manually. Then automatically cookie gets expired and it will be deleted by the browser. We used to set the date 1 January 1970 in the format: “Thu, 01 Jan 1970 00:00:01 GMT” as follows.

```
document.cookie = “cookie_name =cookie_value; expires = Thu, 01 Jan 1970 00:00:01 GMT”;
```

Program 5:

```
<!DOCTYPE html>

<html>

<head>

   <title> Delete Cookie  </title>

            <script language="javasript" type="text/javascript">


function setCookie()

   {

      var i=username.value;

      document.cookie="username="+i;

      alert("Cookie is Created");

   }

function deleteCookie()

   {

      var i=username.value;

      document.cookie="username="+i+";expires=Sun, 20 Aug 2000 12:00:00 UTC";

      alert("Cookie is Deleted");

   }

function getCookie()

   {

      if(document.cookie.length!=0)

      {

      alert(document.cookie);

      }

      else

      {
        alert("Cookie not available");

        }

    }


            </script>

</head>

<body>


Enter Username:<input type="text" id="username"> <br><br>

<input type="button" value="setCookie" onclick="setCookie()">

<input type="button" value="deleteCookie" onclick="deleteCookie()">

<input type="button" value="getCookie" onclick="getCookie()">


</body>

</html>
```

4.1.7 Setting the Expiration Date of Cookie

Q. How to set the expiration date to a cookie. Explain with example.

You can extend the life of a cookie by setting a new expiration date to the cookie. This can be done by setting the 'expires' attribute to a new date and time. The new date object is obtained by using *'new Date()'*. The Date has following function to get the month of the date to set the month of the date and to format the date string.

1. **getMonth() :** this function returns the month of the current date object.
2. **setMonth() :** this function set the month to the current date object
3. **toUTCString() :** this function formats the date in the standard format to set to the expiration date of the cookie.

Program 6:

```
<!DOCTYPE html>

<html>

<head>

  <title> Change Date of Cookie  </title>

        <script language="javasript" type="text/javascript">


function setCookie()

  {

    var i=username.value;

    document.cookie="username="+i;

    alert("Cookie is Created");

  }

function changeDate()

  {

    var i=username.value;


    var now=new Date();

    now.setMonth(now.getMonth()+1);

    document.cookie="username="+i+";expires="+now.toUTCString()+";path=/";

    alert("Expiry Date of Cookie changed");

  }


function getCookie()

  {

    if(document.cookie.length!=0)

    {
        alert(document.cookie);

        }

        else

        {

        alert("Cookie not available");

        }

    }

            </script>

</head>

<body>


Enter Username:<input type="text" id="username"> <br><br>

<input type="button" value="setCookie" onclick="setCookie()">

<input type="button" value="changeDate" onclick="changeDate()">

<input type="button" value="getCookie" onclick="getCookie()">


</body>

</html>
```

4.2 Browser

We can perform various operation with browser window in JavaScript. We can opena new window from currently running javascript. We can give focus to the new window.We can set the position, size and location of new window. We can change the content of the window.We also can perform operation such as scrolling a webpage, multiple window at a glance, creating a webpage in new window and closing a window.

4.2.1 Opening a Window

Q. How to open a window. Describe various style of window.

Q. How to open a window. Explain with example.

We can open a new window from current window on button click. To perform operation with the browser window we have window object. The window object has various function and properties to work with the browser window. To open a new window from current window we use the open() of the window object.

open() function;

This function is used to open a new window from current window. It is invoked on window object.

Syntax :

```
window.open(url, name, style);
```

where,

url : it specifies the URL of the new page going to be open in new Window.

Name : it is used to set the name of the window. It is optional.

Style : Style of the window has various parameters such as scrollbar, toolbar, height, width, location, etc. this is also optional.

If we do not provide optional parameter values, then the window gets default values.

There are various styles of window as follows :

Style	Description
toolbar= yes|no|1|0	To display the browser toolbar
status= yes|no|1|0	To add a status bar
titlebar= yes|no|1|0	To display the title bar
menubar= yes|no|1|0	To display the menubar
fullscreen=yes|no|1|0	To display the browser in full screen mode
resizable=yes|no|1|0	To determine whether the window is resizable
scrollbars= yes|no|1|0	To display scroll bars
height=pixels	The height of window minimum value is 100
width=pixels	The width of the window
left=pixels	The left position of the window
top=pixels	Top position of window

Program 1: Creating new window

```
<!DOCTYPE html>

<html>

<head>

<title> Child Window </title>

<script language="javasript" type="text/javascript">

function openWindow()

{

var new_win=window.open("","","width=300,height=300" );


}

</script>

</head>

<body>

<p> Click to Open a New Window</p>

<input type="button" value="Open New Window" onclick="openWindow()"/>

</body>

</html>
```

Program 2: Giving focus to new window

```
<!DOCTYPE html>

<html>

<head>

<title> Child Window </title>

<script language="javasript" type="text/javascript">

function openWindow()

{

var new_win=window.open("","","width=300,height=300" );

this.focus();

}

</script>

</head>

<body>

<p> Click to Open a New Window</p>

<input type="button" value="Open New Window" onclick="openWindow()"/>

</body>

</html>
```

Program 3: Window position

```
<!DOCTYPE html>

<html>

<head>

<title> Child Window </title>

<script language="javasript" type="text/javascript">

function openWindow()

{

var new_win=window.open("","","left=50,top=50,width=300,height=300" );


}

</script>

</head>

<body>

<p> Click to Open a New Window</p>

<input type="button" value="Open New Window" onclick="openWindow()"/>

</body>

</html>
```

- ## Changing the content of Window:

a. **Changing the content of whole Window:**

Program 4:

```
<!DOCTYPE html>
<html>
<head>
<title> Changing Window Content</title>
<script language="javasript" type="text/javascript">
function changeContent()
{
document.write("<h3><p>After clicking the Button: New Content</p></h3>");

}
</script>
</head>

<body>
<p> Click to change content of whole  Window</p>
<input type="button" value="Change Content" onclick="changeContent()"/>
<h3><p id="para">Before clicking the Button: Old Content</p></h3>
</body>
</html>
```

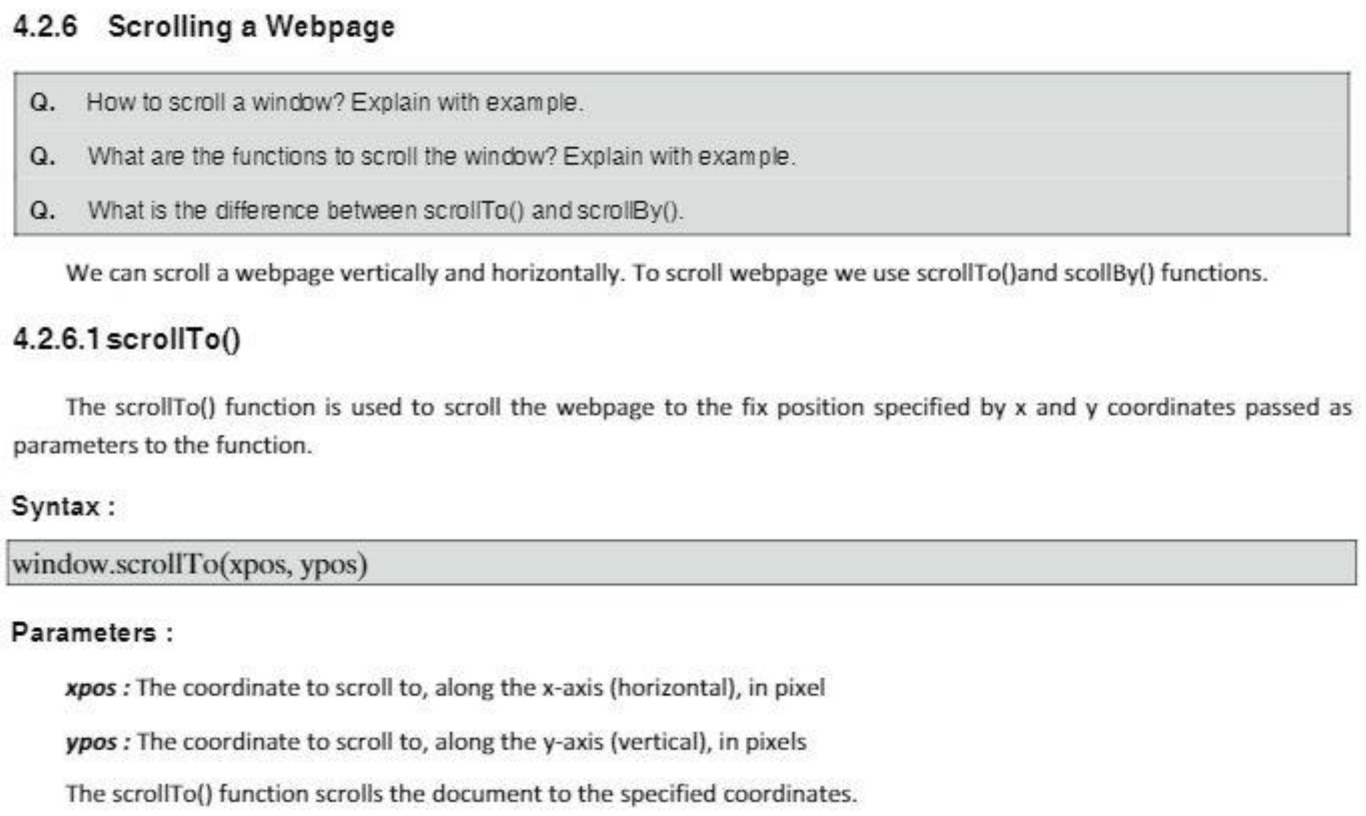
b. Changing the content of specific element in Window:

Program 5:

```
<!DOCTYPE html>
<html>
<head>
<title> Changing Window Content</title>
<script language="javasript" type="text/javascript">
function changeContent()
{
document.getElementById("para").innerHTML=("After clicking the Button: New Content");

}
</script>
</head>

<body>
<p> Click to change content of paragraph</p>
<input type="button" value="Change Content" onclick="changeContent()"/>
<h3><p id="para">Before clicking the Button: Old Content</p></h3>
</body>
</html>
```

Program 6: Closing window

```
<!DOCTYPE html>

<html>

<head>

<title> Child Window </title>

<script language="javasript" type="text/javascript">

function openWindow()

{

var new_win=window.open("","","width=300,height=300" );


}

function closeWindow()

{

  new_win.close();

}

</script>

</head>

<body>

<p> Click to Open a New Window</p>

<input type="button" value="Open New Window" onclick="openWindow()"/>

<p> Click to Close a New Window</p>

<input type="button" value="Close New Window" onclick="closeWindow()"/>

</body>

</html>
```



4.2.6 Scrolling a Webpage

Q.	How to scroll a window? Explain with example.
Q.	What are the functions to scroll the window? Explain with example.
Q.	What is the difference between scrollTo() and scrollBy().

We can scroll a webpage vertically and horizontally. To scroll webpage we use scrollTo()and scollBy() functions.

4.2.6.1 scrollTo()

The scrollTo() function is used to scroll the webpage to the fix position specified by x and y coordinates passed as parameters to the function.

Syntax :

```
window.scrollTo(xpos, ypos)
```

Parameters :

xpos : The coordinate to scroll to, along the x-axis (horizontal), in pixel

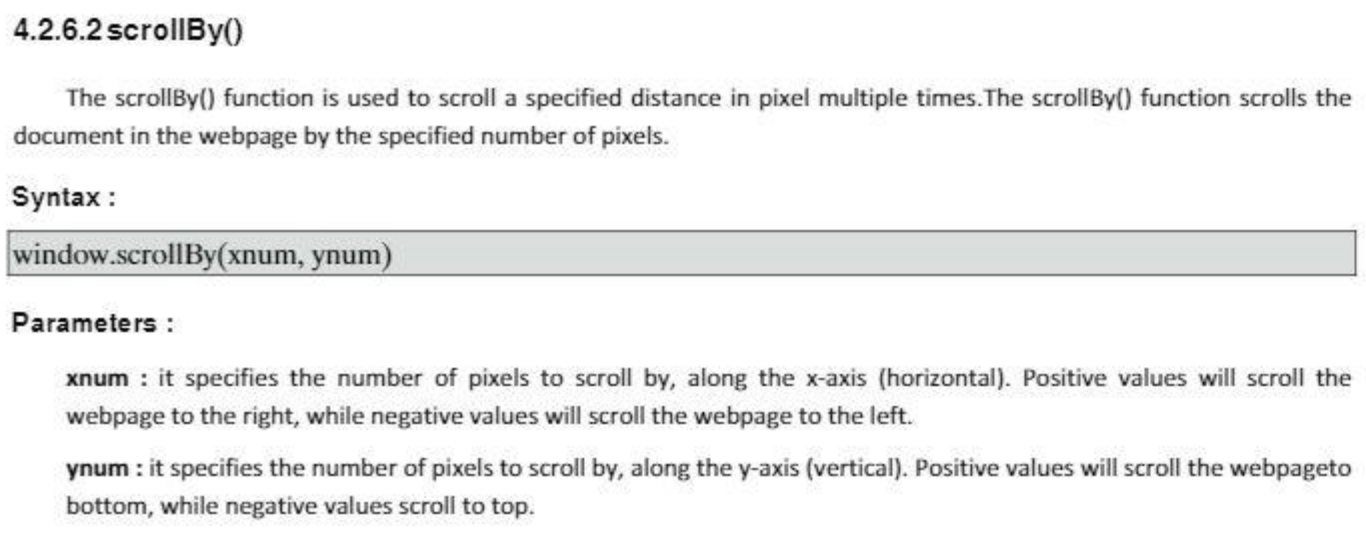
ypos : The coordinate to scroll to, along the y-axis (vertical), in pixels

The scrollTo() function scrolls the document to the specified coordinates.

Program 7:

```
<!DOCTYPE html>

<html>

<head>

<title> Scrolling Window </title>

<script language="javasript" type="text/javascript">

function scrollWindow()

{

scrollTo(0,500);

}
```

```
</script>

</head>

<body>

<p> Click to Scroll a Window</p>

<input type="button" value="Scroll Window" onclick="scrollWindow()"/>

<p>A</p>

<p>B</p>

<p>C</p>

<p>D</p>

<p>E</p>

<p>F</p>

<p>G</p>

<p>H</p>

<p>I</p>

<p>J</p>


<p>------------Window is Scrolled</p>


</body>

</html>
```



4.2.6.2 scrollBy()

The scrollBy() function is used to scroll a specified distance in pixel multiple times.The scrollBy() function scrolls the document in the webpage by the specified number of pixels.

Syntax :

```
window.scrollBy(xnum, ynum)
```

Parameters :

xnum : it specifies the number of pixels to scroll by, along the x-axis (horizontal). Positive values will scroll the webpage to the right, while negative values will scroll the webpage to the left.

ynum : it specifies the number of pixels to scroll by, along the y-axis (vertical). Positive values will scroll the webpageto bottom, while negative values scroll to top.

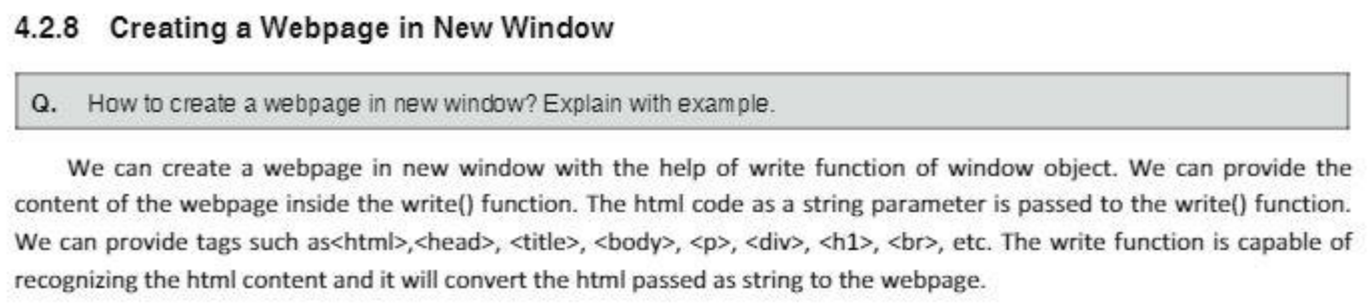
Program 8:

```
<!DOCTYPE html>

<html>

<head>

<title> Scrolling Window </title>

<script language="javasript" type="text/javascript">

function scrollWindow()

{

scrollBy(0,200);

}

</script>

</head>



<body>

<p> Click to Scroll a Window</p>

<input type="button" value="Scroll Window" onclick="scrollWindow()"/>

<p>A</p>

<p>B</p>

<p>C</p>
```

```
<p>D</p>

<p>E</p>

<p>----------------------Window is Scrolled</p>

</body>

</html>
```

Program 8: Multiple Windows

```
<!DOCTYPE html>

<html>

<head>

<title> Child Window </title>

<script language="javasript" type="text/javascript">

function openWindow()

{


for(i=0;i<5;i++)

{

window.open("","","width=300,height=300");

}


}

</script>

</head>

<body>

<p> Click to Open a Multiple Windows</p>

<input type="button" value="Open Multiple Windows" onclick="openWindow()"/>

</body>

</html>
```



4.2.8 Creating a Webpage in New Window

Q. How to create a webpage in new window? Explain with example.

We can create a webpage in new window with the help of write function of window object. We can provide the content of the webpage inside the write() function. The html code as a string parameter is passed to the write() function. We can provide tags such as<html>,<head>, <title>, <body>, <p>, <div>, <h1>, <br>, etc. The write function is capable of recognizing the html content and it will convert the html passed as string to the webpage.

Program 9:

```
<!DOCTYPE html>
<html>
<head>
<title> Child Window </title>
<script language="javasript" type="text/javascript">


function openWindow()
{
var new_win=window.open("New Window","","width=300,height=300" );
new_win.document.write("<html>");
new_win.document.write("<head>");
new_win.document.write("<title>");
new_win.document.write("New Window");
new_win.document.write("</title>");
new_win.document.write("</head>");
new_win.document.write("<body>");
new_win.document.write("<h3><p>Welcome to New Window</p></h3>");
new_win.document.write("<h4><p>This webpage is created in New Window</p></h4>");
new_win.document.write("</body>");
new_win.document.write("</html>");
}
</script>
</head>
<body>
<p> Click to Open a New Window</p>
<input type="button" value="Open New Window" onclick="openWindow()">
</body>
</html>
```

4.2.9 Javascripts in URLs

Q. How JavaScript is used in URLs?

Javascript code can be used with the URL by using 'javascript' pseudo-protocol specifier. This pseudo-protocol specifier specifies that the body of URL is an arbitrary javascript code. This code is going to be interpreted by the javascript interpreter. For example: javascript:alert("Hello World!");.

If the JavaScript code in a javascript: URL contains multiple statements, the statements need to be separated by semicolon. Such a URL looks like: javascript:var now = new Date(); "The time is:" + now; When the browser "loads" these JavaScript URLs, it executes the JavaScript code present in that URL and displays the result. For security reasons the recent browsers are not allowing to execute the javascript code in URLs.

4.2.10 Javascript Security

Q. Explain the JavaScript Security issues and measures.

There are several JavaScript security issues that needs to be considered. As a javascript used in client side, a risk is there for end users. It enables malicious users to send scripts over the web and run them on client computers. There are two measures need to be considered.We should use scripts separately so that the malicious users can only access certain resources and perform specific tasks. The second measure is we should implement the same origin policy i.e. we must use the script from only one site, which prevents scripts of one site from accessing data from script of other sites.

Cross-Site Scripting (XSS) is one of the most common JavaScript security vulnerabilities. Cross-Site Scripting vulnerabilities allow malicious users to manipulate websites to return malicious scripts to the users. These malicious scripts then execute on the client web browser. It can be result in user data theft, account tampering, malware spreading or remote control over a user's browser.

Cross-Site Request Forgery (CSRF) is another common JavaScript security vulnerability. It allows attackers to manipulate users' browsers to take venerable actions on other sites. It happens due to target sites authenticate the request based on the users' cookie. Attackers are able to send requests using users' cookies. This JavaScript security issue causes account tampering, data theft, fraud and more.

4.2.11 Timers

Q.	What are timers ? Explain with example.
Q	Explain the timer functions with example.
Q	Explain setTimeout() and setInterval() with example.
Q	What are the difference between setTimeout() and setInterval().

In a webpage, sometimes there is a situation where you need to generate an event dynamically, or to perform some repeated task without any action of user. JavaScript offers two simple functions that allow you to schedule an event to trigger at a predefined time.

1. setTimeout()
2. setInterval()

4.2.11.1 setTimeout()

The setTimeout() function allows you to schedule a specified function to trigger at a specific time from the current time:

Syntax :

```
setTimeout(function, time);
```

Parameters :

function : specifies the name of the function to trigger, while

time : specifies the amount of time (in milliseconds) for the browser to wait until triggering the function.

Ex. setTimeout(displayFunction, 5000);

As shown in above example the *displayFunction* is a function name which will be triggered after waiting five seconds:

Program 10:

```
<!DOCTYPE html>

<html>

<head>

<title> Timer Example </title>

<script language="javasript" type="text/javascript">


function myFunction()

{

setTimeout(displayFunction, 5000);

}


function displayFunction()

{

alert("Time out");

}

</script>

</head>

<body>

<p> Click to set the Time Out</p>

<input type="button" value="Set Time Out" onclick="myFunction()"/>


</body>

</html>
```

4.2.11.2 setInterval()

When you need to trigger a function repeatedly as a specific time interval, we use setInterval() function. Such as if your application needs to refresh data from the application database on the server. In that case we use *setInterval()* function.

Syntax :

```
setInterval(displayFunction, time);
```

Parameters :

displayFunction : is a function name which will be triggered.

time : is time interval for which the function is called repeatedly.

Program 11:

```
<!DOCTYPE html>

<html>

<head>

<title> Timer Example </title>

<script language="javasript" type="text/javascript">


function myFunction()

{

setInterval(displayFunction, 3000);

}


function displayFunction()

{

document.getElementById("para").innerHTML+="<p> Welcome to Javascript</>";

}

</script>

</head>


<body>

<p> Click to set the Time Interval</p>

<input type="button" value="Set Time Interval" onclick="myFunction()"/>

<p id="para"> </p>


</body>

</html>
```

4.2.12 Browser Location and History

4.2.12.1 Location

Q. Explain window location with suitable example.

Q. Enlist and describe the properties of window location.

When user opens any website, the browser will send the request to the server. Then the server sends the requested webpage to the browser. The webpage is stored on server. If we want know the location or path of the webpage, we use window.location object. It is used to find the location and path of the current webpage. There are various properties of window.location shown in the following table.

Property	Description
Window.location.pathname	It gives thecomplete name of path at which the webpage is located. It includes the folder names and file name.
Window.location.hostname	It gives the name of the host on which the webpage is running. It gives the domain name.
Window.location.protocol	It gives the protocol used for the webpage. Such as HTTP, HTTPS
Window.location.assign	It is used to load the new document in webpage

Program 12: Path

```
<!DOCTYPE html>

<html>

<head>

<title> Path Example </title>

<script language="javasript" type="text/javascript">



function displayFunction()

{

document.getElementById("para").innerHTML="The webpage is located at: "+window.location.pathname;

}



</script>

</head>


<body>

<p> Click to get the Path of webpage</p>

<input type="button" value="Get the Path" onclick="displayFunction()"/>

<p id="para"> </p>


</body>

</html>
```

Program 13: Host

```
<!DOCTYPE html>

<html>

<head>

<title> Host Example </title>

<script language="javasript" type="text/javascript">


function displayFunction()

{

document.getElementById("para").innerHTML="The host of the webpage: "+window.location.hostname;

}

</script>

</head>


<body>

<p> Click to get the Host of webpage</p>

<input type="button" value="Get the Host" onclick="displayFunction()"/>

<p id="para"> </p>


</body>

</html>
```

Program 14: Protocol

```
<!DOCTYPE html>

<html>

<head>

<title> Protocol Example </title>

<script language="javasript" type="text/javascript">



function displayFunction()

{

document.getElementById("para").innerHTML="The protocol of the webpage: "+window.location.protocol;

}



</script>

</head>


<body>

<p> Click to get the Protocol of webpage</p>

<input type="button" value="Get the Protocol" onclick="displayFunction()"/>

<p id="para"> </p>


</body>

</html>
```

4.2.12.2 History

The *window.history* object contains the record of URLs visited by the user within a browser window. The history object is a part of the window object so that it is accessed by window.history.

Property :

length : it is used to return the number of URLs in the list of history.

Methods :

back() : it is used to load the previous URL in the history list.

forward() : it is used to load the next URL in the history list.

go() : it is used to load a specific URL from the history list.

Program 15: History

```
<!DOCTYPE html>

<html>

<head>

<title> Protocol Example </title>

<script language="javasript" type="text/javascript">


function previous()

{

window.history.back();

}


function next()

{

window.history.forward();

}

</script>

</head>


<body>

<input type="button" value="Previous" onclick="previous()"/>

<input type="button" value="Next" onclick="next()"/>


</body>

</html>
```